



MetrikConnect

• AI SECURITY · INSIGHTS

Security Can't Be the Last Sprint

Why AI agents need a secure foundation from day one, and how to build one without slowing your AI work down

Spencer Kryczka

Founder & CEO, Metrik Connect Solutions

October 5, 2026 · 7 min read

metrikconnect.com

IN BRIEF

AI is moving from pilots to production agents that act across real systems. Most programs still schedule security as the final phase, after the architecture is locked. This piece explains why that order fails for agentic AI, and lays out a seven-layer, framework-based approach that runs security in parallel with the build.

Most organizations I talk to are past the AI experiment phase. They have agents summarizing tickets, drafting customer replies, querying data warehouses, and increasingly taking actions on their own: opening a change request, provisioning a resource, moving a record from one system to another. The business case is clear and the pressure to ship is real.

What hasn't caught up is the project plan. In engagement after engagement, security shows up as the last line item: build the platform, deploy the workloads, then bring in the security team for a review before go-live. That sequencing was never ideal for traditional applications. For agentic AI, it breaks down completely.

What changes when software starts acting on its own

A traditional application does what its code tells it to. An AI agent decides what to do based on instructions, context, and whatever data it can reach, then calls tools to do it. That shift introduces three problems security teams aren't used to seeing at the same time.

- **New identities.** Every agent is a non-human identity that needs credentials, permissions, and an owner. Many teams start with a shared service account or a long-lived API key because it's fast. Those shortcuts tend to become permanent.
- **New data paths.** Retrieval pipelines, vector stores, and tool integrations create routes to sensitive data that didn't exist in your original data classification or access model.
- **Chained decisions.** A single user request can trigger a multi-step chain where one agent calls another, which calls an API, which writes to a system of record. Every step is an access decision. As orchestration gets more complex, the blast radius of one over-privileged agent grows with it.

The OWASP Top 10 for LLM Applications captures this well. Alongside prompt injection, it calls out sensitive information disclosure and excessive agency: agents that have more permissions, more tools, or more autonomy than the task requires. Those are architecture problems, not configuration problems, and they're very hard to fix after the fact.

Why "we'll secure it later" gets expensive

When security arrives at the end, the review usually finds the same things: agents running under shared credentials, broad permissions granted during development that were never narrowed, no way to trace an action back to the agent and the person who triggered it, and data flowing into prompts that was never meant to leave its original system.

Fixing those issues at that stage means re-plumbing identity, re-scoping access, and sometimes redesigning how agents talk to each other, all while the business is waiting for a launch date. The alternative is to accept the risk and ship, which is how today's shortcut becomes next year's incident.

“The cheapest time to give an AI agent the right identity and the right permissions is before it exists.”

Start with frameworks your security team already trusts

You don't need to invent a new methodology for AI security. Several well-established frameworks already give you a common language with risk, compliance, and engineering teams:

- **NIST AI Risk Management Framework** for governance: how you map, measure, and manage AI risk across the organization.
- **Google's Secure AI Framework (SAIF)** for extending proven security foundations, like identity, detection, and automation, to AI systems.
- **The OWASP Top 10 for LLM Applications** and OWASP's agentic AI guidance for the application-level threats specific to models and agents.

Anchoring to these frameworks matters for a practical reason: when an enterprise security team reviews your AI program, they want to see controls they recognize. It shortens approvals and keeps the conversation focused on risk instead of terminology.

Seven layers, assessed the same way every time

At Metrik Connect, we organize AI security work into seven layers. Every client gets assessed against all seven, so nothing falls through the cracks between teams:

- 1 **Identity:** human and non-human identities, agent credentials, and least-privilege access for every model and tool call.
- 2 **AI apps and agents:** prompt injection, insecure tool use, and output handling.
- 3 **Data:** classification, access boundaries, and leakage controls for training data, retrieval sources, and model outputs.

- 4 **Cloud:** secure landing zones, workload identity, and posture management for the infrastructure AI runs on.
- 5 **Endpoint and network:** device trust, segmentation, and egress controls around AI workloads and the people using them.
- 6 **Detection and response:** logging and alerting for AI-specific threats, plus playbooks your team can actually run.
- 7 **Governance:** policies, risk registers, and accountability that will hold up to an audit.

Run security as a parallel track, not a final phase

The most important change is timing. Instead of a security review at the end, we run security as its own workstream inside the same timeline as the AI and cloud build. On a typical enterprise engagement, that looks like eight weeks:

- **Discovery (weeks 1–2):** inventory models, agents, data flows, and identities, and baseline risk across all seven layers.
- **Design (weeks 2–3):** agree on the threat model, control objectives, and measurable success criteria.
- **Architecture (weeks 3–4):** define the target security architecture and a prioritized roadmap engineers can build against.
- **Implementation (weeks 4–8):** build the identity and agent controls, and specify and review the cloud-native controls alongside the platform team.

The practical effect is that identity and access decisions get made during the cloud foundation work, before the first agent workload lands, not after. Engineers build on secure defaults instead of retrofitting them.

Identity is the control plane for AI agents

If you only take one thing from this article, make it this: treat every agent like a first-class identity. In practice, that means:

- Every agent gets its own identity, never a shared account.
- Credentials are scoped to the task and short-lived wherever the platform supports it.
- Every agent has a named human owner who is accountable for what it can do.
- Every action is logged with enough context to trace it back to the agent and the request that triggered it.
- Agents are included in the same access reviews as people, and lose access when they're retired.

This is the same discipline mature identity programs already apply to employees and contractors. The difference is that agents multiply faster, change more often, and rarely file a ticket when they need more access.

It scales down, too

None of this is reserved for large enterprises. For smaller organizations, we combine phases and focus on the layers that matter most for their environment. A 50-person company building its first customer-facing agent needs the same principles as a global manufacturer; it just needs fewer weeks to apply them. Many teams start with discovery through architecture only, leave with a full security architecture and roadmap, and treat implementation as a follow-on.

Five questions to ask about your AI program this week

- 1 Do we have an inventory of every model and agent running in production?
- 2 Does each agent have its own identity and a named owner?
- 3 Are agent credentials scoped and short-lived, or shared and permanent?
- 4 Could we trace a specific action back to the agent and the request that caused it?
- 5 Who approves a new tool or data source before an agent can use it?

If any of those answers is "not sure," that's the right place to start, ideally before the next agent goes live rather than after.

Metrik Connect helps organizations design security into their AI programs from day one through our Secure AI Foundation. If you'd like to talk through where your program stands, reach out at contact@metrikconnect.com.

SECURE AI FOUNDATION

Building with AI agents? Let's design the security in from week one.

contact@metrikconnect.com · metrikconnect.com