**Work distribution**

**Description**

You will build a work distribution service that can distribute tasks to agents.

An agent is defined by a unique identifier, and a set of skills they possess.

A task is defined by a unique identifier, a priority, a set of skills an agent needs to possess to handle that task and the agent currently assigned to that task, if any.

The system will assign the tasks to the agents using the following rules:

* The agent must possess at least all the skills required by the task
* An agent cannot be assigned a task if they’re already working on a task of equal or higher priority.
* The system will always prefer an agent that is not assigned any task to an agent already assigned to a task.
* If all agents are currently working on a lower priority task, the system will pick the agent that started working on his/her current task the most recently.
* If no agent is able to take the task, the service should return an error.

The system will expose 2 endpoints:

* 1 endpoint that will allow creating a new task and distributing that task to an agent. This endpoint will accept a task object and will return that task, updated with the assigned agent if one was available.
* 1 endpoint that will allow updating a task to mark it as completed.

For simplification:

* the system will use a static list of agents that you will pref-define.
* The list of skills will be “skill1”, “skill2” and “skill3”
* There are only 2 priorities: low and high.

**Technical requirements:**

* Build an HTTP server using Node.js, Java, or Go (Golang) exposing JSON REST endpoints.
* The endpoints should return the appropriate HTTP status code.
* Your project should be hosted on GitHub and contain instructions to run it.
* You can choose to store the data in the database of your choice.

**Nice to have:**

* Unit tests
* Ability to start the application and all dependencies (database for example) with 1 command (for example using docker-compose)
* 1 endpoint that will return the list of agents with the tasks currently assigned to them if any.