

Agents

Machine Learning Engineer in the Generative AI Era

What is an Agent?

- Definition:
An agent is a system that **acts autonomously** by combining an LLM with external tools, APIs, or workflows to perform complex, multi-step tasks.
- Unlike a standalone LLM, an agent can interact with the environment, make decisions, and execute commands.

What Can Agents Do But LLMs Alone Cannot?

Agents	LLM Alone
Call external APIs & services	Only generate text output
Access up-to-date information	Limited to pretraining data
Perform multi-step reasoning & workflows	Single-turn text generation
Maintain state or memory over interactions	Stateless text generation
Integrate with tools (search, calculator, databases)	No direct integration
Automate real-world tasks (email, scheduling, web scraping)	Can't perform actions

Function Calls in Agents

- Function calls allow the agent to **invoke external functions or APIs** to retrieve information or perform actions.
- Example: Agent receives a prompt → Decides to call a weather API → Returns weather data as part of the response.
- Enables hybrid reasoning: Text + Tool usage.

Popular Agent Frameworks: LangChain & LangGraph

LangChain:

- Python-based framework to build agentic AI applications.
- Provides components for prompt management, chaining, memory, tool integration, and function calls.
- Widely used for building Retrieval-Augmented Generation (RAG) and multi-tool agents.

LangGraph:

- Graph-based approach to design and orchestrate complex agent workflows.
- Models agents as directed acyclic graphs connecting multiple LLMs and tools.
- Enables scalable, interpretable multi-agent and multi-tool systems.

MCP Protocol Overview

- **MCP = Model Context Protocol**
- An open standard enabling LLM applications to seamlessly interact with **external data sources, tools, and servers**.
- Facilitates integration of LLMs with external APIs, local services, or databases for extended capabilities.

Why MCP Protocol Matters

- Enables **dynamic context updates** during LLM conversations.
- Supports **tool-use and chaining** by defining clear message and data passing formats.
- Powers agentic systems like Claude Desktop to use external knowledge and tools.

MCP Protocol Components

- **MCP Server:** Handles communication between the LLM and external resources.
- **Message Format:** JSON-based payloads carrying prompts, tool requests, and responses.
- **Integration:** MCP servers can expose APIs for web search, file system access, or specialized functions.

Example MCP Server Use Cases



WEB SEARCH
INTEGRATION (E.G.,
BRAVE SEARCH API)



LOCAL FILE SYSTEM
ACCESS FOR
DOCUMENT RETRIEVAL



SEQUENTIAL THINKING
OR REASONING TOOL
SERVERS



CUSTOM TOOLS FOR
DOMAIN-SPECIFIC
WORKFLOWS

Summary

- Agents extend LLMs by combining language understanding with **external tools and functions**.
- Function calls enable agents to perform concrete actions beyond text generation.
- MCP is a powerful protocol to build modular, interoperable agent ecosystems.