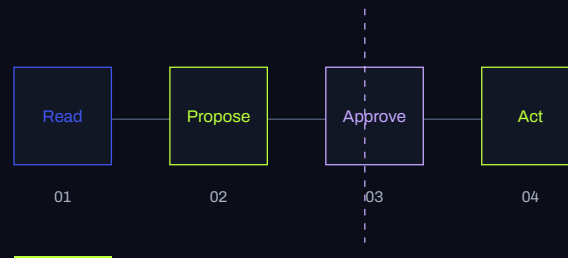


FIELD BRIEFING

Agents need boundaries

A tool call changes the stakes.



When does an assistant become a system that can act?

An assistant that proposes a reply and an agent that sends it have different responsibilities. Tool access changes the consequence of a generated answer. The important design question is not whether a system is called an agent, but what it can read, change, trigger and spend.

Explanatory schematics and worked examples are illustrative. No live model, observed client results or measured market performance are presented.

Read the mechanism.

A typical agent loop selects an action, calls a tool, receives a result and decides what to do next. Some workflows use fixed steps instead. More autonomy is not inherently better: if the route is predictable, a conventional workflow may be easier to inspect and recover.

Take an invented support workflow. The agent can read a ticket and an approved policy, prepare a response and propose store credit. Reading, drafting and issuing credit should not be one undifferentiated permission. An approval screen should show the actual recipient, amount, reason and applicable policy before the credit tool runs.

Put consequential checks at the action boundary. A general instruction to “be careful” is not a substitute for a tool that enforces allowed recipients or maximum amounts. Likewise, an output check after a tool has run cannot undo a message already sent. Approval and validation need to occur before the side effect they control.

Permissions should be narrow enough to describe. Restrict directories, data scopes, destinations and budgets where the environment supports it. Cap retries and make uncertainty visible. A failed action may need an idempotency key or an explicit status check before retrying; repeating a request can otherwise create a duplicate.

External material can contain instructions that conflict with the user's goal. Treat web pages, documents and tool responses as untrusted inputs unless their authority is established. Prompt injection is a system-design problem involving permissions, tool boundaries and information flow, not something a single warning sentence reliably solves.

Keep a useful audit trail: what was proposed, which inputs informed it, who approved it, what actually executed and whether recovery succeeded. Review this with synthetic cases before using real accounts. Neither a sandbox nor a guardrail provides a universal safety guarantee.

Technical references: OpenAI Agents SDK · Human-in-the-loop; OpenAI Agents SDK · Guardrails; OWASP · Excessive Agency (2026); Hugging Face · Secure code execution. Links and source scope appear in the final section.

Try the idea.

INVENTED TEACHING EXAMPLE

A proposal stops at the gate

Synthetic ticket EXAMPLE-004 produces a proposal for £20 store credit. Before the action, a person sees the ticket, amount, policy and reason. Without approval the action remains blocked. After approval the teaching example records a simulated action. No payment, refund or message takes place.

Go deeper.

Approval is before execution

An output guardrail after execution cannot prevent an already-completed side effect. Place required authorisation and parameter checks at the tool boundary. Framework defaults and timing must be checked in the version deployed.

A retry needs state

Before retrying an uncertain operation, determine whether it executed. Idempotency keys and transaction records can help prevent duplicates; their behaviour is specific to the downstream system.

A sandbox limits a surface

Isolation, resource caps and network limits reduce the capabilities available to generated code. They do not make all inputs safe or guarantee that allowed actions serve the user's intent.

Keep the evidence close.

Agent

A system that can select actions and use tools as part of a workflow.

Permission

A capability granted and enforced by the system, not just requested in a prompt.

Approval gate

A decision point before the consequential action it controls.

Idempotency

A design property that avoids duplicate effects when the same operation is retried.

Sources · accessed 4 October 2026

[OpenAI Agents SDK · Human-in-the-loop](#)

[OpenAI Agents SDK · Guardrails](#)

[OWASP · Excessive Agency \(2026\)](#)

[Hugging Face · Secure code execution](#)

Technical and security references support the mechanisms, not our fictional results. KAF's worksheets and examples are original editorial recommendations. No current provider pricing, comparative ranking or legal compliance is asserted. Verify the exact release, configuration and policy for your use.
