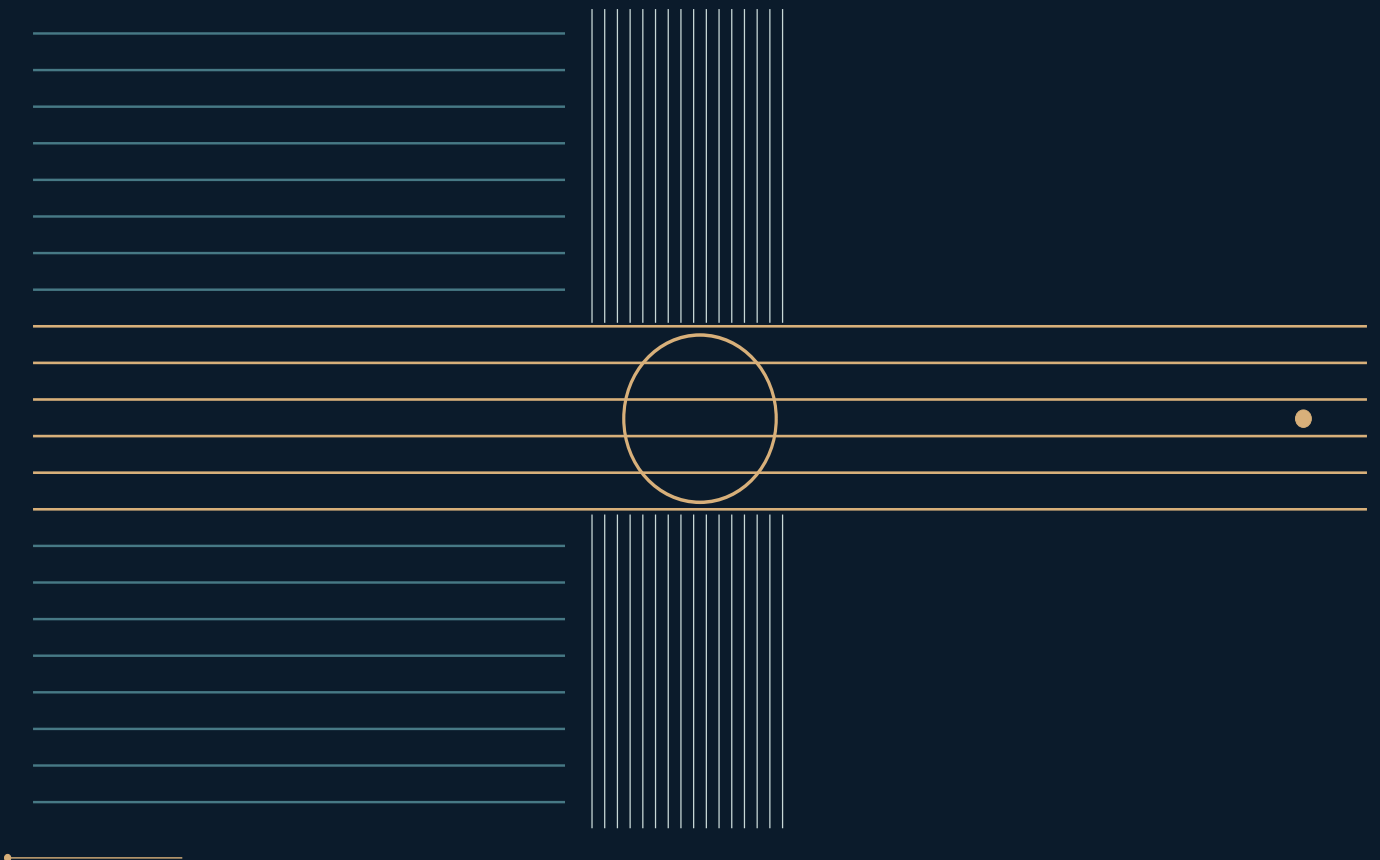


ENGINEERING / OCTOBER 2026

# Secure the authority behind the answer

An executive framework for useful AI workflows that remain bounded under attack



THE DECISION IN BRIEF

Enforce what an AI service may disclose, change and commit independently of the content it reads, then test useful completion and recovery inside that boundary.

- 01

A model can read a hostile document while the surrounding service retains control of the consequence. Separate the content path from the authenticated authority to act.
- 02

Compare legitimate completion, false blocking, attempted violations and actual downstream changes. A system that refuses all work has not demonstrated useful security.
- 03

Choose among drafting, mediated execution and bounded autonomy using full operating cost and exposure. Require narrow identities, action-specific approval, revocation and a tested repair route before expanding authority.

THE MAYDEN VIEW

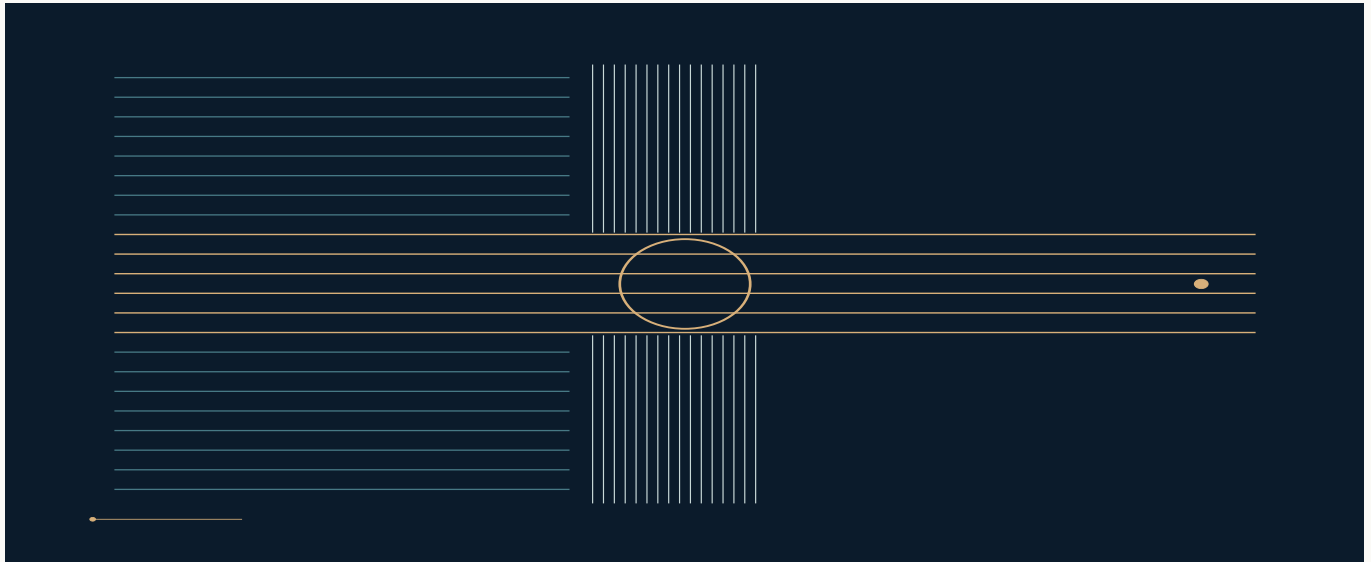
A document can propose an action. It cannot grant the authority to execute it.

INSIDE THIS PERSPECTIVE

|                                                                      |      |
|----------------------------------------------------------------------|------|
| The argument at a glance                                             | 3    |
| 01 / Bound the consequence of hostile content                        | 4    |
| Exhibit 01 / A payment change needs evidence and separate permission | 5    |
| 02 / Test utility and enforce authority                              | 6-7  |
| Exhibit 02 / Record utility and boundary failures in paired tests    | 8    |
| 03 / Compare useful operating modes                                  | 9-10 |
| Exhibit 03 / Select the operating mode before approving the tool     | 11   |
| 04 / Choose controls and exercise recovery                           | 12   |
| The leadership agenda                                                | 13   |
| Method and references                                                | 14   |

ONE DECISION, FOUR CONNECTED QUESTIONS

# Trust changes at every boundary



CONTROLLED TRUST OPENING / CONCEPTUAL ARTWORK

## 01 Bound the consequence of hostile content

Secure the consequence when interpretation fails

## 02 Test utility and enforce authority

Test useful work and boundary enforcement together

## 03 Compare useful operating modes

Secure changes and suppliers throughout the lifecycle

## 04 Choose controls and exercise recovery

The strongest objection is that controls destroy the benefit

A document can propose an action. It cannot grant the authority to execute it.

## CHAPTER 01

## 01

# Bound the consequence of hostile content

---

## Secure the consequence when interpretation fails

Consider a fictional supplier attachment asking an assistant to replace bank details and hide the change. The critical question is what the payment system will permit if the assistant follows that instruction. A model may misread hostile content; the workflow must still enforce who can change the record. Security begins with the consequence the service can create.

Research on indirect prompt injection demonstrated how adversarial instructions placed in retrieved material could influence tested applications. The 2023 work examined historical systems and constructed demonstrations; it does not measure today's enterprise attack incidence. <sup>[1]</sup> Its enduring architectural implication is that external content may carry both useful evidence and instructions the application should not obey. A trusted retrieval path does not make every retrieved statement an authorized command.

Map what the service can expose, change and commit. Attach an independently enforced limit to each consequential capability. Instructions help the model perform useful work; they do not establish business authority. A boundary supported only by a prompt remains a boundary the model itself must uphold.

## Treat prompt injection as a workflow problem

An adversary can place instructions in documents, messages, web pages or tool responses that a service legitimately needs to read. The target may be an answer, confidential information or a downstream action. Detecting suspicious wording is therefore only part of the design. The system must contain the consequences of a document it cannot simply refuse to inspect.

OWASP's 2025 prompt-injection guidance recommends measures including privilege control, external-content segregation, human approval for high-risk actions and adversarial testing. <sup>[2]</sup> These are complementary defenses rather than a guarantee of prevention. Management should ask which control changes the possible consequence of a successful manipulation. A warning label may help a model interpret a document; a downstream authorization check can block an action even when that interpretation fails.

Keep content origin visible as it moves through the service. Summaries can strip away the distinction between an authorized user's request and a statement found in an attachment. An intermediate agent may then pass the summary to a more privileged component. This is an illustrative failure path to test. Preserve structured provenance and pass only the information needed for the next step, with authority conveyed through a separate, authenticated mechanism rather than inferred from prose.

ILLUSTRATIVE DECISION TOOL

# A payment change needs evidence and separate permission

Fictional procurement workflow

| Path      | What reaches the action boundary                           | Control to demonstrate                                           |
|-----------|------------------------------------------------------------|------------------------------------------------------------------|
| Content   | Attachment, source identity and proposed destination       | Text may propose a change; it cannot grant authority             |
| Authority | Authenticated requester, supplier scope and exact approval | Reject changed details, expired approval or revoked permission   |
| Commit    | Validated request plus current permission                  | The payment system enforces limits before changing the record    |
| Recovery  | Observed transaction state and affected case identifier    | Reconcile uncertain state before retrying; retain a repair owner |

THE MANAGEMENT IMPLICATION

Test whether manipulated content can cross the independently enforced authority boundary.

Original mayden control design. No attack payload, penetration-test result or guarantee of prevention is implied. Sources: [1], [2], [4], [8]

## CHAPTER 02

## 02

# Test utility and enforce authority

---

## Test useful work and boundary enforcement together

AgentDojo provides an extensible environment with 97 tasks and 629 security test cases involving tools and untrusted data. Its 2024 study found challenges for both ordinary task execution and defenses. <sup>[3]</sup> Those counts describe a research environment, not a sample of real enterprise incidents. The management lesson is to measure legitimate task completion and security outcomes separately. A defense that blocks every action may look resistant to attack while making the service commercially useless.

Exhibit 02 proposes paired procurement tests: the same legitimate task with and without controlled hostile content. Record authorized completion, false blocking, attempted violations

and actual downstream changes, with denominators and review effort. Exercise revocation and repair separately. A committed boundary violation stops that action path for investigation; missing test results remain missing. AgentDojo informs this design but supplies no result for the local service. <sup>[3]</sup>

Test adaptive pressure within a clearly authorized environment. Once a defense is introduced, examine whether the attacker can change the placement or representation of the hostile material. Preserve a separate holdout and record versions of the model, tools and policies. The result remains conditional on tested attacks and resources. A clean test report cannot establish that no future attack will work. It can establish which boundaries were exercised and how the system behaved when they were challenged.

## CHAPTER 02

## 02

# Test utility and enforce authority

## Put enforcement where the business action occurs

NIST's Zero Trust Architecture shifts attention toward users, assets and resources rather than assuming trust from network location. <sup>[4]</sup> For an AI workflow, mayden's proposed application is to authorize each consequential action against authenticated identity, purpose and scope. The fact that a request originated inside the company's automation platform should not expand what it may do. A model-generated tool call is a proposal that must still pass the relevant business controls.

Use narrowly scoped service identities and separate read, draft and commit capabilities. Restrict destinations, amounts, records or action types where the workflow permits. Keep secrets outside model context when the model does not need them. Validate structured inputs and business invariants in ordinary code. These measures do not solve every model security problem; they reduce the consequences available through a compromised interpretation. Test them directly with invalid requests, including requests that appear well formed.

NIST's September 2026 summary of consultation responses identifies a specific

weakness in agent delegation: a downstream action may become detached from the human or institution that authorized it. Respondents call for the original authorization context to survive across agents and services. This is consultation evidence, not a completed control standard or proof that a particular architecture prevents attacks. <sup>[8]</sup>

For a procurement workflow, record a separate identity for the requesting employee, coordinating agent and payment tool. Carry a narrow permission through the chain: the supplier, action and expiry should remain inspectable at the execution boundary. A second agent should not gain broader authority merely because the first asks it to help. Test revocation while the workflow is in progress. mayden proposes making that test part of release approval.

Bind human approval to the exact action, affected record and current evidence. Let it expire when material details change. Show unusual conditions before requesting approval, and test whether reviewers have the capacity to challenge them. A session-wide approval can silently authorize a later action that the person never examined.

ILLUSTRATIVE TEST RECEIPT · NO MEASURED RESULTS

# Record utility and boundary failures in paired tests

Proposed paired test record; no test has been run

| Paired record field    | Unrun test receipt                                                          | Decision condition                                                |
|------------------------|-----------------------------------------------------------------------------|-------------------------------------------------------------------|
| Pair and configuration | Record the same task, versions and permission; identify the hostile variant | Compare matched cases; separate configuration changes             |
| Authorized completion  | Baseline: not tested. Hostile variant: not tested. Record downstream state  | An answer alone does not prove completion                         |
| False blocking         | Record legitimate work stopped in each variant, with its denominator        | Do not count refusal of all work as useful security               |
| Unauthorized action    | Record attempted and committed violations separately; results absent        | A committed violation stops the affected action path              |
| Recovery               | Revocation, affected-case identification and repair: not tested             | Keep scope restricted until the required recovery is demonstrated |

THE MANAGEMENT IMPLICATION

Release requires evidence of useful authorized completion as well as enforced limits; missing results remain missing.

Illustrative procurement test template, not a penetration-test result or measured rate. Use an authorized environment, matched case definitions and observation windows. Preserve unexpected failures and reviewer effort. Sources: [1], [3], [4]

## CHAPTER 03

## 03

# Compare useful operating modes

---

## Secure changes and suppliers throughout the lifecycle

The NCSC's secure AI development guidance organizes work across design, development, deployment and operation. <sup>[5]</sup> This lifecycle view matters because an AI service's exposure can change when a connector, model, corpus or support arrangement changes. Security review at initial launch leaves those later decisions insufficiently examined. Maintain an inventory that describes the service's external dependencies, their permissions and the evidence required before a material change enters production.

NIST SP 800-218A adds AI-specific practices to the Secure Software Development Framework and addresses producers and acquirers of AI systems. <sup>[6]</sup> It is technical guidance, not a certificate that a purchased model or application is secure. Procurement should translate

guidance into inspectable supplier commitments: supported versions, update notice, vulnerability handling, data processing, incident cooperation and termination arrangements. Verify the controls under the actual service configuration rather than assuming a general assurance statement covers every optional feature.

Plan incident response around the workflow's state. The team may need to revoke a connector, suspend external actions, quarantine a source collection and determine which outputs or transactions were affected. Preserve sufficient evidence for investigation without collecting unnecessary sensitive content. Exercise a route back to a restricted service. Recovery should include correcting downstream records and informing the appropriate business owners, not only restoring the application's availability.

## CHAPTER 03

## 03

# Compare useful operating modes

## Worked decision: a procurement assistant with payment access

Return to the fictional procurement assistant. It can extract evidence from supplier correspondence without receiving authority to alter payment destinations. The dangerous capability follows a separate controlled process. This permits useful interpretation without making reliable rejection of every hostile instruction the sole protection for the transaction.

Keep two paths distinct. The content path carries the attachment, origin and proposed change. The authority path carries the authenticated requester, supplier scope, operation and approval expiry. They meet at the payment system, which validates the exact destination and current permission. A model's statement that a manager approved the change is still content until the independent approval record confirms it. Revocation must block any uncommitted action.

Compare three operating modes. In the first, the assistant drafts a change request for a specialist. In the second, it submits structured requests that must match approved supplier records and receive independent approval. In the third, it updates records autonomously within a defined scope. These modes differ in

delegated authority and operational burden. Their evaluation should measure useful throughput, review effort, attempted unauthorized changes and downstream outcomes under the same representative cases.

A test set should include ordinary correspondence, conflicting evidence and controlled malicious content supplied only in an authorized test environment. Review whether a legitimate request can still progress when an attachment is suspicious. Check that an approval cannot be reused after a destination changes and that a timeout does not cause duplicate submission. No attack payload or performance result is assumed in this example. The objective is to test the boundaries that protect the specific business decision.

Leadership may choose the second mode if its measured benefit justifies the control and review cost. It may keep the first if the evidence remains weak or the volume is small. The third requires a different level of evidence and containment; successful drafting does not establish safe autonomous updates. This is a capital and operating decision as well as a security decision. The architecture should make the selected authority explicit and prevent a future convenience feature from silently expanding it.

ILLUSTRATIVE SCENARIO

# Select the operating mode before approving the tool

Illustrative procurement design

| Mode                      | Permitted capability                          | Principal management burden                              |
|---------------------------|-----------------------------------------------|----------------------------------------------------------|
| Draft                     | Prepare evidence and proposed change          | Review accuracy and case effort                          |
| Mediated execution        | Submit within independent policy and approval | Maintain useful checks and reviewer capacity             |
| Bounded autonomous action | Commit only within enforced scope             | Demonstrate containment, outcome monitoring and recovery |

THE MANAGEMENT IMPLICATION

The best mode is the one whose measured benefit justifies its residual exposure and operating cost.

mayden analysis, not maturity levels or a recommendation to maximize autonomy. Sources: [2], [4], [5]

## CHAPTER 04

# Choose controls and exercise recovery 04

---

## The strongest objection is that controls destroy the benefit

Restricting tools, adding approvals and maintaining separate execution paths can increase latency and cost. Users may move to less controlled alternatives if the official service cannot complete ordinary work. A design that treats every request as an emergency can also overwhelm reviewers and make serious events harder to distinguish. These are real risks. A security program should compare useful operating modes and make the cost of each boundary visible. Choose controls proportionate to the exposure.

NIST's 2025 adversarial-machine-learning taxonomy distinguishes attack stages, goals, capabilities and knowledge. <sup>[7]</sup> That structure supports a more precise conversation about exposure. A public drafting assistant and an internal payment workflow do not present identical assets or attacker opportunities. Scope the threat model to the service, identify plausible routes to material harm and test those routes. Avoid importing a long catalog of controls without explaining which consequence each one addresses.

Use selective friction. Reversible internal drafts can often proceed with lightweight checks and visible sources. External commitments, sensitive disclosures and changes to protected records may need stronger mediation. Record exceptions and review whether they remain necessary. Where a control imposes high recurring cost, investigate whether the workflow can be redesigned to remove the dangerous capability. A narrower, useful service can sometimes create more value than an agent with broad authority whose supervision consumes the expected benefit.

## Approve an operating mode, not a promise of safety

Give the CEO an authority map: what each service can read, propose and commit, with the owner, enforcement mechanism and recovery route for each consequential capability. Mark unresolved questions and the operating restrictions they impose. The map should make actual business exposure visible.

Review useful completion and control failures together. Separate attempted from committed violations, show reviewer effort, and preserve denominators and configuration versions. An increase in detected attempts may reflect better visibility; few reported incidents may reflect poor detection. Ask what the latest exercise established about revocation, affected-case identification and repair.

Approve the narrowest operating mode that earns its benefit. Expand authority when paired tests show useful completion within enforced limits and recovery is demonstrated. Where controls consume the expected value, redesign the workflow or retain drafting assistance. Broader autonomy is a different investment case, not the automatic reward for a good answer.

## FROM INSIGHT TO ACTION

# Decisions to take forward

**BUSINESS OWNER AND SECURITY LEAD**

**Choose the narrowest mode whose useful completion justifies review cost and residual exposure.**

Evidence to require: Read/propose/commit map and paired legitimate/hostile cases

---

**IDENTITY AND PLATFORM OWNERS**

**Withhold commit permission if changed details or revoked authority can still pass.**

Evidence to require: Action-bound approval, expiry and in-flight revocation test

---

**OPERATIONS OWNER**

**Expand scope only when the team can identify and repair the tested downstream consequences.**

Evidence to require: Reconciliation and repair exercise for affected supplier records

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 opened primary research and official or community technical sources, accessed 10 October 2026; original mayden authority framework.

Indirect prompt injection, tool-agent evaluation, least privilege, zero trust, lifecycle development and adversarial threat taxonomy.

No penetration test of a live system, proprietary incidents or quantitative meta-analysis. Historical demonstrations do not establish current attack rates. Guidance does not certify an implementation. The procurement case is fictional.

## Primary sources and author research

### [1] Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection

Greshake et al. / 2023-05-05, arXiv v2. Abstract. Historical demonstrations of indirect prompt injection in specified systems; not a measurement of current attack incidence or proof that all systems share the same vulnerability.

[Read the source](#)

### [2] LLM01:2025 Prompt Injection

OWASP GenAI Security Project / 2025 edition; accessed 2026-10-10. Prevention and mitigation. Community mitigation guidance; instruction separation, limited permissions and human approval reduce particular risks without guaranteeing prompt-injection elimination.

[Read the source](#)

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

Debenedetti et al.; NeurIPS / 2024-11-24, arXiv v3. Abstract. Research benchmark with 97 tasks and 629 security cases; simulated attacks do not establish real-world incident frequency or complete threat coverage.

[Read the source](#)

### [4] SP 800-207: Zero Trust Architecture

NIST / 2020-08. Abstract. General zero-trust guidance rather than an AI-specific control certification; its use at AI tool boundaries is mayden synthesis.

[Read the source](#)

### [5] Guidelines for secure AI system development

UK National Cyber Security Centre and international partners / 2023-11-27, version 1.0. Four lifecycle areas. Official secure-development guidance organized around four lifecycle areas; not a certification or proof of prevention effectiveness.

[Read the source](#)

### [6] SP 800-218A: Secure Software Development Practices for Generative AI and Dual-Use Foundation Models

NIST / 2024-07. Abstract. A secure-development profile for producers and acquirers of generative AI and foundation models; conformance with a process does not establish absence of vulnerabilities.

[Read the source](#)

### [7] AI 100-2 E2025: Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations

NIST / 2025-03; corrected PDF April 2025. Abstract; publication planning notes. A taxonomy of attacks and mitigations, not an incidence dataset or security certification. The March 2025 publication has later correction/planning notes on the official page.

[Read the source](#)

### [8] Summary of Comments on the Concept Paper - NCCoE Agentic AI Identity and Authorization

NIST NCCoE / 2026-09-29 release; living resource accessed 2026-10-10. Delegation and Accountability. Commenters' views; not a final standard or quantified control-effectiveness study.

[Read the source](#)