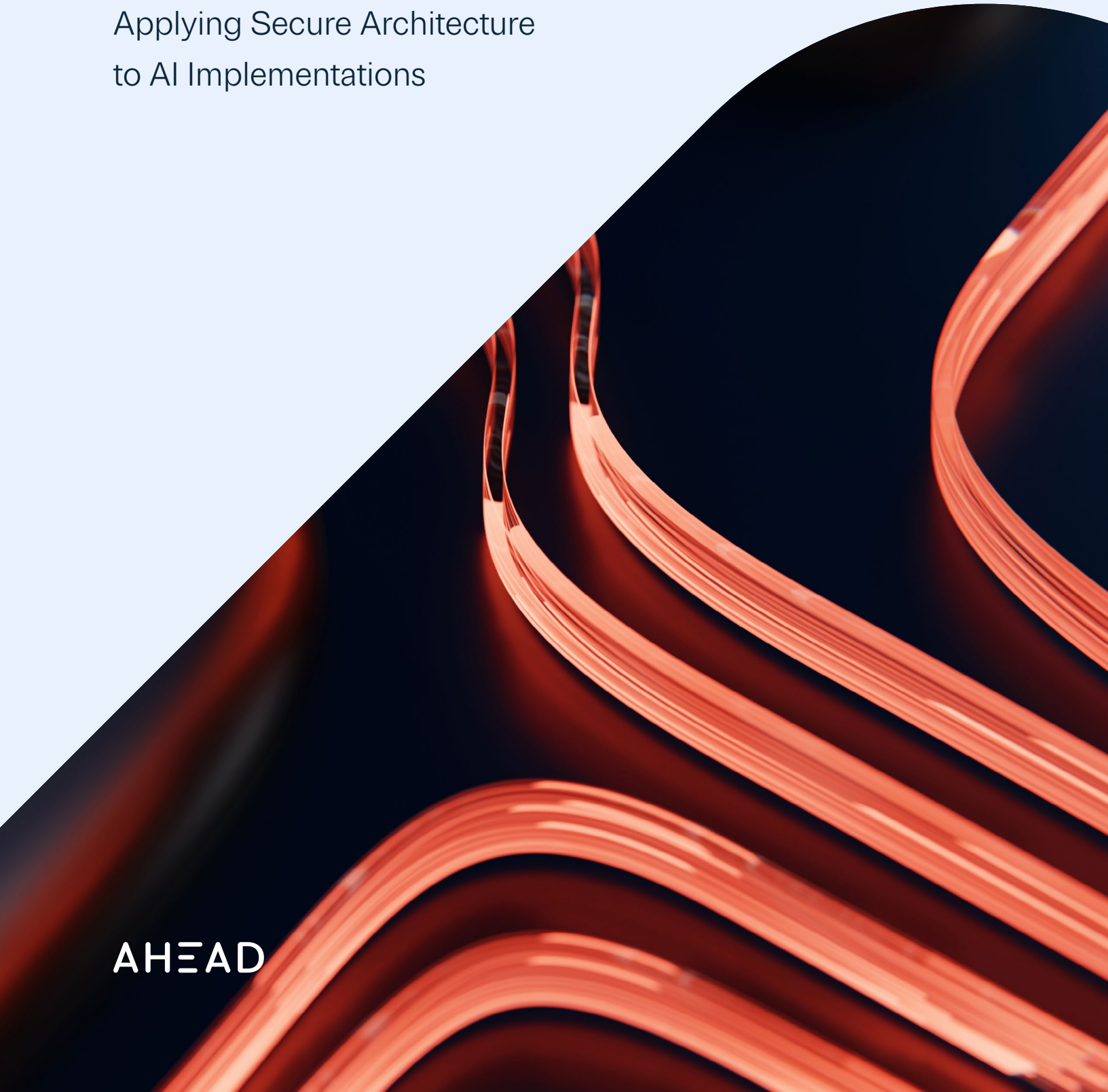
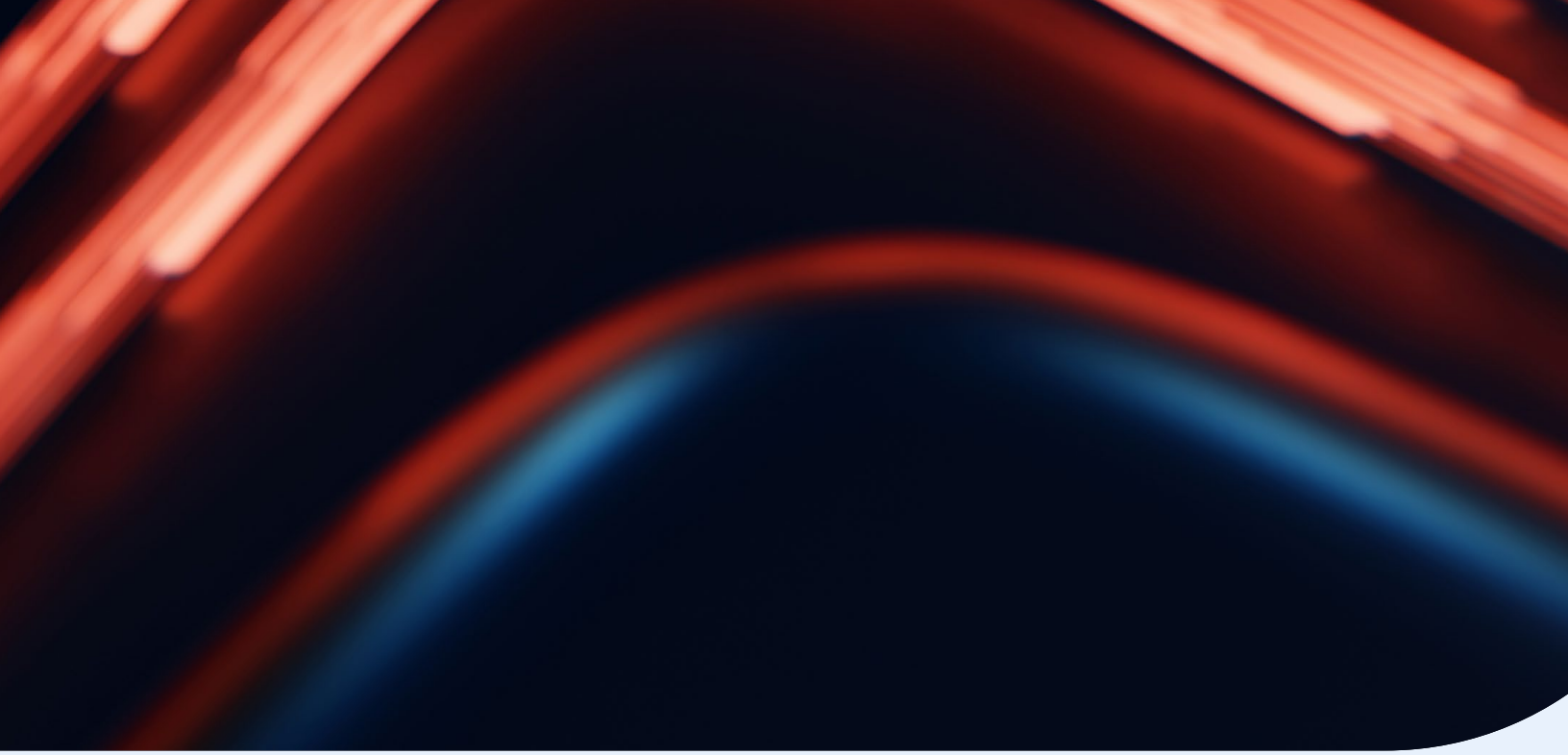
Keeping the Wolves at Bay

Applying Secure Architecture to AI Implementations



AHEAD



Executive Summary

Time is limited, and senior leaders are applying pressure to keep things moving quickly. Complete the risk assessment, approve the AI application, transform the organization with a snap of the fingers. The pressure on security teams is understandable. Meanwhile, artificial intelligence is moving quickly from experimentation to execution, and humans are a creative lot. They will build new models, copilots, agents, and retrieval pipelines faster than many organizations can fully govern them.

That speed creates opportunity, but it also creates exposure.

This is where many security teams lose the thread. Suddenly Zero Trust seems like a good idea, but they start talking about Zero Trust as though it is the destination. It isn't. Zero Trust should be a beacon, but not the main subject. The goal is not to "do" Zero Trust. The goal is to reduce risk to tolerable levels and apply secure architecture in ways that protect data, users, systems, and business outcomes.

That distinction matters even more with AI, because organizations now must protect the AI systems they deploy and protect the enterprise from the AI they just put into production.

If organizations approach AI security as a branding exercise, a control catalog, or a race to adopt the newest tooling, they will miss the point. AI implementations need thoughtful architecture, clear decision rights, strong identity controls, protected data flows, runtime visibility, and response mechanisms that work at machine speed. They also need controls designed to protect the enterprise from the AI system that was just implemented.

That is where security architecture, informed by Zero Trust principles, becomes effective. It provides a practical way to constrain behavior, verify access, limit blast radius, and reduce risk while still moving forward.

The real objective: secure architecture that reduces risk

AI systems are not just another application tier. They often pull together sensitive data, privileged APIs, cloud services, third-party models, embedded agents, human prompts, and downstream automation. That creates new trust boundaries and amplifies existing weaknesses.

A secure AI architecture should do a few things well:

- Verify who or what is interacting with the model, data, and tools
- Limit unnecessary access and constrain privilege
- Protect data before, during, and after model interaction
- Separate components so one failure does not become full compromise
- Monitor for misuse, drift, abuse, and anomalous access
- Enable fast response when a model, plugin, prompt path, or data source starts behaving badly

That's why Zero Trust remains useful here. Not because it gives us a slogan, but because it gives us architectural discipline. It helps teams think in terms of explicit trust decisions, narrow access paths, continuous verification, segmentation, and measurable control effectiveness. Those are exactly the habits AI programs need.

AI changes the attack surface, but not the fundamentals

The wolves are different now, but they still behave like wolves.

Some come after identity. If an attacker can hijack a privileged AI admin, service account, API token, or agent framework credential, they may not need to hack the model at all. Some come after data, where sensitive prompts, embeddings, vector stores, training datasets, output logs, and model responses become leakage paths. Some come after control flow. If an AI system can call tools, trigger actions, or broker decisions across systems, then prompt injection, poisoned retrieval, insecure plugins, and over-permissioned orchestration become architectural problems, not just application bugs. Others come after visibility. If organizations cannot see what prompts were submitted, what data was retrieved, what model was used, what tool calls were made, and what actions followed, they are operating blind.

The defense fundamentals, however, have not changed.

Good architecture still begins with understanding assets, access, dependencies, and risk. It still requires governance, business alignment, and clear accountability. It still depends on security teams being brilliant at the basics while adapting controls to new realities. That is not unusual. Zero Trust guidance has long made clear that tools alone are not enough. Governance, risk management, and program discipline have to mature alongside the technical controls if those controls are going to be effective.

Mapping CISA Zero Trust capabilities to AI architecture

The table below is intended to keep the discussion practical. It maps CISA's five Zero Trust pillars and three cross-cutting capabilities to common AI architecture concerns, and adds a high-level NIST CSF 2.0 overlay. While not a perfect compliance crosswalk, it is meant to show where architectural attention belongs, especially when enterprises need protection not only for the AI, but also from the AI once it is in production.

CISA ZERO TRUST CAPABILITY	AI CONCERN	SECURE ARCHITECTURE MOVES	NIST CSF 2.0 OVERLAY
Identity	Humans, services, agents, pipelines, and model platforms all need explicit identity	Strong authentication for users and admins, workload identity for services, short-lived credentials, token scoping, approval for privileged model changes	Govern, Protect, Detect
Devices	Unmanaged endpoints, developer workstations, inference clients, and edge systems can introduce risk	Device posture checks, conditional access, hardened admin workstations, controlled developer environments, separation of test and production endpoints	Identify, Protect
Networks	AI workloads often span cloud, SaaS, APIs, data platforms, and external model providers	Segment inference paths, isolate model hosting zones, restrict east-west traffic, broker outbound model and plugin communications, inspect egress where appropriate	Protect, Detect, Respond
Applications and Workloads	Models, agents, retrieval layers, prompt services, orchestration engines, and tool connectors are all workload components	Secure software supply chain, signed artifacts, isolated runtimes, least-privilege service design, approval gates for model and prompt changes, strong API mediation	Govern, Protect, Detect
Data	AI is only as safe as the data it can reach, retain, transform, or expose	Data classification, retrieval boundaries, encryption, DLP, redaction, prompt and output filtering, retention limits, segregation of sensitive vector stores and logs	Govern, Identify, Protect, Detect

CISA ZERO TRUST CAPABILITY	AI CONCERN	SECURE ARCHITECTURE MOVES	NIST CSF 2.0 OVERLAY
Visibility and Analytics	Without telemetry, AI misuse can look like normal usage until damage is done	Log prompts, retrieval events, tool calls, model routing, admin changes, data access, and abnormal response patterns; correlate AI events with IAM, endpoint, and cloud telemetry	Identify, Detect, Respond
Automation and Orchestration	AI operates at machine speed, so guardrails and response must keep pace	Automate policy enforcement, block risky tool calls, trigger containment on abnormal behavior, disable compromised connectors, rotate secrets, require human approval for high-risk actions	Protect, Detect, Respond, Recover
Governance	AI security fails when ownership, policy, and risk decisions are fuzzy	Define acceptable use, model approval criteria, data handling requirements, third-party review standards, exception processes, metrics, and accountable business owners	Govern, Identify, Recover

What this looks like in practice

A secure AI architecture does not start with the model. It starts with the use case. Before selecting platforms and controls, teams should ask a few basic questions:

- What data will the AI system access?
- What decisions will it influence or automate?
- What identities will administer, operate, and consume it?
- What external services, plugins, tools, or APIs will it call?
- What could go wrong if outputs are wrong, manipulated, leaked, or over-trusted?
- What evidence will prove the controls are working?

Those questions quickly reveal whether an AI implementation is low-risk experimentation or a high-consequence business system. They also help determine what level of architectural rigor is warranted. A summarization assistant using approved public content is not the same as an agent that can retrieve customer data, write code, open tickets, modify configurations, or trigger financial workflows. One may need sensible guardrails. The other may need strong segmentation, workflow approval, transaction limits, layered telemetry, and active monitoring from day one.

This is the point many organizations miss: what looks like an AI problem is often an identity problem, a data path problem, an orchestration problem, or a control-plane problem. If those layers are weak, the model does not have to fail for the business to be exposed.

Practical design principles for keeping the wolves at bay

Organizations do not need a magic product to secure AI, but they do need disciplined design choices.

1 Treat AI identities like privileged identities

Model administrators, orchestration services, retrieval pipelines, connectors, and agent frameworks should not operate with broad, persistent access. Apply workload identity, scoped credentials, and strict role separation. If an AI component can take action, it should do so with the smallest possible set of permissions.

2 Control the data path, not just the model

Many AI failures are really data failures. Sensitive data reaches the wrong prompt, the wrong retrieval source, the wrong user, or the wrong log store. Protecting the model without governing the data path is like locking the side gate and leaving the front door open.

3 Segment AI components deliberately

Do not let model hosting, retrieval systems, orchestration engines, tool connectors, and administrative planes collapse into one big, trusted zone. Separate duties and communication paths. When something is compromised, blast radius matters.

4 Put policy at the decision points

Policy should exist where access, retrieval, tool execution, and output handling decisions are made. This includes identity-aware gateways, API mediation layers, prompt and content controls, and approval checkpoints for risky actions.

5 Log enough to investigate reality

If a prompt triggers an action, teams should know who initiated it, what data was retrieved, what context was supplied, what model was used, what the output was, what tool calls followed, and whether policy exceptions occurred. If you cannot reconstruct the event, you do not really control the system.

6 Design for misuse, not just intended use

Assume prompts will be manipulated. Assume users will over-trust outputs. Assume connectors will be over-permissioned. Assume context windows will carry things they should not. Good architecture does not depend on everyone behaving perfectly.

7 Govern the lifecycle, not just production access

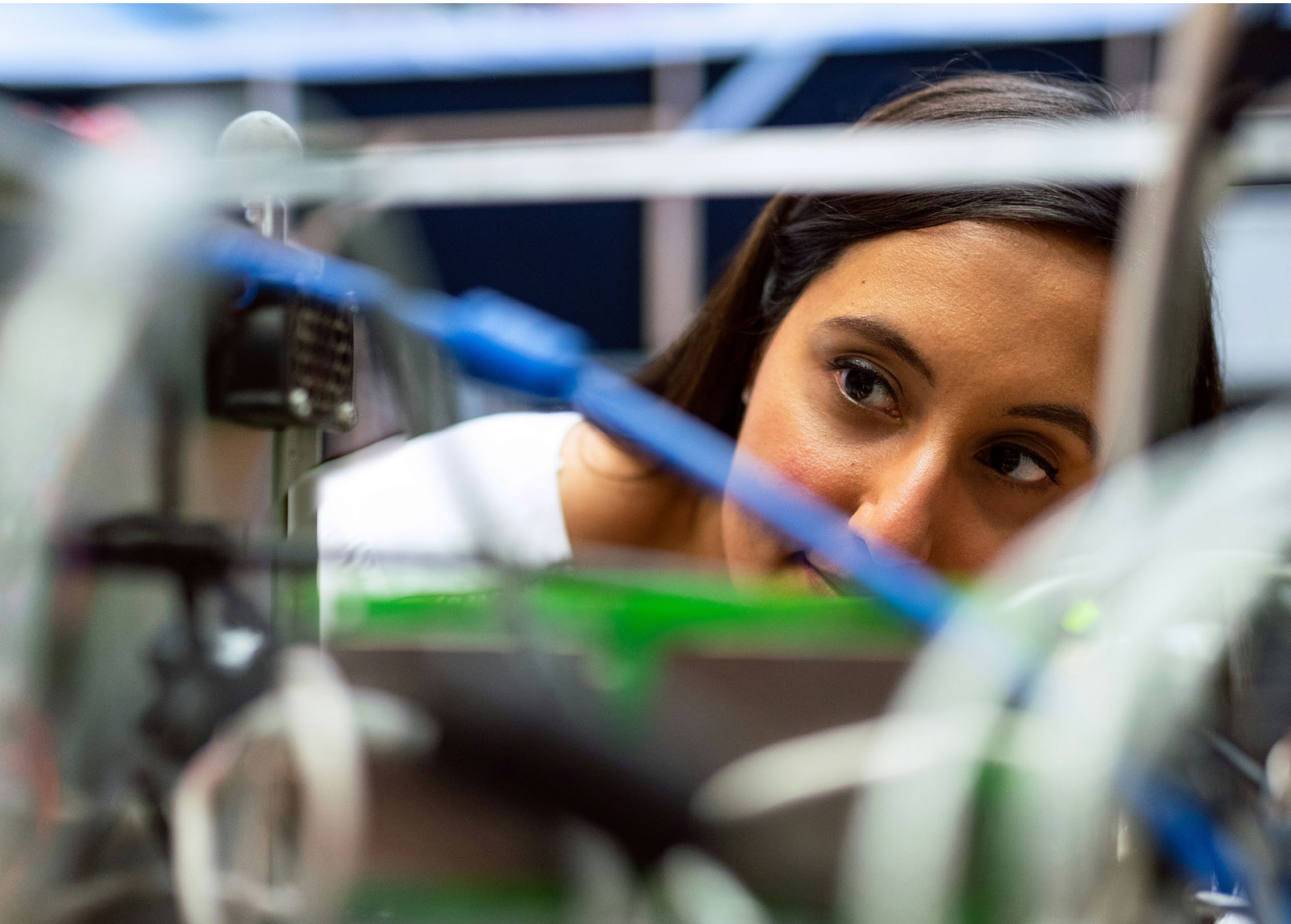
Security does not begin at inference time. It starts with model selection, training data hygiene, software supply chain review, prompt template control, testing, change management, and third-party assessment. Runtime controls matter, but so does the path that got the system there.

Where teams commonly go wrong

A few mistakes show up repeatedly:

- Buying AI tools before understanding business risk and data exposure
- Treating Zero Trust as a product category instead of an architectural method
- Over-focusing on the model while under-securing the orchestration and data layers
- Giving agents or connectors broad standing access for convenience
- Failing to log prompts, retrieval events, and downstream actions
- Assuming existing IAM, network, or DLP controls automatically cover AI use cases
- Measuring activity instead of measuring risk reduction

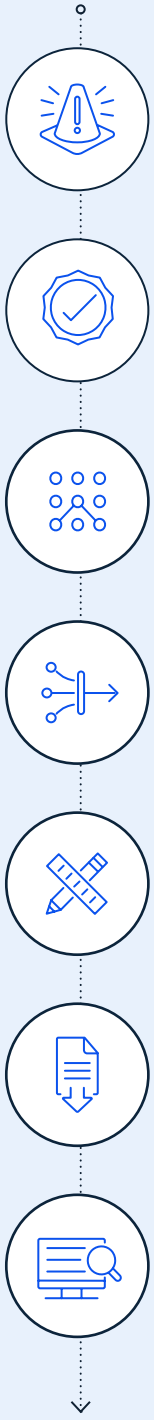
This is where governance and program management matter. AI security is not just a technical exercise. Just as with Zero Trust, implementing and securing AI is a cross-functional discipline involving architecture, legal, privacy, security, operations, engineering, and the business. If ownership is weak, the architecture will usually reflect it.



A practical path forward

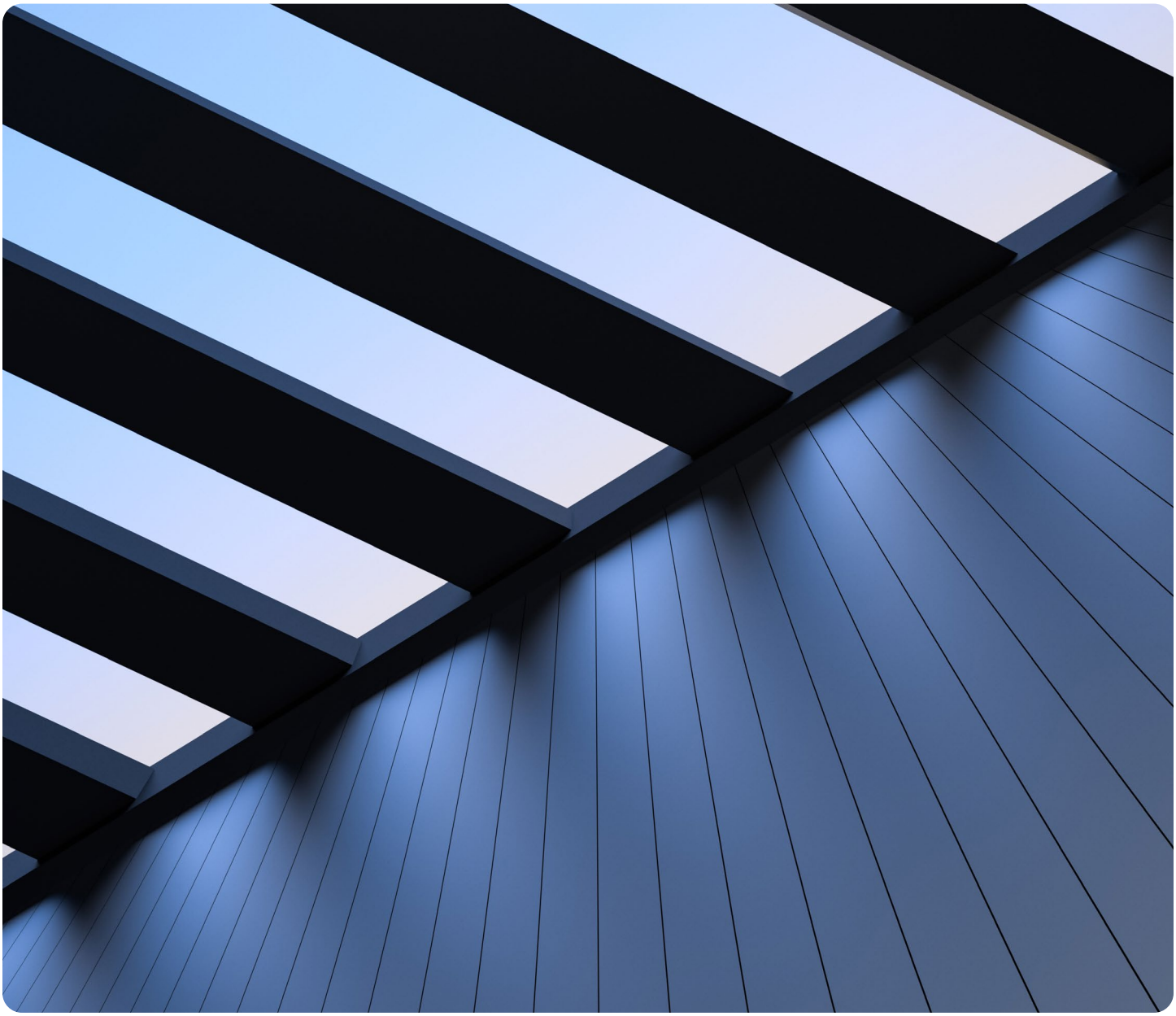
Organizations do not need to solve every AI security challenge at once. They do need to move deliberately.

A practical path often looks like this:



- Classify AI use cases by business impact and risk
- Define approved architectures for low, medium, and high-risk implementations
- Establish identity, data, logging, and segmentation minimums for each pattern
- Integrate identity, data, network, telemetry, and response controls into existing SOC, IAM, cloud, and data protection workflows
- Require architectural review for tool-calling, agentic, and sensitive-data use cases
- Create governance processes for model onboarding, exceptions, third-party review, and change control
- Measure whether risk is actually reduced

That last point matters most. The goal is not to check a Zero Trust box. The goal is to know whether the architecture is making the environment safer, more observable, more controllable, and more resilient.



Final Thoughts

AI will continue to change the way enterprises build, operate, and compete. That part is not in question. The real question is whether organizations will wrap AI in secure architecture early enough to matter.

Zero Trust can help, but only if we treat it correctly. It is not a brand, a product category, or a finish line. It is a design discipline for making explicit trust decisions, narrowing access, segmenting systems, verifying identity, and measuring whether controls actually work. AI doesn't change that mission. It just raises the stakes.

The organizations that succeed will not be the ones that deploy AI the fastest. They will be the ones that build secure architecture early, and build architecture that protects the AI system and protects the enterprise from the AI system once it is in production. If we do that well, the wolves may still circle, but they will find far fewer doors open.

AHEAD

Combining cloud-native capabilities in software and data engineering with an unparalleled track record of modernizing infrastructure, we're uniquely positioned to help accelerate the promise of digital transformation.

Visit us at [ahead.com.](ahead.com)

National Hubs

CHICAGO
444 W. Lake Street
Suite 3000
Chicago, IL 60606

NEW YORK
500 5th Avenue
Floor 17
New York, NY 10010

ATLANTA
1117 Perimeter Center
W406
Atlanta, GA 30338

SAN FRANCISCO
2000 Crow Canyon Place
Suite 250
San Ramon, CA 94583