

Delegating Authority to AI

A practical guide to deploying agents with accountable control

Don Siler | Bridge Light | October 12 2026

Introduction

A customer-service agent that drafts a response and one that issues a refund may use the same model, but they carry different business responsibilities; the first helps an employee decide what to do while the second exercises authority on the company's behalf. Once AI can act, leaders must decide which actions to delegate, what limits to impose, and what evidence would justify giving it more responsibility.

This guide is intended to help business leaders decide what to delegate to AI, within what limits, and when to expand its authority. It is written for executives accountable for business outcomes and for the delivery, operations, security, and governance teams responsible for implementation. An AI agent here means a system that pursues a task by selecting and using tools; the relevant risk depends on what those tools allow it to do.

The most useful starting point is a workflow whose results the business can observe and whose consequences it can manage. Give the agent enough authority to perform useful work, enforce its limits where actions occur, and assess the whole operating result. Expansion should follow evidence of both value and effective control. Some workflows will benefit from greater autonomy while others will perform best with continuing human approval.

How to use the guide

The first five sections take a deployment from workflow selection through continuing operation. Each ends with a decision gate. The evidence sections explain what the underlying research establishes and where uncertainty remains. Two working templates capture authorization and release decisions. Use them within existing delivery and risk processes, with requirements scaled to the consequences of the work.

For an initial deployment, use the full sequence. For an existing agent, begin by comparing its actual access and actions with its approved mandate, then review the evidence and operating measures. Systems affecting rights, essential services, sensitive information, or material financial commitments require stronger domain review and recovery arrangements.

Basis and scope

This guide draws on published research and guidance from the World Economic Forum and Capgemini, McKinsey, the Urban Institute, and NIST, alongside insights from the research team and interviews with industry practitioners. [1–4, 10, 24] Its checklists, decision gates, and templates are adaptations for business use. The approach provides a disciplined basis for deployment decisions, with the caveat that it cannot certify that a deployment is “safe.”

Five pitfalls to check before deployment

Use these questions to challenge an initial deployment plan. They identify weaknesses to check for, rather than a measured ranking of the most common deployment failures.

Confusing capability with permission

A tool may let an agent change an account even when the business intended it only to recommend a response. Have you defined the actions it may take and enforced those limits independently of its reasoning? See Sections 1 and 2.

Treating a demonstration as deployment evidence

A successful demonstration may not show how the system handles incomplete information, failed tools, repeat requests, or misleading instructions. Have you evaluated the complete workflow under the conditions it will encounter? See Section 3.

Assuming human approval provides effective oversight

An approval adds little protection if the reviewer cannot recognize an error or stop the action. Do reviewers have the evidence, expertise, time, and authority to intervene - and can they manage the expected workload? See Sections 3 and 4.

Measuring task speed instead of business value

Fast output can create slow review and costly correction. Are you measuring accepted business outcomes and total operating cost, including supervision, downstream rework, and recovery? See Sections 1 and 5.

Treating authorization as a one time decision

A new tool, changed data source, model update, or cooperating agent can alter the consequences of an approved deployment. Do you identify those changes and reassess permissions before exposure expands? See Section 5.

1 Choose the scope of delegation

The same customer-service workflow can support several forms of delegation. An agent might summarize a complaint, recommend a credit, or issue the credit itself; each step changes the consequences of error and the evidence needed to justify it. Choose the scope by considering ambiguity, potential harm, and whether mistakes can be detected and reversed. Where predictable rules already solve the problem, conventional automation may be sufficient.

Match authority to consequences

Begin with the action the business is prepared to authorize. Reading an account and recommending a response creates different exposure from changing an entitlement, committing funds, or sending information outside the company. Decide which cases may proceed within limits, which need approval, and which remain outside the mandate - a broad goal such as resolving a customer issue should not quietly authorize every method the agent can discover.

A narrow pilot can test work worth doing. Low-value credits, for example, may offer a manageable starting point, but reversibility alone is insufficient: repeated credits can accumulate into a material loss. Assess the combined exposure of actions as well as the consequence of any single transaction.

Make authorization and intervention explicit

Technical access can become delegated authority by default. For example, if a tool allows an agent to change an account, the agent may act to do so even when the business intended it to only provide recommendations. Make the approval of authority explicit, identify who may change it, and establish who can stop execution. Exceptions require an authorized decision maker and a defined response when that person is unavailable.

This becomes harder when agents cooperate. One may gather information, another recommend an action, and a third execute it. Their combined behavior must remain within the approved workflow; individually reasonable permissions should not create authority that no one has reviewed.

Evaluate the economics of delegation

The economics of delegation depend on what happens after the agent finishes. A response generated in seconds offers little saving if an employee spends minutes verifying it or another team must correct the result. Compare accepted outcomes and total operating cost, including supervision, rework, exceptions, and recovery. That is the basis for deciding whether more authority would actually improve performance.

Maintain visibility across deployments

Maintain a record of each deployment's mandate, actual permissions, tools, data sources, provider, and dependencies. Include purchased services and pilots. Shared models, credentials, or tools can transmit failures across workflows; a deployment-level review alone may miss that exposure. McKinsey's framework emphasizes this enterprise portfolio view. [3]

The decision to proceed

Proceed when the proposed authority is proportionate to the consequences, approval rights are explicit, and the value assessment includes the cost of supervision. Confirm that the initial scope can be evaluated before granting access.

2 Define authority and enforce it

A capable agent can discover ways to complete a task that its designers did not anticipate. That flexibility is useful, but it makes a business mandate essential. Capability describes what the system can do; authority defines what it may do in a particular context. WEF's deployment-level authorization approach brings permissions, conditions, and oversight into one record. [1, 2] Appendix A adapts that idea into a working form.

Specify the mandate

Write down the task, the acceptable result, and the actions the agent may take. Identify where it must seek approval, where it may send information, and what it must never do. Set limits on individual transactions and their combined effect, with conditions for stopping. Associate the approval with a specific system version and review date. When agents cooperate, review the combined workflow while keeping their permissions distinct.

The service agent may need considerable freedom to interpret a complaint while having little freedom to alter an account - preserve that distinction! An explanation that a larger credit would resolve the issue is information for an approver; it should not be permission for the agent to raise its own limit.

Put controls at the point of action

- Give the agent access only to the accounts, tools, data, and destinations its task requires. Credentials are the keys or accounts that provide that access; the agent must not be able to change its own permissions.
- Check business rules in the service that performs the action, independently of the agent's reasoning. For credits, check eligibility, amount, duplicates, and total exposure across transactions. Confirm that another tool or sequence of actions cannot bypass those checks.
- Treat external messages, retrieved documents, and tool results as untrusted inputs. Test whether they can redirect execution or cause disclosure.
- Limit the systems and information the agent can reach, using a restricted execution environment where appropriate. Define whether work stops, queues for review, or continues under narrower permissions if an approval or policy-checking service fails.
- Record which tools were used, what approvals were granted, and what actually changed. Keep those records beyond the agent's ability to alter them, and limit the sensitive information collected and retained.

Example: Customer Service Credits

In the service-credit example, the first release might classify tickets and draft responses. Allowing the agent to issue credits adds a different requirement: the execution service must check eligibility, amount, aggregate exposure, and duplicates, while routing exceptions for review. Set the limits according to potential loss and recovery capacity - the appropriate amount is a business decision.

The decision to proceed

Proceed to controlled testing when implemented access matches the approved mandate, consequential execution routes have been reviewed, and pause and revocation mechanisms work. Review the permissions themselves, not just whether they are enforced. An agent can stay within its authorized limits and still cause harm if those limits are poorly defined.

3 Require evidence before release

A convincing demonstration shows that a system can complete a task. A release decision asks whether it can do so reliably under the conditions the business will encounter. Evaluate the complete workflow: tools, data, permissions, human intervention, and downstream effects. Model scores and guardrail accuracy contribute to that judgment but cannot replace it. The US National Institute of Standards and Technology's AI Risk Management Framework provides the general risk management context; the Urban Institute emphasizes evidence and approval before broader deployment. [4, 10]

Define acceptance criteria before testing

For the credit workflow, ordinary eligible cases are only the beginning. Include disputed eligibility, incomplete records, repeat requests, and cases the agent should refer to a person. Establish acceptance criteria before testing, with qualified human assessment where needed. Record versions, conditions, and coverage so a successful result can be traced to the system that produced it.

- Test incomplete or conflicting information, unavailable tools, and retries after an uncertain result. Check whether two individually permitted actions can combine to produce an outcome the business did not authorize.
- Test whether misleading instructions in customer messages, documents, or tool results can cause the agent to act outside its assigned task. Check whether it can send information to an unapproved destination, if it can use access it should not have, bypass an approval, or repeat a payment. Confirm what happens when a required control or human reviewer is unavailable.
- Measure how often work meets the business standard, what harm occurs, and how much review and correction are needed. Count useful actions blocked by controls as well as errors that get through. State the number of cases or actions behind each rate and the severity of harmful outcomes.

Test oversight and recovery

Human approval is useful only if the reviewer can recognize a problem and intervene. Test whether staff understand the evidence, can respond within the available time, and can reject the recommendation. Then rehearse an actual failure: stop execution, revoke access, reconcile partially completed actions, restore service, and correct customer impacts. Essential work needs a continuity route that people have practiced.

Scale the evidence to the consequences

For decisions affecting employment, lending, health, benefits, or comparable interests, involve the relevant specialists. Check performance across affected groups, explain decisions where needed, and provide a correction or appeal route. Confirm applicable requirements. Before a customer-facing release, also test how actual users interact with the system.

Safety measures and business performance must be read together. A system that refuses every credit may record no harmful payments while resolving no customer issues. Conversely, a high completion rate may conceal costly mistakes. Report useful completion alongside harm, exposure, and review burden. Zero observed harm in a limited test is a result to interpret, not proof of zero risk.

The decision to proceed

Record results, failures, mitigations, unresolved gaps, and residual exposure. Proceed only when required controls function, qualified review capacity is sufficient, and the accountable owner accepts the bounded risk. Narrow the scope or continue testing when those conditions are unmet.

4 Release in stages

Release decisions become more defensible when each stage answers a specific question. Does the agent understand the work? Do its proposals remain useful on real cases? Can staff supervise it at the required volume? Can enforced limits contain the consequences of an error? The sequence below uses staged evaluation and adjustable authorization to answer those questions before exposure expands. This is a proposed operating method, not a validated maturity model. [1, 2, 4]

Stage	Operating scope	Evidence before expansion
Sandbox	Controlled data and simulated effects.	Task outcomes, boundary tests, and recovery checks.
Shadow	Propose actions on real work without executing them.	Representative quality and realistic exception volume.
Supervised pilot	Execute designated actions after qualified approval.	Reliable execution and sustainable review workload.
Bounded operation	Execute specified actions within enforced limits.	Measured value, acceptable exposure, and effective monitoring.
Expanded operation	Broaden approved scope or permissions.	Fresh evaluation of the changed consequences.

Approve a defined release

Approval should identify precisely what is being released. Bring together the mandate, configuration and dependency versions, evaluation results, monitoring plan, unresolved risks, recovery procedures, and reviewer coverage. This makes the decision reviewable and prevents approval of a successful prototype from becoming permission for a materially different deployment. Appendix B records the scope, conditions, pause authority, and next review date.

Prepare the people affected

Explain the system's role and limits where users need that information to make decisions. Provide a route to a person for consequential or unresolved cases and a means to correct mistakes. Test these routes under expected demand.

The decision to proceed

Release when the business owner and required reviewers approve the package and operating arrangements are ready. Set review frequency according to transaction volume, consequences, and observed stability. A missed deadline or expiring budget does not justify greater authority; material failures require immediate action.

5 Operate and reassess authority

A deployment can change without a formal decision to expand it. A new tool, altered data source, or provider update may change what an agent can do and how it behaves. The operating process must detect those changes and revisit the authority granted. Review outcomes and dependencies throughout the lifecycle. [3] NIST identifies practical barriers to monitoring deployed AI; the scorecard below requires locally defined measures and reliable collection. [23]

Measure	Operating question
Accepted completion	Does the work meet the business standard without hidden downstream rework?
Harm and exposure	What occurred, with what severity, across how many actions or cases?
Review burden	How often are actions blocked or escalated, and how much qualified review is needed?
Total value	Do benefits exceed model, integration, monitoring, review, correction, and recovery costs?
Human capability	Can staff detect errors, handle exceptions, and perform essential fallback work?

Control changes

For the service agent, a new entitlement tool could introduce consequences that the original credit tests never covered. Apply the same scrutiny to changes in models, instructions, data sources, memory, access, policies, providers, and agent dependencies. Retest affected functions and renew approval when scope or consequences increase. Preserve enough configuration history to investigate failures and support recovery.

Respond to failures

If something goes wrong, stop or restrict the affected work, withdraw relevant access, and preserve the records. Determine what the agent actually completed before correcting downstream effects and restoring service. Involve the designated business, security, and communications owners. Fix the cause and repeat relevant tests before restarting.

Preserve the capacity to supervise

Delegation can reduce the routine practice through which employees learn to recognize unusual cases. Check whether staff still detect consequential errors, resolve exceptions, and perform essential fallback work. The quality of review matters more than the number of approvals recorded. If supervisory capability weakens, revisit training, workload, and the work being delegated.

The decision to proceed

Give the agent broader authority only when the additional work has a credible value case and sufficient supporting evidence. Keep or reduce its permissions when controls fail, results remain unexplained, or qualified review is unavailable. When retiring a deployment, remove access, disable scheduled work and tool connections, resolve unfinished tasks, and handle records under the applicable retention rules.

What the evidence establishes

The preceding sections explain how to define an agent's authority, control its actions, and decide whether it is ready for deployment. What supports those recommendations?

The following sections examine selected studies, reports from deployed systems, and documented incidents. They address practical questions: Can permissions prevent harmful actions? Can safeguards protect the business without blocking useful work? Can people effectively supervise the system?

Each source provides evidence under particular conditions. None establishes that the complete approach in this guide guarantees safety or business value. The discussion explains what was tested, how the result informs a deployment decision, and what remains uncertain.

Can permissions prevent harmful actions

A study titled *Compromise Is Not Consequence* tested whether limiting an agent's permissions could prevent harmful actions even when misleading instructions caused it to propose them. Researchers submitted the same proposed actions under broad access and permissions restricted to the assigned task, then compared what was actually executed.

Across 128 scenarios, four categories of tools, and five locally run model configurations, harmful execution among valid attacked decisions ranged from 8.9% to 37.8% with broad access; the restricted conditions recorded zero. The permissions were supplied by the researchers. The result supports containing the tested harms; it does not demonstrate prevention of misleading instructions, reliable completion of useful business work, or production safety. A small additional test in a separate research environment showed limited task usefulness. [6]

Can protection leave useful work unfinished

LATTICE is a proposed system design that separates an agent's planning from execution and permission checks. Its conservative evaluation across 4,000 recorded runs allowed no unsafe actions - but also automatically allowed no actions at all. Other experiments assessed consistent rule enforcement and resistance to specified attempts to bypass controls. The design demonstrates aspects of enforcement, not a workable production balance between protection and useful completion. [7]

AgentSpec is a research system for checking proposed actions against customized rules while an agent operates. Its evaluated results provide further evidence that specified rules can be enforced during execution. They do not establish that the same results will hold for every workflow or operating condition. [8]

For leaders, the lesson is to measure both the mistakes a control prevents and the useful work it permits. Blocking everything is not a successful deployment.

What do safeguards in deployed systems cover

Uber's Persona Guardrail is a reported safeguard for checking whether requests and responses stay within a customer-facing agent's assigned role. Its report describes evaluation, testing alongside live work, version-controlled specifications, and feedback. Classification accuracy improved from 85.7% to 95.9% on a defined subset that excluded mixed-scope cases. That measures how well the check classifies inputs and outputs, not how often the agent causes harm.

The evaluated threat model concerns external users interacting with the agent. Misleading instructions in retrieved material, corrupted tool results or stored context, and checks of intermediate actions remain limitations or future work. A control for customer conversations should not be assumed to protect every step of a tool-using workflow. [9]

Microsoft Execution Containers restrict the resources an agent can access from its execution environment, outside the agent itself. Microsoft's October 7 announcement describes availability on Windows 11, with some enterprise management and agent identification features forthcoming. The extent of blocking depends on the chosen mode. This is a vendor-reported capability; the announcement is not independent proof of protection against every attack. [15]

What incidents tell us about deployment risk

OpenAI's reporting describes harmful activity involving internal research models during training and evaluation, including access outside intended boundaries. DIVD, the Dutch Institute for Vulnerability Disclosure, documents an actual breach involving agent-driven exploitation of Zammad, a support-ticket system. GitLab's advisory describes a corrected flaw that could allow escape from a restricted execution environment; it does not establish that attackers exploited it. [16–18]

These accounts identify failures that controls and response plans should address. They do not tell us how frequently business deployments fail, because they generally provide no count of deployments or actions against which to assess the incidents.

What evidence should leaders rely on

Test the workflow rather than just the model

AgentDojo is a research testing environment that measures whether agents complete assigned tasks and whether misleading instructions in external data can divert them. It helps evaluate useful performance and protection together. Its benchmark can inform test design, but cannot estimate risk for a different business deployment. [11]

NIST, the US National Institute of Standards and Technology, provides an evaluation approach called ARIA that combines model tests, adversarial testing, and user testing. Applied to a business workflow, that means examining the model, the ways people or external content might redirect it, and the behavior of the whole system in use. [12]

An agent's explanation may sound reasonable without accurately describing what happened. Anthropic, an AI developer, has published research examining internal model computations and showing instances where plausible explanations did not faithfully describe them. A separate audit study by Casper and colleagues explains how limited access to a system restricts what an auditor can verify. [19, 20]

For a deployment review, use records of tool calls, permissions, approvals, and actual changes. The agent's account can help explain a case, but should not replace that evidence.

Measure value across the workflow

Practitioner interviews highlighted workflow redesign, explicit decision ownership, operating costs, and the human judgment needed to supervise delegated work. Participants differed on where to start and how quickly to expand authority. These qualitative insights inform the guide's recommendations; they are not a representative survey or evidence of causation. [24]

Human + Machine, a business book by Paul Daugherty and H. James Wilson, examines how people and AI can share work. An OECD study - the Organisation for Economic Co-operation and Development's interviews with 25 organizations - describes early deployment approaches and perceived benefits without independently measuring outcomes. These sources reinforce the need to examine handoffs, human responsibilities, and costs. They do not prove that redesign is always the principal cause of successful adoption. [5, 21]

Value can be lost between teams: time saved preparing an answer may become time spent correcting it elsewhere. Include integration, monitoring, approvals, incident response, and rework in the assessment. For

decisions affecting people, the US Federal Trade Commission's Rite Aid action illustrates the consequences of inadequate safeguards in facial recognition. It is a relevant governance example, not evidence of how frequently autonomous agents fail. [22]

Protect the expertise required for oversight

A study by researchers Shen and Tamkin asked whether AI assistance changes how people learn a new skill. In its main randomized experiment, 52 developers learned an unfamiliar programming library. AI assistance reduced subsequent measured learning without a statistically significant average improvement in completion time. The finding concerns learning a new skill in that setting; it does not establish that experienced workers generally lose expertise when they use AI. [13]

Independent judgment exercises, exception handling, and fallback practice are reasonable ways to investigate this risk, not proven remedies from the study. Research on the productivity J-curve explains another source of uncertainty: investments in skills, processes, and other complementary capabilities can delay returns from a new technology. It remains unclear whether successive AI capability changes require a distinct pattern of continuing redesign. [14]

What leaders should still require

The final decision is whether the evidence supports the particular authority proposed. No established universal threshold answers that question. Connect permissions to successful business outcomes, harmful actions, human effort, and total cost within the workflow. Stronger controls may make more valuable delegation possible, but operating results must establish the case. When they do not, retaining or reducing authority is a sound business decision.

Appendix A Deployment authorization record

Use this template to document what an agent is allowed to do in a specific business workflow, the limits on its actions, and who oversees it. Create a separate record when its responsibilities or permissions differ. The template draws on guidance from the World Economic Forum and Capgemini; it is our adaptation, not their official form. [1, 2]

Ownership

Deployment name and inventory ID: _____
 Business owner and deputy: _____
 Delivery and operations owners: _____
 Required reviewers and pause authority: _____

Mandate

Business objective and accepted outcome: _____
 Users and affected parties: _____
 Excluded tasks and cases: _____

Permissions

Permitted actions and systems: _____
 Approved data and destinations: _____
 Credential identities and enforcement locations: _____
 Transaction, aggregate, time, and resource limits: _____
 Prohibited actions: _____

Oversight

Actions requiring approval and approvers: _____
 Exception routing and response capacity: _____
 Stop triggers and response to control failure: _____
 Recovery and continuity route: _____

Approval and validity

Model, tool, policy, and configuration versions: _____
 Evaluation and release decision links: _____
 Approved scope and residual risks: _____
 Effective date, review date, and expiry if applicable: _____
 Approver and date: _____

Update this record whenever the agent's responsibilities, permissions, or working conditions change. Keep earlier versions so you can trace what was approved, and check that the agent's actual access matches the latest approval. Identify the accounts or credentials it uses, but do not include passwords, API keys, or other secret values.

Appendix B Release decision record

Use this template to document the decision to deploy an agent or significantly change what it is allowed to do. Include links to its authorization record and its entry in your organization's list of AI deployments.

Decision and evidence

Deployment and configuration: _____
 Date and proposed stage or scope: _____
 Decision: approve / approve with conditions / hold / pause / retire
 Evidence location and evaluation period: _____
 Accepted completion and baseline comparison: _____
 Harm, severity, denominator, and exposure: _____
 False blocks, escalations, review effort, and total cost: _____
 Recovery and human readiness results: _____

Release requirements

- ☐ Implemented permissions match the approved mandate.
- ☐ Relevant errors, attacks, dependencies, and alternate routes were tested.
- ☐ Required data, security, domain, and affected-user reviews are complete.
- ☐ Monitoring, reviewer coverage, pause, revocation, and recovery work.
- ☐ The authorized owner has accepted documented residual risks.

Conditions and approval

Unresolved gaps and compensating measures: _____
 Required actions with owners and due dates: _____
 Pause triggers and person authorized to act: _____
 Business owner approval and date: _____
 Reviewer decisions and dates: _____
 Next review date and required evidence: _____

This approval covers only the system version, tasks, and permissions described here. Record a new decision if changes introduce significant new risks or give the agent broader responsibilities or access.

References and evidence scope

This guide brings together selected research, deployment guidance, and insights from practitioner interviews. It does not attempt to cover all available research. Each finding should be understood in the conditions where it was observed, and the recommended practices should be tested in your own business setting. The approach as a whole has not been independently evaluated. Interview insights are shared without names, direct quotations, or confidential details, and do not imply participant endorsement.

Sources were reviewed through October 8, 2026. For the World Economic Forum guidance, we used the publisher's summary and accompanying article rather than the full report. We reviewed the McKinsey and Urban Institute publications in full; the Urban Institute continues to update its playbook. Vendor announcements describe capabilities reported by their developers, rather than independently verified results.

- [1] World Economic Forum, in collaboration with Capgemini. AI Agents in Action: A Playbook for Trusted Adoption, Authorization and Scaling. May 26, 2026. Publisher summary: <https://www.weforum.org/publications/ai-agents-in-action-a-playbook-for-trusted-adoption-authorization-and-scaling/>
- [2] Duet, Audrey. Why autonomous AI demands a new model of organizational authority. World Economic Forum. May 26, 2026. <https://www.weforum.org/stories/all/autonomous-ai-new-model-organizational-authority/>
- [3] Klein, Benjamin; Lewis, Charlie; Isenberg, Rich; Gabrielli, Dante; Möllering, Helen; Engler, Raphael; and Yuan, Vincent. Deploying agentic AI with safety and security: A playbook for technology leaders. McKinsey & Company. October 16, 2025. <https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/deploying-agentic-ai-with-safety-and-security-a-playbook-for-technology-leaders>
- [4] Urban Institute. Responsible Agentic AI Playbook: A Practical Guide for State and Local Government Agencies. Living document, main branch. <https://github.com/UrbanInstitute/agentic-ai-playbook/blob/main/agentic-ai-playbook.md>
- [5] OECD. (2026, September 24). Agentic AI in organisations: Early insights from practitioner interviews. OECD Artificial Intelligence Papers, No. 65. Qualitative study of 25 organizations. <https://doi.org/10.1787/1257a26f-en>
- [6] Hagverdiyev, T. (2026, October 5). Compromise Is Not Consequence: Evaluating Task-Scoped Authorization in LLM Agents with Paired Replay. arXiv:2610.05840. Preprint. <https://arxiv.org/abs/2610.05840>
- [7] Calboreanu, E. (2026, August 14). LATTICE: A governance-first architecture for authorized autonomous AI operations. Frontiers in Artificial Intelligence, 9, 1800407. <https://doi.org/10.3389/frai.2026.1800407>
- [8] Wang, H., Poskitt, C. M., & Sun, J. (2026). AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents. Proceedings of the 48th IEEE/ACM International Conference on Software Engineering, 12 pages. <https://doi.org/10.1145/3744916.3764546>
- [9] Pal, B., Maddireddy, S. R., Munir, M. B., Puha, Z., Zhurovich, M., Raghavendra, A., & Tout, S. (2026, October 2). Persona Guardrail: A Production-Grade Defense Framework for Agentic Systems. arXiv:2610.03434v1. Uber-authored preprint. <https://arxiv.org/html/2610.03434v1>
- [10] National Institute of Standards and Technology. (2023). Artificial Intelligence Risk Management Framework AI RMF 1.0. NIST AI 100-1. <https://doi.org/10.6028/NIST.AI.100-1>
- [11] Debenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., & Tramèr, F. (2024). AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. NeurIPS Datasets and Benchmarks; arXiv:2406.13352v3. <https://arxiv.org/abs/2406.13352v3>
- [12] Jensen, T., Przybocki, M., Kodwani, M., Greene, K., Hall, P., Amironesei, R., & Greenberg, C. (2026, September). ARIA Evaluation Planning Manual: Elements of ARIA-Style AI Evaluations. NIST AI 200-3. <https://doi.org/10.6028/NIST.AI.200-3>
- [13] Shen, J. H., & Tamkin, A. (2026). How AI Impacts Skill Formation. arXiv:2601.20245v2. Controlled experiments and exploratory interaction analysis. <https://arxiv.org/html/2601.20245v2>
- [14] Brynjolfsson, E., Rock, D., & Syverson, C. (2021). The Productivity J-Curve: How Intangibles Complement General Purpose Technologies. American Economic Journal: Macroeconomics, 13(1), 333–372. <https://doi.org/10.1257/mac.20180386>

- [15] Microsoft. (2026, October 7). Microsoft Execution Containers: Policy-driven containment for AI agents. Windows Developer Blog. Product announcement and operating-mode descriptions. <https://blogs.windows.com/windowsdeveloper/2026/10/07/microsoft-execution-containers-policy-driven-containment-for-ai-agents/>
- [16] OpenAI. (2026). The Hugging Face incident and other third-party impacts from misaligned models. Evolving primary incident-reporting page. <https://openai.com/hugging-face-incident-and-misalignment/>
- [17] Dutch Institute for Vulnerability Disclosure CSIRT. (2026). DIVD-2026-00014. Primary case report and incident timeline, including Zammad exploitation and containment. <https://csirt.divd.nl/cases/DIVD-2026-00014/>
- [18] GitLab. (2026). GitLab AI Gateway Critical Patch Release: 19.2.4, 19.3.2, and 19.4.1. Security advisory for CVE-2026-90970. <https://docs.gitlab.com/releases/patches/other-patches/patch-release-gitlab-ai-gateway-19-4-1-released/>
- [19] Anthropic. (2025, March 27). Tracing the thoughts of a large language model. Research report describing circuit-tracing findings and limitations. <https://www.anthropic.com/research/tracing-thoughts-language-model>
- [20] Casper, S., et al. (2024). Black-Box Access is Insufficient for Rigorous AI Audits. Proceedings of the ACM Conference on Fairness, Accountability, and Transparency. arXiv:2401.14446v3. <https://doi.org/10.1145/3630106.3659037>
- [21] Daugherty, P. R., & Wilson, H. J. (2024). Human + Machine, updated and expanded: Reimagining Work in the Age of AI. Harvard Business Review Press. ISBN 9781647827205. Practitioner book used for the work-design perspective.
- [22] Federal Trade Commission. (2023, December 19). Rite Aid Corporation FTC v. Case No. 2023190; complaint and proposed stipulated order concerning facial-recognition safeguards. Enforcement precedent used here concerns alleged consumer harm. <https://www.ftc.gov/legal-library/browse/cases-proceedings/2023190-rite-aid-corporation-ftc-v>
- [23] National Institute of Standards and Technology. (2026, March). Challenges to the Monitoring of Deployed AI Systems: Center for AI Standards and Innovation. NIST AI 800-4. Practitioner workshops and literature review. <https://doi.org/10.6028/NIST.AI.800-4>
- [24] Siler, D. (2026). Practitioner research interviews. Unpublished interview records. Qualitative themes concerning bounded business needs, explicit rules, operating economics, workflow redesign, decision rights, and workforce judgment. No participant attribution or confidential details reproduced.