Chapter 2: What are AI Agents

The IT department of every company is going to be the HR department of AI agents in the future

- Jensen Huang

As major companies like Meta and Salesforce have noted, 2025 is shaping up to be the era of AI agents. Building on this momentum, this chapter explores what AI agents are, how they function, and their growing impact across industries.

2.1 What are Agents?

In Generative AI, an agent is like a virtual assistant or a digital entity designed to perform specific tasks, make decisions, or solve problems. It usually uses a combination of LLM + some third-party APIs, databases, or even special functions to complete a task. It can also be enabled for task execution and decision-making capabilities.

2.1.1 AI Agents vs LLMs?

To better understand the distinction between AI agents and large language models (LLMs), let’s break it down with a practical example: email generation.

Email Generation Using an LLM

When you use an LLM to generate an email, here’s what happens:

You Provide a Prompt:

For example, you might ask the LLM:

“Write an email thanking a client for a meeting and attach a proposal.”

LLM Generates Text:

The LLM produces the following output:

```
Subject: Thank You for the Meeting

Dear [Client Name],

I wanted to take a moment to thank you for the time you took to meet with us today. It was a
pleasure discussing [specifics of the meeting], and we're excited about the opportunity to w
ork together.

As promised, please find attached the proposal for your review. If you have any questions or
need further clarification, don’t hesitate to reach out.

Looking forward to hearing from you.

Best regards,
[Your Name]
```

Now, as you must have noticed, it just generates the text. A lot of work is still pending for you to

1. *Copy the email text.*
2. *Open your email client.*
3. *Fill in the recipient’s details.*
4. *Attach the proposal manually.*
5. *Click send.*

Using Email Agent : The email agent will not just output the text but also do the remaining steps (if enabled) to complete the task.

That’s not the case with AI Agents …

AI agents go beyond just generating content—they execute tasks by interacting with the tools they’re connected to. For example, a scheduling agent doesn’t just suggest meeting times; it connects with your calendar API, finds available slots, and books the meeting automatically. The real power of AI agents lies in their ability to bridge the gap between generating ideas and turning them into actions, all without human intervention.

2.1.2 How Do AI Agents Execute Tasks?

While AI agents internally use LLMs, their ability to interact with APIs, databases, or other tools is enabled through customized code. Here’s how this interaction typically works:

Example: Weather Agent

User Query: “How is the weather in New Delhi?”

Agent’s Workflow:

1. **Step 1:** The LLM within the agent analyzes whether a tool (e.g., weather API) is required. It may use a system prompt like: *“Check whether the user is asking about the weather of a specific place.”*
2. **Step 2:** If a tool is required, the agent extracts the relevant entity (e.g., “New Delhi”) from the user’s prompt.
3. **Step 3:** A request is sent to the weather API (tool) with the extracted entity as a parameter.
4. **Step 4:** The API returns data (e.g., temperature, humidity), which the LLM uses as context to generate a final answer.
5. **Step 5:** The agent delivers the response: *“The weather in New Delhi is 28°C with moderate humidity.”*

This process of letting the LLM decide when to use a specific function or service is known as **tool calling**. It’s like giving the agent access to a toolbox and letting it pick the right tool based on the task at hand—whether it’s calling a weather API, searching the web, sending an email, or querying a database. Tool calling is what allows the agent to move from just generating ideas to actually performing real actions.

Some examples of AI agents

Agent Type	Description
Weather Agent	LLM + Weather API to fetch real-time weather data.
Perplexity AI	LLM + Internet access for real-time information retrieval.
Customer Support	LLM + Database access to provide personalized responses.
Spanish Agent	LLM + Special internal prompt to answer queries only in Spanish.

2.1.3 Why Are AI Agents Important?

AI agents significantly enhance the utility of LLMs, transforming them from text-generation tools into intelligent assistants. Here’s why they matter:

1. **Task Automation:** Agents automate complex, multi-step workflows, going beyond simple content generation.

2. **System Integration:** They connect AI to external tools (e.g., APIs, calendars) to perform real-world actions.
3. **Decision-Making:** Agents can autonomously choose and execute tasks without constant user input.
4. **Real-Time Action:** Agents interact with live data, making dynamic decisions and updating in real time.
5. **Personalization:** Agents remember user preferences and adapt based on past interactions.
6. **Reduced Human Intervention:** Agents handle tasks end-to-end, minimizing manual effort.
7. **Scalability:** Agents can process repetitive, large-scale tasks efficiently, freeing up human resources.

2.2 Popular AI Agent Frameworks

Since the AI agent hype, there have been a number of AI agent frameworks that have been released on the internet, which you can't even trace out. To ease things up for you, these are the famous and the most popular AI agent frameworks that you should know:

1. **LangGraph:** Developed by LangChain, this framework enables building stateful, graph-based AI agent systems. It specializes in handling cyclic workflows and multi-agent coordination with persistent memory. Ideal for complex applications like automated research, customer service bots, and long-running decision processes where agents need to maintain context.
2. **CrewAI:** A Python-based framework focused on creating specialized AI teams where each agent has defined roles (e.g., analyst, writer, reviewer). It features built-in task delegation, parallel processing, and easy integration with external tools. Perfect for content creation pipelines, data analysis workflows, and other structured multi-step processes.
3. **AutoGen:** Microsoft's framework for creating conversational AI systems with advanced multi-agent capabilities, specific to coding tasks. Supports customizable LLM backends, tool integration, and complex dialog patterns. Commonly used for developing

sophisticated chatbots, coding assistants, and collaborative problem-solving systems.

4. **Google-ADK** (Agent Development Kit): Google's comprehensive toolkit for building enterprise-grade, multimodal AI agents. Offers tight integration with Google Cloud services and specialized APIs for handling real-time data streams. Designed for large-scale deployments in business automation, smart assistants, and data processing applications.
5. **OpenAI Agent SDK**: A lightweight development kit for creating function-calling AI agents powered by OpenAI models. Simplifies connecting LLMs to external APIs, databases, and tools. Best for building responsive agents handling tasks like information retrieval, data processing, and simple automation workflows.
6. **Microsoft TinyTroop & Magentic-One:** Two specialized frameworks - TinyTroop simulates interactions between multiple AI personas for testing scenarios, while Magentic-One focuses on reliable code generation. Together, they support development of simulation environments and AI-powered developer tools with an emphasis on stability.
7. **HuggingFace smolAgents:** A collection of lightweight, modular agent components optimized for small-scale AI applications. Enables rapid prototyping and experimentation with HuggingFace models. Particularly useful for researchers and developers working on constrained hardware or needing simple agent implementations.
8. **AWS Multi-Agent Orchestrator**: Amazon's cloud-native solution for coordinating distributed AI agents at scale. Integrates seamlessly with AWS services like Lambda and Bedrock to manage complex workflows. Tailored for enterprise applications in logistics, IoT systems, and large-scale automation projects.

We won't be discussing these frameworks in detail in this book. Why?

Think of them as early versions—AI agents 1.0. With the emergence of the Model Context Protocol (MCP), we’re on the cusp of a new wave of AI agent development. This book aims to prepare readers for the next generation of AI agents—those powered by MCP—by

building a strong foundational understanding of the evolving landscape.

2.3 Potential Challenges with AI Agents

While AI agents offer incredible potential, some challenges come with their implementation.

1. First, agent systems depend heavily on the ***reliability of external tools***, such as APIs or databases, which means any downtime or error from those tools can cause the agent to fail.
2. Secondly, agents often need ***careful design*** to ensure that their decision-making abilities align with user expectations, particularly in critical tasks like financial transactions or sensitive data handling.
3. Lastly, maintaining ***security and privacy*** when agents access multiple systems is another important consideration.
4. A ***lot of coding*** and usage of multiple frameworks is required to build out complex systems, making it completely unreachable for non-programming folks.
5. ***No standardization*** is present if you are using AI agent frameworks. Every framework works in a different manner, and hence a lot of segmentation.

Concluding, AI agents look to be a promising concept, and if you've been listening to news, you might have heard of this concept way before. Though there are certain lacunae present in this concept, it's still very useful. With the incoming Model Context Protocol (MCP) that we'll discuss in the next chapter, you'll know how MCP is the AI agent 2.0.