RITHMO GUIDE

What is an AI agent?

What AI agents are, what they aren't, and how they actually work.

September 2026 / 12 min read

"AI agent" has quickly become one of the most widely used and loosely defined terms in artificial intelligence. Depending on who is talking, an agent might sound like a chatbot with a few extra capabilities, an automated workflow, a digital employee, or an autonomous system capable of running an entire business function.

There is some truth in all of those descriptions. This guide removes the jargon and explains what an AI agent actually is, how companies build them, where they run, and what they can realistically do today.

What an AI agent is

There isn't one universally accepted definition. A useful business definition is:

> An AI agent is software that uses an AI model to pursue a goal, with some ability to decide what to do along the way.

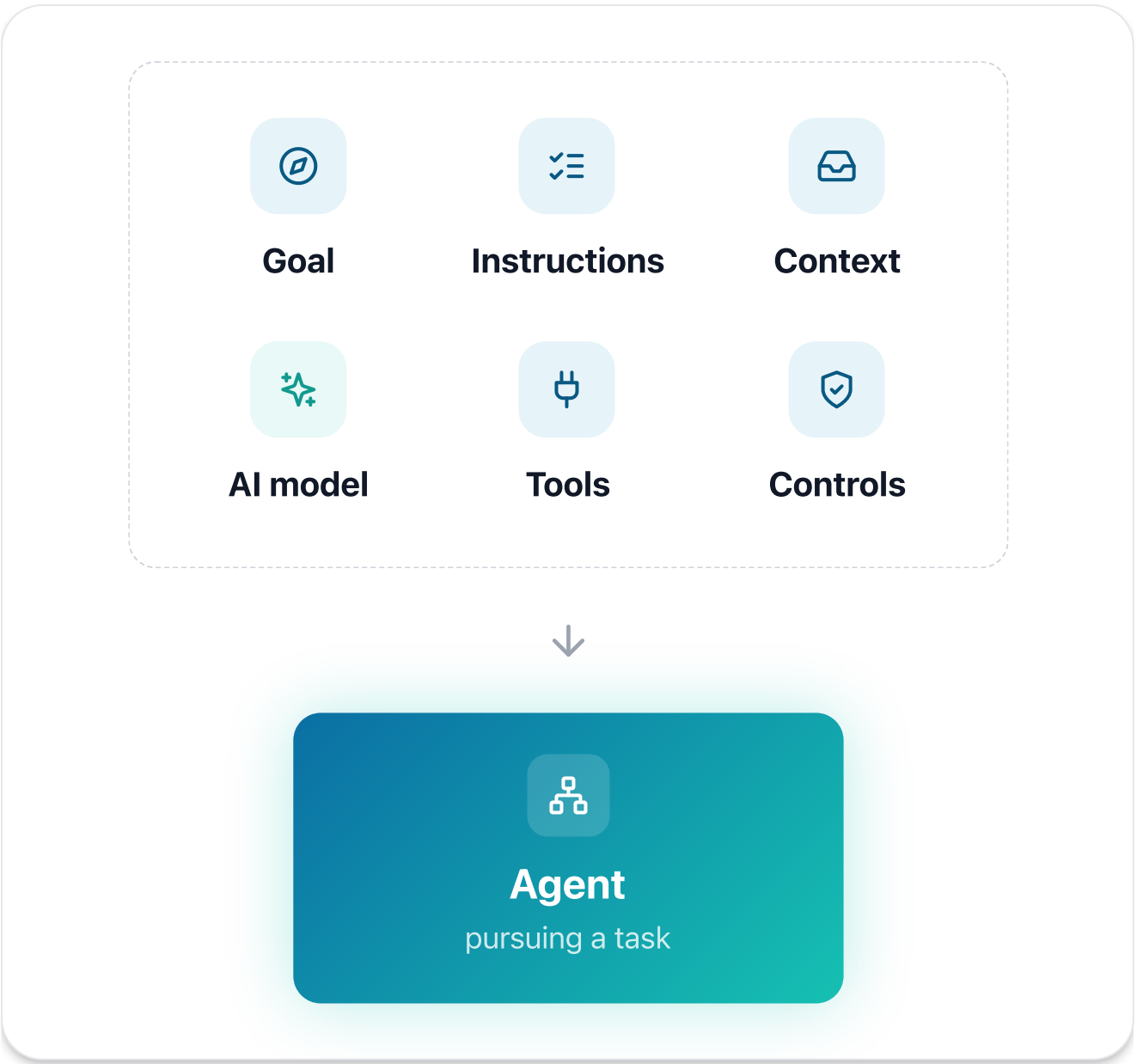
That last part matters. Traditional software generally follows logic that developers explicitly define:

If X happens → do Y.

An agent can instead be given an objective:

Accomplish X, using these tools, while following these rules.

The AI model can then determine what information it needs, which tools to use, what steps to take, and how to respond to what happens. The amount of freedom it gets can vary enormously.



An agent is more than the AI model

This is one of the easiest things to misunderstand. GPT, Claude, Gemini and other large language models are not, by themselves, the entire agent. Think of the AI model as the reasoning engine inside the system. An agent typically surrounds that model with several other components:

Goal. What is the agent supposed to accomplish?

Instructions. How should it behave, and what rules should it follow?

Context. What information does it have available for the current decision?

Tools. What can it access or do? Search the web, read Salesforce, query a database, send an email, modify a file?

AI model. The model evaluates the available information and determines what to do next.

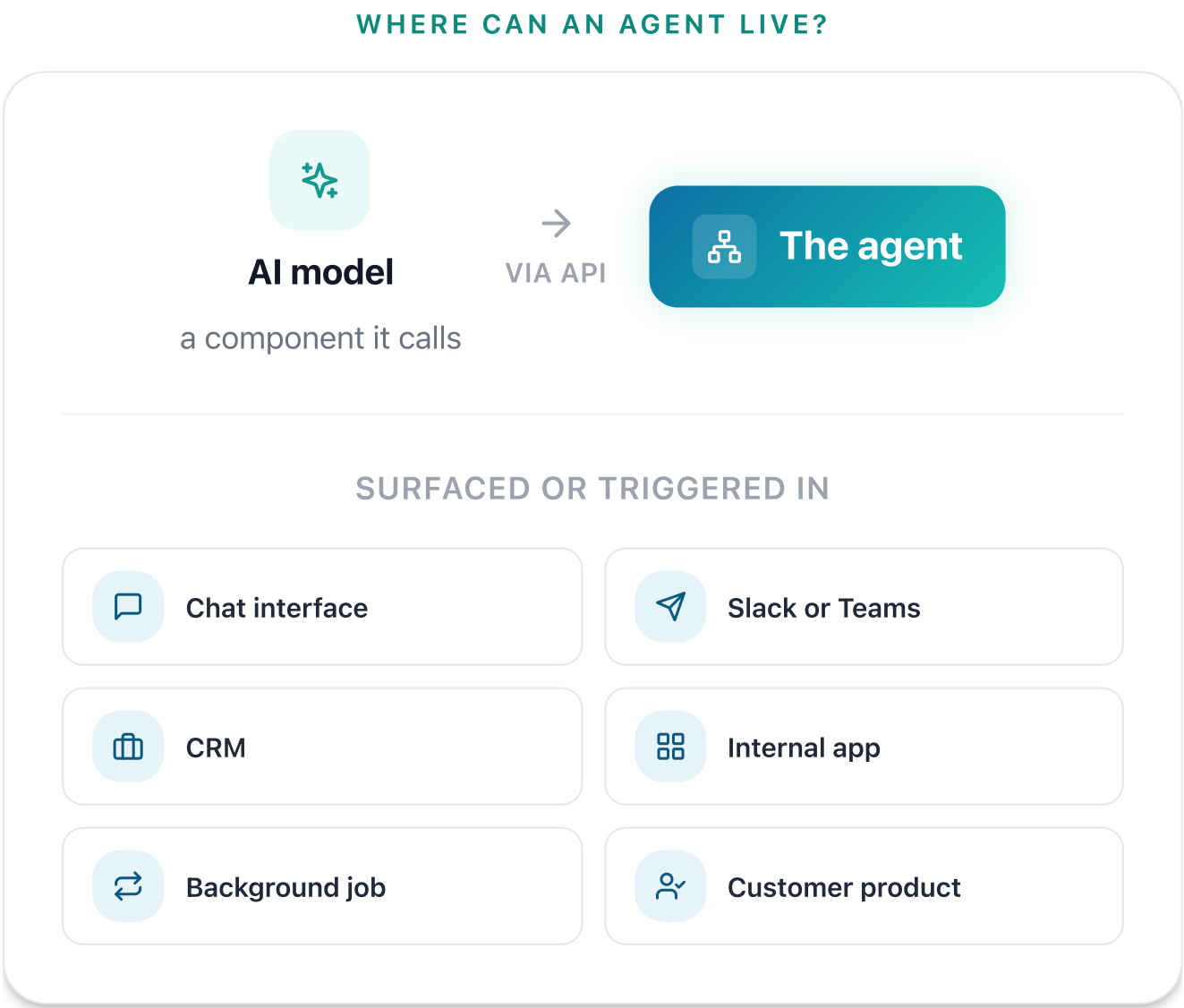
Controls. What is the agent allowed to do independently, and what requires human approval?

The resulting software system is the agent.

So where does an agent actually live?

There is no single answer. An agent does not have to live inside ChatGPT, Claude or another chat application. A company can build an agent as its own software application running in its own cloud environment, inside its own infrastructure, or on infrastructure provided by an AI or agent-platform vendor. The application then connects to an AI model.

For example, a company might build a sales agent running inside its own environment and connect it to an OpenAI model through OpenAI's API. The employee using that agent might never visit ChatGPT at all. The company could put its own interface in front of the agent, trigger it automatically when something happens, expose it through Slack, embed it inside an existing application, or have it operate largely in the background.



Then what is an agent inside ChatGPT or Claude?

Products such as ChatGPT and Claude increasingly provide their own agentic capabilities. In that case, the product provider has already built much of the infrastructure required to let the AI perform multi-step work and interact with tools. That makes agents accessible to people who aren't building their own software.

But that's only one way of deploying an agent. An enterprise building an agent for a core business process may instead create its own application and use models from OpenAI, Anthropic, Google or another provider as components inside it. The underlying idea is similar. The packaging, infrastructure, tools, controls and user experience can be completely different.

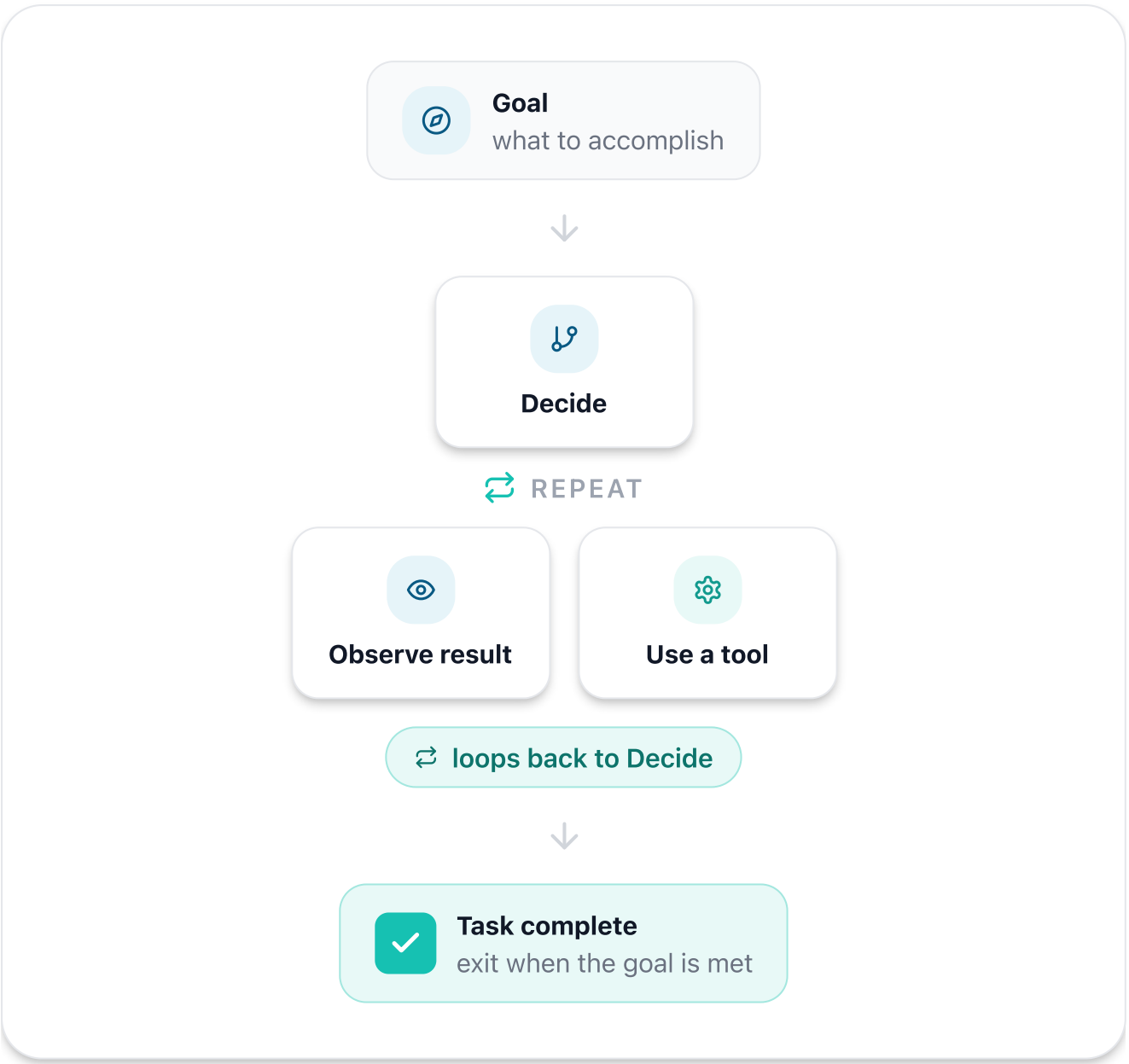
How does the AI model plug into an agent?

Usually through an API. An API is simply a way for one piece of software to communicate with another. The agent's software sends the AI model the objective, instructions, relevant information, available tools, and results from previous steps. The model processes that and returns a response.

That response might be "I have completed the task." Or it might effectively be "I need to check Salesforce before I can continue." The surrounding agent software then performs that operation, gives the result back to the model, and lets the model determine what to do next. This can happen repeatedly.

The agent loop

This repeated process is at the heart of many agents. Given the goal "research this prospect and prepare an appropriate outreach email," the agent might look up the prospect in Salesforce, review previous interactions, research the company, search for recent developments, determine a messaging angle, draft the email, and submit it for approval. A different prospect might cause the same agent to take a different path. That ability to adjust its process based on what it discovers is part of what makes an agent different from a traditional predetermined workflow.



Does an agent live inside one giant chat thread?

Not necessarily. A chat conversation is one way of interacting with an agent, but it is not what defines the agent. An enterprise agent can operate without a traditional chat interface at all: a new lead enters Salesforce, the agent starts on its own, researches the account, prepares outreach, and sends the result to a salesperson in Slack.

There may never be a person sitting in a chat window talking to it. The "conversation" happening behind the scenes is really an ongoing exchange between the agent software and the AI model as the task progresses.

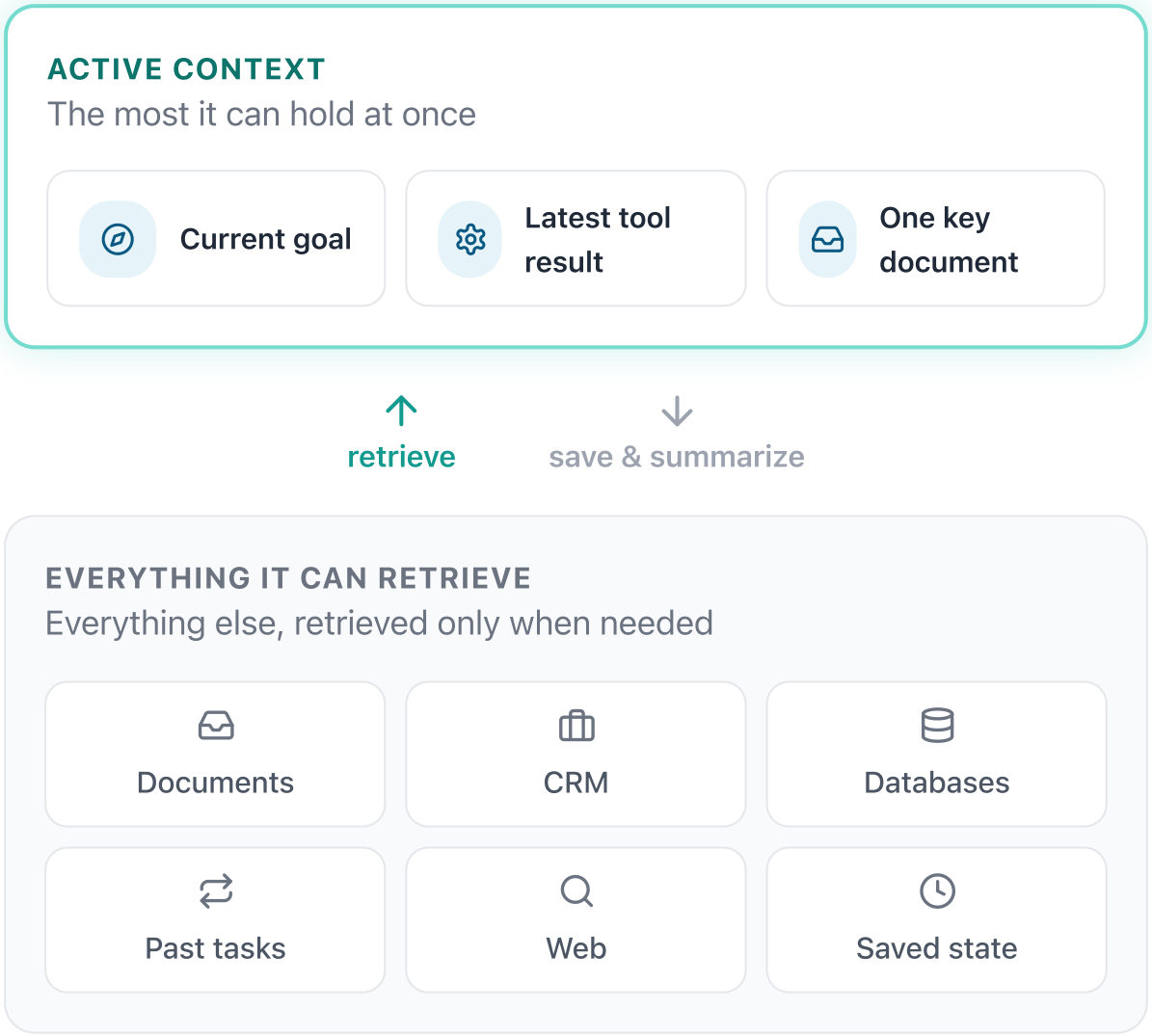
Does an agent eventually run out of context?

AI models have a finite context window. In simple terms, there is a limit to how much information the model can actively consider at one time. So a sophisticated agent cannot simply keep feeding every instruction, document, tool result, previous decision and historical interaction into the model forever.

Agent developers handle this in several ways. They can keep only the information relevant to the current task, summarize previous work, save information outside the model and retrieve it later, let the agent search files and databases when information becomes relevant, and save the state of a long-running task so it can stop and resume. So an agent can operate for much longer than a single context window, even though the model itself still cannot consider unlimited information at once. The surrounding software manages what reaches the model at any given moment.

CONTEXT WINDOW VS. THE AGENT

The model can work with only what fits on the **desk** at once. The agent fetches more from the **library** whenever it is needed.



Does the agent remember what happened yesterday?

It can, but that capability has to come from somewhere. The model itself does not automatically contain a permanent record of everything an enterprise agent has ever done. The application can save information externally in databases, files, task histories or other storage and make it available again later.

How much to save, for how long, and how to retrieve it are architectural choices. This is one reason two agents using exactly the same underlying AI model can behave very differently. The model is only one part of the system around it.

Are all agents highly autonomous?

No. This is probably the biggest misconception created by current AI marketing. There is a broad spectrum.

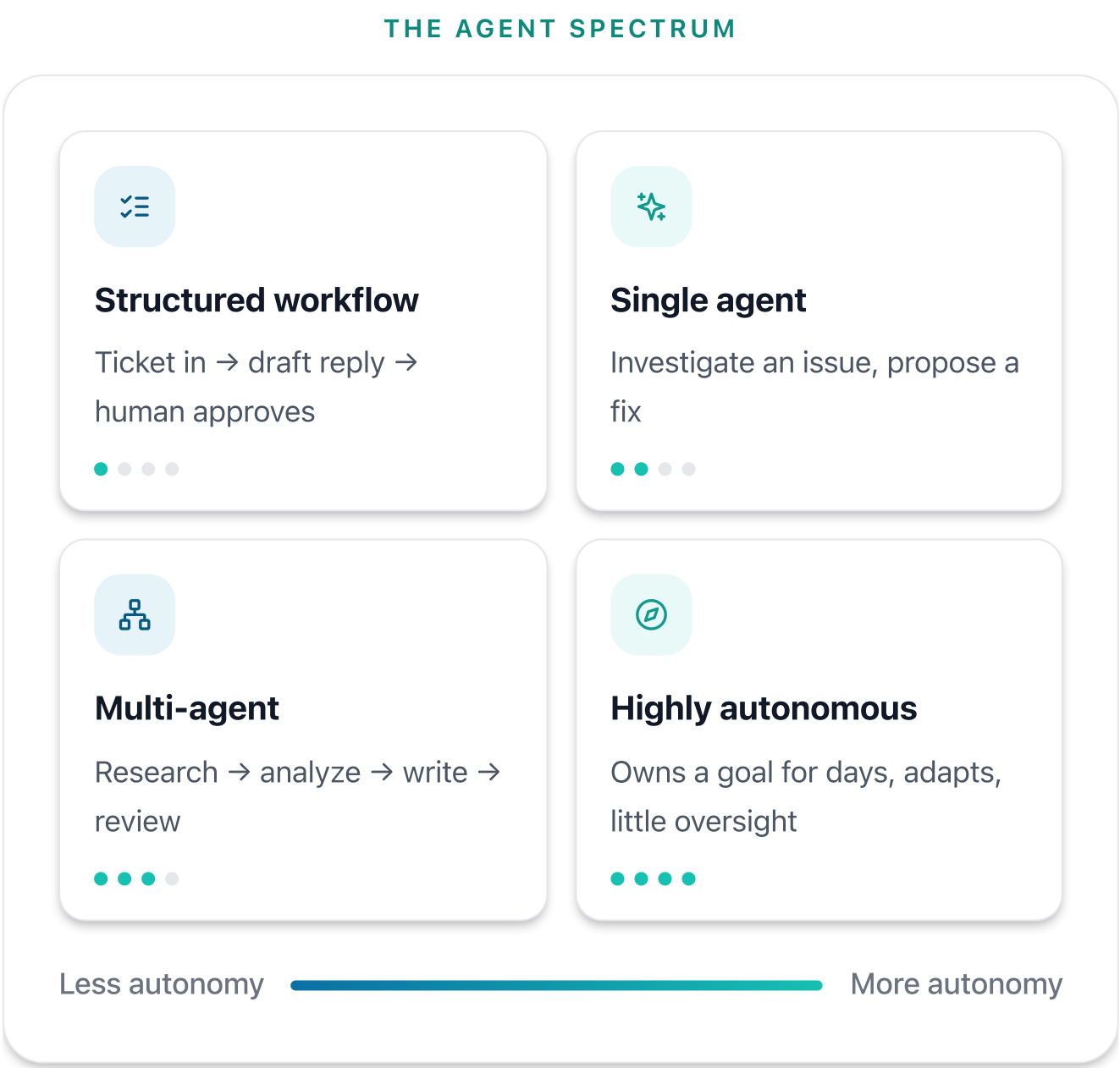
Structured AI workflows. The process is largely predetermined, with AI performing particular steps: new support ticket, classify, retrieve customer information, draft response, human approves.

Single agents. The agent receives a goal and has more discretion about how to accomplish it: "investigate this support issue and propose a resolution."

Multi-agent systems. A larger task is divided among specialized agents (research, analysis, writing, review), and one agent may coordinate the others.

More autonomous agents. At the ambitious end, agents operate for longer periods, handle broader goals, use many systems, adapt their plans, and need progressively less human intervention.

These architectures are not mutually exclusive. Companies combine them depending on the problem they are trying to solve.



What are “tools”?

Tools are simply capabilities the agent is allowed to use. An agent without tools can reason and generate information, but it has limited ability to interact with the outside world. Give it tools and the situation changes dramatically. A sales agent might have tools that let it:

- Search Salesforce
- Read meeting transcripts
- Search the web
- Query a product database
- Create an email draft

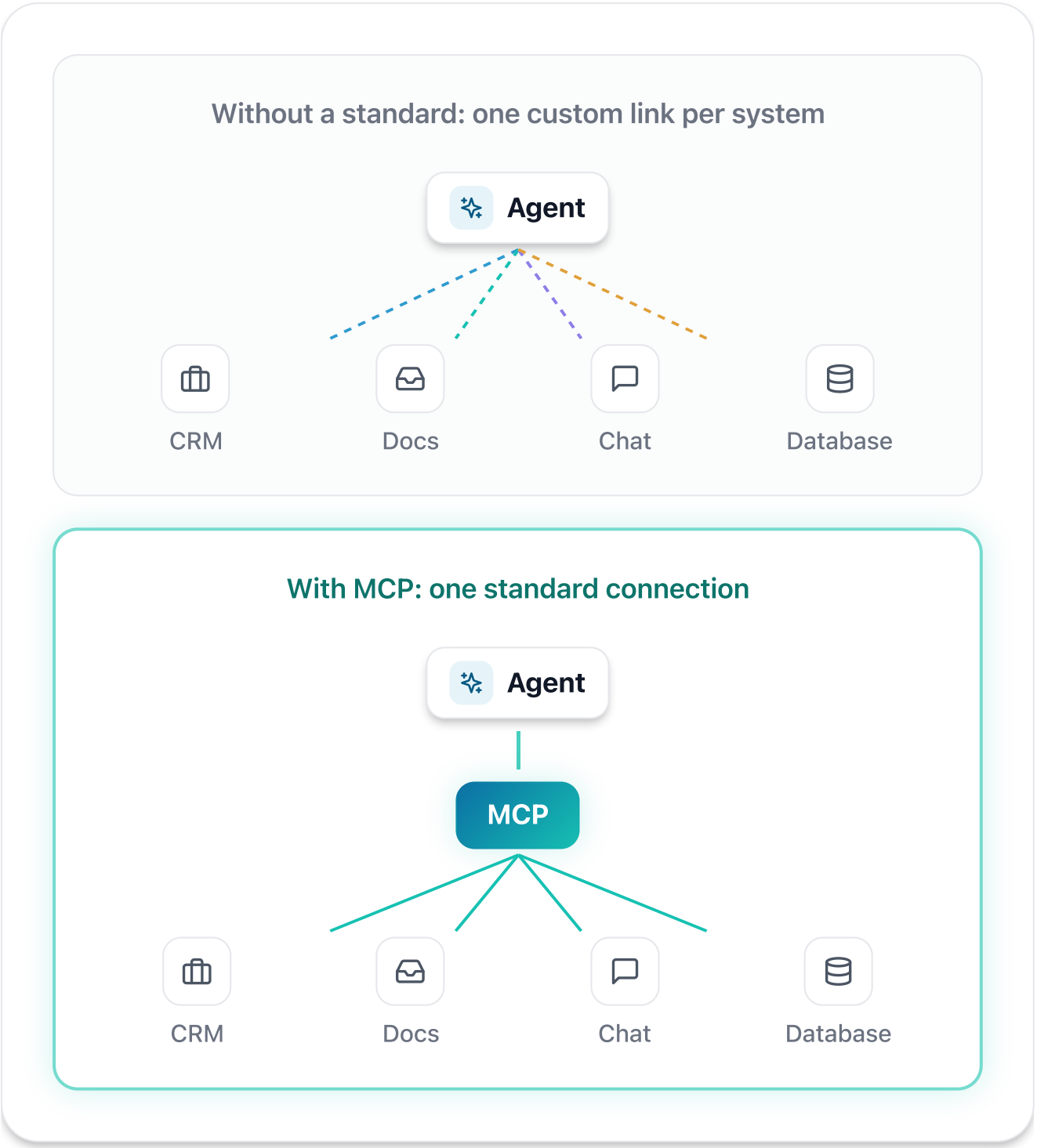
- Update a CRM record
- Send a Slack message

The AI model decides when and how to use those tools within whatever boundaries the developers have established. This is why connecting AI models to business systems is such an important part of building useful enterprise agents.

What is MCP?

You may increasingly hear MCP, or Model Context Protocol, mentioned in conversations about agents. The concept is simpler than the name makes it sound. MCP is an open standard designed to make it easier for AI applications to connect with external tools and information.

Without a common standard, developers may need to create custom connections between every agent and every system it needs. MCP provides a more standardized way for those systems to expose capabilities to AI applications. It doesn't create the agent and it isn't the AI model. It is one way of connecting agents to the things they need to access.



What can agents realistically do today?

Quite a lot, but the reality is generally more bounded than the "AI employee" language suggests. Agents are already useful for things like:

- Researching prospects and preparing outreach
- Handling portions of customer support

- Investigating software problems
- Writing and testing code
- Processing documents
- Gathering information across business systems
- Updating records and preparing reports
- Performing multi-step research

The important distinction is that an agent does not have to reproduce an entire human job to be useful. A salesperson performs dozens of activities requiring judgment, relationships, organizational knowledge and experience. An agent might automate one portion extremely well, or several agents and workflows might collectively automate a much larger portion. Both qualify as meaningful uses of agentic AI.

What an AI agent is not

A chatbot. Chat can be the interface, but agents don't require chat.

An AI model. The model is usually one component of the larger agent system.

A fully autonomous employee. Some agents have significant autonomy. Many intentionally don't.

One specific architecture. Agents can be built and orchestrated in many ways.

A system that knows everything. Agents work with the information and tools made available to them.

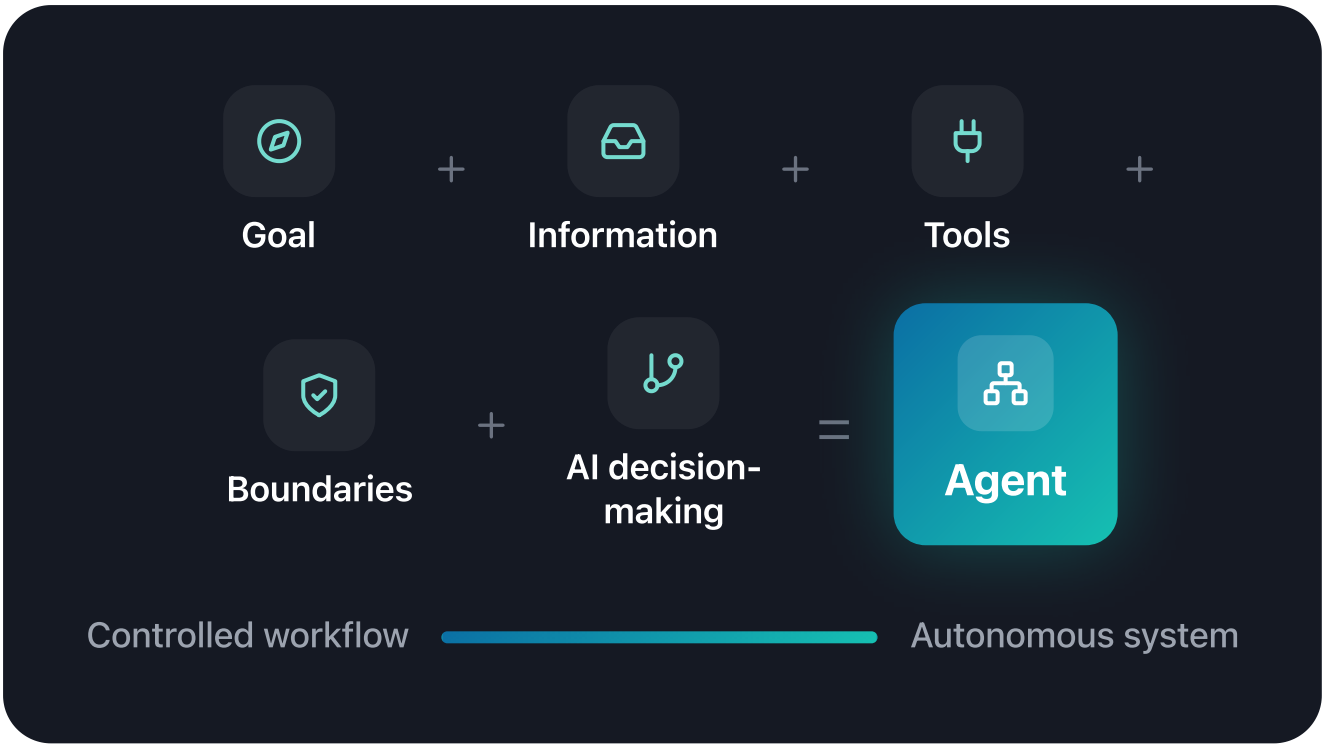
A replacement for traditional software. Most production agents combine AI reasoning with conventional software, APIs, databases, permissions and business logic.

The simplest way to think about it

Strip away the terminology and most agent architectures come back to a relatively understandable idea: give an AI a goal, give it information and tools, establish boundaries, and allow it some discretion over how to accomplish the goal.

How much discretion it receives is what creates the enormous range of systems now being called AI agents. At one end is a tightly controlled workflow where AI makes a few decisions. At the other is the emerging vision of autonomous systems pursuing complicated objectives over long periods with minimal human involvement. Most real-world implementations today live somewhere between.

ONE IDEA BEHIND EVERY AGENT



> "Agent" doesn't describe one specific piece of technology. It describes a way of building software in which AI increasingly participates in deciding what happens next.

ABOUT RITHMO

Rithmo is the fact-checker for AI agents. Before your agents act, it checks what your business has actually decided and gives them the current answer, so they never run on stale, conflicting, or incomplete information.

It reads across the meetings, messages, and systems a company already runs on, keeps a live record of what has actually been decided, and hands that answer to whatever needs it. It runs inside the customer environment, on-premise or in their own cloud, or in Rithmo Cloud.

See how it works hello@rithmo.ai