

Building Trustworthy AI: Data Integrity and Action Integrity

How Runtime Governance Helps Enterprises Complement Trusted Data with Controlled, Auditable AI-Agent Actions

Thank you for your interest
in exploring this content.

Carahsoft is the **Trusted Government IT Solutions Provider**® supporting a broad portfolio of industry-leading technologies through OH MSCA, NASPO ValuePoint, Texas DIR-CPO-5683 and a wide range of other contract vehicles.

As the **Master Government Aggregator**®, Carahsoft connects government agencies, industry partners, and technology providers to deliver innovative, mission-focused solutions.

In partnership with Broadcom, we provide technology solutions that drive modernization, strengthen operations, and ensure compliance with evolving government standards.



To learn more about how Carahsoft can support your technology needs, please visit carahsoft.com



Explore More Resources:
carah.io/BroadcomResources



Join Events & Webinars:
carah.io/BroadcomEvents



Discover Technology Solutions:
carah.io/BroadcomSolutions



Learn About Procurement:
carah.io/BroadcomContracts



Connect With Our Team:
Broadcom@carahsoft.com
(571) 662-3260

Solution Brief

Challenges

Enterprises need more than good models when AI agents can access tools, data, and operational workflows.

Opportunities

Runtime governance helps organizations enforce identity, authority, and accountability for autonomous AI actions.

Business Outcomes

Improve control, traceability, and confidence as AI agents move from experimentation into production use.

Building Trustworthy AI: Data Integrity and Action Integrity

How Runtime Governance Helps Enterprises Complement Trusted Data with Controlled, Auditable AI-Agent Actions

Overview

Bruce Schneier's article, Building Trustworthy AI Agents¹, makes a sharp and timely argument: the missing foundation for trustworthy AI is integrity. He argues that while availability has improved and confidentiality is getting attention, integrity remains underdeveloped, especially as AI systems become more personalized, more persuasive, and more embedded in decisions that affect people's lives.

His proposed answer is architectural. Rather than letting AI providers hold and interpret all personal context directly, Schneier argues for separating personal data stores from AI systems so control, verification, and security can evolve more independently. In that model, trustworthy AI depends not only on smarter systems, but on trustworthy data layers with stronger user control, provable accuracy, and protection against manipulation.

That argument matters well beyond the consumer context, Schneier emphasizes. As enterprises connect AI agents to internal data, tools, APIs, and business workflows, they face a parallel trust problem: not only whether the underlying data is reliable, but whether increasingly capable agents are authorized to act at all, under what conditions, and with what accountability.

A useful distinction emerges here. Schneier is primarily concerned with data integrity: whether the information an AI system uses is accurate, protected, and under the right party's control. AgentMinder addresses a different but complementary problem, action integrity: whether an AI agent can access the right systems, invoke the right tools, stay within approved authority, and produce evidence that its actions were authorized and auditable.

Trustworthy AI increasingly requires both. A secure data layer without runtime controls still leaves room for unauthorized or unaccountable action. A strong authorization layer without trustworthy data still leaves systems vulnerable to bad inputs, manipulation, and incomplete context. For enterprises moving from AI assistance to AI action, the challenge is building both layers into a coherent trust architecture.

1. Bruce Schneier, "Building Trustworthy AI Agents," IEEE Security & Privacy, Jan.-Feb. 2026, <https://www.computer.org/csdl/magazine/sp/2026/01/11369814/2dHi8b9dIAI>.

What Schneier Gets Right

Schneier is right to frame integrity as central to AI trust rather than as a secondary technical feature. If AI systems are going to retrieve personal or organizational context, make recommendations, trigger workflows, or influence decisions, then trust depends on more than model quality. It depends on whether the surrounding architecture preserves accuracy, accountability, and meaningful control over how information is used.

He is also right to argue that architectural separation matters. Treating data stewardship and AI reasoning as distinct functions creates room for stronger controls, clearer accountability, and a more realistic division of responsibility between the systems that store context and the systems that interpret it. That insight becomes especially important as organizations try to scale AI without collapsing data governance, application governance, and model behavior into a single opaque control surface.

His six requirements point toward a broader lesson: trustworthy AI is not just a model problem. It is a systems-design problem involving storage, access, proof, control, usability, and defense against tampering. In that sense, Schneier's article is best read not only as a proposal for personal AI assistants, but as part of a larger argument that AI trust must be grounded in enforceable technical architecture rather than goodwill, convenience, or provider assurances.

That framing aligns with an important enterprise reality. Once AI systems move beyond generating answers and begin interacting with real systems, a new question becomes unavoidable: not only *Is the data trustworthy?* but also *Is the action authorized?* Schneier's article is strongest when it pushes the industry to take that trust question seriously at the architectural level.

Where the Enterprise Problem Differs

The enterprise trust problem is not identical to the consumer trust problem Schneier describes. Schneier focuses on individuals controlling personal data stores and protecting themselves from manipulation, inaccuracy, and hidden inferences. Enterprise organizations, by contrast, must govern how AI agents act across shared systems, delegated authority chains, regulated workflows, and sensitive operational environments.

That difference matters because enterprise agentic systems introduce a second layer of risk. Even if the data layer is well governed, an AI agent may still be over-privileged, use the wrong tool, exceed delegated authority, access an unintended resource, or trigger downstream actions without adequate oversight. In other words, data integrity does not automatically produce action integrity.

As organizations experiment with autonomous and semi-autonomous agents, the central challenge shifts from model output quality to governing machine-executed authority. That includes identity, authorization, delegation, tool access, and the ability to reconstruct what happened after an agent acts inside real workflows.

Where Runtime Governance Matters

Schneier's article focuses on the integrity of personal data, but enterprise AI systems introduce a second trust problem: whether an agent is authorized to act once it is connected to real tools, APIs, and business processes. In enterprise settings, trust does not end with secure storage or accurate context. It also depends on whether the system can enforce who the agent is, what it can do, which services it can reach, and how its actions can be reconstructed after the fact.

That challenge becomes more acute as organizations move from AI assistance to AI action. Once agents can retrieve records, invoke tools, delegate work, update systems, or trigger downstream processes, the central question is no longer only whether the model generates helpful output. It is whether machine-executed authority is bounded by identity, purpose, policy, and evidence that human operators can review.

A trustworthy agentic architecture needs more than prompts, static API keys, or application-level permissions. It needs a control layer that can evaluate whether a requested action aligns with the agent's declared role, approved mission, delegated authority, and permitted access path before allowing it to proceed.

In that sense, runtime governance is to agent actions what data integrity is to AI inputs: one protects the trustworthiness of the information being used; the other protects the trustworthiness of the actions being taken. Enterprises that want reliable AI outcomes increasingly need both.

How AgentMinder Helps

AgentMinder addresses the action-integrity side of this problem by giving enterprises a dedicated control plane for agent identity and authorization. Rather than treating an AI agent as just another application client, it gives each agent a distinct identity, binds its authority to an approved mission, evaluates actions against declared intent and policy, and preserves the evidence needed for oversight and investigation.

That matters because enterprise agentic systems are rarely simple one-step interactions. They often involve orchestration layers, delegated tasks, multiple tools, and chains of action that move across systems and services. AgentMinder is relevant in this environment because it is designed to govern not only whether an action is allowed, but also on whose behalf it occurs, within what scope, and with what traceability.

Several aspects of that architecture map meaningfully to Schneier's concerns, even if they do not solve the same problem directly. Its model-agnostic design aligns with the idea that trust should not be tied to a single AI provider. Its mission-aware authorization provides selective disclosure for actions, and its delegation controls address a multi-hop enterprise problem that consumer-oriented trust models often leave underexplored.

Most importantly, AgentMinder helps organizations move beyond a trust model based on hopeful prompts and broad permissions. It gives enterprises a way to enforce least privilege, preserve accountability, and constrain autonomous behavior at the point where AI agents interact with real systems. That is a narrower claim than *trustworthy AI* in the broadest sense, but it is a practical and increasingly necessary one.

What AgentMinder Does Not Solve

It is equally important to keep the claim boundary clear. AgentMinder does not solve the core data-layer problem Schneier is centered on: it does not provide a personal data store, prove the accuracy or completeness of underlying data, or establish consumer sovereignty over how personal information is collected, inferred, and shared. Those are different architectural and governance challenges.

It also does not resolve model-behavior problems such as manipulation, gaslighting, hallucination, or reasoning over incomplete and inaccurate context. AgentMinder can govern whether an agent is authorized to act, but it does not determine whether the model's interpretation of the world is truthful, fair, or psychologically manipulative. That remains a broader model-safety, data-quality, and product-design problem.

This distinction matters because it strengthens the credibility of the positioning. Schneier's proposal and AgentMinder are not rival answers to the same question. They address different layers of the trust stack: one focused on trustworthy personal data and user control, the other on trustworthy enterprise actions and enforceable authorization.

A mature AI architecture will likely need both kinds of controls, along with additional layers for model evaluation, cybersecurity, privacy, legal governance, and human accountability. The practical message is not that one platform solves trustworthy AI, but that trustworthy AI increasingly depends on multiple control layers working together.

Why This Matters Now

The broader significance of Schneier's article is that it pushes the AI industry to think architecturally about trust. As AI systems become more autonomous, more connected, and more deeply embedded in personal and organizational workflows, trust can no longer be treated as a vague aspiration or a model-quality attribute alone. It has to be designed into the systems that store context, govern access, and authorize action.

For enterprises, that shift is especially urgent. Many organizations are already experimenting with agents that can search internal knowledge, call APIs, operate across business systems, and carry out limited tasks with minimal human intervention. The closer those systems move to real authority, the more important it becomes to ensure that autonomy is bounded by identity, mission, policy, and evidence.

That is why Schneier's argument and AgentMinder's role should be understood as complementary, not competitive. Schneier highlights the need for trustworthy data and meaningful user control. AgentMinder addresses the equally important need for trustworthy enterprise action through runtime identity, authorization, delegation control, and auditability.

The deeper lesson is simple: AI trust will not come from models alone. It will come from architectures that make both data use and autonomous action governable, constrained, and accountable.