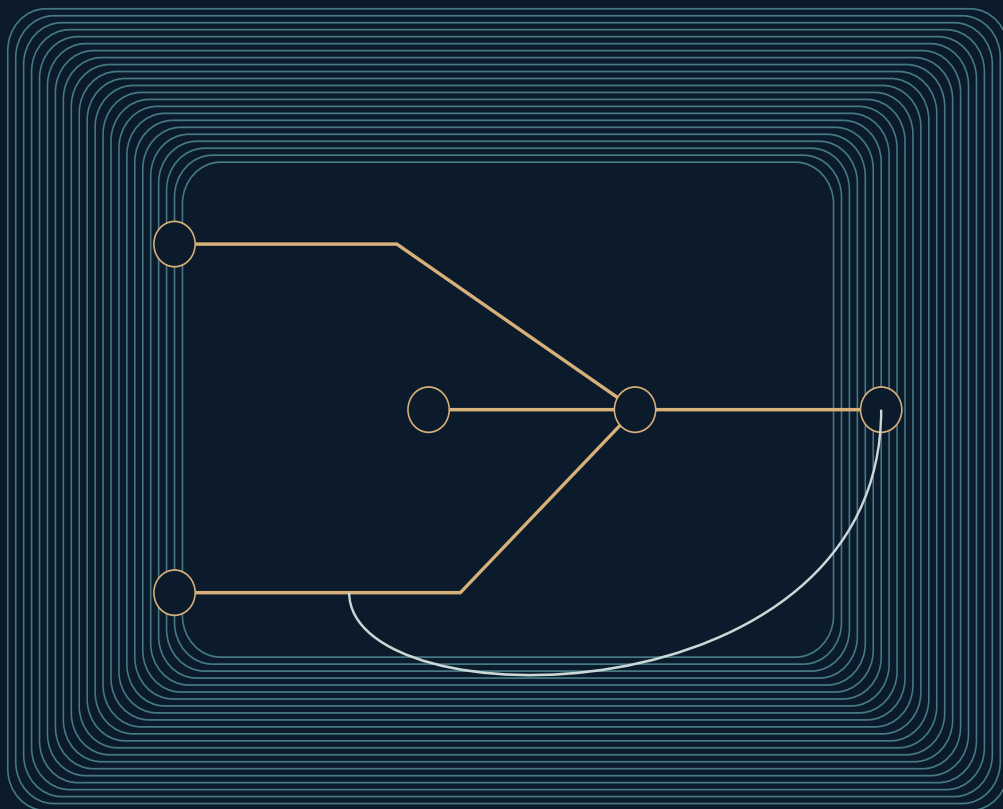




ENGINEERING / OCTOBER 2026

Agents need an operating contract

How to turn flexible reasoning into bounded, recoverable business execution



THE DECISION IN BRIEF

Agent autonomy should expand only when the business can verify completion, contain authority and recover uncertain actions at an acceptable cost.

- 01

Delegate an action class, not an open-ended ambition. Specify the permitted objects, transactions and limits before choosing how much freedom an agent needs.
- 02

Verify the business state after execution. A lost response can conceal a completed transaction; reconciliation must precede any retry that could repeat it.
- 03

Buy autonomy only where it improves accepted outcomes after supervision and recovery costs. Keep simpler workflows as the comparator and expand one permission at a time.

THE MAYDEN VIEW

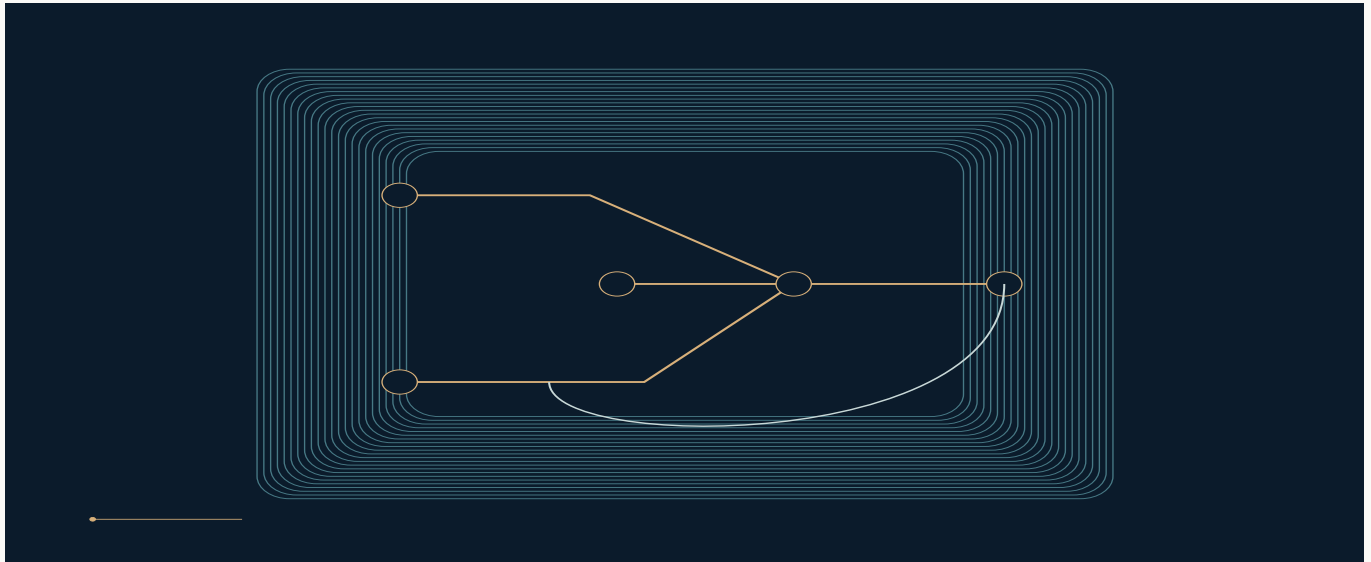
An agent has finished when the business state proves it - not when the agent says so.

INSIDE THIS PERSPECTIVE

The argument at a glance	3
01 / Delegate the action	4
02 / Verify the business state	5-6
Exhibit 01 / An uncertain order mutation must be reconciled before retry	7
03 / Contain authority and recovery costs	8
Exhibit 02 / Five conditions define the agent's execution envelope	9
Exhibit 03 / Historical agent studies distinguish task success from dependable execution	10
04 / Prove the next permission	11-12
The leadership agenda	13
Method and references	14

ONE DECISION, FOUR CONNECTED QUESTIONS

Autonomy needs an envelope



BOUNDED AGENT ROUTE / CONCEPTUAL ARTWORK

01 Delegate the action
Delegate an action, not an ambition

02 Verify the business state
Count the work after apparent success

03 Contain authority and recovery costs
Keep source content outside the authority boundary

04 Prove the next permission
The order changed; the confirmation did not

An agent has finished when the business state proves it - not when the agent says so.

CHAPTER 01

01

Delegate the action

Delegate an action, not an ambition

An agent changes a customer order. The transaction commits, but the response disappears. Retrying may duplicate the change; declaring failure may mislead the customer. This illustrative case captures the production challenge: flexible reasoning must operate inside a service that knows what happened and who may act next. The executive decision is how much authority to delegate, with what completion evidence and recovery capacity.

Anthropic's engineering guidance distinguishes predefined workflows from dynamically directed agents and describes the associated cost and latency trade-offs. This is vendor practice, not independent proof of business impact. ^[1] mayden's recommendation is to preserve deterministic steps wherever the business rule is stable and use model judgment where the path depends on information encountered during execution. An agent can interpret a request while an ordinary service enforces the allowed transaction.

A research assistant may read approved documents and recommend an action. A scheduling assistant may move a meeting within agreed limits. A procurement assistant may prepare an order while a budget owner approves it. These are illustrative designs, not observed mayden implementations. Choose the least discretion that can deliver the intended outcome; additional freedom carries a cost of evidence, supervision and recovery.

Verify the transaction behind the answer

A fluent message saying a task is complete is weak evidence of completion. Business execution should be verified against the relevant state: a booking exists with the required constraints, an approved record changed once, or a draft remains unsubmitted. The original tau-bench evaluates tool-and-user interactions against annotated end states and tests consistency over repeated trials. Its 2024 performance findings describe those models and simulations, not current production capability. ^[2] Its methodological contribution remains useful.

Define acceptable and unacceptable end states before building the agent. Include partially successful outcomes. If a customer asks to change an order and receive a confirmation, changing the order without sending the confirmation is a different failure from sending a confirmation without changing the order. Both are different from changing the wrong order. A single completion flag erases these distinctions and directs engineering effort toward an incomplete view of the service.

Operators need a compact receipt after each consequential action. It should identify the object changed, the actor, the authorization used and the observed result. Preserve uncertainty when the downstream system times out. The correct state may be outcome unknown, requiring reconciliation. A second blind attempt can turn a transient communication problem into a duplicate transaction. The service should be able to inspect whether the first action committed before choosing a recovery path.

CHAPTER 02

02

Verify the business state

Count the work after apparent success

Measure an agent across repeated runs, interruptions and recoveries. A high first-attempt success rate can coexist with unstable behavior on particular customers or long task paths. tau-bench's repeated-trial framing makes that distinction explicit. ^[2] The resulting evaluation should retain case-level variability. Some failures may be concentrated in a narrow, identifiable class that can be routed differently; other failures may be broadly unpredictable. These two patterns call for different operating decisions.

Long-horizon research finds that execution behavior can worsen as task paths lengthen, while more capable reasoning approaches can mitigate some of the observed effects. The

experiments also challenge a simplistic assumption that every step has the same independent chance of failure. ^[3] The enterprise should measure its own path-length and recovery distributions. A mathematical curve based on identical independent steps can illustrate sensitivity, but it should never be presented as an observed reliability forecast.

The hidden workload often sits after nominal completion: reconciling a state, correcting a document, contacting a customer or reconstructing an event. Include this work in cost per accepted outcome. Assign a recovery category and owner when the system fails. Track whether the same operator problem reappears after releases. Measure recurring intervention explicitly in the operations budget and test whether it falls as the service improves.

CHAPTER 02

02

Verify the business state

Enforce the execution envelope outside the model

mayden proposes an execution envelope with five elements: identity, action scope, state checks, resource limits and recovery. This is an original design framework derived from established security and reliability principles. Identity binds the run to the relevant user or service. Action scope constrains the objects and operations available. State checks verify preconditions and results. Resource limits bound time, spend and repeated actions. Recovery determines what the system does when confidence or information becomes insufficient.

OWASP identifies excessive functionality, permissions and autonomy as separate sources of damaging agent behavior. Its recommendations include enforcing authorization downstream of the model. ^[4] For management, this means a prompt telling an agent to respect policy should be accompanied by actual restrictions at the point of action. Removing a dangerous operation from the tool interface can be more dependable than relying

on the model to decline it under every possible input.

NIST's Zero Trust Architecture centers access decisions on resources and identities rather than assuming trust from location. ^[5] Applied here as an engineering principle, each tool action should be authorized in its own context. A research run's permission to inspect a supplier file does not imply permission to send that file externally. These controls constrain the consequences of a reasoning failure. They do not make the reasoning correct or remove the need to test the service.

NIST's September 2026 summary of public comments on agent identity and authorization highlights the need to preserve authority through delegation chains and the concern over making probabilistic models the sole authorization mechanism. ^[8] This is a record of stakeholder feedback, not a completed standard. It reinforces mayden's design position: the agent may propose an action, but an external control should decide whether that specific action is permitted for that principal, resource and context.

ILLUSTRATIVE DESIGN · NO MEASURED RESULTS

An uncertain order mutation must be reconciled before retry

Illustrative decision record

OBSERVED OUTCOMES / BRANCHES, NOT A RETRY SEQUENCE

Proposed / authorised Mutation path Validate scope, object and preconditions before submit Communication and evidence Retain approval and unique operation identifier	Submitted, receipt confirmed Mutation path Verify the authoritative order state Communication and evidence Confirm only the change evidenced by that state
Timeout / outcome unknown Mutation path Query authoritative state with the operation identifier Communication and evidence Say pending verification; timeout is not proof of failure	Terminal operation: no commit Mutation path Retry only after confirmed non-commitment, within authority and limits Communication and evidence An absent receipt or lookup miss does not establish this state
Partially complete Mutation path Reconcile each step; recover or escalate remaining work Communication and evidence Report completed and pending steps separately	Changed, notification failed Mutation path Do not repeat the order mutation Communication and evidence Retry permitted notification separately; preserve verified receipt
THE MANAGEMENT IMPLICATION A successful mutation and successful communication are separate completion conditions.	

Fictional order-change design; branches depend on transaction semantics. Never replay a mutation solely because its response or notification was lost. Reconciliation may require a staffed queue. Sources: [2], [3]

CHAPTER 03

Contain authority and recovery costs

03

Keep source content outside the authority boundary

Agents encounter documents, email, search results and tool responses that can contain hostile instructions. AgentDojo provides an extensible evaluation environment for testing attacks through untrusted tool data. Its task and attack collection demonstrates a way to examine the combined system; it does not estimate the prevalence of attacks in a particular company.

[6] The design implication is to separate what a source says from what the user has authorized.

A supplier document may legitimately contain a bank account or a request to change a delivery address. The agent must distinguish recording this information from executing the requested change. It should present a proposal with a source reference and obtain whatever independent verification the business process requires. A trusted internal storage location does not make every embedded instruction authoritative, particularly when documents originate outside the organization or are editable by a large group.

Test the complete route from intake to side effect. Include instructions embedded in retrieved content, conflicting tool outputs and apparent requests from other agents. Evaluate whether the final service preserves its authority boundary even when the model follows a misleading instruction. Record harmless refusals and blocked legitimate work as well as successful attacks. Controls that stop the service from accomplishing its ordinary purpose create pressure for bypasses, making operational usability part of the security design.

Buy discretion only where it earns its cost

A tightly constrained agent can become an expensive way to run a workflow. Every approval, check and handoff adds latency. For stable processes with clear rules, this objection may be decisive. AI Agents That Matter calls for joint attention to cost and accuracy and warns that benchmark practices can reward unnecessary complexity. [7] The company should require an agent to outperform simpler alternatives under the same definition of accepted work, with the chosen level of autonomy justified by that comparison.

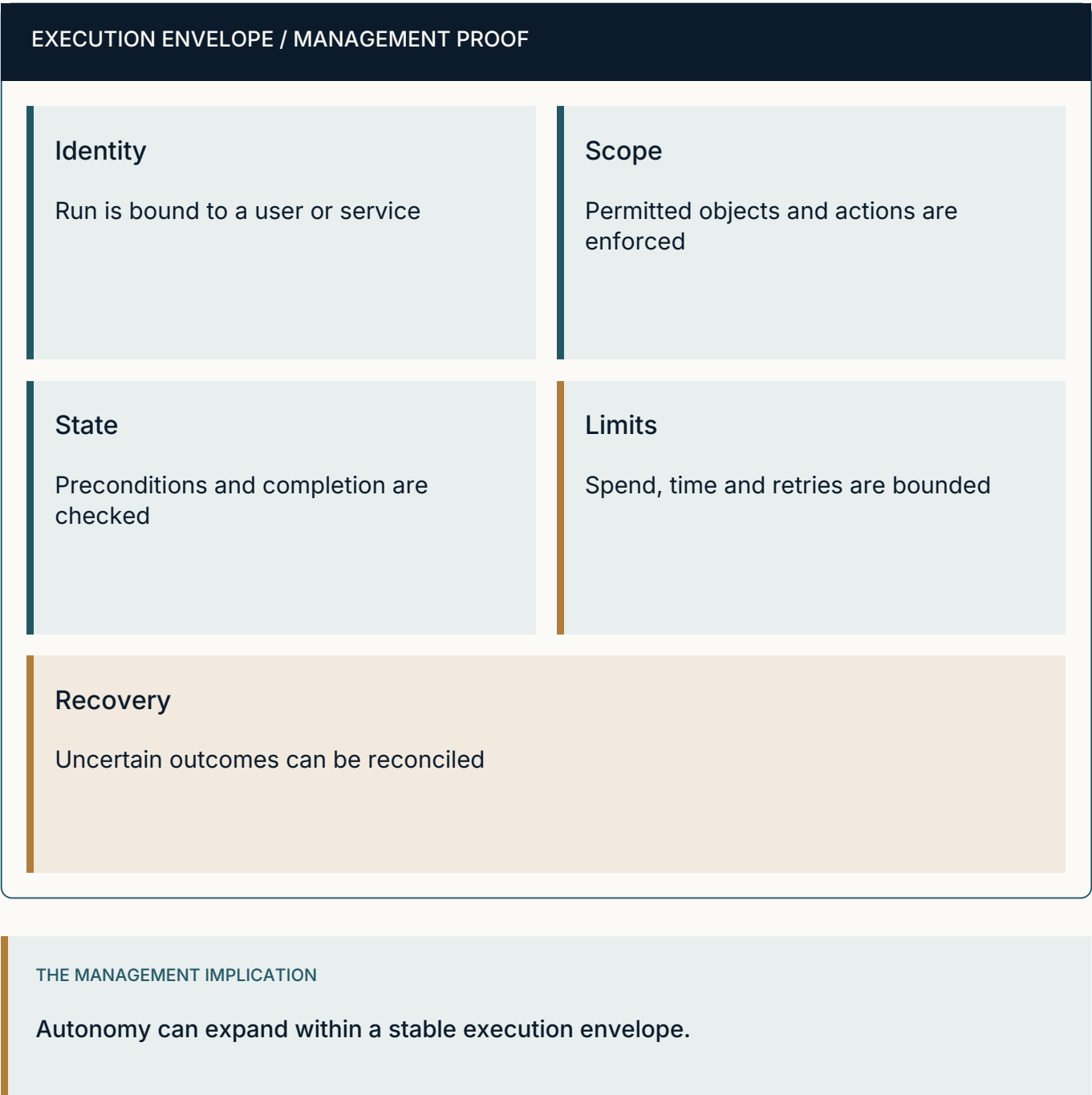
There are also situations in which restricting discretion prevents useful discovery. An analyst working with public information may benefit from exploring sources that were not anticipated. The correct response is to distinguish exploration from commitment. Permit broader inquiry inside a bounded environment, while retaining narrower permissions for changes to company systems and communication outside the organization. This separation can preserve creativity without transferring every exploratory judgment into an irreversible business action.

Review capacity belongs in the autonomy decision. Give a reviewer the source evidence, proposed change and consequences together. Measure waiting time and the frequency of changes or rejections. A queue in which reviewers cannot inspect consequential proposals does not provide the intended control. Reduce scope or redesign the handoff before adding traffic.

MAYDEN FRAMEWORK

Five conditions define the agent's execution envelope

Conceptual architecture



mayden original framework; controls constrain risk but do not guarantee correct outcomes. Sources: [2], [4], [5]

RESEARCH COMPARISON

Historical agent studies distinguish task success from dependable execution

Study-specific findings

tau-bench; June 2024; simulated retail/airline interactions	Published finding Tested agents such as GPT-4o solved fewer than 50% of tasks; retail pass^8 below 25%	What can transfer Repeated consistency needs its own measure; figures are historical bounds
Long-horizon execution; ICLR 2026; controlled execution tasks	Published finding Per-step behavior changes with sequence length; thinking mitigated observed self-conditioning	What can transfer Test long paths; do not assume independent equal-probability steps
AI Agents That Matter; 2024; benchmark-practice analysis	Published finding Accuracy-only comparisons can conceal complexity and cost	What can transfer Compare full configurations and accepted-outcome economics
THE MANAGEMENT IMPLICATION A single successful run and an economical, repeatable service are different claims.		

No current model ranking. pass^8 means consistency across eight trials, not success after eight retries. Conditions and tasks differ; no pooled estimate. Sources: [2], [3], [7]

CHAPTER 04

04

Prove the next permission

The order changed; the confirmation did not

Consider a fictional distributor whose customers ask to amend delivery dates and quantities. The proposed agent can interpret the request, inspect order status and prepare a change. The order service remains responsible for checking inventory, customer authority and whether dispatch has started. A validated change request receives a unique operation identifier. These are illustrative design choices; they are not a description of a tested mayden product or a claim of guaranteed reliability.

Exhibit 01 follows the difficult case: the order service commits a change but its response never reaches the agent. The result is unknown, not failed. Query authoritative order state using the unique operation identifier before retrying; confirmed non-commitment may permit a bounded retry, while a confirmed change must not be repeated. If only part of the request completed, reconcile each step and recover or escalate the remainder. A separate notification failure is a communication problem: retrying a permitted message must not replay the order mutation. Until verification succeeds, tell the customer that the change is pending verification rather than asserting success or failure. If the

service cannot distinguish a missing result from an operation still in flight, keep the state unknown and escalate. A lookup miss alone cannot authorise replay.

The design has costs. Unique-operation handling may require changes to a legacy system. A human queue needs staffing during peak periods. A narrow tool can prevent legitimate amendments that fall outside its schema. Management should compare these costs with the current service, including manual errors and delays. If the legacy system cannot support safe execution, the agent can still prepare a structured proposal for an operator. This narrower role may deliver useful value while the transaction layer is improved.

Expansion should follow evidence from the actual workflow. First test interpretation and proposal quality. Then test verified execution within a limited class of reversible amendments. Include dispatch races, concurrent staff edits, revoked customer authority and repeated delivery of the same request. Examine the full outcome and operator work. The successful experiment authorizes only the tested operating envelope; adding cancellation, refunds or a new warehouse changes the consequences and requires a separate release judgment.

CHAPTER 04

04

Prove the next permission

Expand authority one permission at a time

Select one action class whose result can be checked: preparing an order, amending a reversible field or sending an approved notification. Define the permitted objects, limits and end states. Then interrupt the workflow at each consequential transition. Operators must be able to distinguish completed, failed and unknown outcomes and exercise the appropriate recovery. An untested rollback diagram is not release evidence.

Expand users, tools and financial limits through separate decisions. Each changes a different exposure. Retain the policy, model, tool and evaluation versions behind the accepted

baseline, and compare the next configuration on repeated completion and intervention cost. If a new permission adds supervision faster than useful work, retain the narrower service.

The CEO should approve an operating contract for each agent: the work delegated, the actions permitted, the evidence of completion and the owner of uncertain cases. This makes autonomy an economic and operational choice. Scope can grow when the service earns it, and contract when the recovery burden outweighs the benefit.

FROM INSIGHT TO ACTION

Decisions to take forward

SERVICE OWNER

Delegate only a bounded action class

Evidence to require: Permitted actions and accepted end states

ENGINEERING OWNER

Separate mutation from notification retries

Evidence to require: Timeout, partial completion and reconciliation tests

OPERATING LEADER

Expand only when the exception path works

Evidence to require: Recovery capacity and repeated accepted outcomes

mayden connects senior judgement, research and engineering to turn strategic choices into working capabilities.

EVIDENCE YOU CAN INSPECT

Research notes

Method and boundaries

Narrative synthesis of eight research, standards and technical-practice sources accessed 10 October 2026; no pooled benchmark score.

Sources cover authority, end-state evaluation, repeated execution, long-horizon behavior, security and economics.

Historical benchmarks and vendor guidance do not predict current production performance. No independent enterprise field experiment was conducted. Proposed execution envelope needs local testing.

Primary sources and author research

[1] Building Effective AI Agents

Anthropic / 2024-12-19; living page accessed 2026-10-10. When and when not to use agents; workflows. Vendor engineering guidance originally published in 2024 and subsequently maintained; architectural practice, not causal proof that one pattern is universally superior.

[Read the source](#)

[2] tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains

Yao et al.; arXiv / 2024-06-17. Abstract. Historical benchmark with simulated users and specified models. pass^8 measures consistency across eight trials, not success after eight retries; results are not current model rankings.

[Read the source](#)

[3] The Illusion of Diminishing Returns: Measuring Long Horizon Execution in LLMs

Sinha et al.; ICLR 2026 / 2026-03-13. Abstract, version3. Controlled long-horizon experiments; effects depend on task, model and method and do not supply an enterprise reliability forecast.

[Read the source](#)

[4] LLM 06:2025 Excessive Agency

OWASP GenAI Security Project / 2025 edition; accessed 2026-10-10. Root causes; prevention and mitigation. Community security guidance on excessive functionality, permissions and autonomy; applying it does not guarantee prevention of every attack.

[Read the source](#)

[5] SP 800-207: Zero Trust Architecture

NIST / 2020-08. Abstract. General zero-trust architecture guidance from 2020; the application to AI tool authority in this report is maiden synthesis.

[Read the source](#)

[6] AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents

Debenedetti et al.; NeurIPS / 2024-11-24. Abstract. A research test environment with 97 tasks and 629 security cases; benchmark attack results do not estimate real-world incident prevalence.

[Read the source](#)

[7] AI Agents That Matter

Kapoor et al.; arXiv / 2024-07-01. Abstract. Research preprint on agent evaluation and reproducibility; benchmark economics do not establish the return from an individual deployment.

[Read the source](#)

[8] Summary of Comments on the Concept Paper: Software and Agentic AI Identity and Authorization

NIST NCCoE / Released September 2026; announcement 29 September 2026. Delegation and Accountability; Deterministic v. Probabilistic Authorization; Control Plane vs. Data Plane. Not final normative guidance, law, or empirical proof of mitigation effectiveness; unresolved standardization gaps remain.

[Read the source](#)