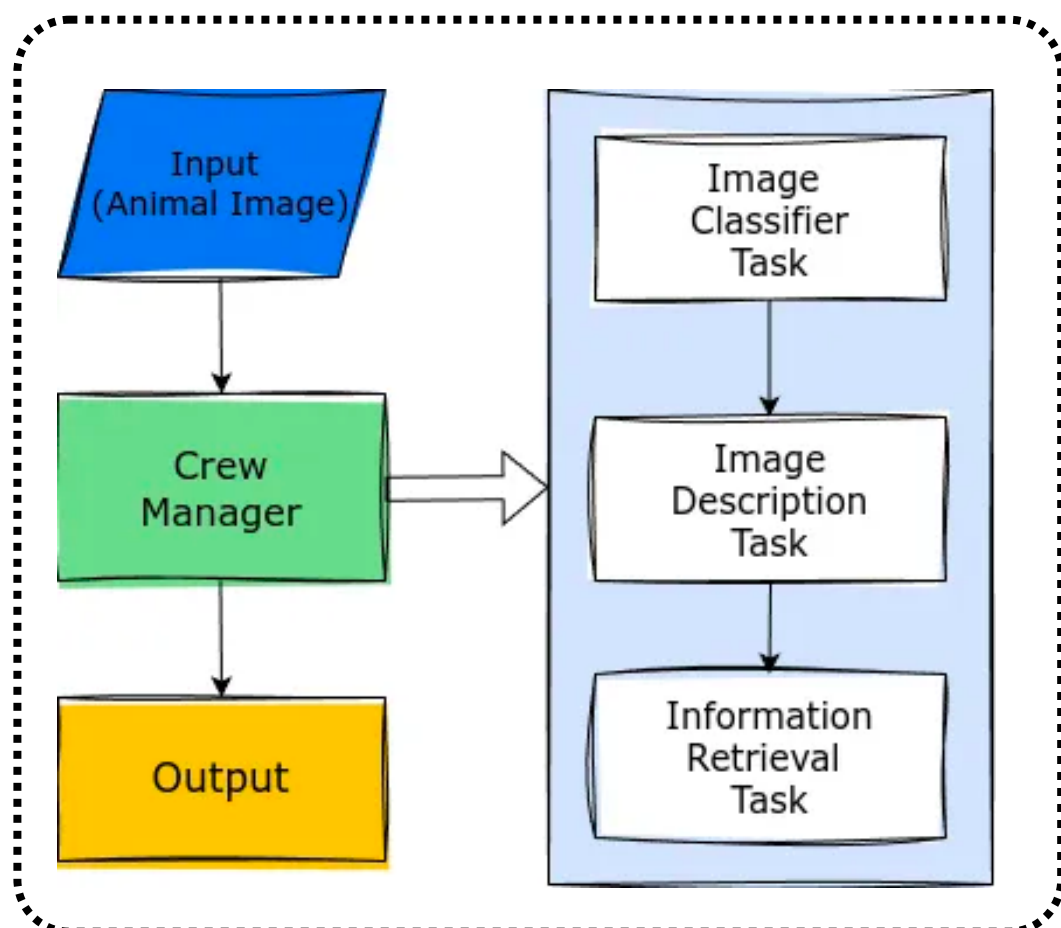
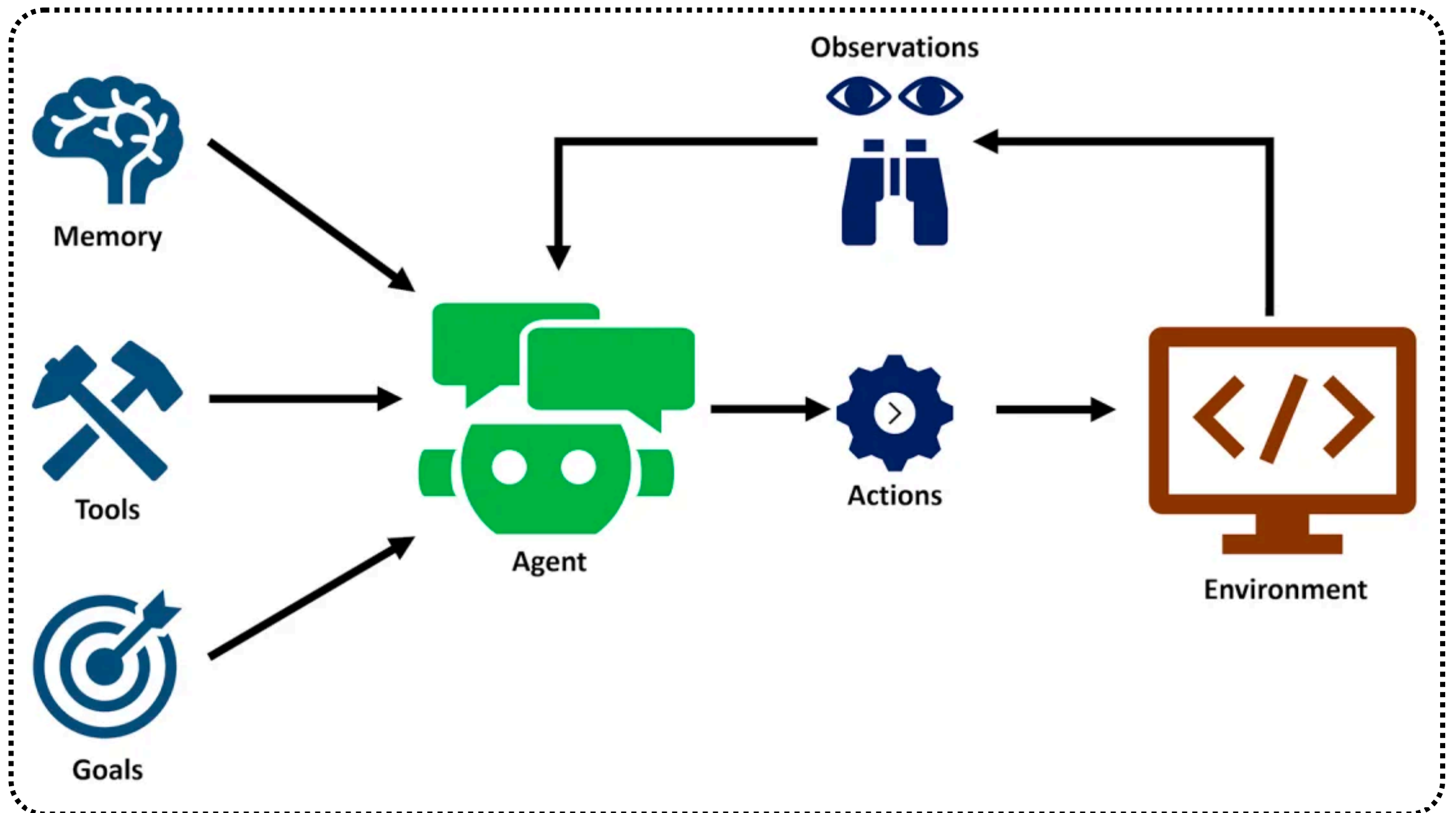
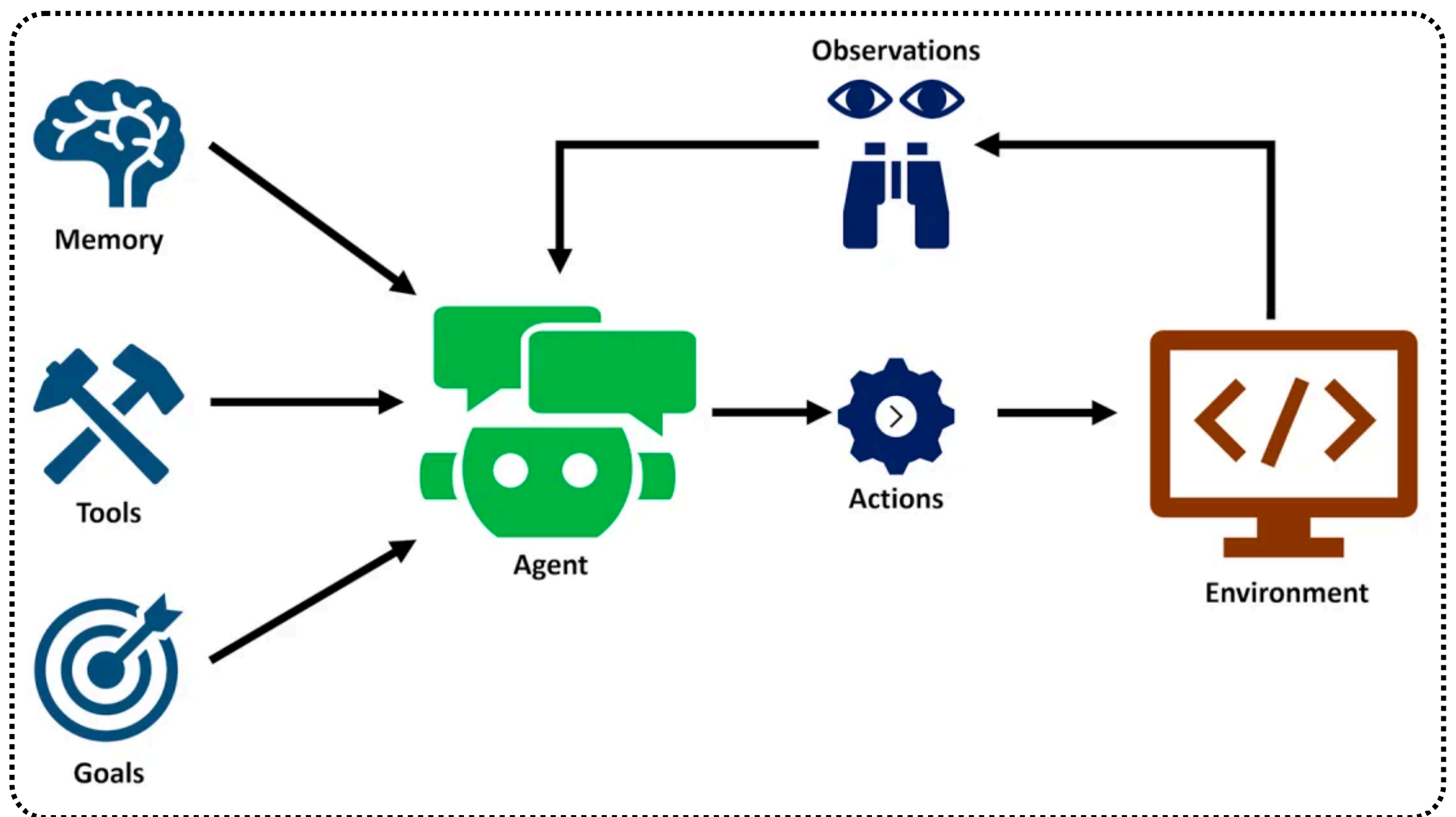


# How to Build Multi-Agent System with CrewAI and Ollama?



# Agents, Agentic Frameworks, and CrewAI

Generative AI has transitioned from basic large language models (LLM) to advanced multi-agent systems. In theory, Agents are autonomous systems capable of planning, reasoning, and acting without human input. These agents aim to reduce human involvement while expanding functionality.



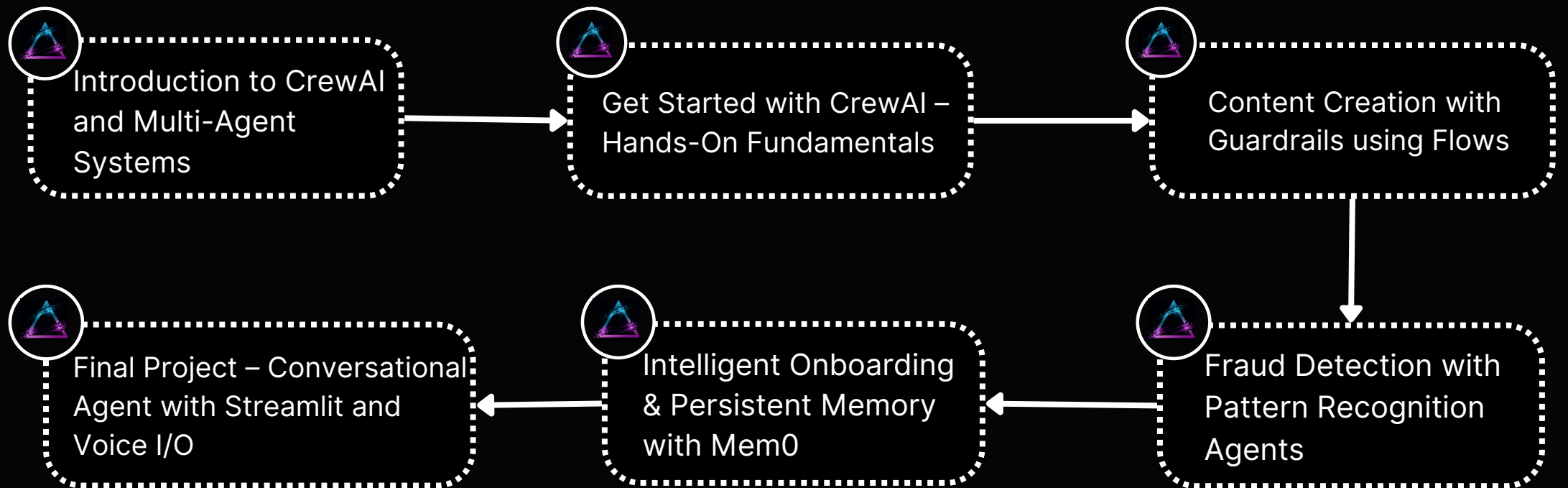
Moreover, if you are looking forward to learn more about “Multi-Agent Applications with CrewAI” we are conducting a workshop at DataHack Summit 2025.





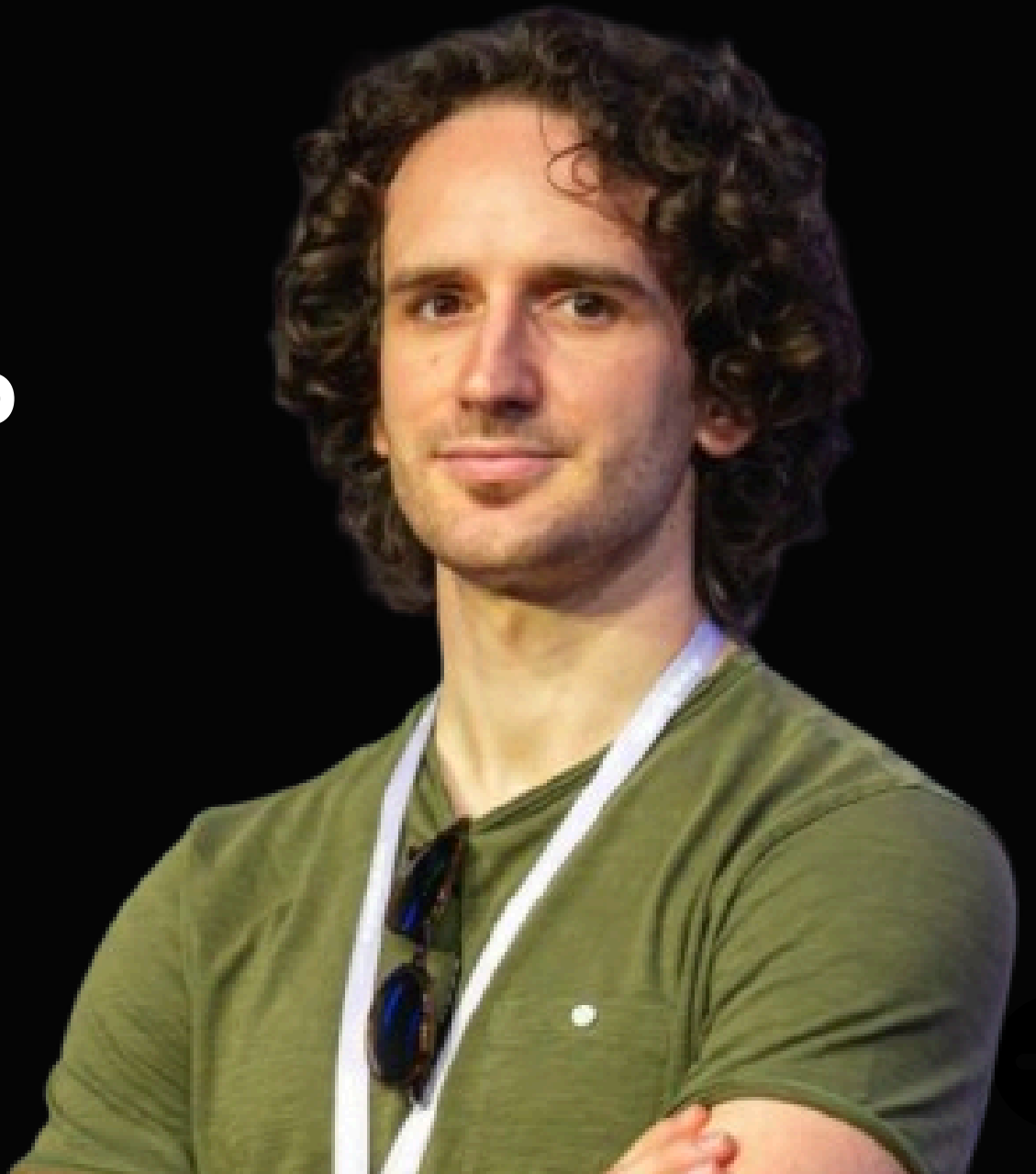
# Workshop Announcement

## Unleashing Multi-Agent Applications with



**Alessandro Romano**

Senior Data Scientist



# Agentic Frameworks

These frameworks utilize multiple agents working in concert, allowing for collaboration, communication, and problem-solving that exceed the capabilities of single-use agents. In these frameworks, agents have distinct roles and goals and can perform complex tasks.

Multi-Agent Framework such as CrewAI and Ollama are essential for large-scale, dynamic, and distributed problem-solving, making them adaptable across industries like robotics, finance, healthcare, and beyond.

## Key Components of Agentic Frameworks

- **Agent Architecture:** Defines the internal structure of agents, including planning, reasoning, and communication protocols.
- **Communication Protocols:** Methods for agent collaboration through messaging and data exchange.
- **Agent Interaction Design:** Mechanisms for agent collaboration, including task allocation and conflict resolution.
- **Environment:** The setting where agents interact, often including external tools and resources.



# CrewAI Framework

crewAI is an advanced multi-agentic framework, enabling multiple agents (called a “crew”) to collaborate through task orchestration. The framework divides agents into three attributes—role, goal, and backstory—ensuring a thorough understanding of each agent’s function. This structured approach mitigates under-specification risk, improving task definition and execution.

## Key Strengths of CrewAI

- **Explicit Task Definition:** Tasks are well-defined, ensuring clarity in what each agent does.
- **Tool Use:** Task-specific tools precede agent-level tools, creating a more granular and controlled toolset.
- **Agent Interaction Processes:** crewAI supports sequential and hierarchical agent collaboration processes.
- **Advanced Memory Management:** The framework provides short-term, long-term, entity, and contextual memory, facilitating sophisticated reasoning and learning.





# Ollama

Ollama is a framework for building and running language models on local machines. It's easy to use, as we can run models directly on devices without needing cloud-based services. There's no concern about privacy.

To interact with Ollama:

We can run the `pip install ollama` command to integrate Ollama with Python.

Now, we can download models with the `ollama pull` command to download the models.

Let's run these:

```
ollama pull llama2
```

```
ollama pull llama3
```

```
ollama pull llava
```



Now, we have 3 of the Large Language Models (LLMs) locally:

1. **Llama 2**: An open-source large language model from Meta.
2. **Llama 3**: The latest iteration of Meta's Llama series, further refining capabilities for complex language generation tasks with increased parameter size and efficiency.
3. **LLaVA**: A vision-language model designed for image and text understanding tasks.

We can use these models locally by running `ollama run model-name`, here's an example:

```
(base) mounsih@avadmin-ThinkPad-E14:~$ ollama run llama3
>>> Explain stable diffusion in brief
Stable Diffusion is a type of image generation algorithm that uses a
process called "stable training" to improve the quality and stability of
generated images. Here's a brief overview:

**Key Idea:** Instead of optimizing for similarity to a target image (like
traditional generative models), Stable Diffusion optimizes for the
likelihood of the input prompt under the generated image.

**How it works:**

1. The algorithm starts with an initial image.
2. It then iteratively applies a series of transformations (e.g., affine
transformations, noise injection) to the image while maintaining its
stability and diversity.
3. At each step, the algorithm evaluates how well the transformed image
matches the input prompt.
4. It adjusts the transformation parameters based on this evaluation to
improve the likelihood of generating an image that satisfies the prompt.

**Benefits:**

1. **Stability:** Stable Diffusion produces more stable and coherent
images by optimizing for stability rather than similarity to a target
image.
2. **Diversity:** The algorithm can generate diverse and novel images, as
it explores different transformations and combinations.
3. **Improved quality:** By focusing on likelihood of the prompt, Stable
Diffusion can produce high-quality images that accurately represent the input text.
```



# Building a Multi-Agent System

Let's work on building an Agentic system that takes an image as an input and gives a few interesting facts about the animal in the system.

## Objectives

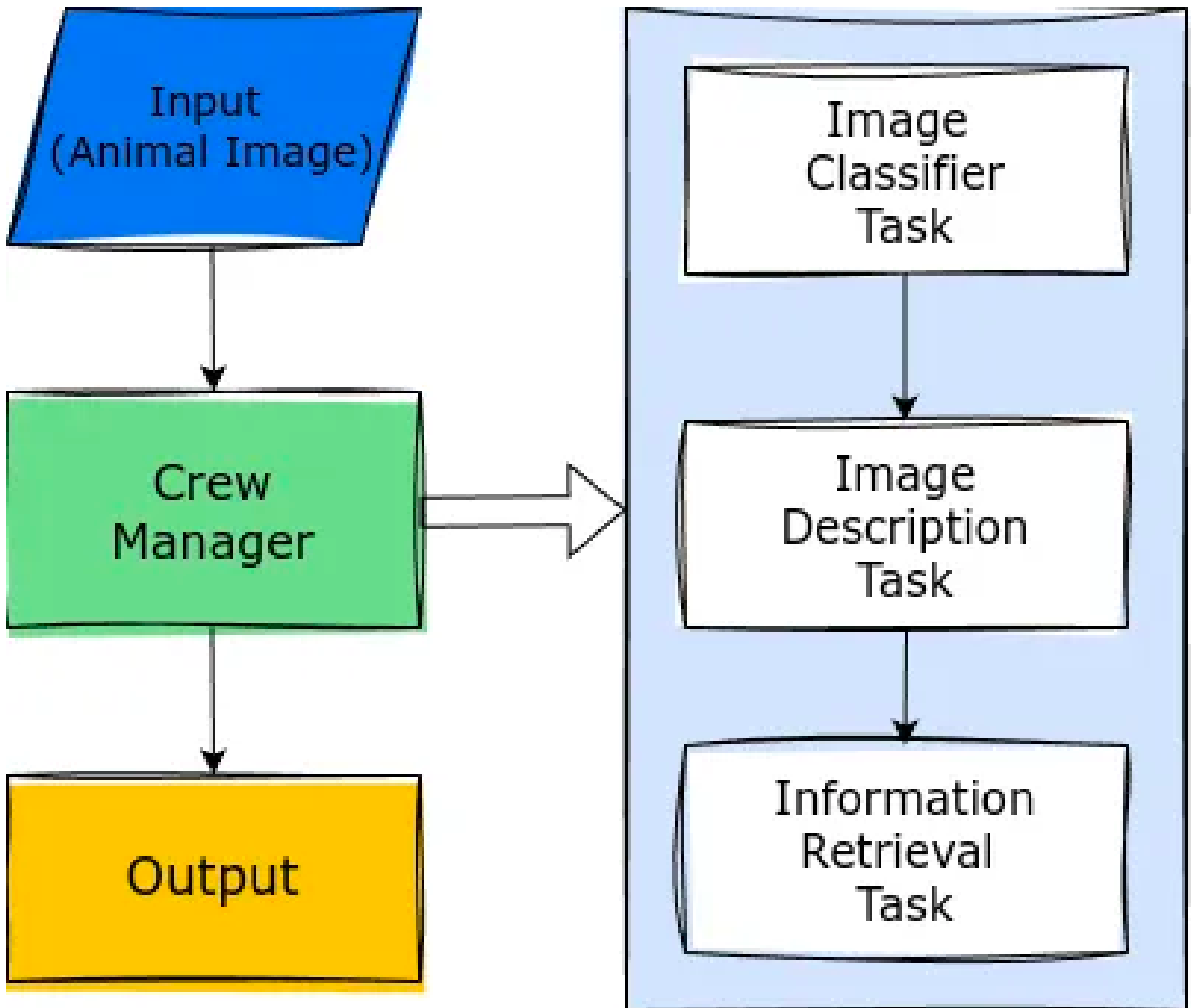
1. Build a multi-agent system for image classification, description, and information retrieval using CrewAI.
2. Automate decision-making: Agents perform specific tasks like identifying animals in images, describing them, and fetching relevant facts.
3. Task sequencing: Coordinate agents through tasks in a stepwise, agentic system.

## Components

1. **Classifier Agent:** Identifies whether the input image contains an animal using the llava:7b model.
2. **Description Agent:** Describes the animal in the image, also powered by llava:7b.
3. **Information Retrieval Agent:** Fetches additional facts about the animal using llama2.
4. **Task Definitions:** Each task is tied to a specific agent, guiding its action.
5. **Crew Management:** The Crew coordinates agent actions, executes tasks, and aggregates results based on the input image







By default, tasks are executed sequentially in CrewAI. You can add a task manager to control the order of execution. Additionally, the `allow_delegation` feature allows an agent to ask its preceding agent to regenerate a response if needed. Setting memory to `True` enables agents to learn from past interactions, and you can optionally configure tasks to ask for human feedback about the output.



For more information, please visit this [article](#)



Advanced

Generative AI

Large Language Models

LLMs

## How to Build Multi-Agent System with CrewAI and Ollama?

Build agents and Multi-Agent System with CrewAI and Ollama, using local LLMs like Llama2, Llama3, and LLaVA.

Mounish V 12 Oct, 2024

