What is agentic architecture?

What is agentic architecture?

Agentic architecture refers to the structure and design of agentic artificial intelligence (AI) frameworks. An agentic architecture is one that shapes the virtual space and workflow structure to automate AI models within an agentic AI system.

Agentic AI is a system or program that uses AI agents to autonomously perform tasks on behalf of a user or another system. Agentic architecture works to support and regulate the behavior of AI-powered agents working within a generative AI (gen AI) system. Agentic AI systems require its agents to be adaptive and navigate dynamic environments to achieve wanted outcomes.

The model is not so different from human psychology—agency refers to the ability to intentionally make something happen based on one's actions.[1] To achieve wanted outcomes, one must use planning, action, memory and reflection. These characteristics align with that of modern AI agents that are used in both single and multi-agent frameworks.

Advancements in machine learning (ML) algorithms and large language models (LLMs) such as OpenAI's GPT have driven the development of AI agents. It is the goal of the agentic architecture to provide a structure for an LLM to automate agents to complete complex tasks.

The autonomous or decision-making behavior of an AI agent depends on the infrastructure that enables it. Agentic architecture is designed to adapt to dynamic environments, enhancing interoperability.

For example, agents can interface with diverse data sources and formats, application programming interfaces (APIs) or systems. This adaptable behavior allows agents to make informed decisions.

Your subscription will be delivered in English. You will find an unsubscribe link in every newsletter. You can manage your subscriptions or unsubscribe here. Refer to our IBM Privacy Statement for more information.

How agentic architecture works

Agentic AI architecture should be composed of components that address the core factors of an agency: Intentionality (planning), forethought, self-reactiveness and self-reflectiveness.[2] These factors provide autonomy to AI agents so that they can set goals, plan, monitor their performance and reflect to reach their specific goal.

Agentic technology uses backend tool calling to gather up-to-date information, performs optimization of complex workflows and automatically generates tasks to achieve complex goals.

As it operates, the autonomous agent adapts to the user's preferences over time, offering a more personalized experience and delivering more detailed responses. This tool calling process can run without human input, unlocking broader possibilities for real-world AI applications.

Agentic vs. non agentic

Agentic architectures support agentic behavior within AI agents. AI agents are adaptable systems driven by machine learning models that can interact with external environments and use tools to complete specific goals. Not all AI agents are agentic. It depends on the complexity and capabilities of the orchestration framework or system.

Agentic architecture enables AI agents to act with a degree of autonomy and make decisions based on goals without the constant need for human input.[3] Autonomous AI agents require little to no human intervention to complete their specific tasks.

In nonagentic architectures, LLMs are capable of singular or linear tasks.[4] The AI model's function in a nonagentic architecture is to provide outputs based on input and context.

Without explicit orchestration, LLMs cannot retain new information in real time and often struggle with complex problems due to their limited context. For example, a few common AI applications that don't require complex agentic workflows include semantic analysis, chatbots and text generation.

The ideal agent architecture depends on the requirements of the application and use case. Single-agent systems excel at addressing focused, specific problems—essentially acting as individual problem solvers.

However, some challenges might call for the unique expertise of a specialized agent, while others might benefit from a collaborative approach involving multiple agents working together as a team.

Types of agentic architectures

The table provides a clear comparison of different AI agent architecture system types: vertical, horizontal and hybrid. It highlights their structures, key features, strengths, weaknesses and best use cases to help determine the most suitable approach for various tasks.

Single-agent architectures

A single-agent architecture features a single autonomous entity making centralized decisions within an environment.

- **Structure**

    - A single-agent architecture is a system where a single AI agent operates independently to perceive its environment, make decisions and take actions to achieve a goal.

- **Key features**

    - **Autonomy**: The agent operates independently without requiring interaction with other agents.

- **Strengths**

    - **Simplicity:** Easier to design, develop and deploy compared to multi-agent systems. Requires fewer resources because it does not need to manage multiple agents or communication protocols.
    - **Predictability**: Easier to debug and monitor because the agent operates independently.
    - **Speed**: No need for negotiation or consensus-building among multiple agents.
    - **Cost**: Less expensive to maintain and update compared to complex multi-agent architectures. Fewer integration challenges when deployed in enterprise applications.

- **Weaknesses**

    - **Limited scalability**: A single agent can become a bottleneck when handling high-volume or complex tasks.
    - **Rigidity**: Struggles with tasks that require multistep workflows or coordination across different domains.
    - **Narrow**: Typically designed for a specific function or domain.

- **Best use cases**

    - **Simple chatbots**: Chatbots can operate independently, don’t require coordination with other agents and perform well in self-contained, structured user interactions.
    - **Recommendation systems**: Personalized content recommendations such as the ones experienced at streaming services are straightforward enough for a single agent architecture.

Multi-agent architectures

Multi-agent architectures go beyond the AI capabilities of traditional, single-agent setups, bringing several unique benefits. Each agent specializes in a specific domain—such as performance analysis, injury prevention or market research—while seamlessly collaborating to solve complex problems.

Agents adapt their roles based on evolving tasks, helping to ensure flexibility and responsiveness in dynamic scenarios.

Multi-agent systems are more flexible. One agent might use natural language processing (NLP), another might specialize in computer vision. An agent might use retrieval augmented generation (RAG) to pull from external datasets.

There are many multi-agent framework providers such as crewAI, a Python-based multi-agent framework that operates on top of LangChain. Another AI solution is DeepWisdom, which offers MetaGPT, a framework that uses a structured workflow guided by standard operating procedures.

Vertical AI architectures

- **Structure**
    - In a vertical architecture, a leader agent oversees subtasks and decisions, with agents reporting back for centralized control.[5] Hierarchical AI agents know their role and report to or oversee other agents accordingly.
- **Key features**
    - **Hierarchy**: Roles are clearly defined.
    - **Centralized communication**: Agents report to the leader.
- **Strengths**
    - **Task efficiency**: Ideal for sequential workflows.
    - **Clear accountability**: Leader aligns objective.
- **Weaknesses**
    - **Bottlenecks**: Leader reliance can slow progress.
    - **Single point of failure**: Vulnerable to leader issues.
- **Best use cases**
    - **Workflow automation**: Multistep approvals.
    - **Document generation**: Sections overseen by a leader.

Horizontal AI architectures

- **Structure**
    - **Peer collaboration model:** Agents work as equals in a decentralized system, collaborating freely to solve tasks.[6]
- **Key features**
    - **Distributed collaboration**: All agents share resources and ideas.

    - **Decentralized decisions**: Group-driven decision-making for collaborative autonomy.

- **Strengths**
    - **Dynamic problem solving**: Fosters innovation.
    - **Parallel processing**: Agents work on tasks simultaneously.
- **Weaknesses**
    - **Coordination challenges**: Mismanagement can cause inefficiencies.
    - **Slower decisions**: Too much deliberation.
- **Best use cases**
    - **Brainstorming**: Generating diverse ideas.
    - **Complex problem solving**: Tackling interdisciplinary challenges.

Hybrid AI architectures

- **Structure**
    - Combines structured leadership with collaborative flexibility; leadership shifts based on task requirements.
- **Key features**
    - **Dynamic leadership**: Leadership adapts to the phase of the task.
    - **Collaborative leadership**: Leaders engage their peers openly.
- **Strengths**
    - **Versatility**: Combines strengths of both models.
    - **Adaptability**: Handles tasks requiring both structure and creativity.
- **Weaknesses**
    - **Complexity**: Balancing leadership roles and collaboration requires robust mechanisms.
    - **Resource management**: More demanding.
- **Best use cases**
    - **Versatile tasks**: Strategic planning or team projects.
    - **Dynamic processes**: Balancing structured and creative demands.

Agentic frameworks

Agentic frameworks refer to design architectures or models that define how agents (whether artificial or natural) can perform tasks, make decisions and interact with their environment in an autonomous, intelligent manner. These frameworks provide the structure and guidelines for how agents operate, reason and adapt in various settings.

Reactive architectures

Reactive architectures map situations directly to actions. They are reflexive, making decisions based on immediate stimuli from the environment rather than drawing on memory or predictive capabilities. They can’t learn from the past or plan for the future.

Deliberative architectures

A deliberative architecture is an AI system that makes decisions based on reasoning, planning and internal models of the world. Unlike reactive agents, deliberative agents analyze their environment, predict future outcomes and make informed choices before acting.

Cognitive architectures

A cognitive agentic architecture is an advanced AI system that mimics human-like thinking, reasoning, learning and decision-making.

These agents incorporate elements of perception, memory, reasoning and adaptation, each represented by individual modules, enabling them to operate in complex, uncertain environments while improving over time. This is the most advanced type of agentic architecture.

A **BDI architecture** (more commonly referred to as a model or framework) is designed to model rational decision-making in intelligent agents, and it's based on the belief-desire-intention (BDI) framework.

This architecture models human-like reasoning in a BDI agent, with:

- **Beliefs (B)**: The agent's knowledge of the world, which could include its understanding of the environment, current situation and any sensory data.

Example: "The door is closed."

- **Desires (D)**: The agent's goals or objectives, representing what it wants to achieve. Desires are not necessarily actions but high-level goals.

Example: "I want to enter the room."

- **Intentions (I)**: The course of action that the agent commits to in order to achieve its desires. Intentions represent planned actions that the agent is actively pursuing, considering its beliefs and desires.

Example: "I will open the door to enter the room."