# Velocity

V5 2026

Booz Allen

INSIGHTS FOR INNOVATORS

# AI vs AI

# REIMAGINING CYBER FOR A FASTER FIGHT

IN THIS ISSUE: a16z Interview p 4 / Zero-Knowledge Proofs p 10 / Trusted Agents p 40 / Zero Trust for OT p 54

# How Can You Trust Agentic AI? **Start With Engineering**

## Why trust must be designed, governed, and validated—not assumed

*By Jonathan Lebert, Marie Nguyen,* and Sahil Sanghvi

**Can I trust agentic AI to handle this operation?" It's a simple question. And a reasonable one to ponder.**

The push to consider this question at all is due to AI's remarkable ascent in just the last decade. It has progressed from predicting things to creating things, and now with agentic AI, to actually managing and doing things.

For IT leaders, the implications are enormous. Agentic AI can accelerate strategic planning, automate complex workflows, improve operational awareness, and augment workforces in ways that were impractical just a few years ago. These systems have the potential to transform how we analyze information, coordinate operations, and deliver services to citizens and consumers.

But as AI accumulates more capability, something counterintuitive happens: our trust in it wavers.

Upon reflection, this makes sense: as AI becomes more versatile, we assign it more complex and challenging tasks, pushing ever higher the stakes of those tasks—whether it is diagnosing a patient, countering cyberattacks, or orchestrating drone and satellite operations. And these higher stakes leave less room for error.

This ambivalence is particularly pronounced in the case of agentic AI. The very features that make agentic AI so compelling—its ability to make decisions and take actions independently, access data sources, invoke tools, interact with other systems, and autonomously adapt to changing conditions—are also what make it challenging to deploy and operate safely and securely in both government and regulated environments.

Agents, for example, introduce new identity and access requirements and, like human users, must be governed, audited, and tracked. This requires that we rethink our approach for how to manage these new actors. More to the point, we should monitor agents as potential insider threats. In fact, new agent behavioral analytic capabilities are already emerging in the marketplace to assist with this.

Another challenge is that agentic systems introduce a wide range of cyber threat vectors. Also, because they rely on large language models (LLMs), their reasoning is non-deterministic and often unpredictable. Moreover, their deployment in multi-cloud, multi-agent environments expands the addressable attack surface that must be protected.

These risks can be daunting for executives weighing whether to employ this powerful technology in service to their missions. This does not mean we should not be aggressively moving forward with agentic AI systems. As with any new technology, we need to lean into it, while also making sure we understand it and are managing how it is deployed.

So, the question becomes: Is it possible for agentic AI systems to operate securely, safely, reliably, and predictably in service to federal missions and business operations? And, if so, how?

The short answer is yes, it is possible, but it requires that we apply new approaches for how we design, build, and manage agentic AI systems.

## A Closer Look at Agentic AI's Risk Profile

Building organizational trust in agentic AI requires addressing four critical risks:

**Privacy:** Agentic systems often operate across large volumes of sensitive information—ranging from operational data to personally identifiable information (PII) and classified intelligence. Agencies must ensure agents access only the data they are authorized to use and sensitive information can't leak through prompts, outputs, or system interactions.

Identity management, access controls, and strict data governance become foundational capabilities when agents operate autonomously across multiple systems.

**Security:** Agentic AI systems introduce new cybersecurity risks that extend beyond traditional application security. Because agents frequently interact with external tools and services, adversaries may attempt to exploit those integrations to gain access to systems or sensitive data.

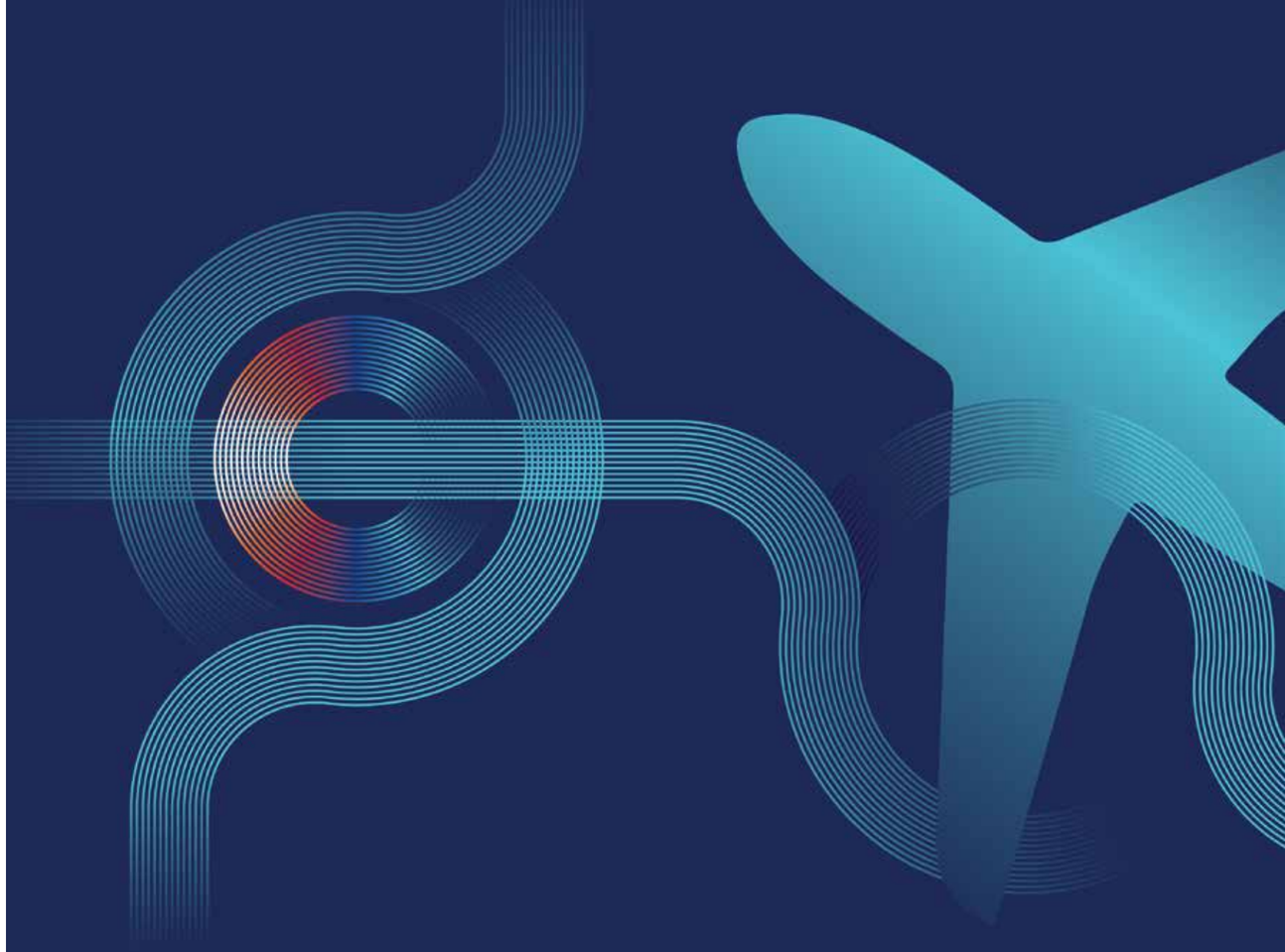
For example, attackers may attempt to craft malicious prompts that manipulate agent behavior or exploit tool integrations to execute unauthorized actions. Other threats include memory poisoning, where attackers rewrite the long-term memory that agents rely on to function; privilege escalation, where an attacker forces an agent with high-level access to perform an action on behalf of a low-level user, effectively escalating the user's privileges; and tool misuse, where attackers manipulate AI agents to abuse their integrated tools through deceptive prompts or commands, all while operating within authorized permissions.

**Performance:** Confidence in AI also depends on reliability. Generative models can produce hallucinations, inaccurate reasoning, or biased outputs under certain conditions. Over time, models may drift as operational environments change.

When agents are responsible not just for generating insights, but also for executing actions across systems, the consequences of these errors increase. Leaders must ensure agentic systems consistently produce outcomes that are accurate, auditable, and aligned with mission objectives.

**Resilience:** Agentic AI systems also introduce new operational dependencies. Multi-agent environments can involve dozens—or eventually hundreds—of interacting agents performing specialized tasks. If a component fails or behaves unexpectedly, agencies must be able to recover quickly and maintain continuity of operations.

Resilience requires robust fallback mechanisms, replay capabilities, and contingency plans that allow mission operations to continue even when AI systems encounter unexpected conditions.



## Trust by Design: What Aviation Teaches Us About AI

**When digital fly-by-wire systems were first introduced on commercial airliners in 1987, many observers were skeptical.**

At that point, commercial airliners were mechanically controlled. Their control surfaces were operated via cables and pushrods connected to the pilot's control sticks and rudder pedals. In a fly-by-wire system, a computer collects sensor data from the pilot's controls and relays those signals via wires to actuators that move the aircraft's control surfaces accordingly.

The question many had was: "Can we trust this new technology with our lives?" Perhaps the code could fail or the system could overrule the pilot. And what if the system operated unpredictably?

Engineers didn't solve this with assurances. They solved it with design. They built layers of trust into the system via:

- Multiple redundant computers running in parallel
- Dissimilar redundancy to ensure no single event can cause all parallel channels to fail simultaneously
- Strict "flight envelope" protections to prevent unsafe actions
- Clear, deterministic system responses—no surprises
- A rigorous configuration control process to produce and maintain quality software
- Continuous validation by regulators

The result wasn't just a working system; it was a trustworthy one. Fly-by-wire aircraft like the Airbus A320 are now among the safest ever built—not because they rely less on automation, but because their automation is tightly governed, observable, and constrained.

The lesson for agentic AI is clear: Trust doesn't come from capability alone. It comes from engineering systems whose behavior is bounded, predictable, and controllable—by design.

## Building Executive Trust in Agentic AI Demands a Different Approach

Before generative AI arrived, developing trust in AI meant evaluating the model thoroughly: training it, testing it, measuring it for accuracy, and then deploying it. That worked when systems were static, deterministic, and largely predictable.

Today, that model-centric playbook for building trust in AI no longer applies. Not because it was wrong, but because the technology has changed.

Now we have multiple agents working in tandem, each with specific roles; live connections to enterprise systems and external tools; real-time reasoning and adaptation; and autonomous decision making and execution.

Agentic AI doesn't just generate answers; it decides, acts, and coordinates—and most importantly, it operates within a larger, more complex system, not as a standalone model.

Moreover, the workflow of an agentic AI system can vary dramatically depending on the tools it invokes, the data sources it queries, and the other agents it coordinates with. While an agent's identity remains fixed, the permissions it exercises and the context it retrieves may differ based on the operational scenario. These dynamic interactions make strong identity governance, entitlements management, and observability essential.

In other words, the model is still important, but now there are many more components within a larger, dynamic system that must be vetted and monitored.

### Ensuring Trusted Performance Requires a Wider Focus

To deliver secure, safe, reliable, and predictable outcomes with agentic AI, leaders must shift their lens from models to systems, from point solutions to architectures, and from validation events to continuous assurance.

For federal leaders, for example, there is a pathway from experimentation to mission-scale impact with confidence. But it requires thinking bigger, designing holistically, and engineering trust into the system, not just the model.

## A Roadmap for Ensuring Secure, Safe, Reliable, and Predictable Agentic AI Performance

The best way to ensure trusted performance from an agentic AI system is to never trust it.

In other words, remove the trust element entirely by designing a system that, at every point of risk, has adequate safeguards built in to mitigate those risks throughout the lifecycle of that system.

This, of course, is the fundamental approach that underlies zero trust architectures as they apply to networks, applications, and critical infrastructure.

### An Architectural Approach Guided by Zero Trust Principles

Applying the principles of zero trust—never trust, always verify, micro-segmentation, least privilege, and assume there is a breach—to agentic AI is a critical step toward ensuring trusted performance. Embedding these principles into your architecture will mitigate many of the biggest security-related risks that can plague agentic AI systems, such as adversarial cyberattacks, improper data exposure, and access to systems without authorization.

But zero trust, by itself, is not enough. Agentic AI introduces a broader range of risks beyond the security concerns typically addressed by zero trust frameworks. In addition to security, it's important to address performance risks that stem from potential flaws in the AI models themselves. These can include algorithmic bias, hallucinations, model drift, or anything that might generate unexpected outcomes.

Here is a roadmap for ensuring secure, safe, reliable, and predictable agentic AI performance at five key stages: risk assessment, design, development, deployment, and future-proofing your system.

#### 1. RISK ASSESSMENT

Start by defining—and constraining—the risk surface.

Agentic AI introduces new actors (agents), new pathways (tools and APIs), and new behaviors (autonomous, non-deterministic action) that expand exposure to risk. Before a single line of code is written, be sure to understand and shape that exposure by thinking through these four factors.

## Claws: Autonomous AI That Doesn't Wait—Are We Ready?

**What if you had a platoon of autonomous AI agents on your computer that never stopped working?**

No pausing. No waiting for prompts. No asking permission. Just acting… continuously.

While you are sleeping or at a meeting, these digital workers prepare daily intelligence briefings for you every morning; monitor contract deliverables and compliance; alert you with bulletins when priority messages arrive; reply to emails; update your social media posts; and book your next trip.

Welcome to the age of AI claws.

This technology exists thanks mainly to OpenClaw, a fast-growing, free, and open-source autonomous AI agent launched in November 2025. OpenClaw performs real work—email, scheduling, reporting, coding—on your local computer using your tools, credentials, and data on a continuous basis. It executes tasks via large language models (LLMs) using messaging platforms as its main user interface. And, perhaps most important, it retains memory over time, capturing user preferences, ongoing projects, and past email threads.

This persistent state of AI autonomy adds value, but it also scales risk. Here are some risks to consider:

**Identity challenges:** Who acted, the human or the machine? Without clear identities and audit trails, accountability blurs fast.

**Attack surfaces expand:** Always-on systems do not just increase exposure; they live in it. Prompt injection, tool abuse, and memory manipulation move from edge cases to persistent threats.

**Access dangers:** Claws need real credentials to deliver value. This gives them real authority to act, but there are real consequences when something goes wrong.

**Unpredictable behavior:** Claws learn, adapt, and infer. This leads to improved performance, but it can—and does—lead to occasional unintended actions and missteps.

**Exposed supply chains:** Plugin ecosystems and external integrations create invisible dependencies that serve as new paths for compromise.

NVIDIA's NemoClaw beefs up security and privacy protections around OpenClaw. It includes features such as sandboxing, default-deny network egress, policy enforcement, and audit logging. Additional focus continues on credential management, security maturity, integration stability, cost predictability, and usability.

**The bottom line:** NemoClaw is not yet fully production-ready, but claws are evolving rapidly and they will no doubt assume an ever-greater amount of our focus as this technology matures.

## Applying Zero Trust Principles to Agentic AI

**Zero Trust Architecture is built on a simple premise: never trust, always verify.**

Every user, agent, device, and system interaction is treated as untrusted until authenticated and authorized—and then they are continually re-authenticated and re-authorized as needed.

In highly dynamic agentic AI systems, zero trust principles are just as critical as they are in traditional network systems. AI agents must authenticate not only with users, but with each other and with every tool they invoke. Access must be dynamically enforced based on identity, context, and mission need.

Equally important is the zero trust principle of least privilege. Each agent should have access only to the specific data and tools needed to complete its task—nothing more—to limit the impact of any compromise if there is one.

Combined with micro-segmentation and continuous monitoring, zero trust transforms agentic systems into highly controlled environments where trust is continuously validated, not assumed. In addition, an effective zero trust architecture for an agentic AI system must focus on securing the five key pillars of identity, devices, networks, applications, and data for a layered security approach that covers the entire agentic ecosystem.

What is different about zero trust for agentic systems is in how it is implemented. For example, zero trust safeguards must also be designed and implemented for the orchestration and management components of an agentic system.

Also, authenticating and re-authenticating agents as they operate can be challenging because they operate non-deterministically, at machine speed, at scale, and can be either persistent or ephemeral. So, special considerations must be given to the tools and methods used to implement zero trust protections to agentic ecosystems.

**Credentials for agents**
Just like human users on a traditional network, AI agents must have identities and be authorized to do what they do: interact with other agents or business platforms, invoke tools, and access data. Modern identity and access management (IAM) approaches use role- and attribute-based controls, policy-as-code, and just-in-time credentialing to ensure agents can interact only with the systems, tools, and data required for their approved workflows.

But deciding the entitlements for a particular agent may be one of the most complicated deliberations you will have. Is the agent persistent or temporary? What entitlement constraints will be placed on it? How will its behavior be monitored? Will it have a cached memory? If so, how much and for how long? In most cases, these questions will be determined by the specifics of the use case scenarios the agent will be operating within.

However those questions are decided, it will be important to assign the appropriate non-person identifiers to agents and to monitor their behaviors continuously.

**Memory constraints**
Memory for an agent is a double-edged sword: it improves performance but increases risk. As a result, calibrating an agent's memory parameters is akin to finding the right balance between performance and risk. Think of memory as enabling an agent's continuous learning: a repository of contextual information specific to tasks and enterprises that enhances an agent's proficiency.

The flipside is that long-term memory can unintentionally store sensitive data or inadvertently leak information. Planners will need to deliberately define what gets stored, for how long, and under what conditions it is purged. For some mission environments, such as edge devices operating on a battlefield, minimal memory may be the safer default.

**Data governance**
Agentic systems can access a variety of sensitive datasets. Agencies must define and enforce clear policies on data access, lineage, and usage to ensure agents interact only with data they are authorized to access. One way to support data governance is through a "composite identity," which combines the identity of a human user initiating a query and the identity of the AI agent supporting that query. A key component of attribute-based access control (ABAC) for AI systems, composite identities ensure that a user can't gain elevated privileges to access restricted data through an AI agent.

Data governance rules must be clear and enforced to ensure input validation, personally identifiable information (PII) protection, and output governance.

**Kill switch for containment**
Autonomy demands control. To limit the scope of a potential failure or compromise, agencies should create the ability to immediately terminate an agent, sever a connection, or halt a workflow if anomalous or unsafe behavior is detected.



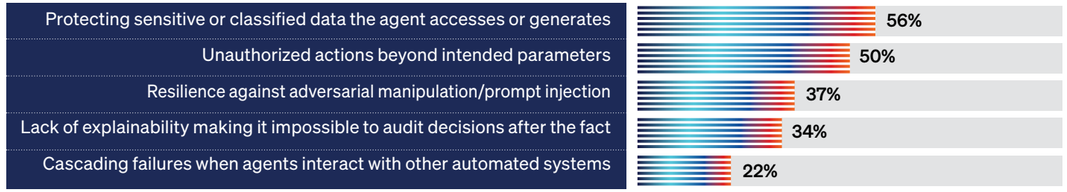
## Federal Leaders on Agentic AI: Trust, Risk, and Readiness

An exclusive survey of over 100 federal IT and cybersecurity leaders found that more than half—58%—have either deployed or are piloting AI agents at their agencies. Another 22% planned to deploy in the next 12 months. However, few expressed high confidence in their ability to manage these systems securely.

**CONFIDENCE IN MANAGING AGENTIC AI SECURELY**

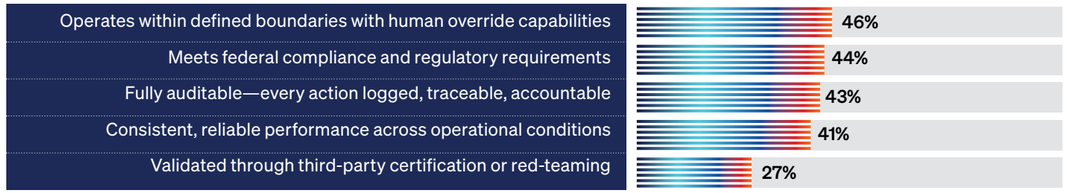
| 3% | 25% | 50% | 19% | 4% |
|---|---|---|---|---|
| Extremely confident | Very confident | Moderately confident | Slightly confident | Not confident |

### Top Risks Leaders Need to Address




### Factors Needed to Expand Agent Use


### How Can AI Agents Be Trusted?




**WHO IS ULTIMATELY RESPONSIBLE FOR SECURITY INCIDENTS INVOLVING AI AGENTS?**

| 26% | 22% | 22% | 21% | 10% |
|---|---|---|---|---|
| The business, mission, or program owner | The IT organization | Accountability is unclear—an unresolved gap | A dedicated AI governance function | A cybersecurity or risk function |

## 2. DESIGN

The design stage involves thinking through the operations of an agentic system in exhaustive detail to understand what acceptable, expected performance looks like and how to engineer for risk and performance through safeguards, guardrails, monitoring, and oversight. Start by defining the business process and expected performance as well as identifying when a human needs to be involved.

**Defining the re-engineered business process**
Start with the mission or business workflow. How will humans and agents work together and what will be the roles and functions of each? What decisions are being made and by whom? What actions are being executed and how will they be orchestrated? Where does autonomy accelerate outcomes—and where does it introduce risk? This step requires deep stakeholder engagement to map, validate, and refine the process.

**Defining expected agentic AI behaviors and performance**
Agents must operate within clearly defined boundaries. What does "good" look like, and what level of accuracy, explainability, and timeliness is required? These expectations must be explicit, measurable, and aligned to mission needs. This step will inform where to design in appropriate monitoring and oversight mechanisms.

**Identifying appropriate human-in-the-loop (HITL) checkpoints**
Not every decision or agent should be automated. High-risk actions, ambiguous outputs, and mission-critical steps require human validation. When designing the orchestration and management layer of an agentic system, decide where checkpoints must be designed into workflows for human review, approval, or override of agent actions.

## 3. DEVELOPMENT

This is where trusted performance moves from concept to evidence.

Developing agentic systems requires disciplined engineering practices that embed modern cybersecurity practices and protections throughout and validate not just outputs, but also behaviors. Key components include DevSecOps, MLOps, SBOM scanning, and testing

**DevSecOps**
DevSecOps embeds automated security, compliance, and validation checks throughout an agent's lifecycle using techniques such as continuous monitoring, specialized LLM auditing, and sandboxed execution. When applied to agentic AI, DevSecOps combines traditional standard pipeline security—including security testing, vulnerability scanning, and compliance checks—with AI-specific testing—like data poisoning checks and agent reasoning audits—to mitigate risks like prompt injection and uncontrolled autonomy.

**MLOps**
AI models evolve and sometimes drift beyond their original intended scope. MLOps is the set of practices—such as data preparation, continuous monitoring, retraining, and performance management—that helps ensure the LLM models that run agentic systems remain accurate and reliable over time.

**Software bill of materials (SBOM) scanning**
Agentic systems rely heavily on open-source and third-party components. SBOM scanning provides visibility into these dependencies, helping identify vulnerabilities and manage supply chain risk proactively.

**Tested performance**
Once developed, agentic systems must be tested against real-world conditions, including edge cases and adversarial scenarios, to validate the re-engineered business processes and ensure they meet the expected behaviors and performance.

## 4. DEPLOYMENT AND OPERATIONS

When it is time to deploy, observability and oversight take a central role.

The goal is to ensure the performance you are getting mirrors expectations—and to course-correct when it doesn't. Performance includes all aspects of the system, including agent interactions and orchestration, ICAM enforcement, and AI model functioning.

It is critical to employ monitoring and oversight throughout the entire lifecycle of the agentic system—because AI models can drift or get corrupted at any time—but the intensity of the monitoring and oversight can be adjusted as needed.

**Continuous log monitoring**
Monitor and analyze the system's activity logs to see where it is performing well or not and then adjust as needed. Employ tools, such as to log tool interactions, token usage, decision paths, error rates, and latency.

**Human-in-the-loop oversight**
Start your deployment with a robust human-in-the-loop (HITL) presence so human experts can review and assess in real time the system's performance at key points in the workflow. Again, adjust as needed until the performance reliably meets expectations.

"Developing agentic systems requires disciplined engineering practices that embed modern cybersecurity practices and protections throughout and validate not just outputs, but also behaviors."

## Model Context Protocol (MCP) Security: What Leaders Need to Know

**Agentic systems rely on multiple protocols, but the most prominent one is MCP, an open standard that enables AI applications to connect seamlessly with external data sources, tools, and services. This makes MCP foundational in any agentic system.**

The challenge is that MCP introduces new security risks because, unlike traditional APIs, MCP interactions are dynamic and driven by AI decision making, creating a broader and less predictable attack surface.

So, securing agentic AI necessarily means securing MCP. This requires a full lifecycle approach that combines traditional cybersecurity best practices—such as those cataloged in the NIST SP 800-53 controls—with exotic AI security techniques, such as prompt filtering, for example.

MCP security must be applied across three lifecycle phases:

- Runtime: To protect live interactions between AI and tools.
- Build-time: To prevent insecure code from entering the environment
- Deployment-time: To secure infrastructure and the execution environment

At runtime, every request must follow zero trust principles, such as authenticate users, enforce role-based access, validate connections, and apply centralized policy controls. AI-specific safeguards—such as prompt injection defenses, human-in-the-loop approvals for sensitive actions, and rate limiting—are essential to prevent misuse or runaway automation.

At build and deployment, organizations should enforce container hardening, artifact scanning, red team testing, and strict network isolation. Only vetted, approved MCP servers should be allowed.

Done right, MCP acts as a guardrail, ensuring agents operate within controlled, auditable boundaries while still enabling the flexibility that makes agentic AI powerful.

**Other methods**
In addition to HITL and log audits, other monitoring and oversight mechanisms include:

- Citations: Require agents to show the sources of their reasoning for verification.
- SQL-based validation tools: Fact-check AI outputs, particularly LLM-generated code or data insights, by executing, parsing, or comparing generated queries against a trusted database schema.
- Automated anomaly detection: Continuously assess performance and detect drift against established baselines.
- Stakeholder feedback mechanisms: Capture real-world input to refine system behavior.

**Precautions for sensitive workloads**
For highly sensitive environments, agencies should consider deploying models within internal networks rather than relying on external commercial services. Data is processed on premises, reducing exposure and strengthening control.

**Start small, then scale up**
Wherever possible, begin your agentic deployment with contained, low-risk use cases. Refine it as needed, prove value, build confidence, and then expand to more complex, mission-critical applications. This phased approach reduces risk while accelerating learning.

**Stateful memory backups**
Continually back up the stateful memory of your agentic system. Resilience requires continuity. Systems should maintain backups of workflows and memory states—allowing agents to resume operations without having to restart from scratch. This supports both operational efficiency and mission reliability. And, of course, regularly back up critical data that is accessed and modified by agents to ensure that, if something goes wrong, nothing critical is permanently lost or impacted.

**Evaluate mission value**
Performance is not just technical—it is operational. Agencies must continuously evaluate whether systems are delivering meaningful improvements to mission outcomes.

## 5. FUTURE-PROOFING

Agentic AI isn't static, nor are the risks and opportunities. Ensuring trusted performance also means preparing systems to evolve tomorrow in response to changing circumstances. Two emerging best practices can help agencies stay ahead of change:

**Audit agents for high-risk operations**
In safety-critical software engineering, "dissimilar redundancy" is a long-standing approach: Two independent implementations perform the same operation, and discrepancies trigger immediate investigation. Agentic AI needs an equivalent.

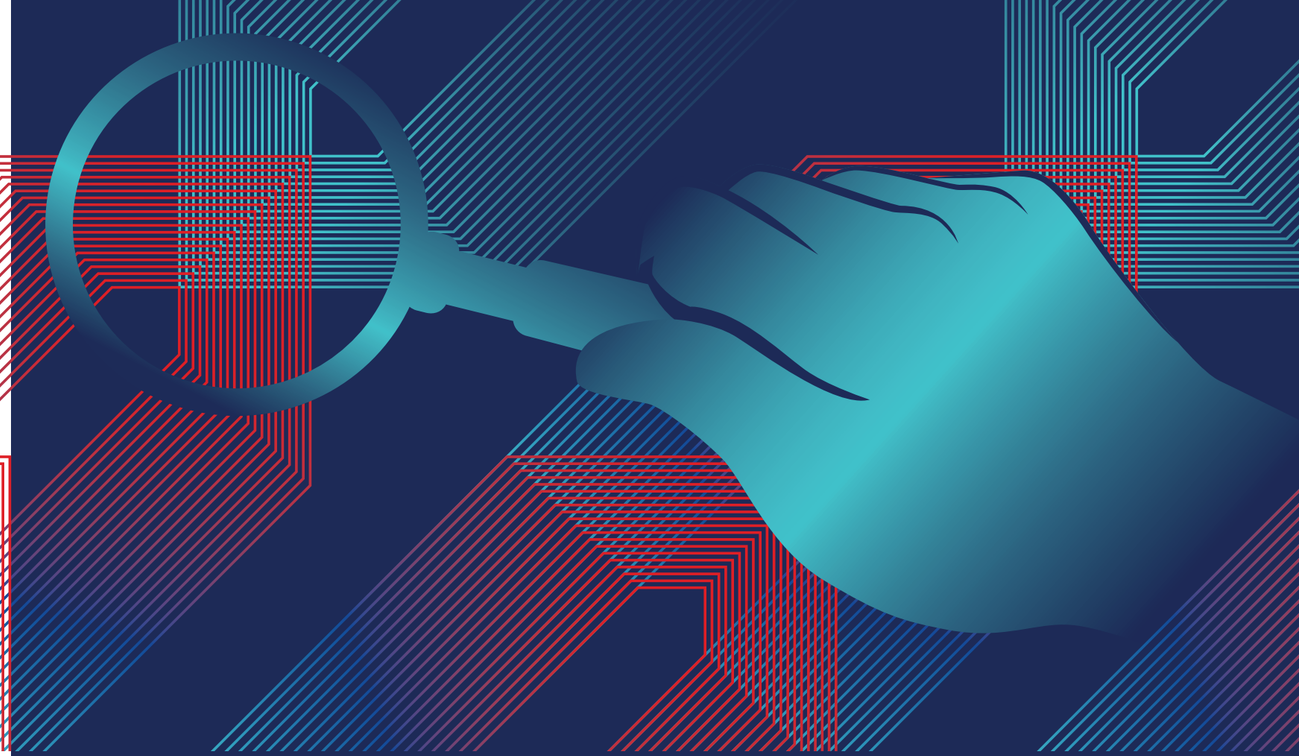
Audit agents—ideally running on different LLMs or toolchains—can independently re-execute sensitive operations such as data mutation, privilege escalation, financial actions, or mission-critical updates. When results diverge, the system flags the action for review. This ensures autonomy never outruns accountability, particularly when agents have real entitlements and access to operational systems.

**Model and component agility**
The pace of AI innovation makes LLM agility as essential as cryptographic agility has been for secure systems. Architectures should be designed and built so agencies can swap out models—or simulation engines, rendering engines, mapping services, or data stores—with minimal friction.

New models will excel in some areas, lag in others, and sometimes require targeted tuning or retrieval augmentation, much like onboarding and training a new employee. Agility ensures agencies can quickly pivot to safer, faster, cheaper, or more robust models as technology, costs, or operational requirements change. But it's not enough to simply design model agility into the system; it's important during the operational phase to leverage that agility by keeping models and other system components modern and optimized for peak performance.

The goal is to understand and govern emerging technologies while deploying them quickly and correctly. When upgrades or new technology adoption lags, shadow IT—and the risks that come with it—grows. These two practices—designing in model agility and auditing agents for sensitive tasks—will give leaders confidence not just in today's agentic systems, but in whatever comes next.





# Red Teaming: A Critical Component of Trusted Agentic Performance

**Agentic AI systems are powerful. But that power comes with risk—and that's where red teaming comes in.**

At its core, red teaming means thinking like an attacker, probing agentic systems with specialized prompts and tools to see how they might fail, misbehave, or be exploited, ideally long before real attackers do.

Employing red teams with expertise in agentic systems is critical to ensuring they perform as needed because traditional security controls don't fully account for the additional vulnerabilities and risk that the AI components present.

**How red teaming works**

Part sleuthing, part experimentation, a red team operates like a squad of detectives. The team will start by learning everything it can about the agentic system: how it is built, what it connects to, and what it's supposed to do.

From there, the team launches targeted and exploratory attacks; injects malicious or misleading inputs (such as "memory poisoning" or context manipulation); scours the outputs and logs for the tiniest of anomalies; identifies points of vulnerability; and follows up with more experiments.

The process is iterative. Early findings lead to deeper probing, often uncovering issues no one anticipated.

Common issues that red teams discover include:

- Data exfiltration: an ability to manipulate the AI to extract sensitive information
- Tool misuse: vulnerabilities that allow agents to be directed to take harmful actions
- Hallucinations: the creation of false or misleading outputs
- Design flaws: bugs, infinite loops, or unintended behaviors
- Identity gaps: agents operating without clear accountability or traceability

Many of these are security risks; some are reliability or trust issues. All of them matter.

Through rigorous testing, red teams help organizations discover vulnerabilities before attackers do; validate whether safeguards actually work; improve system reliability and trustworthiness; and build confidence in deploying AI at scale.

The bottom line is red teaming takes assumptions out of the equation by forcing agentic systems to prove they are safe and trustworthy. And in a world where AI can act, decide, and access critical resources, that shift—from assumption to validation—is everything.

## Engineering Trusted Performance to Unlock Mission Value

We now live in an era of agentic AI, and the stakes of this technology performing as needed—securely, safely, reliably, and predictably—will only grow as we continue to harness it in support of federal missions.

The path forward is not to slow down adoption because of this, but rather to change how we engineer trust in these new capabilities. Building executive trust in agentic AI systems is not a model problem; it is a systems challenge—one that spans the entire lifecycle, from risk assessment through design, development, deployment, and operations.

It requires a fundamental shift in mindset from trusting outputs to engineering outcomes, from validating models to governing systems, and from point-in-time testing to continuous assurance.

In short, as the stakes of agentic AI performance increase, so too will the effort and complexity needed to ensure that performance meets our expectations.

This lifecycle approach helps resolve one of the most pressing concerns that federal leaders must grapple with as they consider how to safely harness this promising technology on behalf of their missions and business operations: how to balance autonomy with accountability.

The answer is not to choose between them. It is to architect for both.

When agentic systems are built with zero trust principles at their core and embedded with clearly defined guardrails, least-privilege access, and continuous observability, autonomy becomes bounded and predictable. Agents can act independently, but only within the scope they are authorized to operate.

At the same time, accountability is embedded into the system itself. Every action is logged. Every decision is traceable. Human-in-the-loop checkpoints ensure oversight where it matters most. And built-in safeguards—such as escalation thresholds and kill switches—provide immediate control when conditions deviate from expectations.

In this model, autonomy is earned, accountability is enforced, and trust grows over time. This is not just a technical approach. It is an operational one.

Both federal agencies and commercial enterprises are pursuing calculated investments in agentic AI with the expectation that it will deliver high value to their missions and business operations. This systems-wide architectural approach to building assured performance in these agentic AI systems offers a path toward achieving that goal.

---

*Sahil Sanghvi, a vice president; Jonathan Lebert, a solution director; and Marie Nguyen, a senior AI solution architect, lead efforts in the company's Chief Technology Office to advance next-generation agentic AI solutions.*

*The authors would like to thank Eric Ferguson, Matt Keating, Bill Murrin, and Amy Wagoner for their important contributions to this article.*

*Survey Methodology*

*Market Connections and Booz Allen partnered to design an online survey of over 100 federal government decision makers/influencers, involved with IT and cybersecurity, regarding AI's impact on government cybersecurity practices. The survey consisted of 14 questions and was fielded in April of 2026. Due to rounding, responses may not add up to exactly 100%.*

### SPEED READ

Agentic AI is rapidly reshaping government and commercial enterprises, but its expanding capabilities also amplify risk, uncertainty, and the need for new forms of executive trust.

Trust can't be assumed or promised—it must be engineered, producing agentic systems whose behavior is bounded, observable, and controllable by design.

Leaders can ensure secure, safe, reliable, and predictable agentic AI performance by adopting a lifecycle systems approach that embeds identity governance, guardrails, monitoring, red teaming, and continuous assurance throughout the ecosystem.

Booz Allen®