

Team Requirements Document

Platform and Application UI, Scheduler and Load Balancer

Group 1, Team 2

May 14, 2021

Divyansh Shrivastava, Abhishek Gorisaria, Aman Nautiyal

Mentor and Professor : Prof. Ramesh Loganathan

TA: Pratik Tiwari, Shubham Agarwal, Jay Krishna

Contents

1	Functional Overview	3
2	List of sub-systems and their services	3
2.1	Platform and Application UI	3
2.2	Scheduler	3
2.3	Load Balancer and Service Node Manager	4
3	Interaction with other modules	4
4	Block diagram	5

1 Functional Overview

- Platform and Application UI provides a GUI for the Application developers as well as Application users to interact with our platform. It is more of a luxury than a necessity but it makes the overall UX much smoother.
- Scheduler is used to schedule the user requests for running specific algorithms at some specific time or day. It also takes in any runtime information from the user that might be needed and delegates it forward into the platform.
- Load Balancer and Node Manager is responsible for managing the load of the several worker nodes which will be used for actually executing the requested algorithms. Load balancer checks the load of all the worker nodes and optimally assigns the nodes to the requests for execution.

2 List of sub-systems and their services

2.1 Platform and Application UI

- Platform and Application UI provides a GUI for the Application developers as well as Application users to interact with our platform.
- UI provides different roles to the users like Application Developer and Application User. Both have different level of permissions.
- An Application developer can use any deployed application and can also upload his/her application on the platform. Application User can only use the applications that are deployed on the platform.
- UI of the platform is made using the following languages/frameworks/tools : Bootstrap, css, html, python flask, Sqlite and MongoDB.

2.2 Scheduler

- Scheduler is responsible to scheduling the starting and stopping of requested application services/algorithms.
- Application User can request for scheduling of algorithm execution with the desired sensors.
- The input to this component will be the user request json file which will specify which application and algorithm to run with specific place_id and the actions that are associated with the execution of the selected algorithm.

2.3 Load Balancer and Service Node Manager

- The Load Balancer module returns the service node with the least resource usage for the deployer to deploy a service on. .
- The Service Node Manager keeps track of what service nodes are up and which are down. It informs the monitoring module of service nodes that are down.
- There should be a persistent state storage so that the modules can pick off at restart from what state they were in before closing.

3 Interaction with other modules

- Our module will interact with UI/CLI module to get the config file from the Application developer who wants to deploy his application.
- Our module will also interact with the host machines as they will be used for deploying the application as chosen by the load balancer.
- Platform Initializer will be used to start up all the components in this module.
- Fault tolerance module will keep a watch over all the components in this module.

4 Block diagram

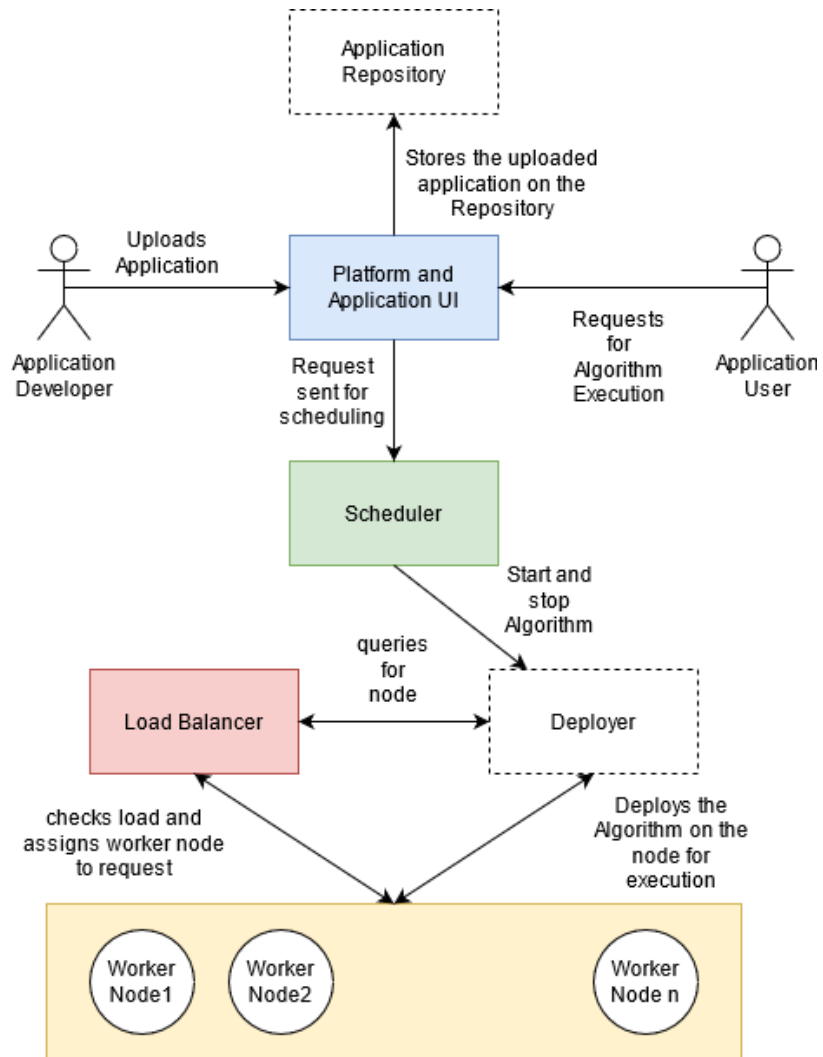


Figure 1: Interaction between the Modules