**Execution Server v1.0**

**Requirements:**

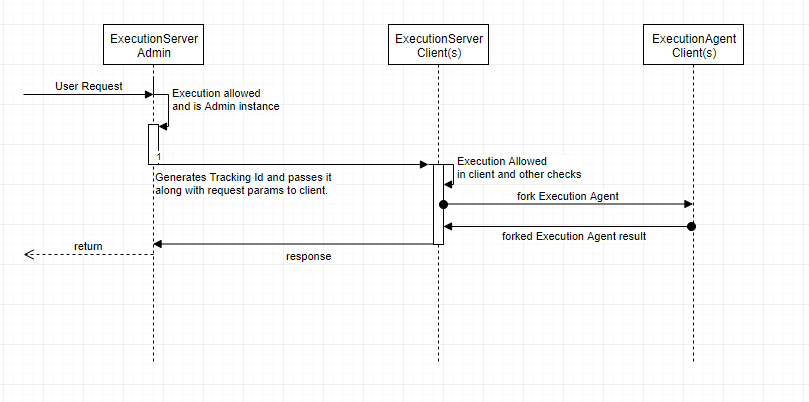
1. Authentication and Authorization for received requests.
2. Validate against black listed commands before forking execution agent.
3. Ability to stop execution (forking of execution agents) in clients.
4. Ability to run commands in multiple/all clients.
5. Ability to get status of submitted job by suppling tracking id for that job.

**Execution Server Design and Implementation Details:**

Language: Java (Spring Boot Application)

Architecture: Multi-Instance

**Sequence Diagram:**



**Tracking Id:**

In order to distinguish the requests in various interacting subsystems involved, the Execution Server in Admin will embed each request with an additional Tracking Id parameter before passing it down to the subsequent subsystems.

This Tracking Id along with the response collected from client Execution Servers will then be communicated back to the requestor.

Tracking Id Generation:

In order to generate a tracking Id, we would be leveraging [UUID type 4 (random)](https://en.wikipedia.org/wiki/Universally_unique_identifier#Version_4_(random)) implementation in Java.

**Authentication/Authorization:**

<Akash to add his section here>

**Assumptions/Constraints:**

1. If BMS on any of the intermediary nodes crashes, then the message would be considered lost and the job would have to be submitted again from the “Client App”.
2. If the Execution Agent dies for some reason unexpectedly (host crashes or any other reason), then the command it was running will be assumed as killed only. It is based on the same assumption as to what we have currently in the run\_manually.pl scenario.