AGENTIC AI

WEEK 3

Exploring CrewAI: Autonomous Agents for Complex Tasks

Presentation Agenda

01

CrewAI Overview

Understanding the core offerings and philosophy.

02

Key Concepts & Terminology

Agents, Tasks, and Crews explained.

03

Project Structure & Workflow

How to create and manage CrewAI projects.

04

Memory Management

Exploring different types of memory in CrewAI.

05

Real-World Applications

Examples of CrewAI in action.

What is CrewAI?

CrewAI is a lean, lightning-fast Python framework built entirely from scratch, completely independent of Langchain or other agent frameworks. It offers two main approaches for orchestrating high-performance AI agents with ease and scale:

- **CrewAI Crews & Autonomous Solutions:** Teams of AI agents with different roles for autonomous problem-solving, creative collaboration, or exploratory tasks.
- **CrewAI Flows:** Structured automations by breaking down complex tasks into precise workflows and fixed workflows.

CrewAI is ideal when you need auditable outcomes and precise control over execution.

CrewAI Offerings

Crew Enterprise

A multi-agent platform for deploying, running, and monitoring AI solutions.

GensAI UI Studio 7

A no-code/low-code product for creating multi-agent solutions.

CrewAI Open-Source Framework

Orchestrating high-performance AI agents with ease and scale. This is the core framework.

These offerings provide flexible options for various user needs, from enterprise-level deployment to open-source development.

Key Concepts in CrewAI

Agent

An autonomous unit with a role, a goal, a backstory, and memory. Agents can be created by code or defined in a YAML file.

```
agent: Financial Researcher
goal: Research companies, news, and market info.
backstory: Expert in financial analysis.
llm: gemini-2.0-flash
```

Task

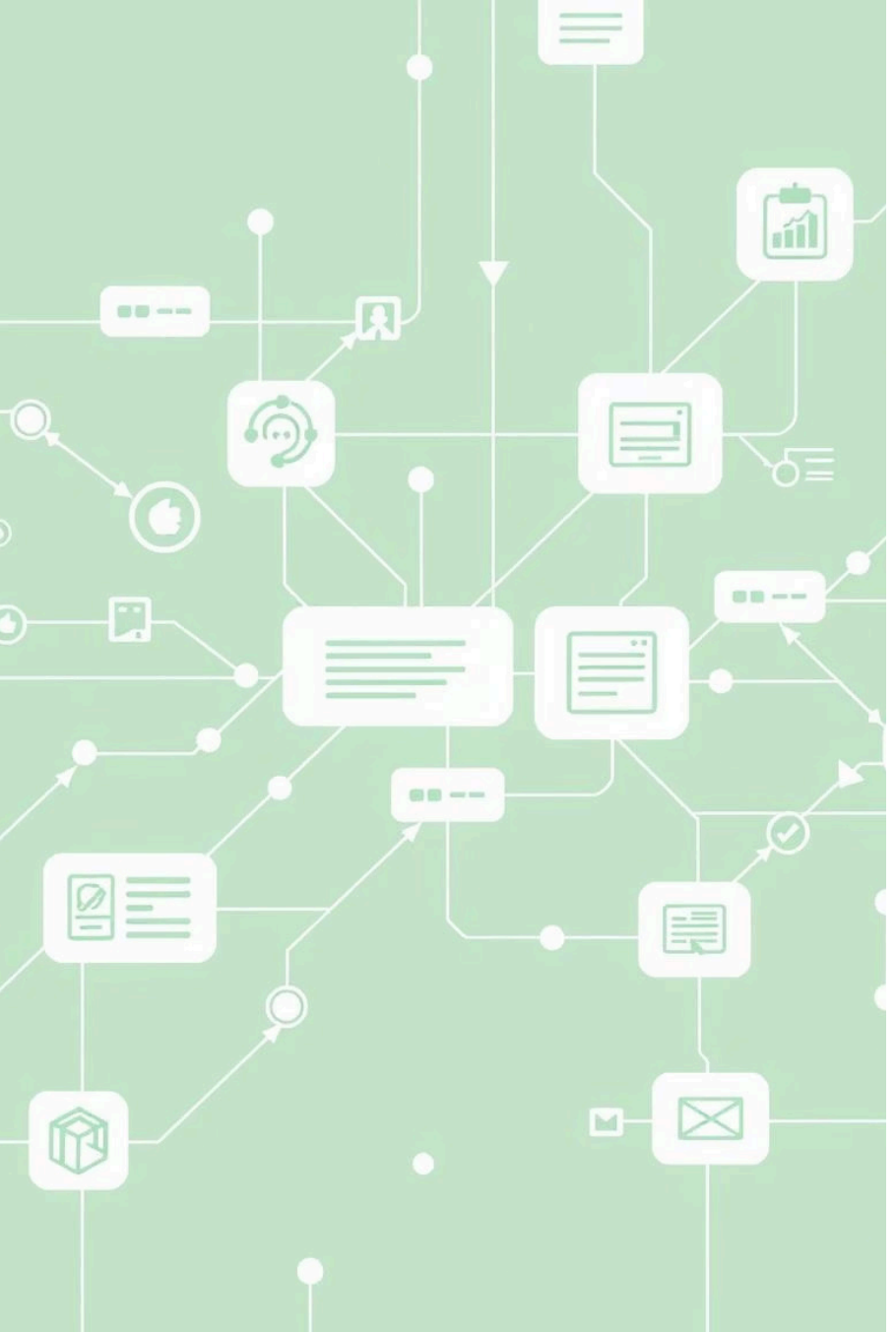
A specific assignment to be carried out, with a description, expected output, and assigned agent.

Crew

A team of Agents and Tasks. Crews can be:

- **Sequential:** Run tasks in the order they are defined.
- **Hierarchical:** Use a manager agent to assign tasks.

CrewAI uses the super-simple LiteLLM under the hood to interface with almost any LLM.



Creating a New CrewAI Project

Follow these steps to set up and run your CrewAI project, ensuring a structured and efficient development process.

1. **Create the project:** Use the command `crewai create crew my-project` to generate a new directory with subdirectories.
2. **Define Agents and Tasks:** Fill in the `config.yaml` file to define your agents and tasks.
3. **Implement the Crew:** Complete the `crew.py` module to define the Agents, Tasks, and crew, referencing the config.
4. **Run the project:** Update `main.py` to set any additional configurations and run your crew.

> For LLM API keys, use Serper.dev, a free open-source platform for search at an unbeatable price. Copy your API key from Serper and put it in your environment file.

Memory Management in CrewAI

CrewAI supports various types of memory to enhance agent performance and enable more complex interactions.

Temporary Memory

Stores more recent information and outcomes, allowing agents to access relevant information during the current execution.

Long-Term Memory

Preserves valuable insights and learning, building knowledge over time for more informed decision-making.

Entity Memory

Stores information about people, places, and concepts encountered during tasks, facilitating deeper understanding and relationship mapping. Use RAG for context.

Memory is added with `crew.py` file code execution set to `True` for the current project.

Project Example: Stock Picker

This project demonstrates how CrewAI can be used to create a structured financial research and stock picking solution.

Agents:

- **Financial Researcher:** Gathers company, news, and market information.
- **Stock Analyst:** Analyzes data and identifies potential stocks.

Workflow:

- Researcher collects data.
- Analyst processes data and recommends stocks.

This project leverages CrewAI's ability to orchestrate complex tasks into precise workflows, providing auditable outcomes.

Project Example: Engineering Team

Agents:

- **Engineer Lead:** Oversees the project and assigns tasks.
- **Backend Engineer:** Develops server-side logic.
- **Frontend Engineer:** Builds user interfaces.
- **Test Engineer:** Ensures code quality and functionality.

This project showcases a hierarchical process where a manager agent assigns tasks, and agents use memory to maintain context and build knowledge.

Key Takeaways

- **CrewAI is a powerful, independent framework** for building autonomous AI agents.
- **Flexible for various needs:** from autonomous problem-solving to structured automations.
- **Robust memory management** enhances agent intelligence and context awareness.
- **Simple project creation** and integration with LLMs.
- **Versatile applications** across industries like finance and software development.

Unleash the power of AI with CrewAI!