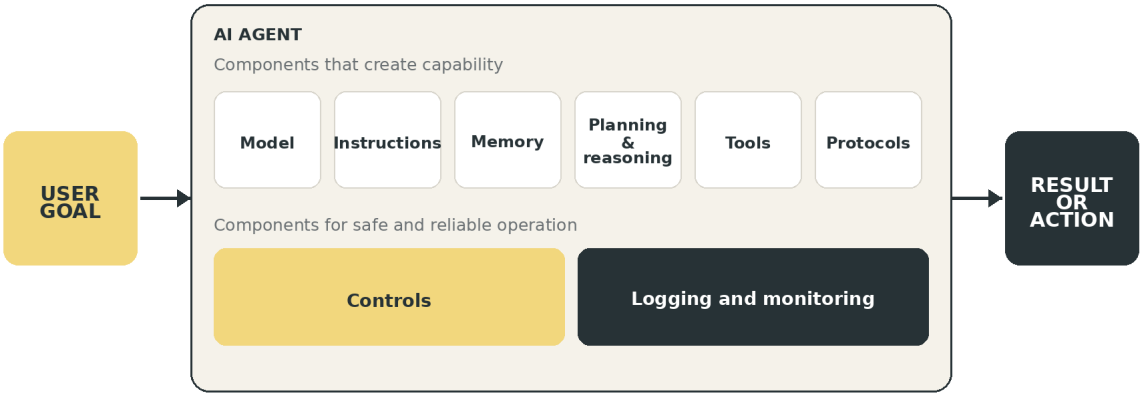
AI agent fundamentals

This guide explains what an AI agent is, the components it is built from, and the choices and configuration that shape what it can do.

There is no universal definition for "AI agent." For this guide, an AI agent is a software system that can plan, make decisions, and take actions over multiple steps to achieve a goal set by a user. Most enterprise agents combine some AI-driven decision-making with rules, permissions, and human involvement.

A simple AI agent can be understood through eight components: Six components create the agent's **working capability**; two supportive components keep its behavior **controlled and monitored**. We use these same component names throughout this guide.



Eight components of a simple AI agent

Component	What it means
Model	The AI model that interprets inputs and produces outputs in the form of responses or decisions.
Instructions	The directions given by a human that define the agent's role, task, and behavioral limits.
Memory	Information the agent can retain or retrieve from earlier interactions or stored information.
Planning and reasoning	How the agent works out the steps needed to complete the task.
Tools	Functions the agent can use to retrieve information or take action in another system.
Protocols	Standard ways the agent connects and communicates with tools or other agents. Frequently used protocols include MCP (Model Context Protocol) and API (Application Programming Interface).
Controls	Restrictions that limit what the agent may access or do, including access rules, guardrails, and necessary human approvals.
Logging and monitoring	Records of the agent's actions and interactions so activity can be reviewed and problems can be detected.

AI agent components in practice

Example: an internal Legal Q&A agent that answers routine questions from approved policies and routes anything outside scope to the Legal department.

Model	Understands the question (input) and drafts a response (output).
Instructions	Answer only defined policy topics, cite the approved source, and route uncertain or sensitive questions.
Memory	Keep the current conversation in context. Do not retain information beyond the approved retention setting.
Planning and reasoning	1. Identify the topic, 2. find the relevant source, 3. check whether the question is in scope, 4. answer or route to the Legal department.
Tools	Search the approved policy library and create a ticket for review by the Legal department when escalation is needed.
Protocols	Use the approved connection between the agent and the policy or ticketing system, such as an API or MCP connection.
Controls	Keep access read-only where possible. Limit tools and permissions. Require human approval for sensitive actions and high-risk decisions.
Logging and monitoring	Record the source used, tool calls, routing decision, and corrections so the workflow can be reviewed.

Two design choices determine the agent's reach

Action-space (authority or capabilities)
The range of actions the agent can take. This depends mainly on its tools and permissions. Read-only search is narrower than editing a record or sending an external message.

Autonomy (decision-making)
How much the agent can decide for itself while working toward the goal. This depends mainly on its instructions and the level of human involvement.

How to configure an agent

Ready to configure an agent? Ortaire can help you apply these choices to a defined workflow.

1. Define the job. Write the goal and the instructions the agent must follow.

2. Decide what it may remember. Set the memory needed for the task and no more.

3. Choose its tools and connections. Give it only the systems and functions needed for the job.

4. Set its action-space, autonomy, and controls. Decide what it may do on its own and where human approval is required.

5. Log, monitor, and test the workflow. Check the full sequence of steps before relying on it and again when the configuration changes.

Contact us: www.ortaire.com/contact or email hello@ortaire.com

Key takeaway: The model is only one component. An agent's practical capability comes from the full configuration, especially its instructions, tools, permissions, controls, and level of human involvement.