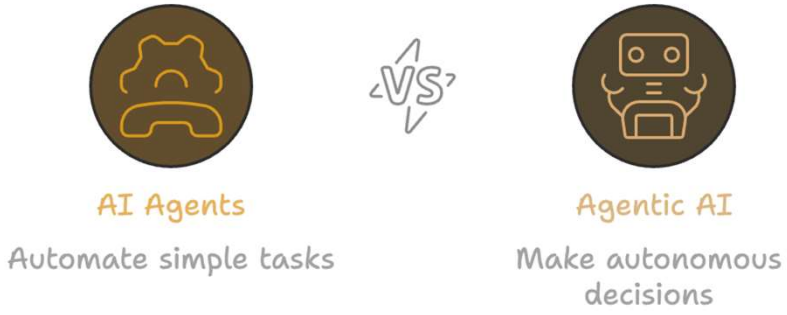
Introduction to Agentic AI and Autonomous AI Agents



What is Agentic AI?

What is an AI Agent?

A system that does more than just process information

- Takes actions to achieve a goal

Key abilities

- Observe: It perceives its environment and gathers information
- Reason: It thinks and makes a plan based on its goal and observations
- Act: It uses tools or APIs to execute its plan in the environment

AI Agent vs Agentic AI

AI Agent

- Operates based on a pre-defined set of rules it has learned from historical data

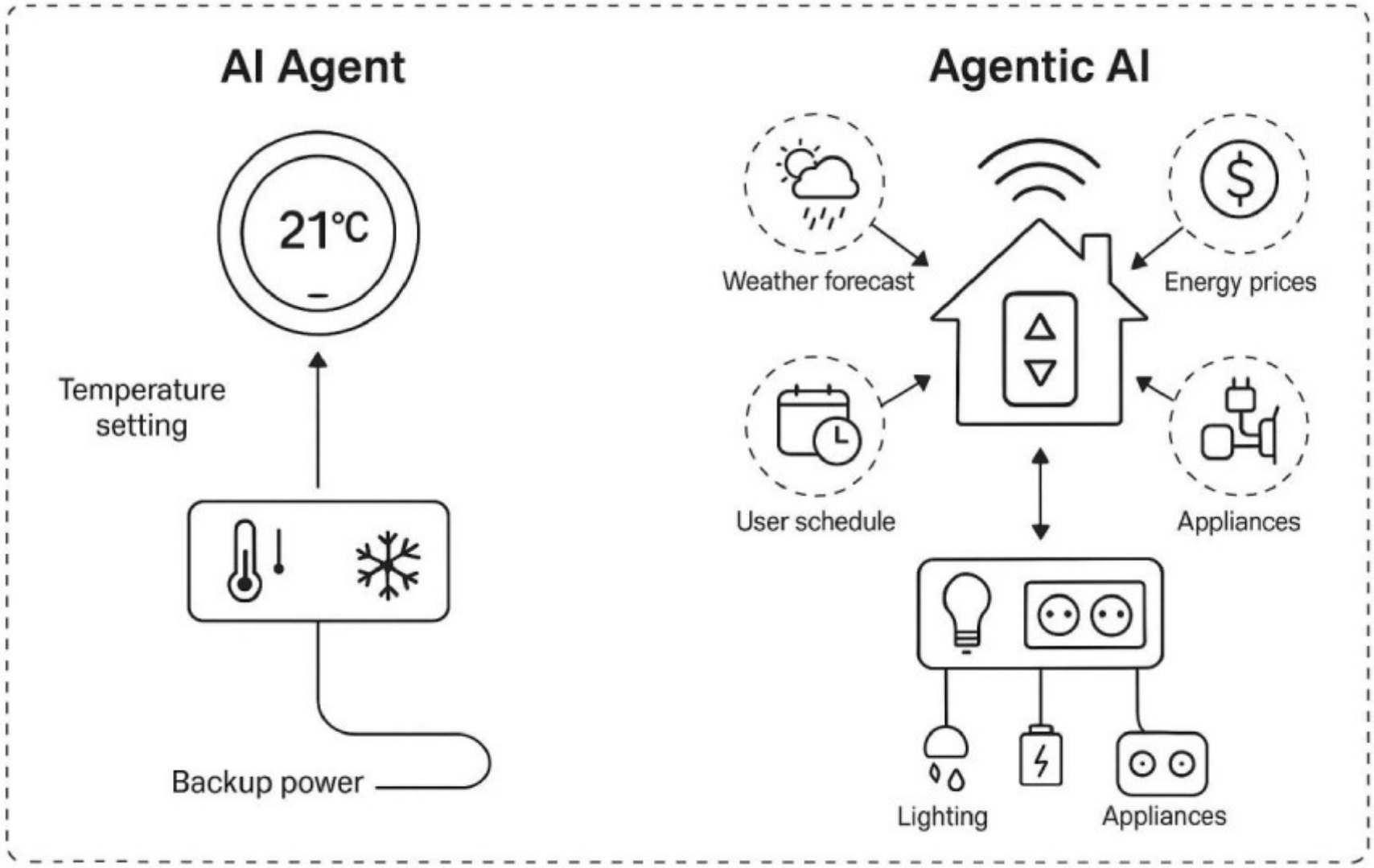
Agentic AI

- More advanced system that can reason, plan, and adapt dynamically

An AI Agent executes a plan given to it

Agentic AI creates and adapts its own plan

AI Agent vs Agentic AI



Components of Agentic AI

Perception

- Gathers and interprets data from the environment

Reasoning & Planning

- Processes information, makes decisions, and formulates actions to achieve goals

Memory

- Stores and retrieves knowledge, context, and past interactions

Action & Tool Use

- Executes tasks and interacts with external systems or tools

Learning & Adaptation

- Continuously improves performance through feedback and experience

Orchestration

- Coordinates and manages the interaction of multiple AI agents within the system

What does it mean by - Autonomous?

The ability to operate and complete tasks without direct human control

- For every single step

Not Autonomous (Remote-controlled car)

- A human makes every single decision: turn left, accelerate now, brake now

Autonomous (Self-driving car)

- A human provides only the high-level goal ("Drive to the supermarket")
- The car then makes all the smaller, intermediate decisions on its own—when to switch lanes, when to stop at a light, how fast to go, etc

Components of AI agents

Memory

- To store and recall information

Reasoning

- Cognitive engine
- Powered by an LLM

Planning

- Ability to break down a complex into a sequence of smaller steps

Action

- To interact with its environment by executing specific tasks

Memory

The agent's ability to store and recall information

- Both short-term (for the current task) and
- Long-term (for accumulated knowledge)

Provides context

Learns from past interactions

Maintains a consistent understanding of the world

Reasoning

The central "brain" or cognitive engine

Typically powered by an LLM.

Analyzes the current situation

Evaluates its options based on its plan and memory

Makes decisions about what to do next

Planning

The ability to break down

- Complex, high-level goal into
  - Sequence of smaller
  - Manageable steps

Creates a strategic roadmap for the agent to follow

- Ensuring its actions are logical
- Directed toward the final objective

Action

The agent's ability to interact

- With its environment by
  - Executing specific tasks

To carry out the steps defined in its plan:

- Calls APIs
- Runs code
- Searches databases, or
- Uses other external tools

Industry examples of autonomous AI agents

Copilot

- Suite of AI agents deeply embedded across Microsoft's ecosystem
- Summarize meetings
- Draft documents
- Write code, and
- Reason over your private data within a secure environment

AutoGPT

- An open-source project demonstrating a fully autonomous agent.
- Generates a plan
- executes steps using tools
- Analyzes the results, and
- Continues its loop until the objective is met

Cohere

- Platform providing models and tools
- Developers use Cohere's API to connect the model to their own data and APIs.

Frameworks for Building AI Agents

Introduction to LangChain

- OpenSource framework
- LLMs have an impressive general knowledge
- Limited to their training data.
- Allows you to connect an LLM like GPT-4 to your own sources of data.

LangChain: General Pipeline

A user asks a question.

The question is sent to the LLM.

A vector representation of the question is used to do a similarity search in the vector database.

Relevant chunks are fetched from the vector database and feed to the LLM.

Now the LLM has both the initial question and the relevant information from the vector database

Capable of providing an answer or taking an action

LangChain Use-Cases

- Chat Bots
- Question Answering Systems
- Summarization Tools
- Medical Doctor Assistant App
- Lawyer Assistant App

LangChain Main Concepts

LangChain Components

- LLM Wrappers
  - Provides a unified way to interact with different LLM providers (OpenAI, Google, Hugging Face, etc.)
- Prompt Templates
  - Reusable "fill-in-the-blanks" text for creating prompts.
- Indexes
  - Vector Store, which stores numerical representations (embeddings) of your text
- Memory
  - Gives application a "short-term memory" to remember previous interactions

Chains

- Allow us to combine multiple components together

Agents

- This process involves taking an action, observing the result, and then repeating the cycle until completion

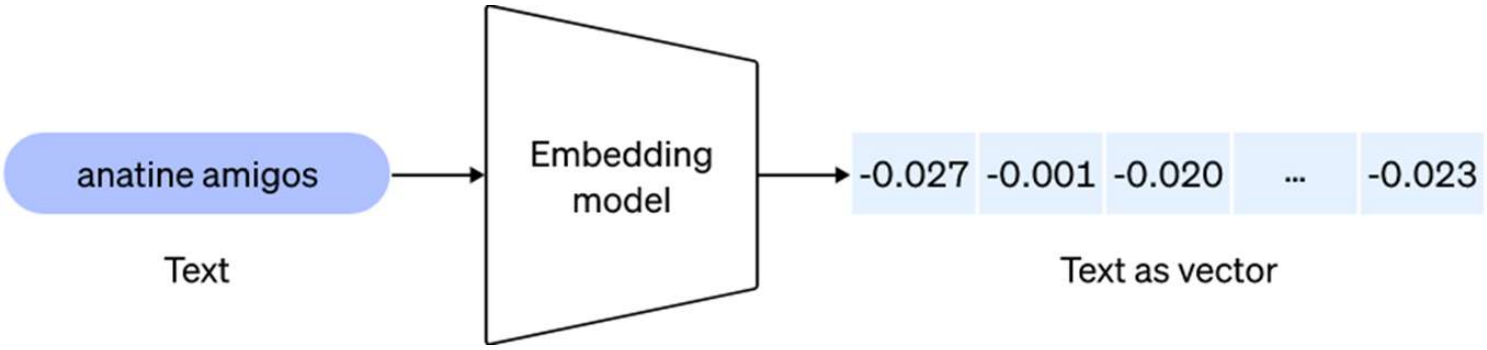
LangChain in Action: AI Travel Agent

User Goal: "Plan my weekend trip. I like beaches and I fly IndiGo from Noida."

- 🧠 **Memory:** Remembers the user's preferences ("likes beaches," "flies IndiGo").
- 📚 **Indexes:** Looks up the user's private profile ("Home city is Noida").
- 🤖 **Agent:** Decides the plan: "I will find flights from Noida to a beach city on IndiGo."
- 📝 **Prompt Templates:** Fills in a "script" to use its tools: `Search for [IndiGo] flights from [Noida] to [Goa]`.
- 🔗 **Chains:** Executes the sequence of actions (Search Flights -> Get Price).
- 📡 **LLM Wrapper:** Ensures all communication with the AI's "brain" is standardized.

Embeddings

- Embeddings are the core of building LLMs applications.
- Text embeddings are numeric representations of text and are used in NLP and ML tasks.



- An open-source project
- Released in early 2023
- Given a single high-level goal, it independently generates a plan, executes actions using tools (like web search), analyzes the results, and repeats this loop until the goal is achieved.
- The simple yet powerful loop it employs to operate
  - This process, often called an agentic loop

LangChain vs AutoGPT

- LangChain is like a full box of LEGOs  .
- It provides all the different bricks, wheels, and special pieces (the components like Indexes, Chains, and Agents) you need to build anything you can imagine
- It’s for builders

- AutoGPT is like a finished, self-driving LEGO car 
- It's a specific, fully assembled creation designed to achieve a goal on its own
- It's for end-users who want to give it a task and watch it go

BabyAGI

A simplified, task-driven autonomous agent

- Designed to accomplish a single, well-defined objective

It operates on a continuous loop

- It pulls the highest-priority task
- Executes it using an LLM
- Creates new tasks based on the result, and then
- Re-prioritizes the entire task list.

Its main purpose is to break down a large problem into smaller sub-tasks

LangChain vs BabyAGI

Feature	**LangChain**	**BabyAGI**
Primary Purpose	A **framework** for developers to build a wide variety of custom LLM applications.	A specific, simple **autonomous agent** script designed for task management.
What it is	A large library of tools and components.	A single, standalone Python script.
Function	Provides the building blocks to connect LLMs to data, tools, and memory.	Runs a loop: creates tasks based on a goal, prioritizes them, and executes them.
Complexity	Can be as simple or as complex as the application you build with it.	Very simple in its design, focused on its specific task loop.

Understanding agent-environment interaction

The Agent is AI decision-maker

Environment is the world it operates in

The agent perceives the environment through an "observation“

Based on its observation, the agent performs an Action.

This action changes the environment and results in Feedback

- Like a reward or penalty
  - That tells the agent how good its action was.

The agent's purpose is to learn the best strategy by repeating this

- Observe → Act → Receive Feedback loop
- Aiming to maximize its total rewards over time

How AI agents use tools and APIs to make decisions?

The agent's "brain" (an LLM) analyzes a goal and decides which tool to use

- e.g., "I should call the weather API"

It then tells your application code which function to run.

Code executes that function and gets the real-world result.

The code shows this result back to the agent, which then uses this new information to give the final answer.

Hands-on

- Building a simple AI agent using LangChain

- Connecting AI agents to external APIs (Google Search, Wolfram Alpha)

Thanks