

Agentic AI: What CIOs Need to Know



A practical guide to delegating work without losing control

Bounded authority. Verified execution.
Practical decisions for Saudi enterprise technology leaders.

September 2026 | English edition

Start with the work, not the agent

The first CIO decision is whether a task needs adaptive execution at all.



Enterprise AI is moving beyond producing answers toward carrying out work. For a CIO, the consequential change is the connection between a model and the systems it can affect. A useful answer can be reviewed before anyone acts. An agent with operational permissions can change a record, trigger a process or create an obligation while the task is still running.

Anthropic distinguishes predefined workflows from agents that dynamically choose their process and tool use. That distinction is useful for investment decisions: flexibility should solve a real execution problem, because additional autonomy can also add latency, cost and failure paths. [1]

SOHOB POINT OF VIEW

Treat an agent as a bounded delegation of work. Define the task, permitted actions, evidence of completion and route back to a person before choosing the platform.

Choose the simplest execution pattern that fits

Task characteristics	Preferred starting point	Example
Fixed rules and predictable steps	Conventional automation	Validate an order against a known set of mandatory fields.
Judgment is needed within a stable process	A workflow with an AI step	Classify a service request, then route it through an existing approval process.
The next step depends on findings	A bounded agent	Investigate a delivery exception across approved order, inventory and carrier systems.

A promising first task has a recognizable finish, accessible evidence and a limited consequence when something goes wrong. An order investigation can produce a documented explanation without authorizing a refund or changing a delivery commitment. Separating investigation from execution allows the team to test useful capability before expanding permissions.

For Saudi enterprises, build the task sample around actual operating conditions: Arabic and English correspondence, local entity names, branch-level permissions and the applications staff really use. A fluent demonstration with clean sample data does not establish that the agent will handle conflicting records or incomplete requests.

The investment question is specific: which difficult step will adaptive execution improve, and how will the enterprise know? If the team cannot answer, simplify the proposed solution before funding a larger agent deployment.

Define the authority to act

Autonomy should be specified per action, not granted to the whole application.



An agent may be allowed to read a service record, prepare a proposed update and create a draft ticket while still being prohibited from changing access rights. These are different permissions within one task. Describing the entire application as either autonomous or human-supervised hides the decisions that matter.

SOHOB recommends an action contract for each deployment. It should name the resources in scope, the allowed operations, approval conditions, spending and runtime limits, and the events that force a handoff. The contract belongs in enforceable application controls as well as in instructions to the model.

Permission	Operating boundary	Evidence required
Read and investigate	Use only approved data sources under the requesting user's authorized scope.	Record sources and access decisions.
Prepare a change	Create a draft that cannot silently become a committed transaction.	Show the proposed change and affected records.
Execute a bounded action	Permit specified low-impact operations within explicit limits.	Record authorization, request and resulting state.
Escalate a consequential action	Require an authorized person to approve the exact action before execution.	Bind approval to the action, parameters and current context.

Make approval an informed decision

An approval screen should show what will change, for whom, why the change is proposed and whether it can be reversed. If the target record or material parameters change after approval, require a new decision. A generic "continue" button is weak evidence that the reviewer understood the transaction.

Assign the execution a traceable identity with narrow permissions and a clear relationship to the initiating user or service. Keep credentials outside model-visible content. A delegated task should never acquire broader privileges merely because it moved to another agent or tool.

DESIGN FOR UNCERTAINTY

When identity, authority or the target record is ambiguous, stop the action and return a structured handoff. Uncertainty is not permission to improvise.

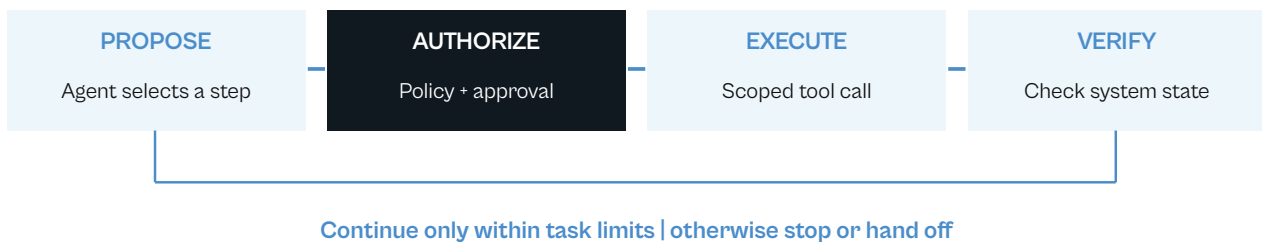
Review the action contract whenever tools, data access or business consequences change. A model upgrade does not automatically justify broader authority; the approved task boundary remains the governing constraint.

Build a controlled execution path

Put enforceable controls between the agent's proposal and the enterprise system.



The model should not be the final authority on whether its proposed action is permitted. SOHOB's recommended design separates task interpretation from authorization, execution and verification. This separation gives the CIO identifiable control points even when the sequence of investigative steps varies.



Supply only task-relevant context with identifiable sources. Before every tool call, the policy gateway checks identity, resource scope, action parameters and required approval. Tool adapters expose narrow operations with validated inputs. Verification checks the resulting business state and routes uncertain or partial completion to recovery.

Treat side effects as a first-class design problem

A timeout does not prove that an action failed. A ticket may have been created even if its response never reached the agent. For operations that change state, use a unique request identifier where supported, check the resulting record before retrying and prevent duplicate execution. Where reversal is possible, define a compensating action; where it is not, define containment and human recovery.

Keep task memory separate from the authoritative business record. Saved notes can help continuity, but they should not silently become approved policy or verified facts. Specify retention, permitted reuse and how corrections or stale information are handled.

Require a run record that links the initiating request, configuration version, tool calls, approvals and verified outcome. Retain useful operational evidence with appropriate access and retention controls; do not depend on access to hidden model reasoning.

ONE AGENT BEFORE MANY

Introduce multiple agents only when a demonstrated need outweighs the coordination cost. Each handoff needs an explicit scope, inherited permission limits and a clear owner for the final result.

Secure the whole action chain

Protect the path from external content to privileged operations.



Agent security requires attention to more than the text returned to a user. OWASP's Agentic AI guidance frames the problem through threats and mitigations across agentic systems. [2] For a CIO review, translate that perspective into concrete abuse cases against the proposed task and its connected tools.

The following scenarios are SOHOB review prompts, not a reproduction of an external threat taxonomy. Each asks whether an attacker-controlled input can influence an action the enterprise did not authorize.

Abuse scenario	Design response	Test before release
A retrieved document tells the agent to export internal records.	Treat retrieved content as untrusted data; independently restrict destinations and tool permissions.	Insert hostile instructions into a realistic document and verify that export is blocked.
An agent tries to access another department's case.	Enforce resource-level authorization at the tool boundary.	Request a valid-looking record outside the initiator's scope.
A stored note changes the interpretation of an approved policy.	Preserve source authority and constrain what memory can update.	Seed a conflicting note and verify that authoritative policy prevails.
A delegated agent receives wider access than its parent task.	Carry scope and approval constraints across delegation.	Attempt a prohibited operation through a second agent.
A repeated failure causes endless retries or duplicate changes.	Set budgets, stopping rules and safeguards against repeat execution.	Simulate timeouts after a committed transaction.

Give operations a usable emergency response

The team must be able to stop new actions, revoke tool access and identify in-flight transactions without relying on the agent to cooperate. Rehearse who can suspend a deployment, how affected users are informed and how the service returns to a safe state. A shutdown button is incomplete without a recovery procedure.

For deployments in Saudi Arabia, document where prompts, retrieved content, traces and backups travel and who can access them. Have the relevant privacy, security and compliance owners validate that complete data flow against applicable requirements. The location of the primary application alone is insufficient evidence about every connected service.

RELEASE QUESTION

Can the team demonstrate that a malicious input cannot bypass the task's permissions, and can it contain the impact when another control fails?

Evaluate completed work

A convincing explanation is not proof that the right action occurred.



Anthropic's evaluation guidance separates an agent's execution record from the final state of its environment. It also notes the need for repeated trials because results can vary. [3] CIOs should use that distinction to demand evidence about completed work, rather than accept a polished response as the sole measure of success.

For a service ticket agent, success might require the correct ticket, correct priority, correct owner and no unauthorized changes elsewhere. Scoring only the written summary would miss a ticket assigned to the wrong team or a duplicate created during a retry.

Evaluation lens	Evidence to collect	Failure example
Business outcome	Check the resulting record against the intended task.	The agent says a ticket exists, but no ticket was created.
Boundary compliance	Inspect attempted actions and authorization outcomes.	A forbidden lookup is attempted during an otherwise successful run.
Resilience	Exercise missing data, conflicts, timeouts and interrupted runs.	A resumed run commits the same change twice.
Human handoff	Check the completeness and timing of escalation.	A reviewer receives a request without the evidence needed to decide.
Operational performance	Measure end-to-end latency, cost and review effort.	The answer is correct but arrives too late to be useful.

Test the deployment configuration, not just the model

Test routine work, legitimate exceptions and misuse attempts in the languages and formats the service will encounter. Keep a held-out set for release decisions so tuning does not turn evaluation into a rehearsed demonstration.

Evaluate the model, instructions, tools, retrieval and permissions together; re-test when they change. Use deterministic checks for records and authorization, with calibrated human review for judgments that require it.

SET ACCEPTANCE CRITERIA BEFORE THE PILOT

Agree which failures are release blockers and which can be handled through escalation. A strong average task score must not conceal unauthorized actions or failure on a critical class of work.

Continue sampled review in production. Separate safe refusals, execution failures, independent completions and tasks rescued by a person. This reveals whether reliability is improving or work is shifting to reviewers.

Deploy for value, with an exit route

Buy evidence of useful execution, not the appearance of autonomy.



Agent economics should be measured over the complete task. Model usage is only one component; tool services, retries, observation, human review and error correction also consume resources. Compare the agent-enabled process with the current way of delivering the same accepted outcome.

A PRACTICAL COST MEASURE

Cost per accepted task = total operating cost for the measured period / tasks completed to the agreed acceptance standard. Include unsuccessful runs in the cost numerator; and report human-assisted completions separately.

Start with a service desk investigation

Consider an illustrative internal IT pilot that investigates recurring application incidents. Begin by allowing the agent to read approved tickets and knowledge articles, assemble evidence and draft a recommendation. Keep access changes, production configuration changes and external messages outside its initial authority.

Compare findings with historical cases in an isolated environment, then run alongside staff without executing changes. Move to a limited live group when the evidence supports it. Evaluate each added permission separately.

Before committing	Evidence to request
Choose a supplier or platform	Demonstrate permission enforcement, trace export, configuration versioning and revocation in your environment.
Expand the pilot	Show verified task results, reviewer workload, failure categories and total cost against the baseline.
Add operational permissions	Approve the exact new action and test its failure and recovery paths.
Plan an exit	Export task definitions and evidence, revoke credentials and demonstrate a workable manual fallback.

Predefine reasons to pause: unauthorized actions, untraceable failures, excessive review effort or an unacceptable cost per completed task. Expansion must follow operational evidence.

The CIO's decision

The objective is a useful service with explicit limits and provable outcomes. SOHOB can support the assessment of candidate tasks, the design of execution controls and the evaluation of a bounded pilot. Start with one task whose result can be checked and whose authority can be contained; expand only when the evidence warrants it.

Evidence & further reading

Primary sources supporting a practical CIO perspective.



This article presents original recommendations and illustrative enterprise scenarios developed for SOHOB. External publications support the concepts identified by numbered references. The examples are proposed designs, not claims of measured client results.

[1] Anthropic | Building effective agents

Published December 19, 2024. Supports the distinction between predefined workflows and dynamically directed agents, and the trade-off between complexity and task performance.

[Read the source](#)

[2] OWASP | Agentic AI - Threats and Mitigations

Published February 17, 2025. A threat-model-based reference for agentic security. The scenarios in this article are SOHOB's practical review examples, not the OWASP taxonomy.

[Read the source](#)

[3] Anthropic | Demystifying evals for AI agents

Supports evaluation of execution traces and final outcomes, repeated trials and evaluation of the deployed agent system.

[Read the source](#)

CONTINUE THE CONVERSATION

Which task could your enterprise safely delegate, and what evidence would justify that decision? Explore a bounded agentic AI pilot with SOHOB.

Sources checked September 2026. Technical designs require validation in the target environment. Photographs depict illustrative enterprise settings.