Core Anatomy of an AI-Agent:

Core Anatomy of an AI Agent

An AI Agent isn't just a model—it's an intelligent decision-making system. This presentation breaks down its internal components: the LLM, Tools, Memory, Prompt Template, Agent Executor, and Loop Logic. By understanding each part, you'll gain clarity on how AI agents reason, act, and learn dynamically.

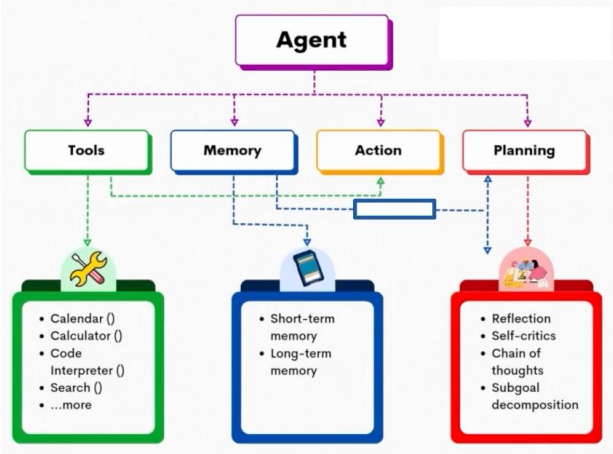
TTAO: Tools, Thoughts, Actions, Observations

TTAO is a foundational structure used in LLM-based agents like ReAct and LangChain agents. It defines how an agent processes information and interacts with the world.

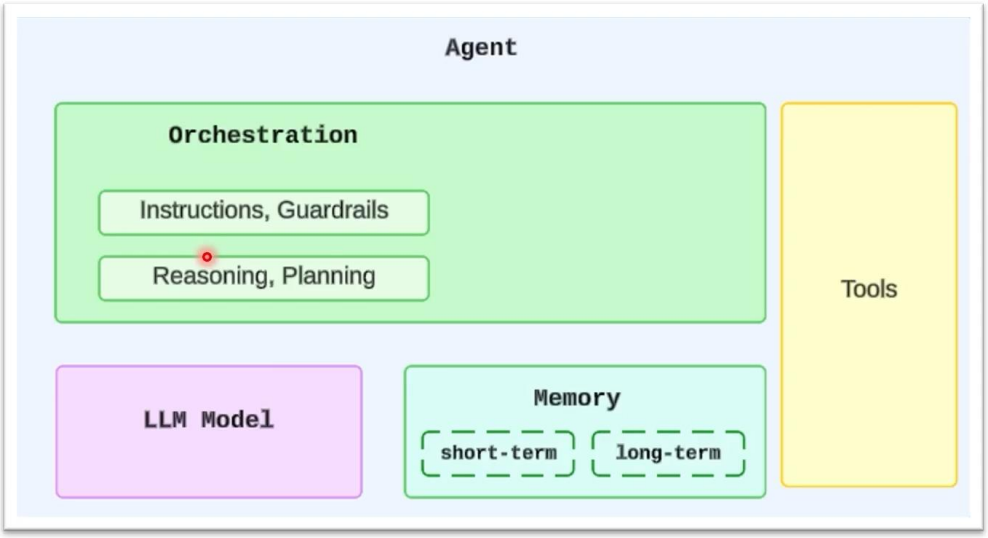
Component	Description
Thoughts	The internal reasoning process of the agent. The model "thinks out loud" using language to decide what to do next. Example: "I need to find the capital of France."
Tools	External resources the agent can use to complete tasks (e.g., calculator, web search, API, database, code executor).
Actions	Calls to tools or APIs based on the current thought. Example: search("Capital of France").
Observations	Feedback or result returned by the tool after an action. The agent reflects on this before generating the next thought.

Example Flow (TTAO)

- Thought: The Eiffel Tower is in France.
- Action: Search("Capital of France")
- Observation: The capital of France is Paris.
- Thought: I now know the answer.
- Answer: Paris



Reasoning and Planning Modules



Reasoning and Planning Modules

LLM agents must decide what to do and how to do it—this is where **reasoning** and **planning** modules come in.

Reasoning

- ❑ Involves **logical steps** and **inference** using natural language.
- ❑ Enables the agent to solve problems step-by-step (e.g., math, decision-making, tool selection).
- ❑ Often implemented through **prompt engineering** like chain-of-thought reasoning or the ReAct loop.

Planning

- ❑ Used when tasks involve **multiple steps** or require **task decomposition**.
- ❑ The agent might:
  - Generate a plan first (e.g., "Step 1: Get data, Step 2: Analyze, Step 3: Summarize").
  - Then execute each step using reasoning + actions.

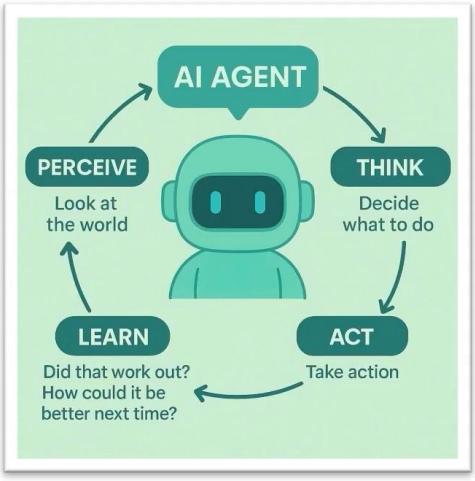
Frameworks like **LangChain (Plan-and-Execute agent)** or **AutoGPT** enable this behavior by separating the planner and executor roles.

Agent-Environment Interaction Loop

Agents work in a **loop** where they continuously interact with their environment:

- ❑ **Perceive (Observation)**: The agent receives input from the environment or from a tool/API.
- ❑ **Think (Reasoning)**: The agent reasons about the observation and decides what to do.
- ❑ **Act (Tool Usage)**: The agent executes an action (e.g., fetches data, runs a function).
- ❑ **Reflect (New Observation)**: The result of the action becomes the next input.
- ❑ **Loop Continues** until the task is complete.

This forms the **agent loop** and mirrors how humans interact with the world (sense → think → act → observe → adapt).



Agent–Environment Interaction Loop

Loop Example

User: “Find the stock price of Google and tell me if it went up today.”

→ Agent observes input

→ Thinks: "I need to get the latest stock price and compare it with yesterday’s”

→ Uses stock API to fetch data

→ Observes result

→ Thinks again and concludes

→ Responds to user