

# Distributed Agent Challenge

## Requirements :-

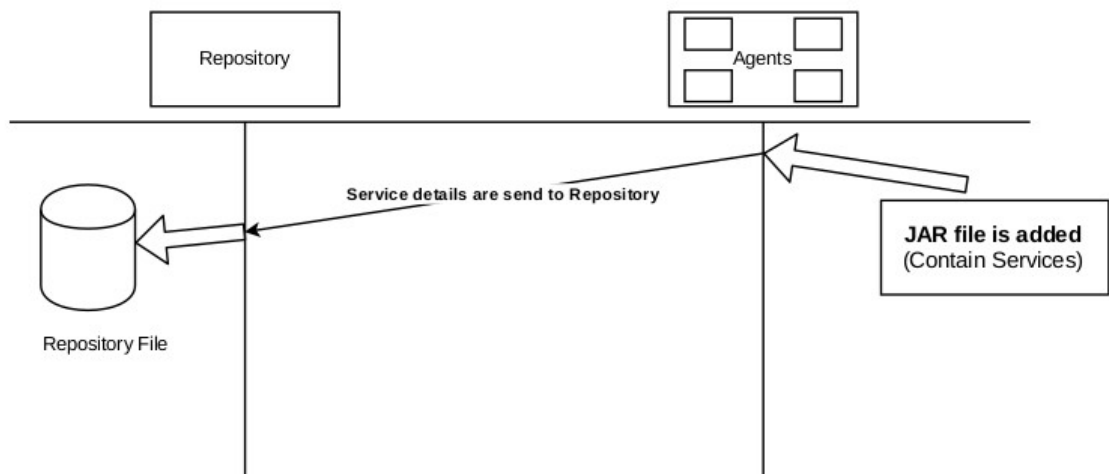
- Design and build an agent that can dynamically setup an artifact and execute a function or operation.
- There can be one super agent that can have the master/life cycle operations like load a service (Java?), execute the service and return the values back.
- The master agent will talk to the individual agents in different machines.
- Define the specific functionality you want, define the protocol and implement.

## Technologies Used :-

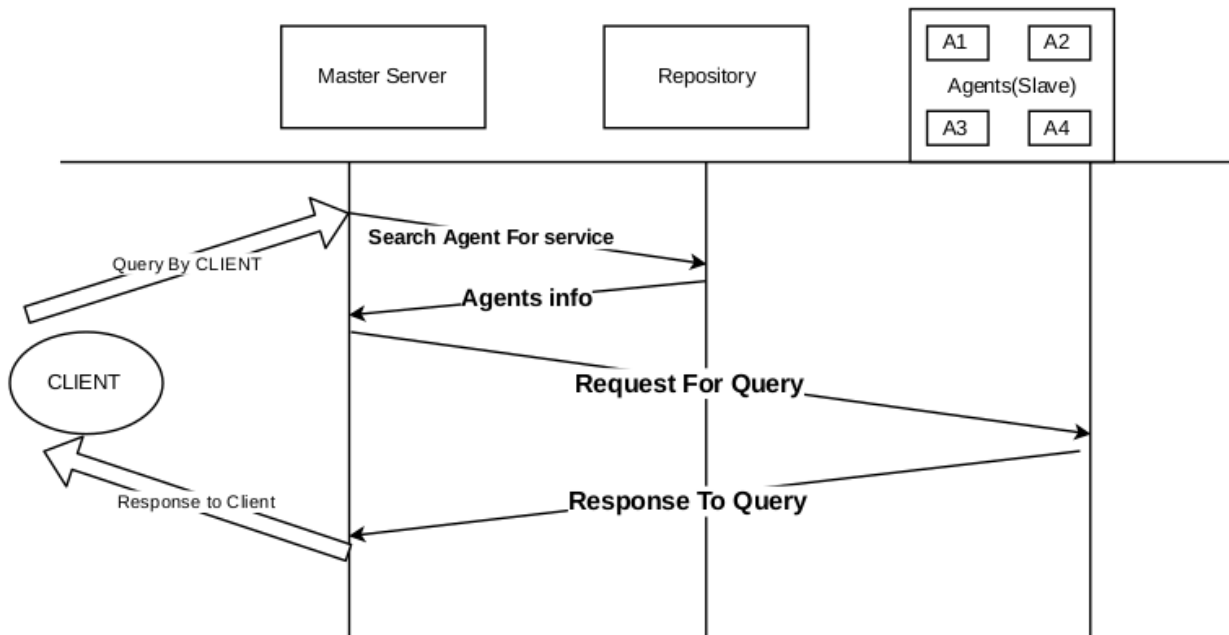
1. Rest API - jersey
2. Apache Tomcat v8.0 onwards
3. Java

## Architecture :-

- Master
  - The Master Server listen for requests from the end user.
  - This request for executing the service, is forwarded to the Repository Server to check if its a valid service request. Also, it gets the slave agent which would service the request.
  - Once it receives the response from the Repository Server, it invokes the slave and gets the response of the service. It sends the result of execution of the service back to the user.
- Repository Server
  - This Server holds the mapping of Service to SlaveAgents identity. i.e IP, Port of the agent and their corresponding services.
  - This server processes the request from the Master and provide the master with the information of agents that has particular requested service .
  - Registration of any new service is communicated by the slave agent to the repo server.
  - The Repo-Server depending upon the number of services can be dist



- Slave Agents -
  - One task for the Slave is to continuously poll a particular directory for any newly added Jar files (service) which contains implementation of the services.
  - Another task is to listens for service requests from the Master Server (corresponding to the user request) and executes the requested service.
  - Once executed, it send back the response back to the Repository Server which gets propagated back to the end user.



**FLOWCHART OF SYSTEM DEPICTING WORKFLOW**

---