Introduction to Large Language Models

Spring 2026

Agentic AI

(Some slides adapted from Ralph Grishman at NYU, Yejin Choi at UWashington, N. Tomura at UDepaul, Jurafsky and Martin, CS224N, CS224, CME295 at Stanford and other resourses on the web)

The rise, and the divide

Bill Gates

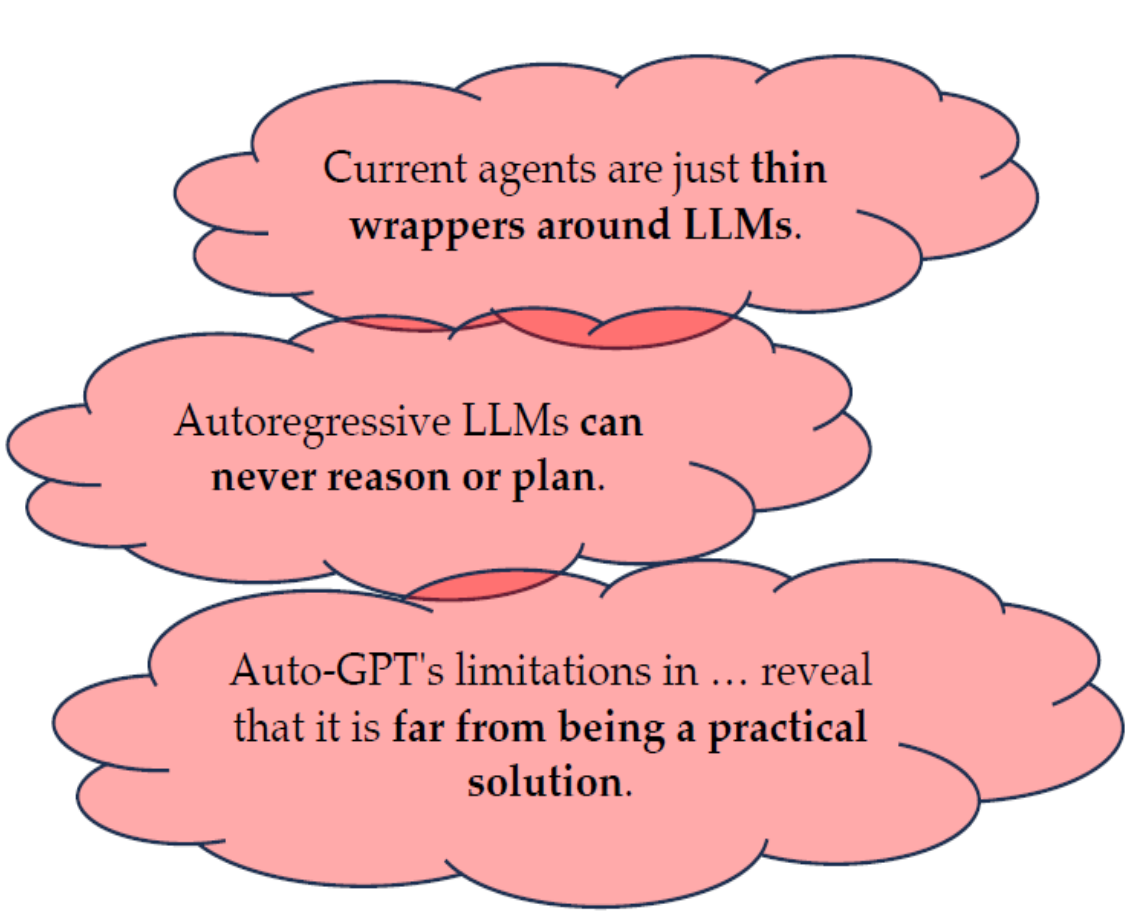
Agents are bringing about the **biggest revolution in computing** since we went from typing commands to tapping on icons.

Andrew Ng

I think AI agentic workflows will drive **massive AI progress** this year.

Sam Altman

2025 is when **agents will work**.

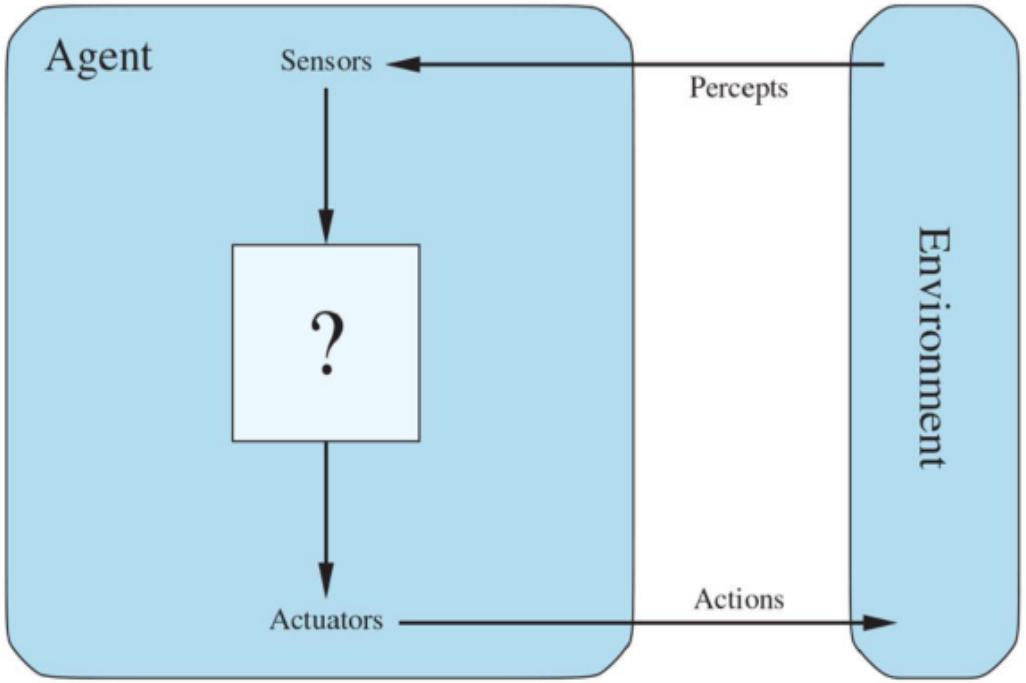


Agents in AI

What Is an Agent in AI?

In **artificial intelligence**, an **agent** is a system that **perceives its environment, makes decisions, and takes actions to achieve goals**.

A standard definition used in AI (e.g., Russell & Norvig) is:



"An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators."

-- Russell & Norvig, *AI: A Modern Approach* (2020)

Core Elements of an AI Agent

An entity is considered an *agent* if it has these key components:

1. Environment

The external world the agent operates in
(e.g., a game board, the internet, a physical room, a software system)

2. Perception (Sensors)

How the agent receives information
(e.g., text input, images, system state, sensor data)

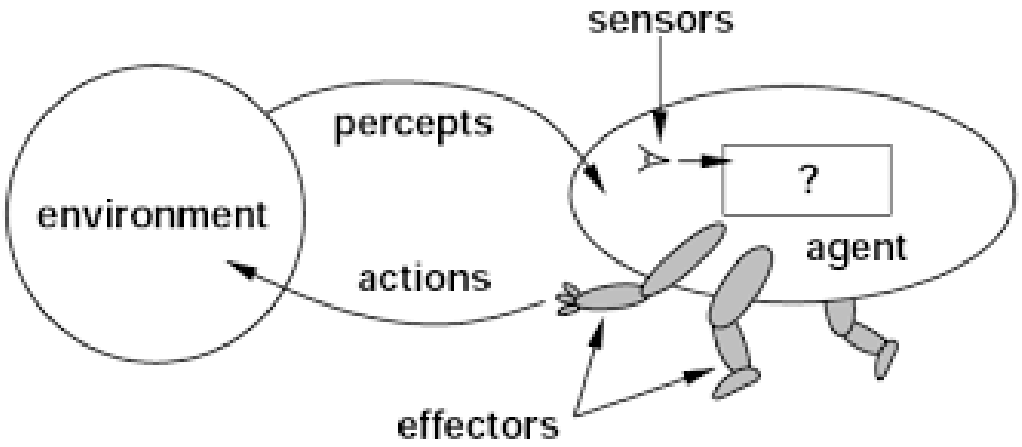
3. Decision-Making

How the agent selects actions based on:

- Observations
- Goals
- Internal state or memory
- Rules, policies, or learned models

4. Action (Actuators)

How the agent affects the environment
(e.g., sending commands, moving a robot, calling an API, writing files)



An **AI agent** can be modeled as a tuple:

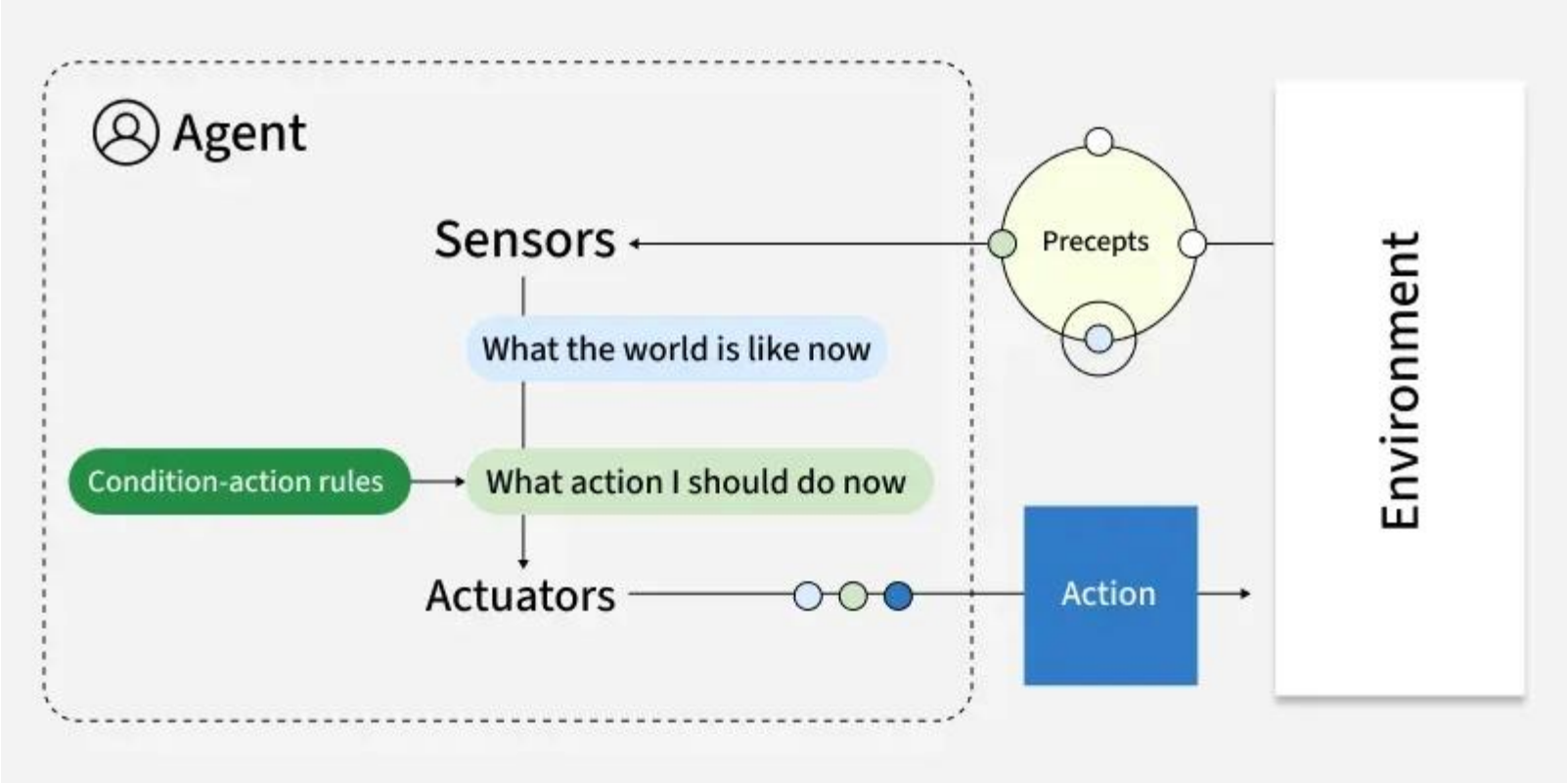
$$\langle E, P, A, \pi, M \rangle$$

Where:

- E: environment
- P: percept space
- A: action space
- π: policy (possibly learned or inferred)
- M: memory or belief state

Types of Agents in AI

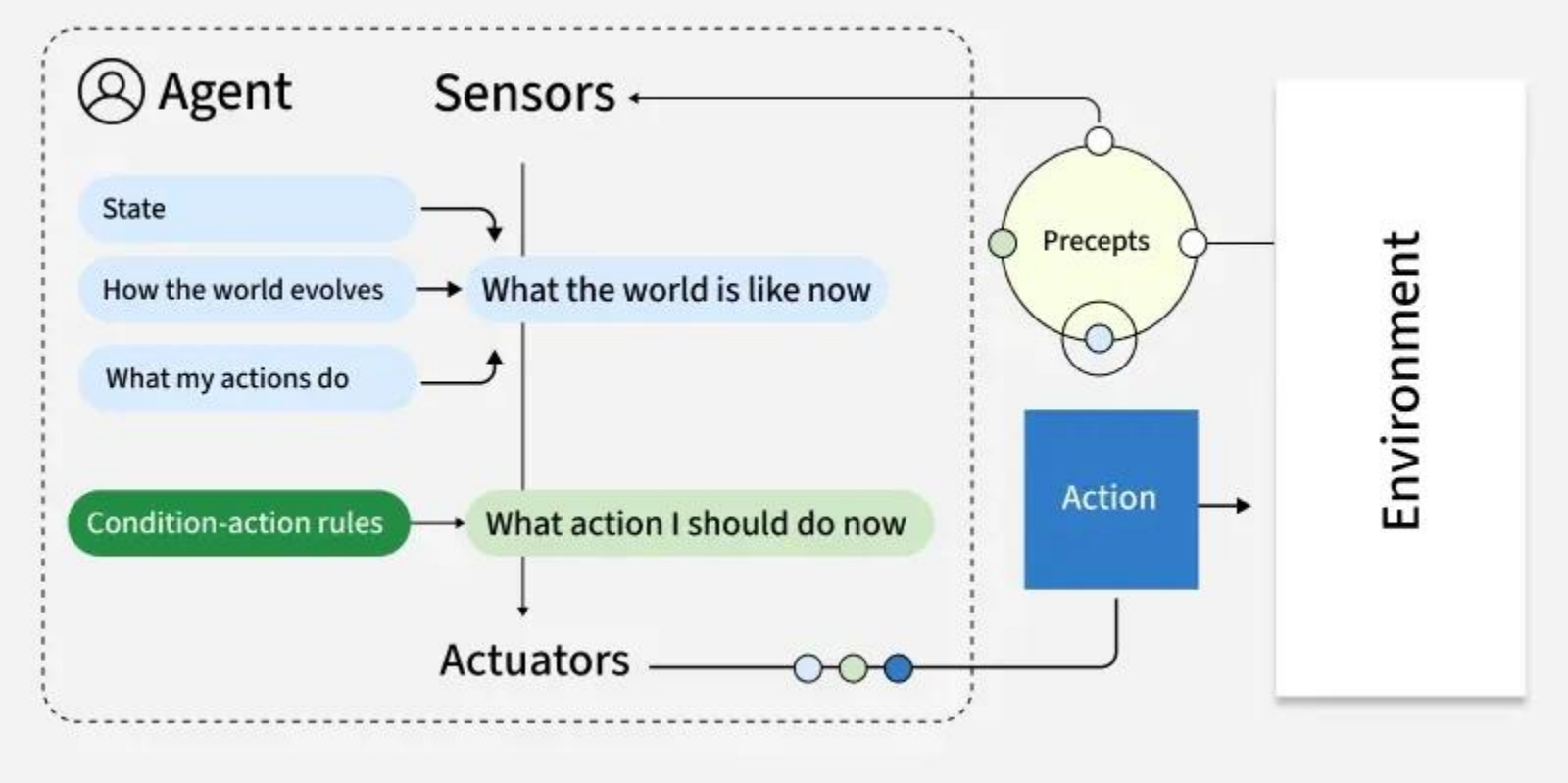
- Simple reflex agents act solely on the current percept using predefined condition–action rules, without storing or considering any history.
- They are fast and easy to implement, making them suitable for fully observable, stable environments with clear and simple rules.
- **Example:** Traffic light control systems that change signals based on fixed timing.



https://www.geeksforgeeks.org/artificial-intelligence/types-of-agents-in-ai/

Types of Agents in AI

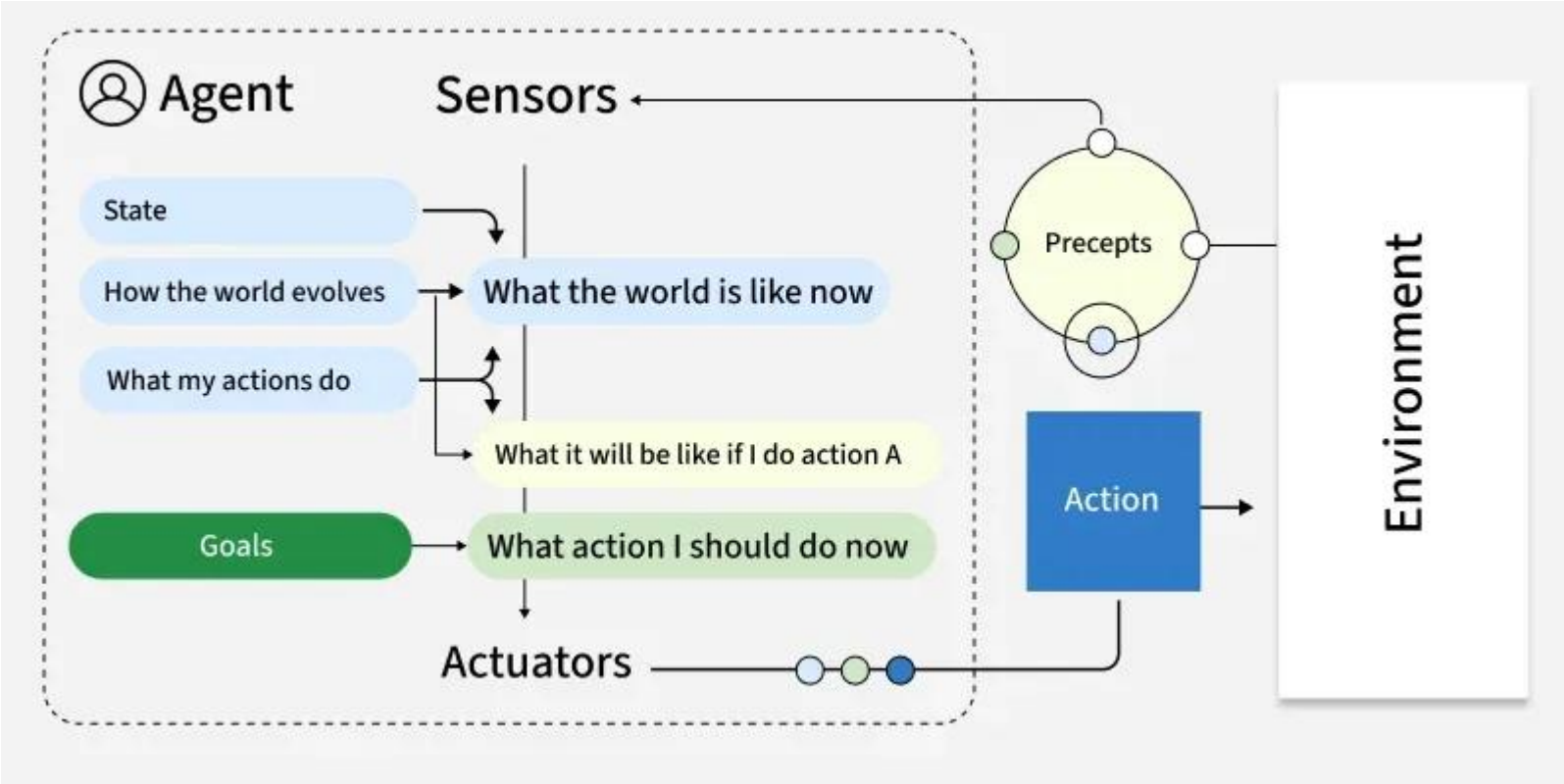
- Model-based reflex agents enhance the simple reflex approach by maintaining an internal state or model of the world, that tracks aspects of the environment not directly observable at each moment.
- This enables them to deal with partial observability and dynamic changes more effectively, although their decisions are still largely reactive and dependent on the accuracy of the model they maintain.
- **Example:** Robot vacuum cleaners that map rooms and tracks cleaned areas.



https://www.geeksforgeeks.org/artificial-intelligence/types-of-agents-in-ai/

Types of Agents in AI

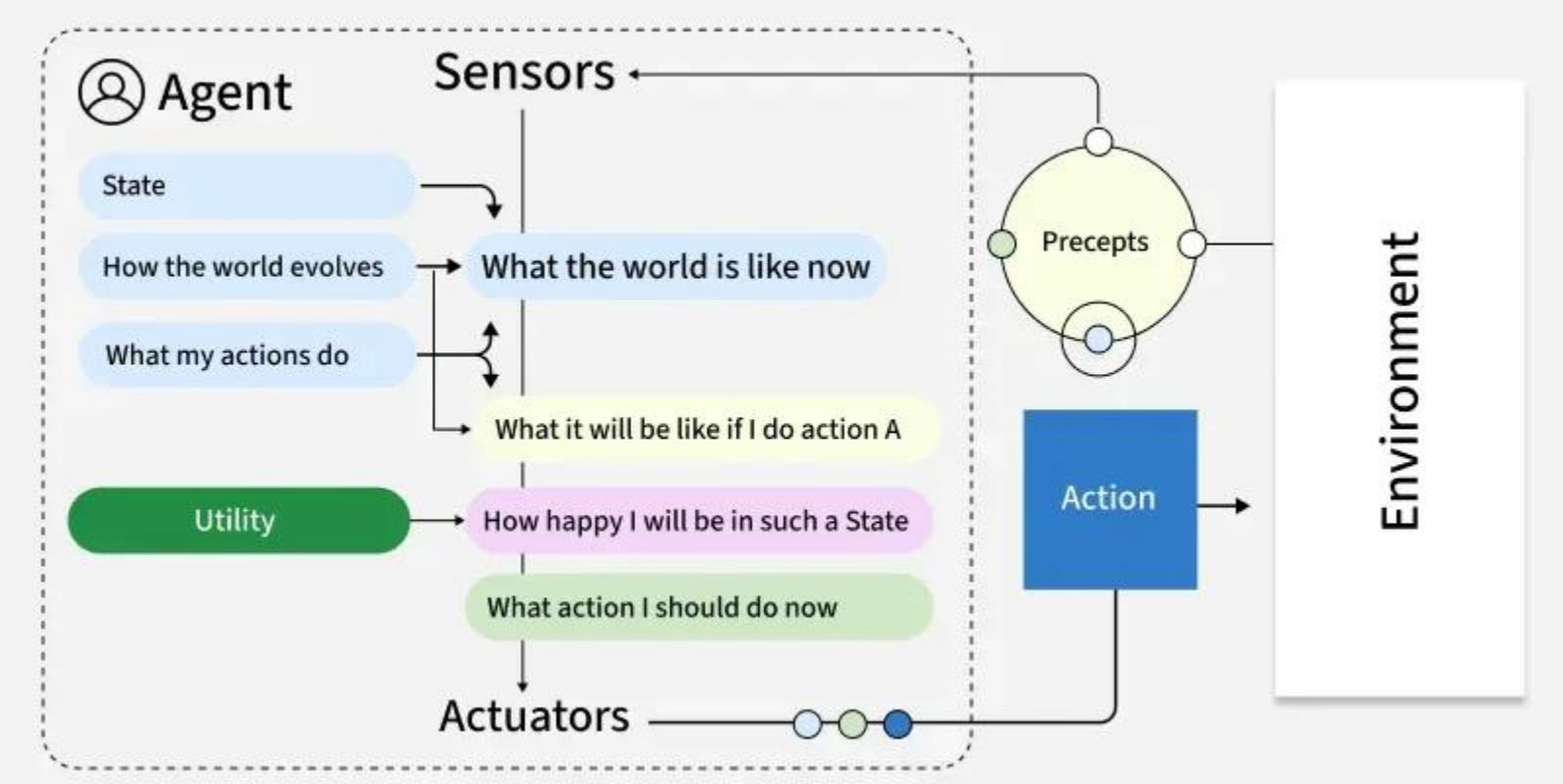
- Goal-based agents select actions by considering future states relative to explicit goals. They are capable of planning sequences of actions to reach these goals rather than just reacting to the current state which enables more flexible and intelligent problem-solving.
- However, they require well-defined goals and effective planning algorithms to perform well in complex domains.
- **Example:** Logistics routing agents that find optimal delivery routes based on factors like distance and time. They continuously adjust to reach the most efficient route.



https://www.geeksforgeeks.org/artificial-intelligence/types-of-agents-in-ai/

Types of Agents in AI

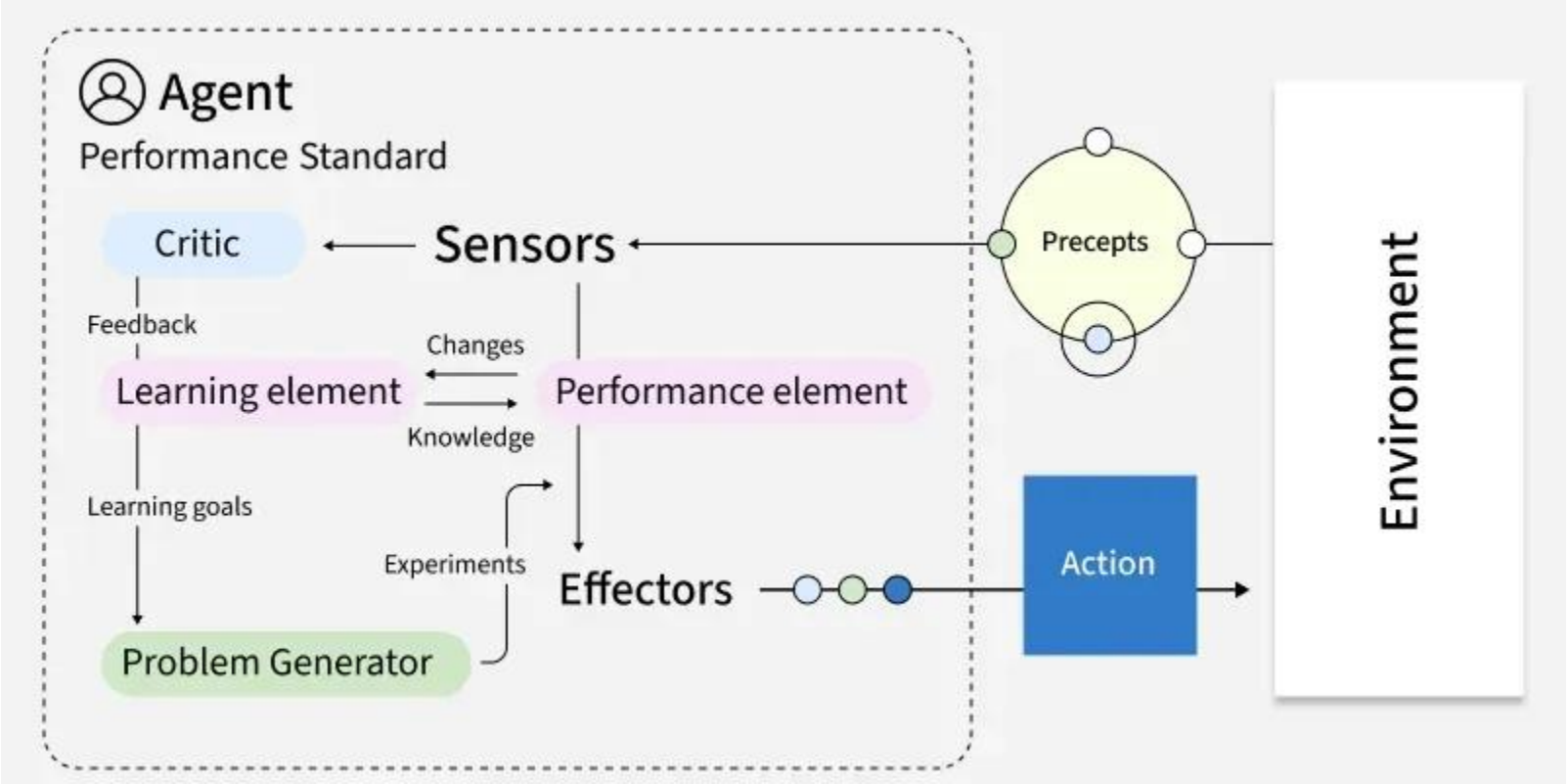
- Utility-based agents extend goal-based reasoning by considering not only whether a goal is met but also how valuable or desirable a particular outcome is.
- They use a utility function to quantify preferences and make trade-offs between competing objectives, enabling nuanced decision-making in uncertain or resource-limited situations. Designing an appropriate utility function is crucial for their effectiveness.
- **Example:** Financial portfolio management agents that evaluate investments based on factors like risk, return and diversification operate by choosing options that provide the most value.



https://www.geeksforgeeks.org/artificial-intelligence/types-of-agents-in-ai/

Types of Agents in AI

- Learning agents improve their performance over time by learning from experience and updating their internal models, strategies or policies. They can adapt to changes in the environment and often outperform static agents in dynamic contexts. Learning may involve supervised, unsupervised or reinforcement learning techniques and these agents typically contain both a performance element (for acting) and a learning element (for improving future actions).
- **Example:** Customer service chatbots can improve response accuracy over time by learning from previous interactions and adapting to user needs.



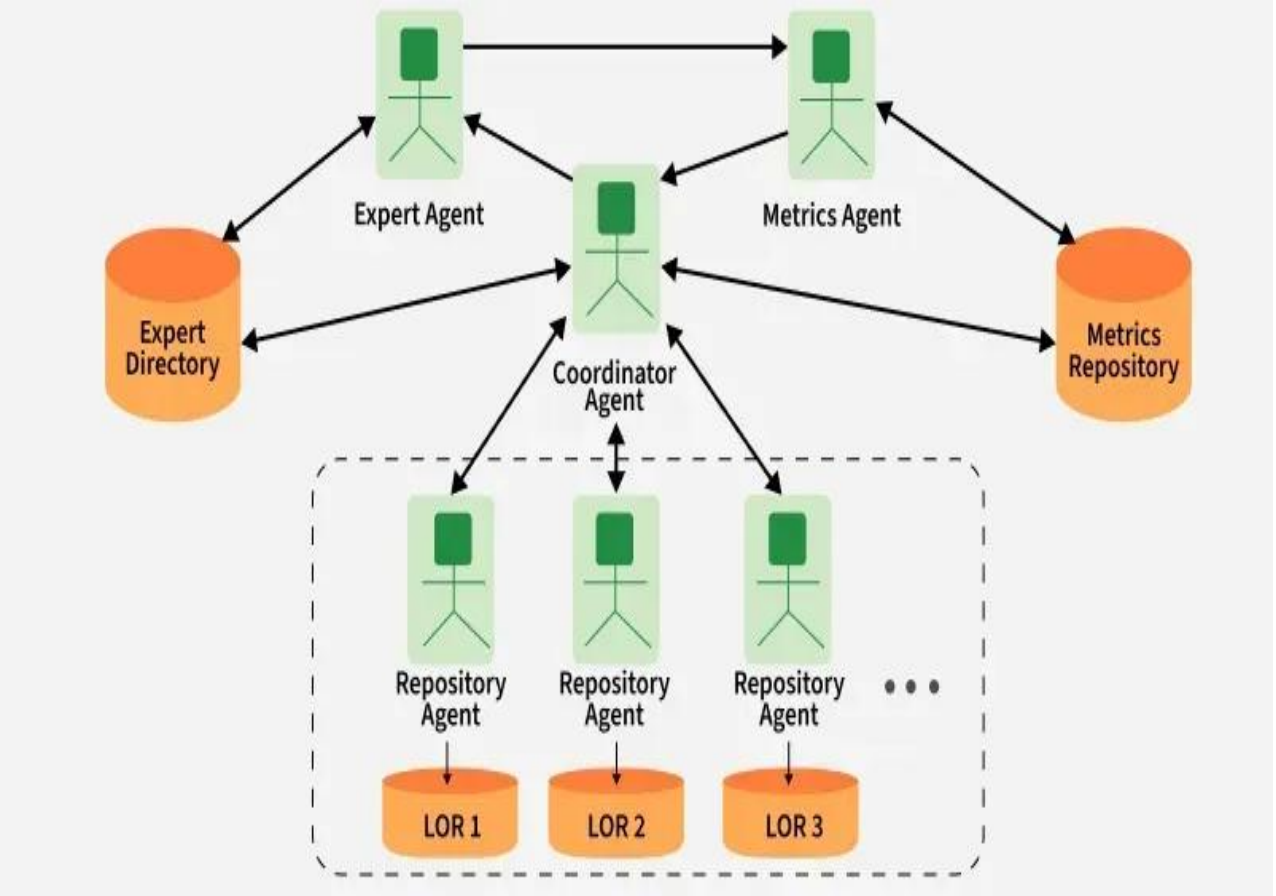
https://www.geeksforgeeks.org/artificial-intelligence/types-of-agents-in-ai/

Types of Agents in AI

- Multi-agent systems operate in environments shared with other agents, either cooperating or competing to achieve individual or group goals. These systems are decentralized, often requiring communication, negotiation or coordination protocols. They are well-suited to distributed problem solving but can be complex to design due to emergent and unpredictable behaviors. Types of multi-agent systems:
- **Cooperative MAS:** Agents work together toward shared objectives.
- **Competitive MAS:** Agents pursue individual goals that may conflict.
- **Mixed MAS:** Agents cooperate in some scenarios and compete in others.

Example: A warehouse robot might use:

- Model-based reflexes for navigation
- Goal-based planning for task sequencing
- Utility-based decision-making for prioritizing tasks
- Learning capabilities for route optimization

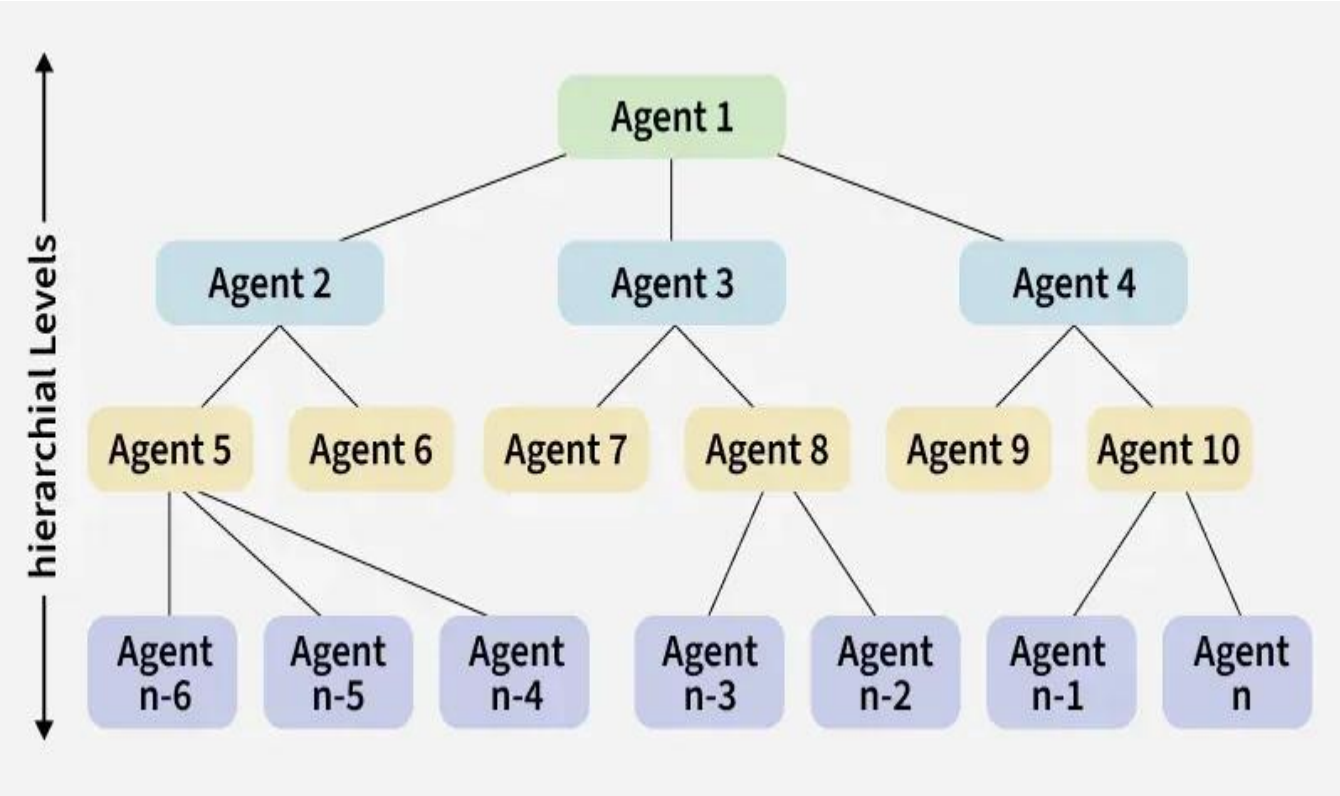


Types of Agents in AI

Hierarchical agents organize behavior into multiple layers such as strategic, tactical and operational. Higher levels make abstract decisions that break down into more specific subgoals for lower levels to execute. This structure improves scalability, reusability of skills and management of complex tasks, but requires designing effective interfaces between layers.

Key Characteristics:

- **Structured Decision-Making:** Decision-making is divided into different levels for more efficient task handling.
- **Task Division:** Complex tasks are broken down into simpler subtasks.
- **Control and Guidance:** Higher levels direct lower levels for coordinated action.
- **Example:** Drone delivery systems in which fleet management is done at top level and individual navigation at lower level.



https://www.geeksforgeeks.org/artificial-intelligence/types-of-agents-in-ai/

Comparison of AI Agent Types

Agent Type	Main Strength	Limitations	Best For	Example
Simple Reflex Agent	Instant reaction based on fixed rules	No memory or learning; fails in dynamic environments	Fully observable, stable and simple environments	Traffic light timers
Model-Based Reflex Agent	Handles partial observability with internal state	More computational demand; depends on model accuracy	Dynamic or partially observable environments	Robot vacuum cleaners
Goal-Based Agent	Plans ahead to achieve specific objectives	Needs clear goals and planning algorithms	Strategic tasks with defined goals	Logistics route planning
Utility-Based Agent	Balances multiple factors for best outcome	Requires complex utility functions	Multi-criteria decision-making	Financial portfolio management
Learning Agent	Improves over time via experience	Needs data and training time	Dynamic environments with changing conditions	AI chatbots
Multi-Agent System (MAS)	Distributed problem-solving with cooperation or competition	Complex interactions; unpredictable behaviors	Decentralized, multi-entity systems	Smart traffic control
Hierarchical Agent	Breaks complex tasks into levels for efficiency	Requires well-defined interfaces between layers	Large-scale, multi-level operations	Drone delivery management

https://www.geeksforgeeks.org/artificial-intelligence/types-of-agents-in-ai/

What is Agentic AI

Agentic AI is a type of AI system that can autonomously make decisions, plan actions and execute tasks to achieve specific goals with minimal human intervention. Instead of just responding to inputs, it actively works toward completing objectives by reasoning, using tools and adapting based on results.

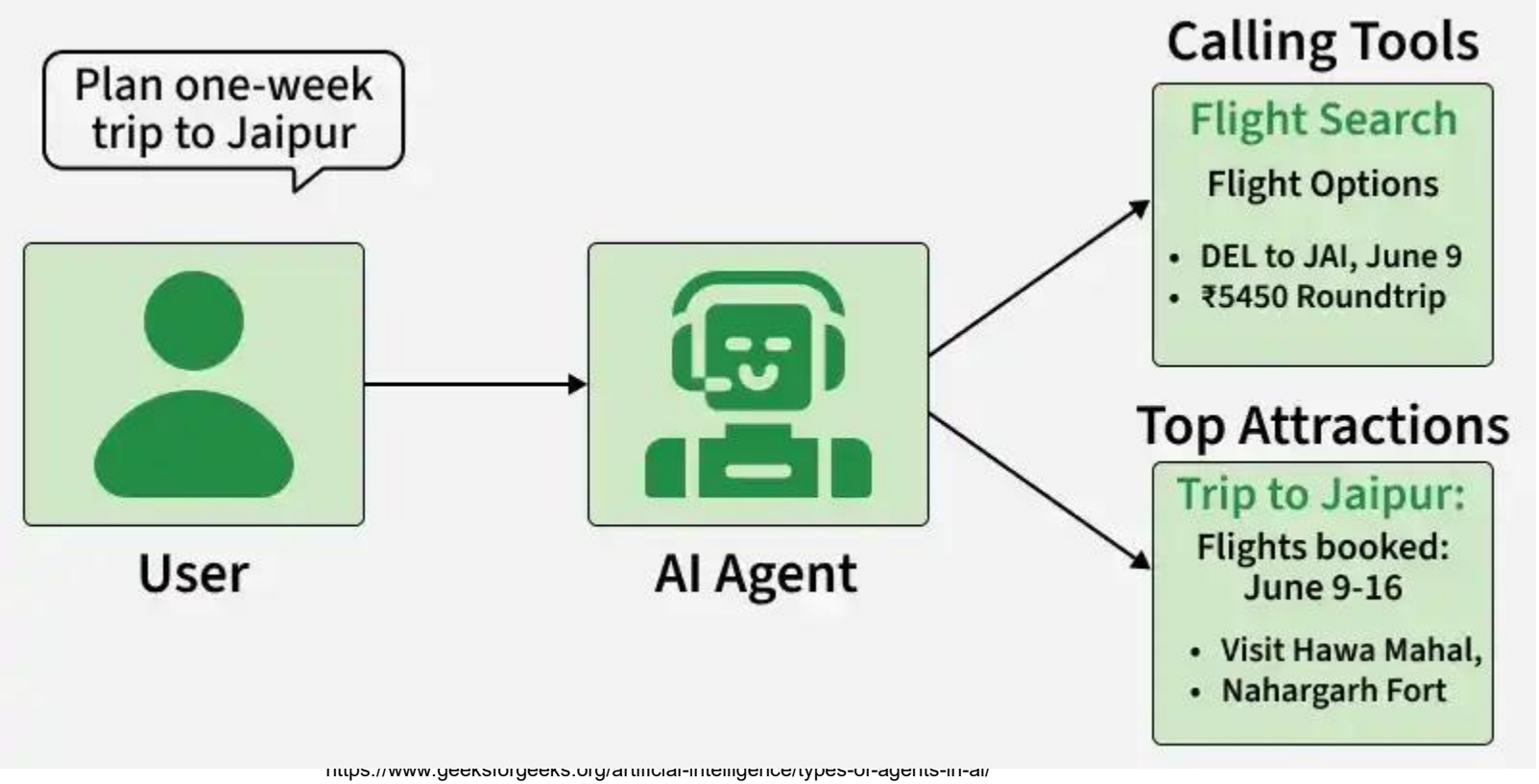
- **Specialised mastery:** It's trained and fine-tuned to handle a particular type of problem with great accuracy.
- **Tool usage:** It can connect with and use specific tools like software, APIs, databases, etc to achieve its goal.
- **Goal-oriented actions:** Instead of just giving information, it actively takes steps toward completing the task.
- **Efficient problem-solving:** Because it's specialised, it can work faster and more effectively in its field than a general AI.
- **Traditional vs. Agentic AI:** Unlike traditional AI, which follows predefined rules, Agentic AI can independently make decisions and take actions to achieve a goal.

Traditional AI	Agentic AI
Stateless or short-lived	Persistent memory
One-shot outputs	Multi-step behavior
Human-in-the-loop	Self-directed actions
Prompt → Response	Observe → Plan → Act → Reflect

https://www.geeksforgeeks.org/artificial-intelligence/types-of-agents-in-ai/

What is Agentic AI

- For example, a travel-planning Agentic AI won't just give us flight options, but it can search multiple platforms, compare prices and book tickets automatically.



https://www.geeksforgeeks.org/artificial-intelligence/types-of-agents-in-ai/

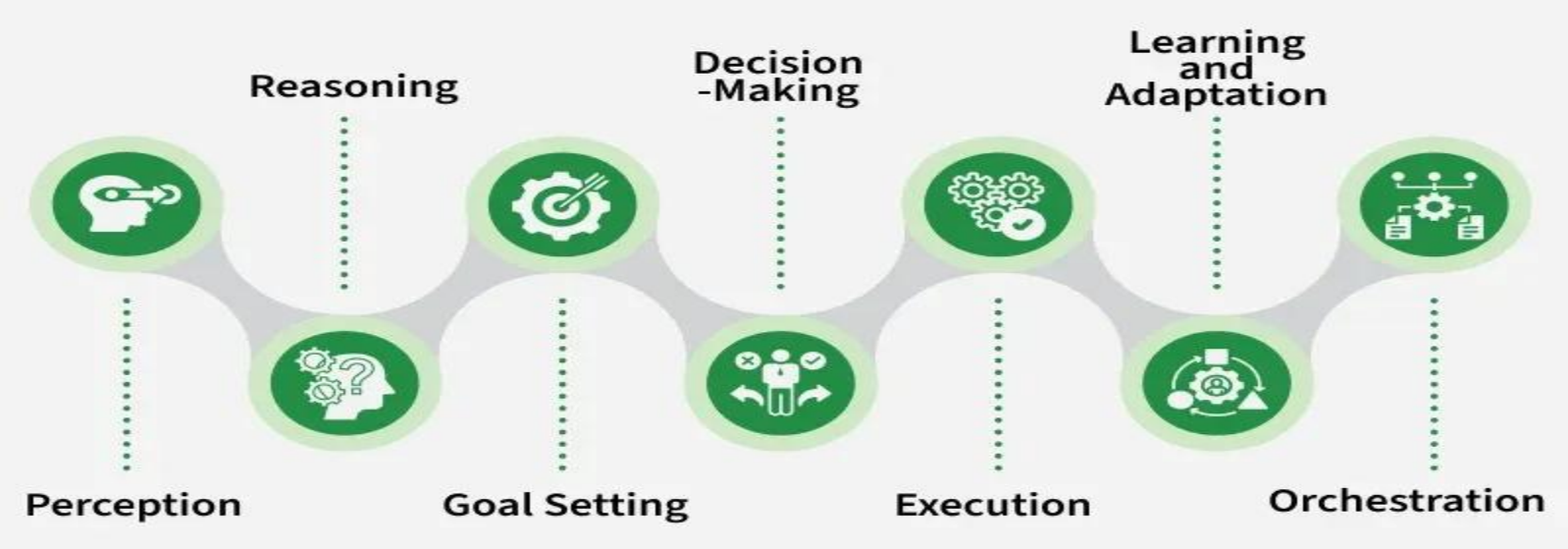
Key Characteristics of Agentic AI

- **Autonomy and Goal-Oriented Behaviour:** Agentic AI systems act independently and make decisions to achieve predefined goals without human intervention.
- **Adaptive Learning and Complex Decision-Making:** These systems learn from experience and adapt their behaviour to handle complex situations effectively.
- **Environment Interaction and Perception:** Agentic AI collects real-time data from its surroundings to understand and respond to the environment.
- **Information Processing:** It analyses data using algorithms and models to make informed decisions.
- **Action Execution:** The system performs tasks automatically using software commands or physical mechanisms based on its decisions.

https://www.geeksforgeeks.org/artificial-intelligence/types-of-agents-in-ai/

How Agentic AI Works

- **Perception**: Collects relevant real-time data and retrieves past information from memory to provide context for the task.
- **Reasoning**: Interprets inputs using domain-specific knowledge and patterns stored in memory or knowledge bases.
- **Goal Setting**: Defines clear objectives and creates a focused plan based on input and past outcomes.
- **Decision-Making**: Chooses actions based on efficiency, accuracy and safety using past successful strategies.
- **Execution**: Performs tasks using tools like APIs and records results for future reference.
- **Learning and Adaptation**: Improves over time by learning from feedback and storing useful experiences.
- **Orchestration**: Collaborates with other agents in multi-agent systems to complete tasks efficiently.



https://www.geeksforgeeks.org/artificial-intelligence/types-of-agents-in-ai/

The Agent Loop

An agent continuously operates in a feedback loop:

```
Observe → Decide → Act → Observe → ...
```

This loop is what distinguishes agents from passive systems.

Agent behavior approximates:

$$a_t = \arg\max_{a \in A} \mathbb{E}[U(s_{t+1}) \mid s_t, a]$$

Operational loop:

- Observation → belief update
- Planning → policy selection
- Acting → environment transition
- Reflection → belief revision

LLMs approximate inference over this loop.

Classical AI Agents vs Modern (LLM-Based) Agents

Classical AI Agents

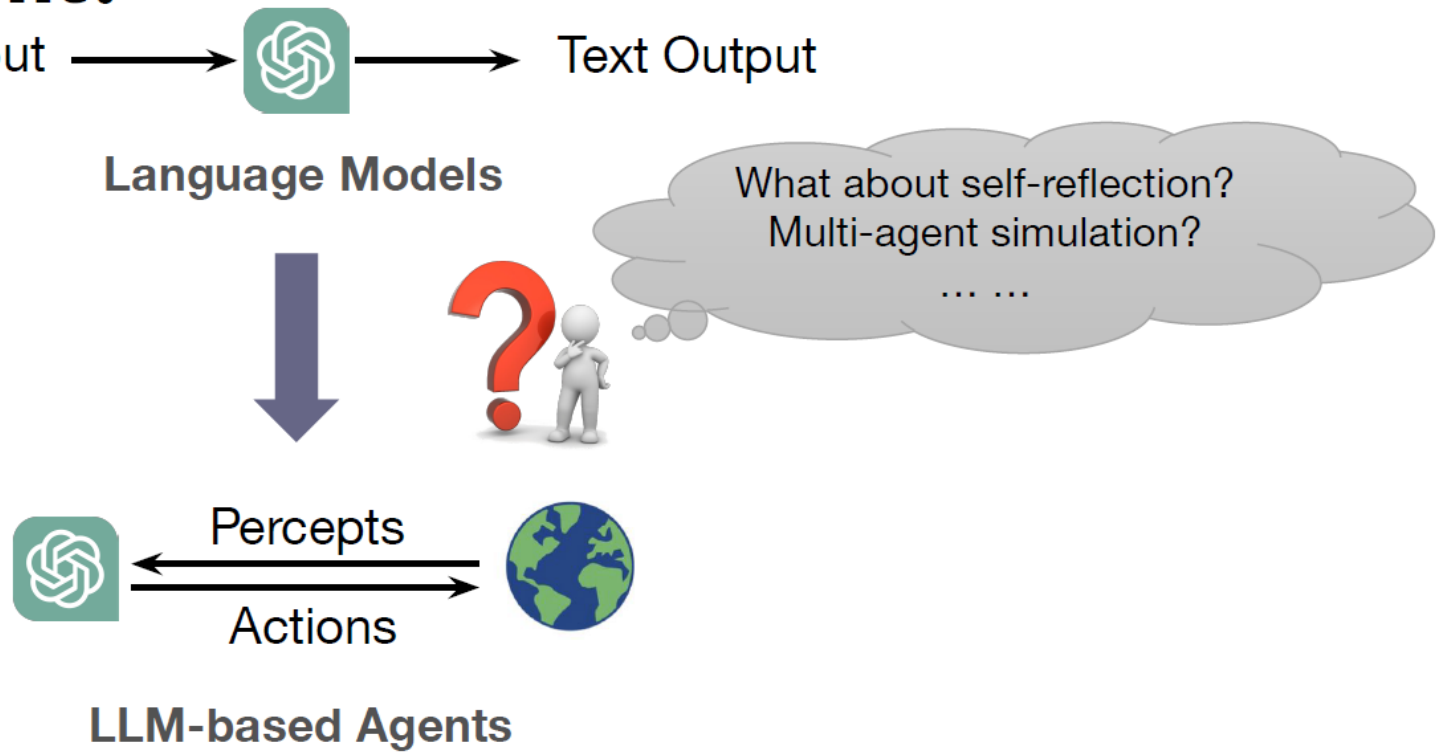
- Rule-based agents
- Planning agents
- Reinforcement learning agents

Modern (LLM-Based) Agents

- Use a language model for reasoning
- Perceive via text or tool outputs
- Act via tools (APIs, databases, code execution)
- Often have memory and planning

Modent Agent

'Modern' agent = LLM + external environment?



https://language-agent-tutorial.github.io/

Two competing views

LLM-first view: We make an LLM into an agent!

- Implications: scaffold on top of LLMs, prompting-focused, heavy on engineering

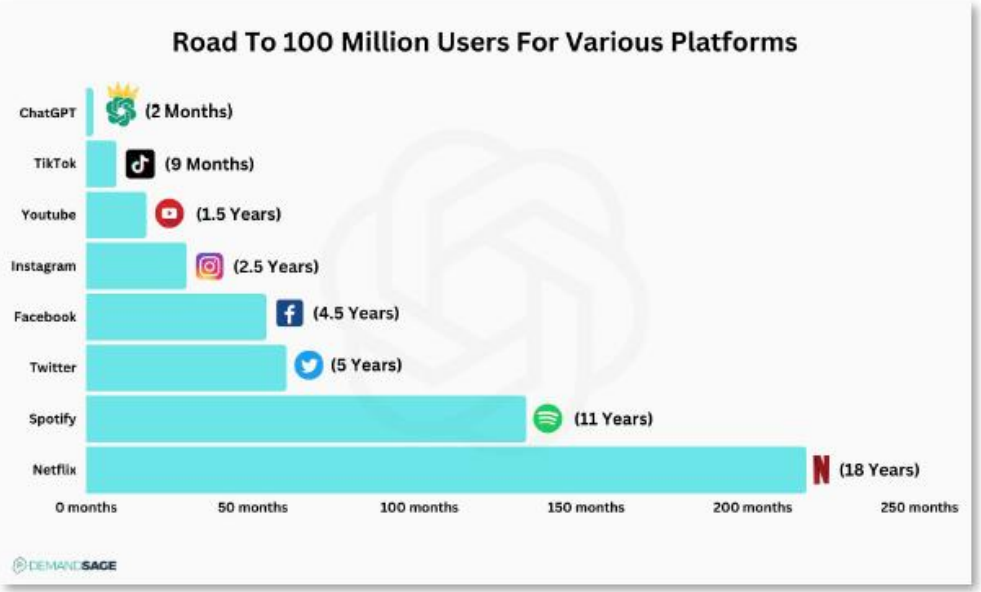
Agent-first view: We integrate LLMs into AI agents so they can use language for reasoning and communication!

- Implications: All the same challenges faced by previous AI agents (e.g., perception, reasoning, world models, planning) still remain, but we need to **re-examine them through the new lens of LLMs** and tackle new ones (e.g., synthetic data, self-reflection, internalized search)

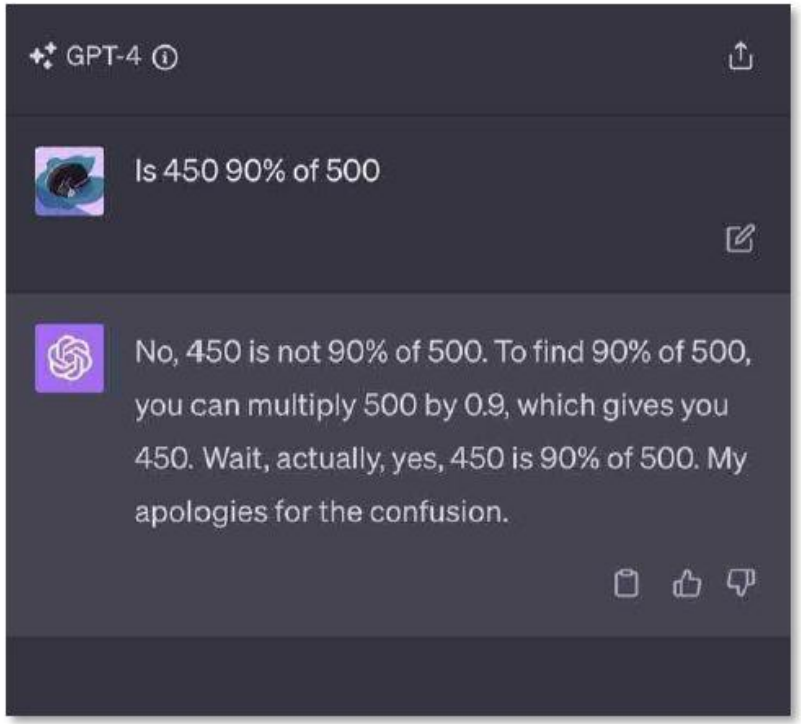
What's fundamentally different now?

Contemporary AI agents, with integrated LLM(s), can *use language as a vehicle for reasoning and communication*

- Instruction following, in-context learning, output customization
- Reasoning (for better acting): state inferences, self-reflection, replanning, etc.



https://www.demandsage.com/chatgpt-statistics/



https://www.reddit.com/r/ChatGPT/comments/16jvl4x/wait_actually_yes/

A generalized notion of ‘reasoning’

Unlike humans, LLMs (mostly) only have one mechanism (token generation) for perception, intuitive inferences, and symbolic reasoning; everything is effortful and takes a forward pass

GPT-4o

In this image, we see a young boy sitting on the floor, looking down at a pile of broken dishes. He appears to be sad or remorseful. In front of him, a woman, likely his mother or caregiver, is kneeling and looking at him with a concerned, questioning expression. Her body language, with her hands open and her facial expression, suggests that she's asking him something or expressing surprise or frustration about the broken dishes.

Perception

Intuitive Inference

Reasoning

One may alternatively call this ‘thought’ to avoid the over-loaded term of ‘reasoning,’ at the risk of further anthropomorphizing machines

Language agents: a new type of AI agents

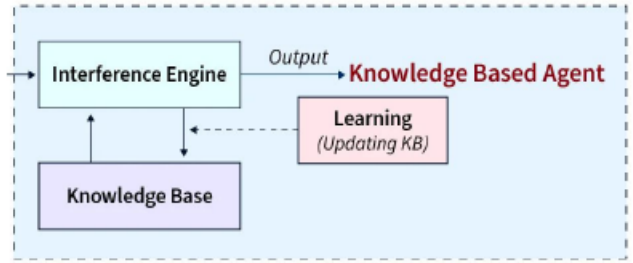
- These contemporary AI agents capable of using language for reasoning and communication are best called "**language agents**." They are qualitatively a different type of AI agents with language being their most distinct trait.

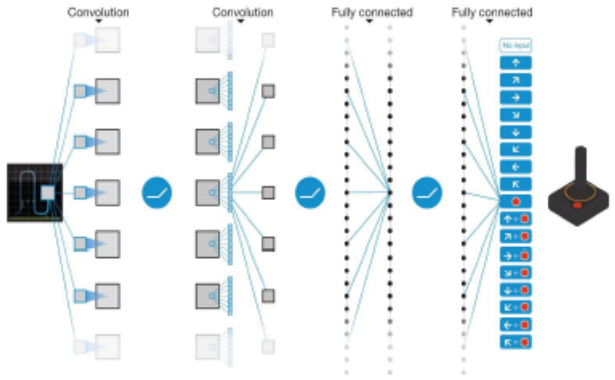
- What about *multimodal agents*?
  - While there's perception of other input modalities, language is still doing the heavy lifting (i.e., reasoning and communication)

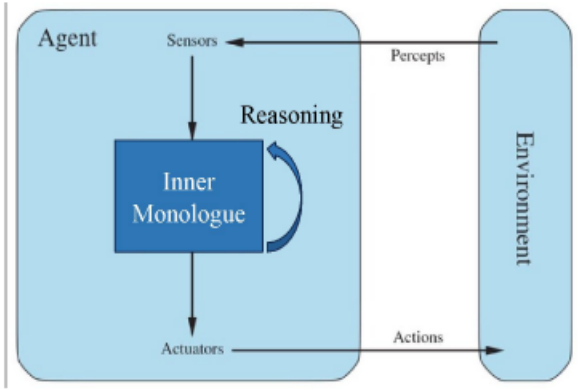
- What about simply *LLM agents*?
  - The key is using language for reasoning and communication, but that doesn't have to come from an LLM; that may turn out to be a means to an end
  - Maybe in a few years, we will move beyond LLMs, but the need for universal language understanding and production in agents will remain

https://language-agent-tutorial.github.io/

Evolution of AI agents







	Logical Agent	Neural Agent	Language Agent
Expressiveness	Low bounded by the logical language	Medium anything a (small) NN can encode	High almost anything, esp. verbalizable parts of the world
Reasoning	Logical inferences sound, explicit, rigid	Parametric inferences stochastic, implicit, rigid	Language-based inferences fuzzy, semi-explicit, flexible
Adaptivity	Low bounded by knowledge curation	Medium data-driven but sample inefficient	High strong prior from LLMs + language use

Image sources: https://www.scaler.com/topics/artificial-intelligence-tutorial/knowledge-based-agent/,
Mnih et al., "Human-level control through deep reinforcement learning." Nature (2015)

Role of LLMs in Agentic AI

Large Language Models (LLMs) play a **central but not sufficient** role in Agentic AI. They act as the *cognitive engine* of agentic systems, while autonomy, action, and interaction emerge from the surrounding architecture.

Large Language Models act as:

- Reasoning engines
- Planners
- Natural language interfaces
- Meta-controllers

LLMs enable:

- Task decomposition
- Tool selection
- Self-reflection and critique

Role of LLMs in Agentic AI

1. LLMs as the Reasoning Core

LLMs provide capabilities that were previously hard to engineer explicitly:

- **Natural language understanding** of goals, instructions, and context
- **Reasoning and abstraction**, including decomposition of tasks
- **Planning in language**, such as producing ordered steps or strategies
- **Generalization**, allowing agents to operate across domains

In agentic systems, the LLM functions as the *decision-making brain* rather than a final answer generator.

2. From Text Prediction to Decision Support

By themselves, LLMs:

- Predict the next token
- Operate passively
- Do not observe or act

Role of LLMs in Agentic AI

1. LLMs as the Reasoning Core

LLMs provide capabilities that were previously hard to engineer explicitly:

- **Natural language understanding** of goals, instructions, and context
- **Reasoning and abstraction**, including decomposition of tasks
- **Planning in language**, such as producing ordered steps or strategies
- **Generalization**, allowing agents to operate across domains

In agentic systems, the LLM functions as the *decision-making brain* rather than a final answer generator.

2. From Text Prediction to Decision Support

By themselves, LLMs:

- Predict the next token
- Operate passively
- Do not observe or act

In Agentic AI, the LLM is embedded in a loop where its outputs are interpreted as:

- **Plans**
- **Decisions**
- **Action selections**
- **Reflections or self-critiques**

This transforms pure language generation into **goal-directed behavior**.

Role of LLMs in Agentic AI

3. Planning and Task Decomposition

LLMs enable agentic systems to:

- Break high-level goals into **subtasks**
- Adapt plans when actions fail
- Generate alternative strategies

Example:

Goal: "Deploy a web service"
LLM-generated plan:

1. Check repository
2. Configure environment
3. Run tests
4. Fix errors if any
5. Deploy

The agent executes these steps using tools, not just text.

4. Interface Between Agents and Tools

LLMs act as a **translator** between human goals and machine actions:

- Human intent → natural language
- LLM → structured commands or tool calls
- Tools/APIs → real-world effects
- Observations → fed back to the LLM

In this sense, LLMs connect:

Language ↔ Action

Role of LLMs in Agentic AI

5. Memory and Reflection

In agentic systems, LLMs often support:

- **Short-term memory** (context, recent actions)
- **Long-term memory** (past tasks, preferences, failures)
- **Self-reflection**, such as evaluating success or diagnosing errors

This allows agents to improve performance over time without retraining the model.

6. What LLMs Do *Not* Provide

LLMs alone do **not** give:

- Autonomy
- Persistent state
- Tool execution
- Safety guarantees
- Control loops

These come from system design, orchestration logic, and constraints around the LLM.

Important distinction:

LLMs enable agentic behavior; they do not guarantee it.

Summary

LLMs provide language-based reasoning and planning; Agentic AI turns that reasoning into autonomous action through control loops, tools, and feedback.

Aspect	LLM Alone	Role in Agentic AI
Reasoning	✅ Yes	✅ Core function
Planning	Implicit (text)	✅ Explicit, executable
Action	❌ No	✅ Via tools
Autonomy	❌ Low	✅ High (system-level)
Learning	Static	Adaptive via feedback

Tools and Actions in Agentic Systems

Tools are **external capabilities** that an agent can invoke to gather information or affect the world.

They extend the agent beyond pure reasoning.

Typical Tool Categories

🔍 Information Tools

Used to observe or query the environment.

- Web search
- Database queries
- File reading
- Log inspection
- API calls for status or data

🛠️ Action Tools

Used to change the environment.

- Writing or editing files
- Executing code
- Calling APIs that modify state
- Sending emails or messages
- Triggering workflows
- Controlling software or hardware

🧠 Memory Tools

Used to persist or retrieve state.

- Vector databases
- Key–value stores
- Task histories
- Long-term summaries

Tools and Actions in Agentic Systems

What Are Actions?

An **action** is a **specific execution of a tool** chosen by the agent at a particular moment.

- Tool = *capability*
- Action = *invocation*

Example:

- Tool: run_tests
- Action: run_tests(test_suite="unit")

Actions produce **observable effects**, which feed back into the agent's next decision.

Tools vs. Actions (Key Distinction)

Concept	Description
Tool	What the agent can do
Action	What the agent actually does
Policy	How it chooses actions
Observation	Result of an action

Role of LLMs with Tools and Actions

In agentic systems, **LLMs do not directly act**. Instead, they:

1. Interpret observations
2. Decide which tool to use
3. Generate structured action requests
4. Reflect on outcomes

The surrounding system executes the action and returns feedback.

LLM = decision-maker
Tools = actuators

The Action Loop in Agentic Systems

A simplified control loop:

Observe → Reason → Select Tool → Execute Action → Observe Result → ...

This loop:

- Enables multi-step behavior
- Allows adaptation when actions fail
- Distinguishes agentic systems from chatbots

Examples

Example 1: Coding Agent

Goal: Fix a failing build

- Tool: run_tests
- Action: Execute tests
- Tool: edit_file
- Action: Modify source code
- Tool: commit_changes
- Action: Save fix

The agent iterates until the goal is met.

Example 2: Research Agent

Goal: Summarize recent literature

- Tool: web_search
- Action: Query recent papers
- Tool: extract_text
- Action: Read PDFs
- Tool: write_summary
- Action: Produce report

Tool Selection as a Decision Problem and Safety and Control Considerations

Tool Selection as a Decision Problem

Choosing which tool to use—and when—is part of the agent's **policy**.

Good agentic systems:

- Avoid unnecessary actions
- Respect costs and constraints
- Handle failures gracefully
- Retry or re-plan when needed

Poor tool use leads to:

- Infinite loops
- Unsafe actions
- Inefficiency

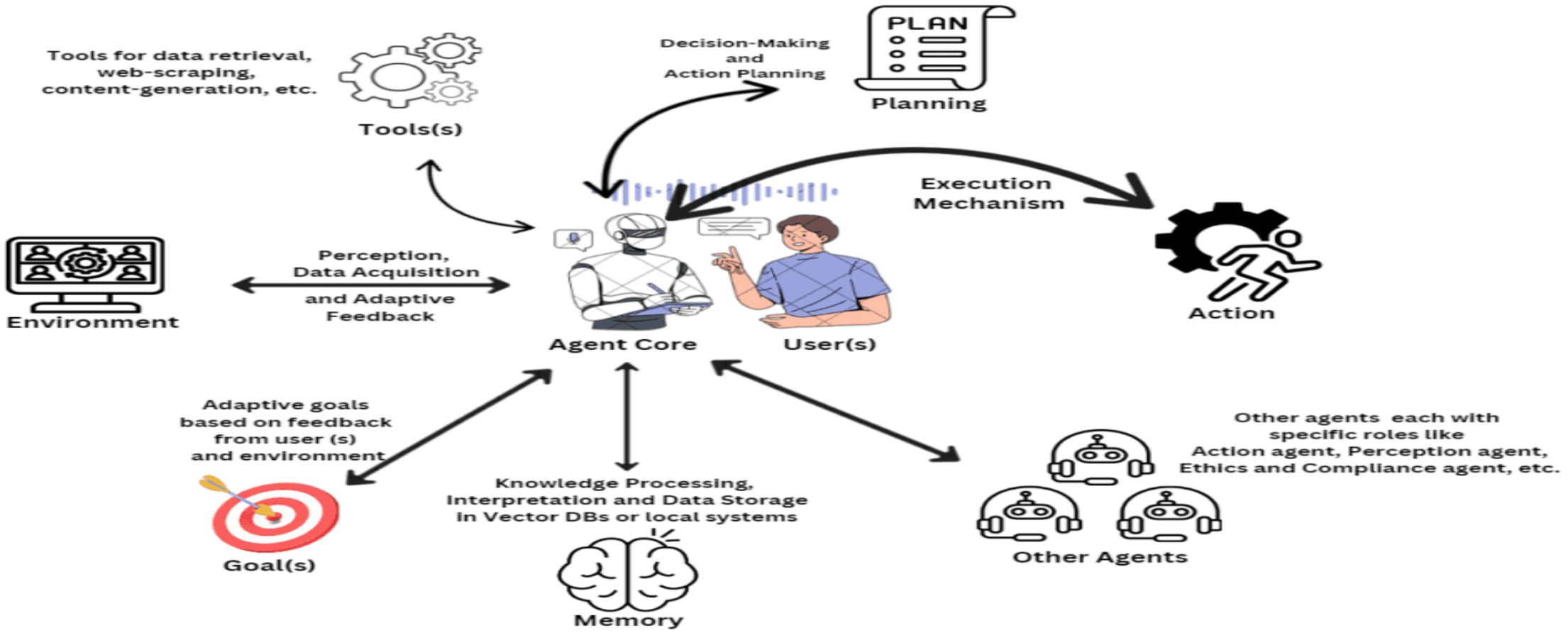
Safety and Control Considerations

Because tools can change real systems, agentic architectures often include:

- Tool permissioning
- Action validators
- Rate limits
- Human-in-the-loop checkpoints

Summary

- Tools provide capabilities, actions execute those capabilities, and agentic systems use both to turn reasoning into real-world impact.



https://learnopencv.com/agentic-ai/

THANK YOU FOR YOUR ATTENTION