

Introduction to Agentic AI

Definition

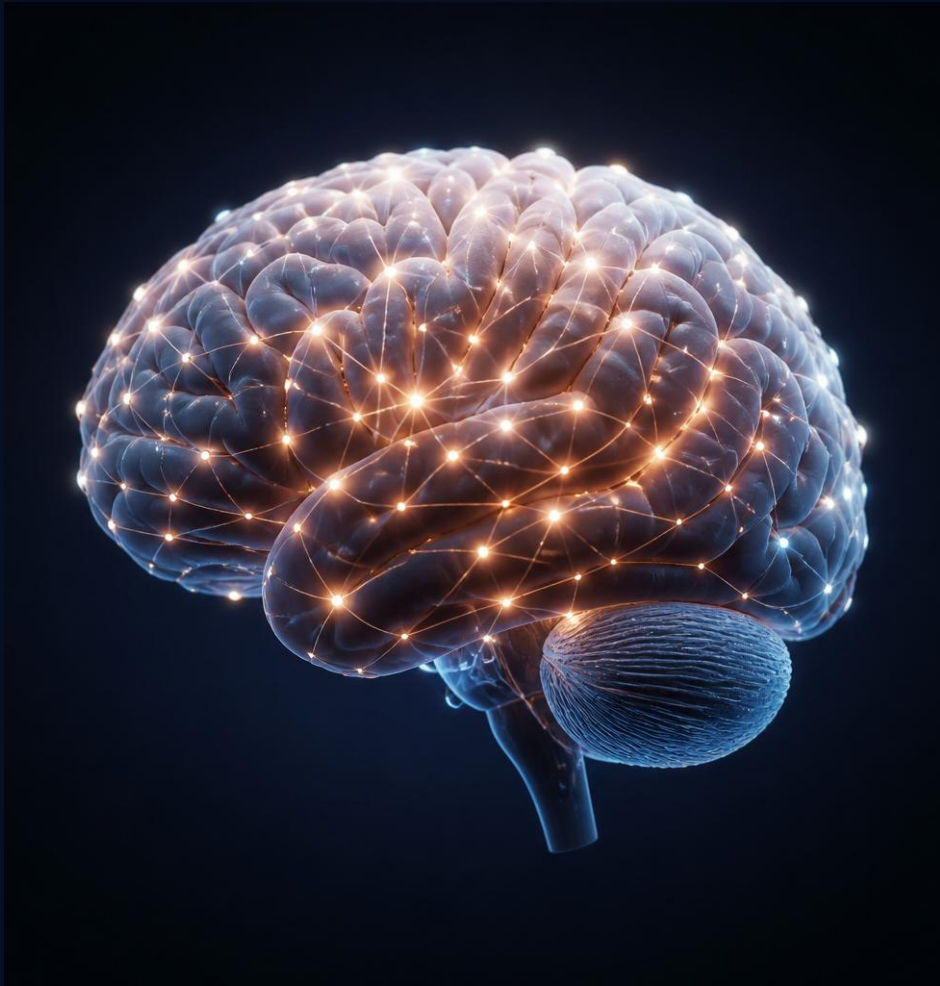
Agentic AI represents a paradigm shift from passive Large Language Models (LLMs) that primarily generate text to proactive systems capable of perception, reasoning, decision-making, and autonomous action.

Key Difference: Conversation vs. Execution

- Traditional Conversational AI: Focuses on predicting the next token; responds only when prompted; limited to information synthesis.
- AI Agents: Task-oriented; utilizes external tools (APIs, files, web tools) to interact with the environment; autonomously iterates to achieve specific, goal-directed outcomes.

Agent Architecture: The 'Brain' and Beyond

Core components driving autonomous agent intelligence



- **Brain (LLM)**

Acts as the central reasoning and planning engine, translating intent into actionable task structures.

- **Memory**

Manages short-term context windows and long-term persistent knowledge via vector databases.

- **Planning**

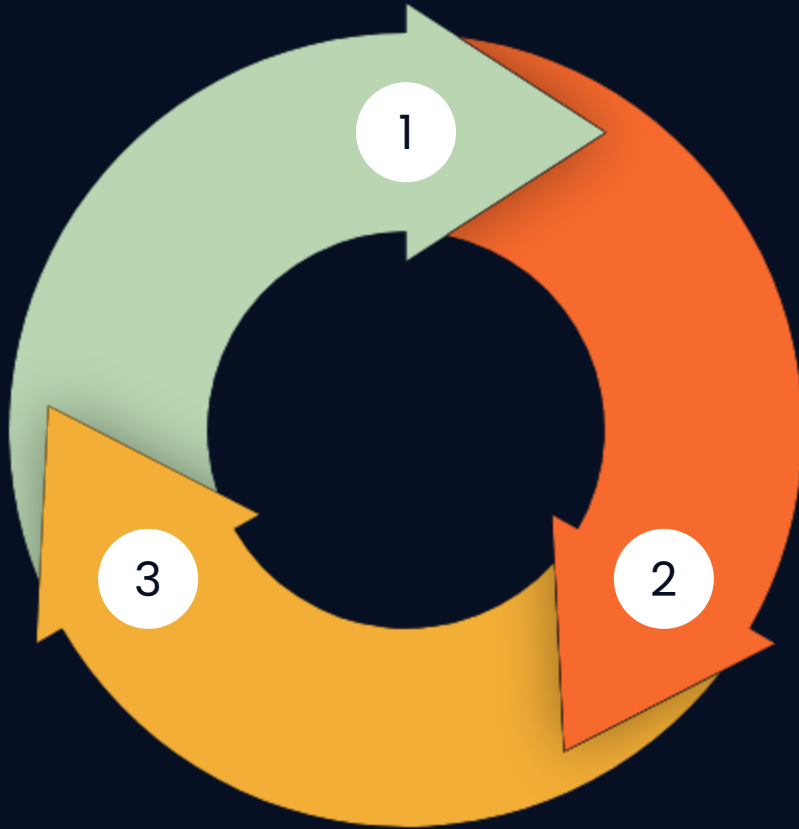
Decomposes complex goals into sub-tasks using methodologies like Chain of Thought (CoT) and Tree of Thoughts (ToT).

- **Tools**

Provides interfaces for external interaction, including API calls, web search, and code execution.

Reasoning and Feedback Loops

Mastering the Agentic Cycle



1

Thought

The agent processes incoming data and formulates a strategic plan based on internal reasoning capabilities.

2

Action

Based on the refined thought process, the agent executes a specific operation or tool call.

3

Observation

The agent captures the outcome of the action from the environment, creating a new data point for analysis.

By leveraging iterative refinement, adaptive strategy, and error correction, agents achieve high-precision execution in complex, multi-step tasks.

Tool Use and Function Calling

Enabling LLMs to interact with external systems through structured data outputs.



Web Search

Accesses real-time information to supplement training data.



API Connectivity

Enables seamless interaction with external SaaS platforms.



Database Interfaces

Allows querying SQL and NoSQL databases for dynamic data retrieval.



Sandbox Code Execution

Provides a secure environment to run scripts and solve complex logic problems.

Agentic Patterns and Architectures

Foundational design patterns for building intelligent agent systems



ReAct (Reasoning + Acting)

Integrates reasoning traces with action execution to improve decision-making and transparency.



Human-in-the-loop

Ensures critical oversight by integrating human intervention points into automated workflows.



Planner-Executor

Decouples high-level strategic reasoning from operational task execution for better modularity.



Multi-agent Systems

Orchestrates a network of specialized agents to distribute complex tasks and leverage collaborative intelligence.



Reflection

Employs self-critique loops to evaluate, refine, and iterate on outputs independently.

Frameworks for Building Agents

Key tools for developing LLM-powered autonomous systems

- **LangChain**

Versatile orchestration framework for building complex LLM-powered applications.

- **LangGraph**

Designed for stateful, multi-actor applications, enabling complex agentic workflows.

- **CrewAI**

Specialized framework for role-playing, collaborative multi-agent systems.

- **AutoGen**

Optimized for conversational patterns and autonomous agent interactions.

- **LlamaIndex**

Data-centric agent development with a deep focus on RAG (Retrieval-Augmented Generation).

- **Semantic Kernel**

Integration-focused framework for embedding agents into enterprise software ecosystems.

Advanced Concepts: RAG, MCP, and Security

Key frameworks and safety measures for modern AI agents



Agentic RAG

Evolves beyond basic retrieval by enabling autonomous, multi-step document research and synthesis without human intervention.



Model Context Protocol (MCP)

A standardized, open-source framework for connecting AI assistants to diverse data repositories, reducing integration friction.



Agent Security Framework

Mitigates autonomous execution risks through prompt injection defense, strict tool access control, and sandboxed environments.

Capstone Project: Multi-Agent Research Assistant

Automated Business Intelligence Reporting



Researcher Agent

Performs targeted web searches and data gathering to compile raw source material.



Analyst Agent

Processes gathered data via reasoning, filtering, and synthesis of key insights.



Writer Agent

Translates analyzed data into professional-grade reports and actionable summaries.

Leverages specialized agents working in tandem to automate complex intelligence workflows.