Requirement Document

## **Group-4**

## **Team-4**

**Team Members:**

ADITYA SHARMA 2021201016

TEJ MITTAL 2021202021

SHRAVAN SHARMA 2021201058

ASHWIN VAISH 2019114014

**INDEX**

**1. Introduction**

**1.1 Scope**

**2.Intended Use**

**2.1 Assumptions and dependencies**

**3) System Features and Requirements**

**3.1) Platform Requirements**

**3.1.1) Different actors on the platform**

**3.2) Functional Requirements**

**3.2.1) AI Model**

**3.2.2) Registry & Repository**

**3.2.3) Packaging details**

**3.2.4) Configuration files details**

**3.2.5) Interaction of different actors with our subsystem**

**3.3) Non-Functional Requirements:**

**3.3.1) Scalability**

**3.3.2) Accessibility of data**

**4) Key Functions**

**4.1) Block Diagram of all components**

**4.2) Brief description of each component**

**5) List what the users can do with the solution**

**6) Primary test case**

**6.1) Name of use case**

**6.2) Company executing the use case**

**6.3) Description of the use case purpose, interactions and what will the users**

**benefit from**

**7) Functional Overview**

**8) Sub-systems**

**9) Services/Capabilities**

**10) Interactions between this and other parts**

**1) Introduction**

According to McKinsey Global Survey on AI indicates that AI adoption continues to grow and the benefits remain significant— though, in the COVID-19 pandemic’s first year, they were felt more strongly on the cost-savings front than the top line. As AI adoption grows, AI model building is also growing, and therefore applications that are using AI models are also increasing.

AI model development and application development are two separate entities. So we are building a platform that connects these data scientists (who build the AI model) and the application developer.

We as a team 4 handle model uploading, logging, and bootstrapping. Model uploading is responsible for model upload to the platform. The logging service is responsible for storing all logs in the database and bootstrapping service is responsible for starting the platform.

**1.1) Scope:**

The scope of our team is to build three services, Model upload service, logging service, and bootstrapping service.

The model upload service is responsible for validating zip files and contract files uploaded by the data scientist. Then this service creates an API file for the model and an onboarding script for model running.

The logging service will stores all logs created by different services and bootstrapping service will start the platform.

**2) Intended Use:**

These Three services are used by different microservices are as follow:

* **Model upload service:** This service will be used by the Request manager to upload the model given by the data scientist.
* **Logging service:** This service will be used by all micro-services that want to add some log.
* **Bootstrapping service:** This service is the start point of all services available on the platform.

**2.1) Assumptions and dependencies:**

We are doing some basic validation of the zip file and contract files with the assumption that the data scientist will not change the template given for the contract. We are also assuming that data scientists will provide all requirements that are required to run the model.

**3) System Features and Requirements**

**3.1) Platform Requirements**

**3.1.1) Different actors on the platform**

**Data Scientist:**

Data Scientist will build ML/AI Model which will be deployed on the platform and used by Application Developers.

**Platform Developer:**

Platform Developer will develop and integrate all the major components of the platform and simultaneously do the testing.

**Platform Administrator:**

Platform Administrator will be responsible for core administration, support platform technologies, and will ensure overall performance, availability. Along with this administrator also handle sensor registration that will be supported by our platform.

**Application Developer:**

Application Developer will build and deploy the application using the ML/AI model present in the platform and sensor available on the platform.

**User:**

All the applications deployed on the platform will be used by this actor.

**3.2) Functional Requirements**

**3.2.1) AI Model**

**1. Development of AI Model**

* Data scientists will build the required machine required model in Python (currently only Python 3 is supported). They must separate the preprocessing and postprocessing logic from their prediction code using 3 functions.
* Data Scientist will save the final model as a .pkl file.

**2. Packaging of Model**

Data scientists will have to bundle up the required files and upload the zip for the model to deploy on the platform. The zip must contain the following files:

* **Pickle file**: This file will have the final saved model.
* **Contract.json:** This file will contain all the procedures which need to be generated on the platform to run the .pkl file, which will eventually run the AI model built by Data Scientists.
  + **Other required files:** This will contain details about the initial configuration that is required before running the model on the platform, like the .py file which contains class logic in which pre/post-processing, etc functions will be defined.

The above files will be zipped and uploaded on the platform.

**3. Upload and deployment of the model**

Model upload:

* The data scientist will provide a zip file containing the files mentioned above.
* Before uploading the model, we validate files present in a zip file.
* After validation, the AI model uploader service creates two files, an onboarding script for running this model and an API for this model.
* AI model uploader will store in a database the model name, model ID, and path where these files are stored on the server.

Model deployment:

* The model deployer will receive the model ID, and by using this ID it will fetch the model.
* Then the model deployer will run an onboarding script that presents on fetched address from the database.
* This onboarding script will install all prerequisite modules and get the model in a runnable state.
* After deployment of the AI model, it will be present to predict value at a given endpoint.

**3.2.2) Registry & Repository**

This Registry and repository are required by bootstrapping service to boot the platform and also required by logging and model upload service to store log and model respectively.

**3.2.3) Packaging details**

We will require one package:

**Model Package-**This will contain all the files required for a model (like a contract file, pickle file, and other required files).

**3.2.4) Configuration files details**

**Model Configuration File**

It will contain configuration and executable files related to a model following a predefined contract between Data scientist and Platform.

**3.2.5) Interaction of different actors with our subsystem**

**Data Scientist:**

Data Scientists will interact with the platform to upload the ML models built by them. The data scientist will send a request to the request manager then the request manager will forward the request to the model upload service.

**Platform Administrator:**

It will run the bootstrapping script to start the platform.

**Application Developer:**

This actor has no direct connection with our subsystem. But it will be provided with the list of models uploaded by the data scientists.

**End-User:**

This actor also has no direct connection with our subsystem.

**3.3) Non-Functional Requirements:**

**3.3.1) Scalability**

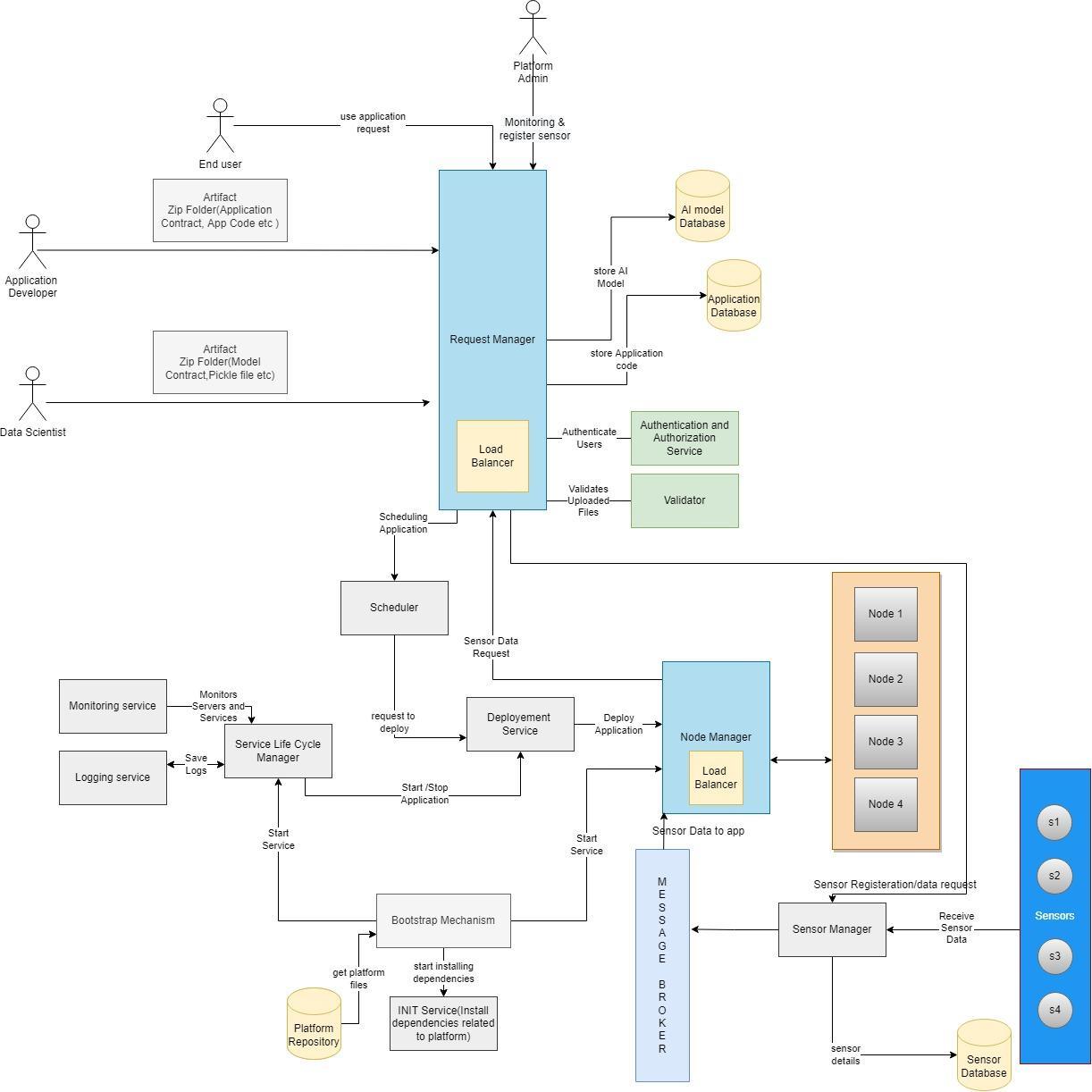
The system our team is building will have scalability, that is, a data scientist can upload any number of models to the platform.

**3.3.2) Accessibility of data**

All the models uploaded to the platform will be available for the Application developer to use in their application.

**4) Key Functions**

**4.1) Block Diagram of all components**



**4.2) Brief description of each component:**

* **Authenticator:** The actors(Data Scientist, App Developer) would Register and log in to the Platform. The Authenticator will authenticate and Authorize the user so that they can perform their respective tasks.
* **Request Manager:** Users' requests will now be routed to the Request Manager, who will handle all of the routine tasks. The Request Manager would also be in charge of Load Balancing, ensuring that traffic is distributed evenly across all servers and that no server is overburdened.
* **Service Lifecycle Manager:** The Service Lifecycle Manager will be in charge of all service-related tasks such as starting a new service instance, recovering a service instance, terminating the entire service, or terminating a service instance. It will also be in charge of monitoring all services. It is also in charge of providing an easy-to-use general-purpose logging system that records all events associated with a specific service.
* **Scheduler**: It will perform the scheduling of services on the platform. It will schedule model deployment or app deployment. It will schedule the model for a particular time interval (between start time and end time).
* **Deployer:** It will perform deployment of Machine Learning Models created by the data scientist and Applications created by App Developers. ItIPwill packages the model and deploys it to any server instance.
* **Platform Manager:** The platform manager will perform tasks related to sensors and controllers. Such as adding a new sensor and controller to the database. Sensor Management, Fault Tolerance, Controller, Communication will be done through this.
* **Repository Manager:** All the data will be present in the Repository (User data, as well as some model-related data like .pkl file and config file). So all this data related management will be done through the Repository Manager.
* **UI Manager:** All the data that the user should provide should be through an interface. So the task related to the management of the user interface is done by UI Manager.
* **Validator:** Checking if the input and output are in the correct form and valid is another important function. This will be done by Validator. The validator will also be responsible for the validation of the contracts (checking if the data given is as per the contract or not).
* **Bootstrap Mechanism:** It will do all the platform startup-related tasks. It will start all the essential services which will be present in the INIT service and access the corresponding files in the Platform Repository.
* **Sensor Manager:** It will manage all the sensor-related data like Sensor Type, Location (IP Address, Port number, Physical Location), and Assign a Sensor ID to it, which will be used when the end-user will be using the app. It will also send these details to the Message Broker.

**5) List what the users can do with the solution**

Components that are built by our team will be used by data scientists to upload the model to the platform to make it available to the application developer. Admin can access logs to check if the platform is working fine or not.

**6) Primary test case**

**6.1) Name of use case:**

Create an AI application using an ML model.

**6.2) Company executing the use case:**

Consumer technology services company.

**6.3) Description of the use case purpose, interactions and what will the users benefit from:**

* Machine Learning models will be generated by the data scientists, who will aggregate the data, and utilize an ML algorithm.
* App developers who wish to utilize will create applications utilizing available models and adding them to their projects.
* Users benefit as the scientists creating the model and developers creating the applications have a common platform to speed up development.

**7) Functional Overview:**

* **ML Model uploader:** This service is responsible for uploading the model to the platform. It takes a zip file on the */upload\_model* endpoint and validates it. This zip file contains the model pickle file, contract.json, and other required python files. This service generates model API and created an onboarding script for a given model that requires at the time of model deployment.
* **Logging:** This service is responsible for storing all logs created from all services. At the given endpoint, this service takes logs and stores them in the database.
* **Bootstrapping:** This service is responsible for starting the platform. It is basically a script file that first installs required modules and then starts all services.

**8) Sub-systems:**

* **ML Model uploader**
* **Logging**
* **Bootstrapping**

**9) services/capabilities:**

This part is responsible for :

* Validation of zip file and contract file.
* Creation of API for each model to be uploaded.
* Creation of onboarding script that is responsible for the deployment of the AI model.
* Creation of logging API.
* Creation of bootstrapping script file.

**10) Interactions between this and other parts:**

#### 