What Is an Agentic System?

An AI agent is one moving part. An agentic system is the whole machine — the agents, the tools, the memory, the orchestration, and the humans who sit above it. Here is how that machine is actually put together, and why the shift from generative to agentic changes what software can do.

Alain AI Lab Research · Published July 13, 2026 · 10 min read

Most people meet the word “agent” before they meet the word “system.” They picture a single clever program that reads a goal, thinks, and acts. That picture is correct but incomplete. In practice, almost nothing useful is built from one agent alone. It is built from an **agentic system** — an arrangement of one or more agents, the tools they can call, the memory they share, the code that routes work between them, and the human oversight that decides when the machine is allowed to act on its own. Understanding the system, not just the agent, is what separates a demo from something you would trust with a wallet. If you want the ground-level definition of the single building block first, start with what an AI agent is in crypto and then come back to see how those blocks are assembled.

AT A GLANCE

CORE IDEA	BUILDING BLOCK	SYSTEM SCOPE
Generative responds; agentic acts	**One agent = perceive, reason, act, loop**	**Agents + tools + memory + orchestration**
ARCHITECTURE	NAMED TREND	TOP RISK
Single-agent or multi-agent	**“Agentic AI,” Gartner, Oct 2024**	**Prompt injection & excessive agency**

01

The Word Everyone Uses and Few Define

“Agentic” became one of the most repeated words in technology across 2024 and 2025. The research firm Gartner named agentic AI a top strategic trend in a press release dated October 21, 2024, and from there the term spread into every product deck and pitch. But repetition is not definition. An **agentic system** is any software arrangement in which an AI model does not merely produce an answer but pursues a goal — taking actions, observing results, and adjusting — with some measure of independence. The key phrase is “pursues a goal.” A system is agentic to the degree that it can be handed an objective rather than a script, and still make progress.

That definition deliberately spans a range. A single agent calling one tool in a loop is a minimal agentic system. A fleet of specialised agents coordinating through a shared plan is a large one. What they have in common is not size but posture: the software is oriented toward *doing*, not just *describing*. Once you hold that idea, the marketing noise gets easier to filter. The question is never “does it say agentic?” It is “can it be given a goal and left to act?”

02

Generative Responds, Agentic Acts

The cleanest way to understand agentic systems is against the thing they grew out of: **generative AI**. A generative system takes an input and returns an output — a paragraph, an image, a block of code. It responds. It is brilliant and it is passive; it waits for you, produces one turn, and stops. Everything it does happens inside the conversation. Nothing happens in the world unless a human copies the output and acts on it.

An agentic system inverts that relationship. The model is still generative underneath — the same kind of language model produces the text — but it is wired into a loop that lets it *act*. It can call a function, read the result, decide what to do next, and repeat until the goal is met or a limit is hit. The generative model becomes the reasoning core; the surrounding system gives it hands. This is the paradigm shift the industry keeps pointing at: not a smarter model, but a model placed inside machinery that turns its words into actions. Generative AI answers the question. Agentic AI goes and does the thing the answer described.

> A generative model tells you how to rebalance a portfolio. An agentic system reads the balances, decides the trades, and — if you let it — submits them. Same underlying model. Completely different consequences.

03

The Anatomy of an Agentic System

Strip an agentic system to its parts and you find a small, recurring set of components. First, one or more **agents** — the reasoning units, each a language model wrapped in a loop of perceive, reason, act, observe. Second, **tools** — the functions, APIs, and integrations that let an agent affect the outside world: a search endpoint, a database query, a wallet transaction. Third, **memory** — the store that lets the system carry context beyond a single model call, because the model itself is stateless and forgets everything between turns. Fourth, **orchestration** — the code and logic that decides which agent runs when, how results pass between them, and when to stop. Fifth, **oversight** — the humans and guardrails positioned to approve, veto, or bound what the system does.

None of these is exotic on its own. What makes the arrangement powerful, and dangerous, is the loop that binds them. Because the agent observes the result of each action before choosing the next, the system can handle tasks nobody scripted in advance. If you want the mechanical detail of how that perceive-reason-act cycle runs inside a single agent, the piece on how AI agents work walks through the loop step by step. The system view simply asks: how many of these loops are running, and who is coordinating them?

04

Single-Agent Versus Multi-Agent

The first real architectural fork is how many agents you run. A **single-agent system** is one reasoning unit with access to a set of tools. It is simpler to build, easier to reason about, and easier to debug because there is one place where decisions happen. For a large share of real tasks, a single well-equipped agent is not just enough — it is the better choice, precisely because there are fewer moving parts to fail.

A **multi-agent system** breaks the work across several agents, each often specialised: one plans, one researches, one writes, one checks. The appeal is modularity — each agent can be tuned to its narrow job, and the division mirrors how human teams split labour. The cost is coordination. Every hand-off between agents is a place where context can be lost, instructions can drift, and errors can compound. More agents is not automatically more capability; frequently it is more surface area for things to go wrong. The honest engineering rule is to reach for multiple agents only when a single one genuinely cannot hold the task, not because a diagram with more boxes looks more impressive.

05

Orchestration: The Part That Actually Matters

If agents are the workers, **orchestration** is the management layer — and it is where most of the real design happens. Anthropic's engineering essay "Building Effective Agents," published December 19, 2024, drew a distinction that has become a reference point: **workflows**, where language models are orchestrated through predefined code paths, versus **agents**, where the models dynamically direct their own processes. A workflow is predictable and auditable because a human wrote the route in advance. A truly agentic loop is flexible but less predictable because the model chooses the route as it goes.

Between those poles sit named patterns. **Prompt chaining** passes the output of one step into the next in a fixed line. The **orchestrator-workers** pattern uses a central agent to break a goal into subtasks and delegate them to worker agents, then assemble the results. Each pattern trades autonomy for control in a different ratio. The craft of building an agentic system is choosing the least autonomous pattern that still solves the problem — because every degree of freedom you hand the model is a degree of freedom you can no longer fully predict.

06

The Frameworks That Build Them

You rarely wire an agentic system from raw model calls. A layer of frameworks now exists to handle the plumbing — the loops, the tool calls, the message passing, the state. In late 2024 Anthropic introduced the **Model Context Protocol**, an open standard published November 25, 2024 for connecting AI systems to tools and data sources through a common interface, so that every integration need not be bespoke. On the orchestration side, Microsoft announced its **Agent Framework** in public preview on October 1, 2025, unifying its earlier AutoGen and Semantic Kernel efforts into a single stack for building multi-agent systems.

These frameworks are not the intelligence; they are the scaffolding that lets the intelligence be assembled reliably. If you want to see how the major open frameworks compare — how they structure multi-agent coordination, roles, and hand-offs — the report on AI agent frameworks and multi-agent systems lays the landscape out. The point for the system view is that the tooling has matured to the stage where assembling several coordinated agents is an engineering choice, not a research project.

07

Where Humans Sit — and Where Crypto Raises the Stakes

An agentic system is defined as much by its brakes as by its engine. Oversight is usually described in three postures. **Human-in-the-loop** means the system pauses for approval before consequential actions. **Human-on-the-loop** means it acts on its own but a person monitors and can intervene. **Human-out-of-the-loop** means it runs unattended. Most responsible deployments keep a human in or on the loop wherever an action is expensive or irreversible — and this is exactly where blockchains sharpen the design.

In ordinary software, a bad automated action can often be rolled back. On a public chain, a confirmed transaction is final; there is no support desk to reverse it. That finality is why crypto agentic systems lean on constrained execution rails — account-abstraction wallets under the ERC-4337 standard, session keys that grant narrow time-boxed permissions, and multi-party computation that splits signing authority so no single agent holds an unrestricted key. The design goal is blunt: give the system enough authority to be useful and not one degree more. A crypto agentic system without hard limits on what it can sign is not autonomous. It is unsecured.

08

The Honest Risks of Systems That Act

Every property that makes an agentic system useful also makes it exposed. The OWASP security project ranks **prompt injection** as the number-one risk to LLM applications in its 2025 list — the class of attack where hostile text, buried in a webpage or a document the system reads, hijacks the agent's instructions. In a single agent that is bad enough. In a multi-agent system it is worse, because a poisoned instruction or a wrong result can propagate: one agent's error becomes another agent's trusted input, and the mistake compounds down the chain. OWASP names a companion failure, **excessive agency** — granting a system more permission and autonomy than its reliability justifies.

History is a useful corrective here. The first wave of fully autonomous agentic systems — AutoGPT in March 2023, BabyAGI the same year — showed both the promise and the fragility: they would loop, lose the thread, repeat actions, and burn resources without finishing. The tooling has improved enormously since, but the underlying lesson holds. An agentic system is only as trustworthy as its narrowest permission and its most watchful human. The right question to ask of any such system is not "how capable is it?" but "what is

the worst thing it is allowed to do without asking?” A system you would trust answers that question with a short, deliberate list.

“Prepare thy work without, and make it fit for thyself in the field; and afterwards build thine house.”

PROVERBS 24:27

METHODOLOGY & SOURCES

This report was produced by a parallel multi-agent research process, with each claim independently checked and labelled verified, partial, or uncertain before inclusion. Figures are stated as approximate ranges where precise public data does not exist; no specific decimals, dates, or dollar figures were invented, and unverified “nine-figure AI-agent hack” claims were excluded.

Primary references include Gartner’s October 21, 2024 strategic-trends press release naming agentic AI; Anthropic’s “Building Effective Agents” (December 19, 2024) on workflows versus agents and the orchestrator-workers pattern; the Model Context Protocol announcement (November 25, 2024); Microsoft’s Agent Framework public preview (October 1, 2025); the OWASP Top 10 for LLM Applications (2025) on prompt injection and excessive agency; and the ERC-4337 account-abstraction standard. Educational content only; not financial advice.

Related reading: What Is an AI Agent in Crypto? · How Do AI Agents Work? · AI Agent Frameworks & Multi-Agent Systems

Alain AI Lab — independent crypto & AI research.
intelligencecrypto.org · This document is for educational purposes only and is not financial advice.