Analytics Vidhya

Introduction to the Multi-Agent Pattern

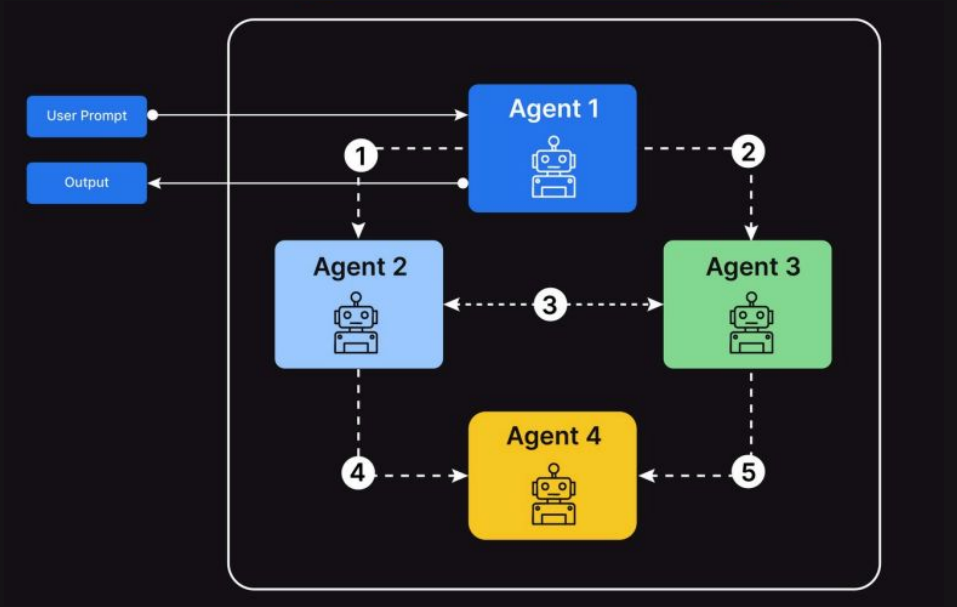
Created by:
Eleni Verteouri
Gen AI Tech Lead @ UBS

Created & Narrated by:
Dipanjan Sarkar
Head of Community & Principal AI Scientist @ Analytics Vidhya
Google Developer Expert - ML & Cloud Champion Innovator
Published Author

Analytics Vidhya

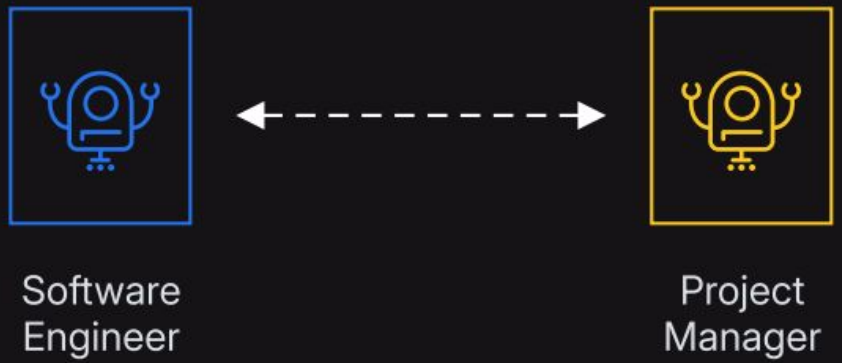
What is the Multi-Agent Pattern?

The **Multi-Agent Pattern** is a framework where complex tasks are broken down into **smaller subtasks, each handled by specialized agents** working collaboratively to achieve a common goal.



Analytics Vidhya

How does the Multi-Agent Pattern Work?

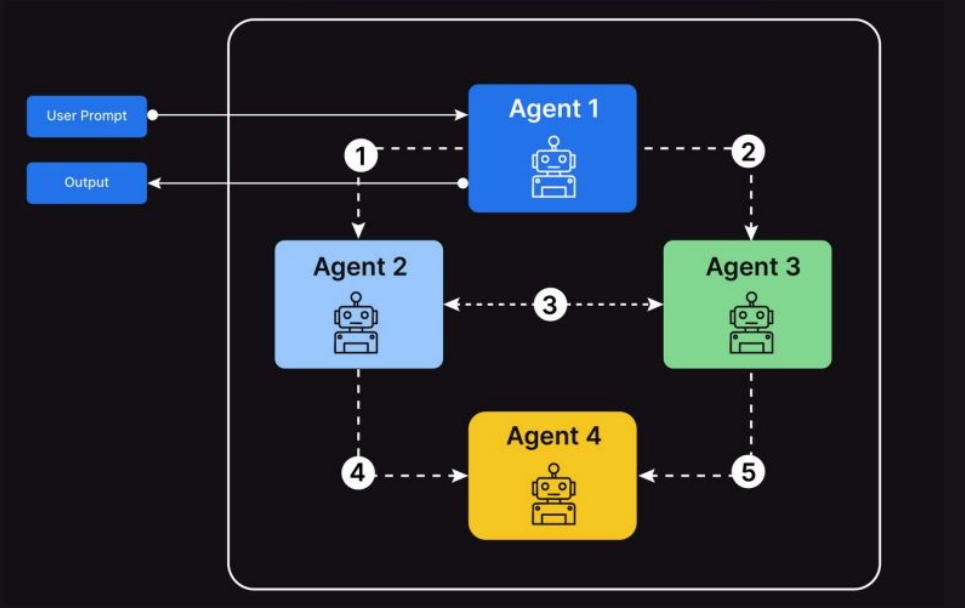


Task Division: Tasks are divided into subtasks and assigned to agents with specific roles

Example: One agent can be a software engineer, another a project manager, etc.

Why are Multi-Agent Systems Essential?

- A single agent-based system can face challenges where there are **too many tools to handle**, **too many specialized tasks to handle** and **context states start growing too large**.
- A multi-agent system has **several AI Agents that work together or independently** to solve a larger complex problem.



Analytics Vidhya

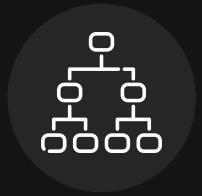
Frameworks for Multi-Agent Patterns

- **CrewAI:** Enables the creation of multi-agent applications.
- **AutoGen:** Implements variations of the Multi-Agent Pattern for task automation.
- **LangGraph:** Facilitates dynamic communication and collaboration between agents using stateful graphs.

Types of Architecture in Multi-Agents

Network Architecture

Supervisor Architecture



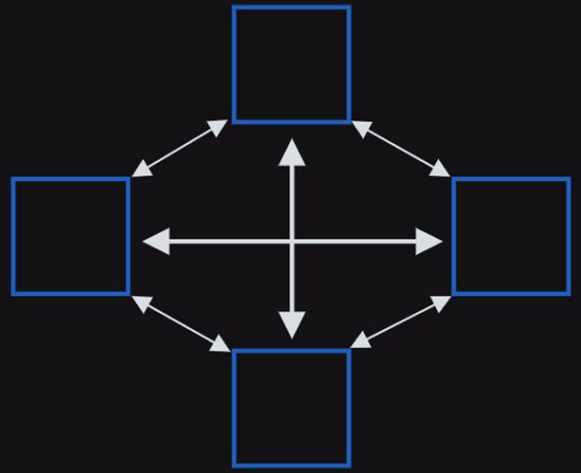
Hierarchical Architecture

Custom Architecture

Types of Architecture in Multi-Agents

Network Architecture:

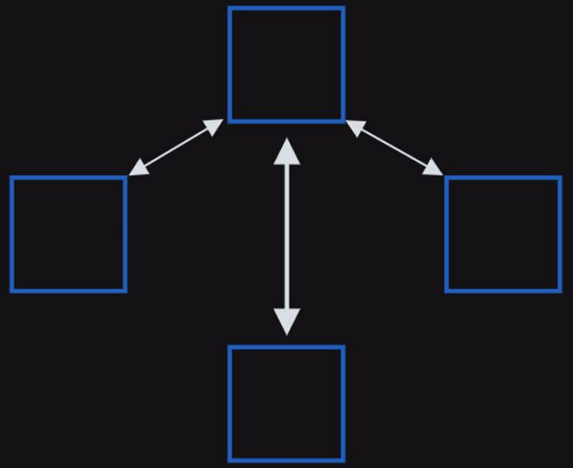
- Each agent can **communicate** with **every other agent**.
- Any agent can decide which other agent to call next.
- Typically also known as **collaborative multi-agent systems**.



Types of Architecture in Multi-Agents

Supervisor Architecture:

- We first build separate agents (known as **sub-agents**) to tackle a specific set of tasks.
- We can then choose to use an **LLM to orchestrate** the different agents.
- Each sub-agent communicates with a **single supervisor agent**.
- Supervisor agent makes decisions on which agent should be called next.

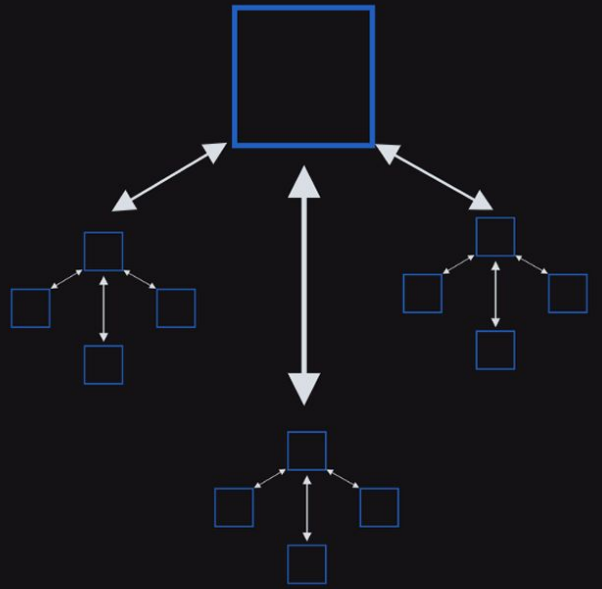


Analytics Vidhya

Types of Architecture in Multi-Agents

Hierarchical Architecture

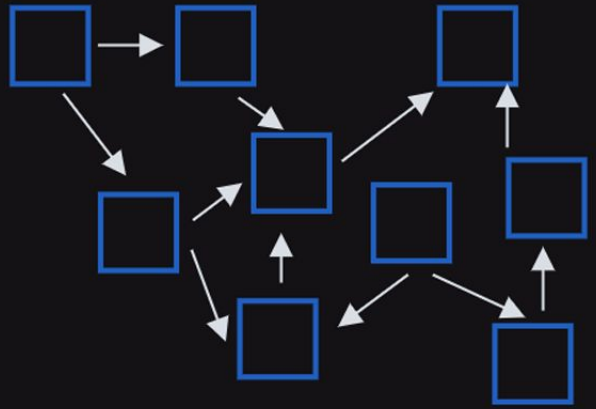
- Think of it as **multiple supervisory** multi-agent systems with a **primary supervisor overseeing them**.
- Useful when tasks are too complex to be just routed to single sub-agents.
- Specific sets of tasks are handed to specific supervisor agents.
- The **supervisor of supervisors controls the overall delegation and flow**.



Types of Architecture in Multi-Agents

Custom Architecture:

- Each agent **communicates with only a subset of agents**.
- You can **combine** different architectures and patterns based on your use-case.
- Some parts could also be deterministic AI workflows.



Benefits of Multi-Agent Patterns

Modularity

Separate agents make it easier to develop, test, and maintain agentic systems.

Specialization

You can create expert agents focused on specific domains, which helps with the overall system performance.

Control

You can explicitly control how agents communicate.

Thanks!

Analytics Vidhya