06.10.2025 7 MIN READ

AI Agents: Cutting Through the Noise

AI AND ML

AI agents are transforming how systems operate autonomously. In this article, Denis Tomilin, our Solutions Architect, breaks down the essentials of agentic AI, its core components, and practical insights for building proactive, decision-making AI solutions.

Summarize with AI →

ARTICLE AUTHORS

Denis Tomilin
Solutions Architect

Colombia / Mexico / Remote.LATAM / Uruguay

AI/ML Engineer

Colombia / Mexico / Remote.LATAM / Uruguay

The article ends with a quiz — a quick way to test your understanding of agentic AI. So read carefully and quiz yourself afterward — **or jump straight to the quiz if you already feel confident about the topic!**

What Is Agentic AI?

Agentic AI refers to systems that can perform tasks independently on behalf of a user or another system. Unlike generative AI, which focuses on creating new content in response to prompts, agentic AI prioritizes decision-making and proactivity. It responds to context, plans actions, and adjusts its behavior dynamically.

AI Agent Components

An effective AI agent is built on four key elements:

Goals: Input, instructions, information, and/or context to operate. At least one of these is required. But the more, the better.

Tools: Access to external systems that enable the agent to act, exchange data, and execute tasks across different platforms.

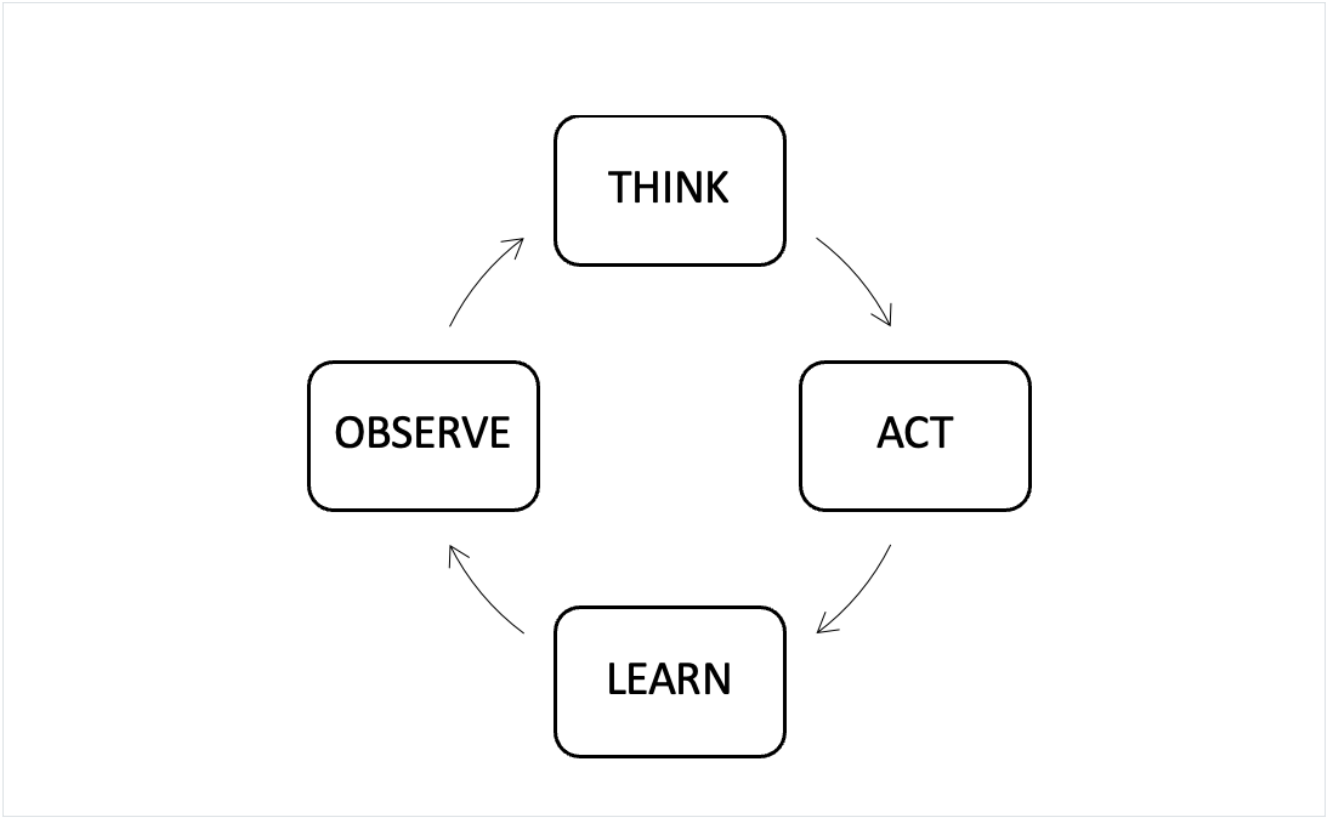
Memory: This is the storage of findings, insights, and context, supporting the feedback loop essential for multi-step tasks.

LLM: The reasoning engine that plans, decides, and leverages tools to achieve the goal.

AI Agent Core Loop

These components interact within the agent's core loop: the agent plans, executes, observes the outcome, and adapts.

Mistakes can occur. For example, someone may not be available for a scheduled call. There are many possible scenarios. That's why the iteration is critical: Agents refine their actions until the task is completed successfully.



Example: Payment Risks Assessment

Here's an example of an agent we are currently developing.

Goal: Evaluate the risk of a payment and assign risk classifications. Based on the evaluation, the agent either allows or rejects the payment.

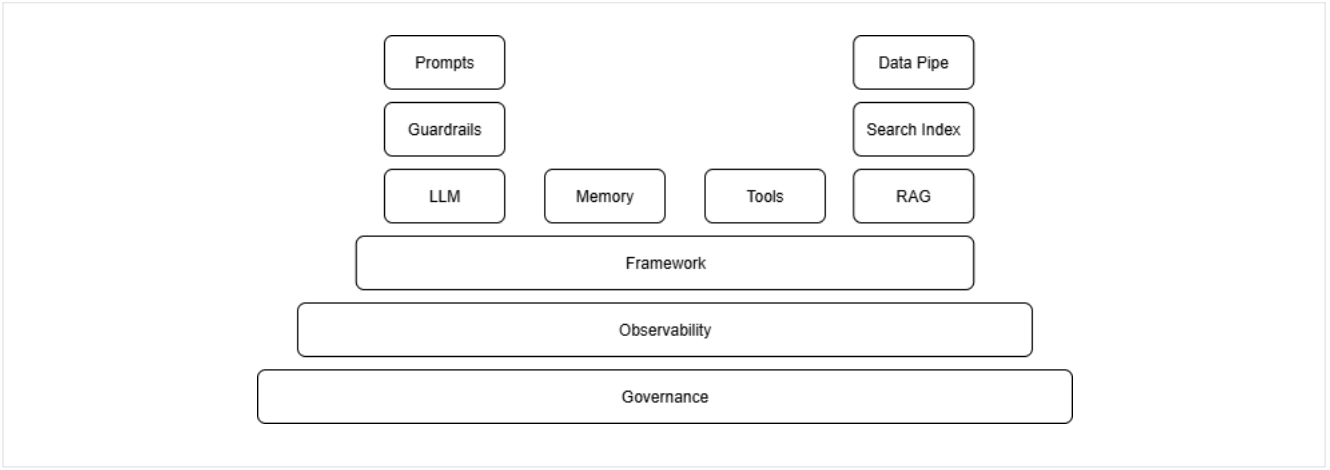
To achieve this, the agent must interact with external systems using **tools** such as:

- ID verification
- Fingerprinting
- Behavior Statistics
- Payments System Data

Loop: Each time new insights are collected, the data is re-evaluated, and the following action is determined. This cycle repeats until enough information is available to confidently allow or reject the payment.

Agentic AI Structure

Let's explore what makes up an AI agent.



Choosing the Right LLM

In the agentic AI structure, LLMs determine how effectively an agent can reason and plan. With so many LLMs available, it can be tempting to select one at random. However, your choice should be guided by your specific requirements, agreements, or partnerships. The wrong LLM can undermine your project.

Key technical factors include:

Model: Choose a model that fits your needs. The most expensive model is not always the best choice. Consider reasoning capabilities, size (small or large), and specific functionalities.

Context Window: Determines how much information the model can process at once.

Hosting: Some models can run on-premises, while open-source options like Mistral and Ollama are also available.

Hardware Needs: CPU vs. GPU performance requirements.

Guardrails: Set boundaries on how LLMs interact with people to prevent harmful or unintended behavior.

Memory

Memory enables continuity. At its core, memory is a collection of recent messages or insights that serve as context. Without it, agents will simply repeat actions blindly. Options include short-term, long-term, rolling buffers, and windowed memory.

When designing memory, consider:

Variability: Implementations and frameworks will vary, affecting how agents interact.

Larger Memory: The more data you have, the more hallucination-prone it is.

Memory Compression: Be careful—over-compression may result in loss of crucial information.

Short-Term Memory: May overlap with long-term memory.

RAG (Retrieval Augmented Generation)

RAG, also known as CAG (Cache Augmented Generation), essentially refers to enabling agents to interact with your data. There are two primary techniques that agents use to access external information:

Search Engine: Agents can query mainstream search engines to retrieve data.

Vector Database: Stores data as mathematical representations, allowing agents to search and retrieve relevant information.

Establishing a robust database is important. The success of RAG hinges less on the technology chosen and more on the strength of the database: data mapping, chunking, relevance scoring, and semantic consistency are crucial. When dealing with varied data types—images, text, tables—the process can become resource-intensive and slow.

Tools

Agents require tools to act on stored or retrieved data. You may encounter different terms, such as actions or MCP, but they all refer to **API interactions,** granting agents the ability to connect with external systems.

Think of tools as API clients (REST API, OpenAPI, GraphQL). Given challenges like security, encryption, multi-tenancy, or the lack of an available API, building custom tools is often necessary. Use protocols and patterns like MCP, A2A, or ACP to facilitate these interactions.

Framework

Nowadays, there are many different frameworks available. Essentially, frameworks bind everything together. So, you can pick whichever you like and have a functional agent up and running quickly.

However, remember that both technology and the market are **continuously evolving**. Industry leaders like NVIDIA (with AgentIQ), Microsoft (with Autogen), and OpenAI (with Agents) offer robust starting points. It makes sense to leverage their solutions where possible.

Frameworks also **differ** in the technologies they support. Some are geared toward Python, while others focus on TypeScript or .NET. Python is no longer the sole language dominating this space. Additionally, infrastructure and distributed computing challenges persist, so it's worth considering your architectural requirements carefully.

Observability

This is what differentiates expert systems from amateur ones. Without observability, agents become black boxes, which can be dangerous in enterprise settings.

Be mindful of:

- Silent features
- Infinite feedback loops
- Lack of reasoning traceability
- Compliance issues

APM tools such as **DataDog** can support observability, though many are still maturing in this area. Maintaining these systems also adds cost. Just as important, transparency around data use is essential. In agentic AI, observability must extend well beyond API call tracing to provide a complete view of agent behavior and overall system health.

Governance

This is often one of the major hurdles in the adoption of AI agents. Privacy, ownership, compliance, and security must all be addressed. Many companies are in the process of figuring out how to accomplish that. For example, Microsoft is actively working on enterprise readiness for AI in data handling.

Enterprises typically define their own policies, but client-facing transparency is non-negotiable. Some rely on vendors like Microsoft for enterprise readiness, while others self-host for greater control.

Power Techniques: AI Teams

Agent Mesh (or AI Team/Agency)

Sometimes a single agent isn't enough. Agent Mesh multi-agent system (also called AI teams, swarms, agencies) enables more complex workflows, horizontal scalability, and distributed computing, helping to mitigate the limitations of relying on a unique agent.

How to reduce the entropy of a single agent:

- Distribute the load
- Narrow down the scope for each agent
- Simplify troubleshooting
- Increase reusability

Two main collaboration models exist to implement the agent:

- **Hand-off (Delegation)**

  One agent passes its context or task to another agent. This approach reduces the data size and allows the context to be adapted for the receiving agent. However, passing the context around can cause several issues:

  - An agent may get stuck
  - Bias in context hand-off
  - Hallucinations, potentially creating echo-feedback loops

Coordination

Triage or coordination agents organize the workflow and manage multiple tasks in parallel. These agents are typically more advanced, complex, and expensive than regular agents. Keep in mind that there are too many coordination patterns to consider. No need to overthink, just use the ones you get out of the box.

Human-in-the-Loop

As we've seen, agents can get stuck, require additional actions with external systems, or make mistakes. Sometimes, human judgment is necessary. This is where the human-in-the-loop concept comes in. When needed, agents can use tools to request human input by:

Creating a task for a human

Request input within the app

Send a notification

When available, frameworks should provide humans with the necessary context to make a decision. Sometimes, text may not be enough. The trade-off: added reliability but potential slowdown.

Quiz

Got the basics of Agentic AI? Great — now put yourself to the test!

DataArt

Conclusion

Agentic AI is more than an industry buzzword; it's a practical framework for building autonomous, decision-making systems. By focusing on the essentials: clear goals, the right tools, reliable memory, and thoughtful governance, organizations can design agents that are both powerful and trustworthy.

If you're exploring how to apply agentic AI in your projects, now is the time to start experimenting, testing, and building pilots that can grow into production-ready systems.

You may also find this interesting

BLOG POST

"We're about a year away from a critical moment"

BLOG POST

"I find pleasure in coding. So why should fun part to AI?"

MOST WANTED

1 of 3

Senior Power BI Engineer

ARMENIA, BRAZIL, BULGARIA, COLOMBIA, CYPRUS, GEORGIA, INDIA, KAZAKHSTAN, LATVIA, MEXICO, POLAND, REMOTE.LATAM, ROMANIA, SERBIA, UKRAINE, URUGUAY

We are looking for a Senior Business Intelligence Specialist to design impactful data visualizations and dashboards that support senior leadersh across Fixed Income departments, playing a key role in our global technology transformation

Senior AI Consultant

USA

DataArt AI Lab seeks a Senior AI Consultant to collaborate with leadership and cross-functional teams in designing and implementing AI solutio boost productivity, streamline workflows, and drive innovation across diverse industries

Senior Java Developer

INDIA

We're seeking an experienced Senior Java Developer to build scalable, real-time microservices using the latest Java technologies in a collabor cutting-edge environment

Senior Business Analyst with NetSuite Experience

ARMENIA, BRAZIL, BULGARIA, COLOMBIA, CYPRUS, GEORGIA, INDIA, LATVIA, MEXICO, POLAND, REMOTE.LATAM, ROMANIA, SERBIA, URUGUAY

Advanced technological solutions, skilled technical specialists, and the importance of human relations are what make the work on projects of thi company comfortable and interesting

DataArt supports tech talent at every stage with continuous learning, challenging projects, and career paths tailored to individual goals. Flexible remote work and a global community create a culture where people thrive — and future-proof their careers.

Vacancies
About Us
What we Do
For Beginners
Internships
For Pros
Work at DataArt
How to join us?
Workplace options
English at DataArt
Centers of Development
Stories

IT Museum →
IT engineering heritage

Proggy-Buggy →
Programming Olympiad

IT Camp →
Conference for IT beginners

Try me! →