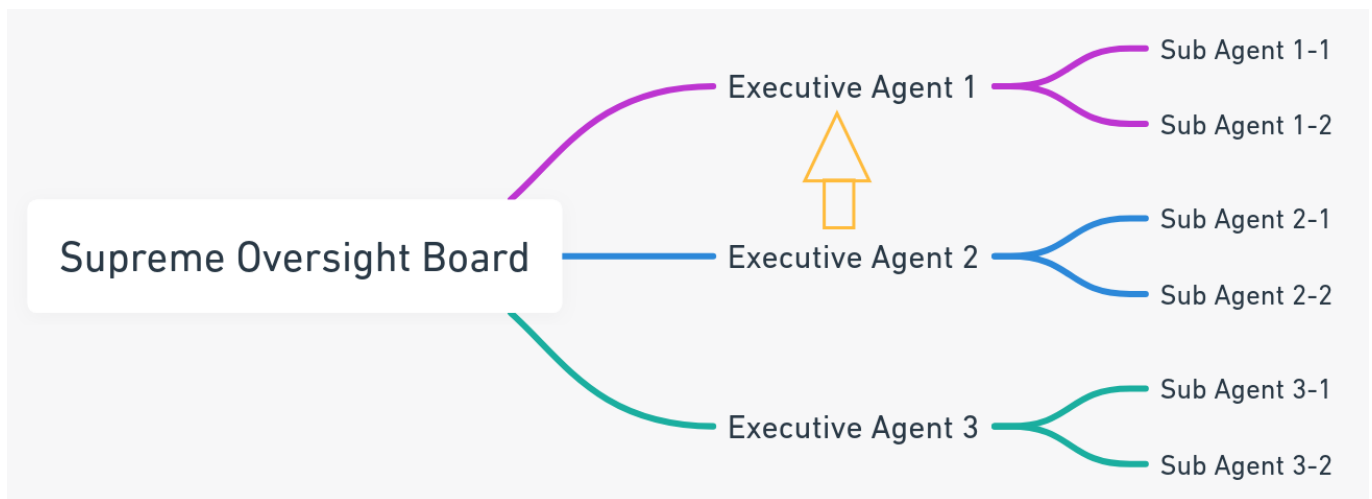


Overview

This project aims to create an AI agent swarm with an emphasis on orchestration and cohesion between them. The eventual goal being to have an autonomous system that can handle tasks accurately with minimal human oversight.

Structure

The overall structure of the project is centred on the communication between agents and orchestrating how they will interact. Using the experimental features of OpenAI's API the system attempts to limit the scope needed from each agent and as a result provide relevant context in order to increase accuracy and decrease costs from usage.



System Architecture

Executive agents

These agents are responsible for orchestrating and instructing sub-agents. Executive agents are responsible with analysis of the user's question of query, and subdividing the problem into separate steps of discipline. From that point this agent will do the following:

- create the sub-agents
- field responses from sub-agents, using threads
- summarise the sub-agent responses and return to the user
- answer user questions based sub-agent responses

Sub-Agents

These agents are created to be specialised and limited in scope. Rather than the administrator of a project they can be thought of as experts in their field. The hope with this approach being that they will be both more accurate and more efficient when giving answers on a given problem.

Instructions

Agents are given a set of instructions that outline their methodologies, goals, definitions of done, KPIs, and other operational directives.

Conversation Structure

Interactions with agents are structured in a conversational format, with user inputs leading to agent actions and responses.

Core technologies

The current implementation uses OpenAI's GPT models and API for the underlying agents. This provides the benefit of convenience as well as guaranteed future improvements as they release new models. However the goal is to create systems that allow interchangeability of different types of AI models to allow sub-models to be created that use domain specific pre-trained models which tend to be more accurate and far cheaper to operate.

Controlling Agents

The current goal is to fully implement a modular single tier orchestration system with control of agents only done through the Executive agent. Eventually the goal would be to scale this system and introduce an oversight board to coordinate tiers on different branches to allow for more direct control of individual agents.

Usage - tool creator + tool user

Tool Creation

Run the following commands to get started:

```
pip3 install -r requirements.txt

python -m agents.tool_maker.unit_manager
```

- From the `tool_creator` script:
 - chat with the bot about what you want the tool to do, and it will create the tool for you.
 - The tool will be saved in the `tools` directory with both the `.json` and `.py` files
 - The assistant will be saved in the `assistants` directory as `tool_creator.json`.

Tool Usage

- From the `tool_user` script:
 - The assistant will use all the tools in the `tools` directory.
 - Interact with the assistant in the chat to use the integrated tools.
 - The assistant will be saved in the `assistants` directory as `tool_user.json`.