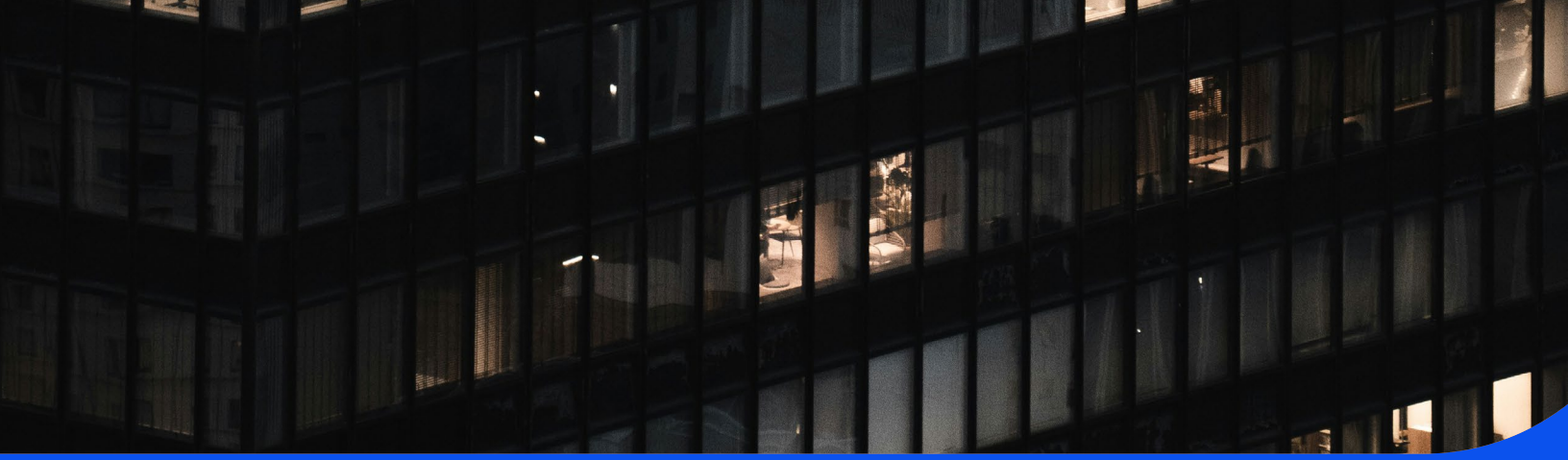
From CNAPP to AI Application Security

AHEAD



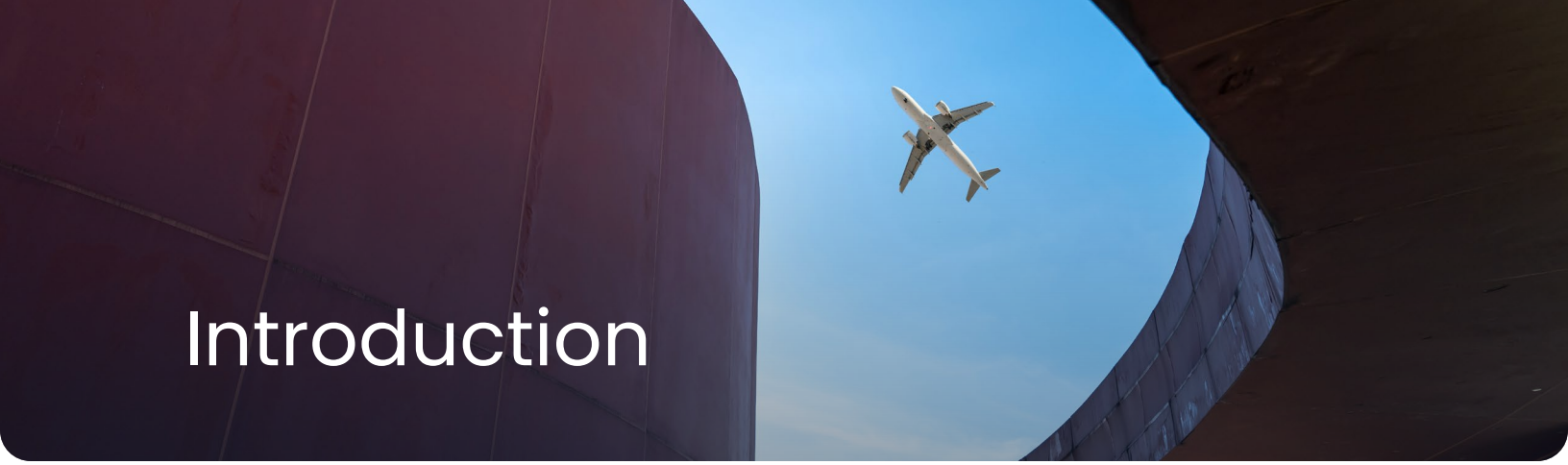
Executive Summary

Several merging forces are reshaping how enterprises need to think about AI security:

- AI adoption is moving quickly from pilots to production use cases embedded inside business applications, employee workflows, customer experiences, and automated decision paths.
- The attack surface is broadening beyond infrastructure misconfiguration into data poisoning, prompt misuse, retrieval abuse, model supply chain risk, excessive agent permissions, insecure API exposure, and weak runtime controls.
- Traditional Application Security and Cloud Security disciplines remain relevant, but they are not sufficient on their own because AI systems are probabilistic, data-dependent, and increasingly autonomous in how they invoke tools and act on context.
- Governance pressure is increasing at the same time that technical complexity is rising. Organizations now need to align security with AI governance, privacy, compliance, model risk, and operational resilience.

Visibility alone is not enough. The organizations that create lasting value are the ones that turn AI security into an operating capability with clear ownership, repeatable workflows, and controls that prevent the same risks from recurring.

The real-world implication is straightforward: AI application security should be treated as the next maturity layer after CNAPP, not as a separate side project. Successful programs will extend proven cloud security disciplines into AI-specific controls for data, identities, models, agents, and runtime behavior, but controls and best practices alone are not the same as an operating model. The operating model is the organization-specific way those disciplines are assigned, governed, and executed day to day.



Introduction

Cloud Security has always evolved in response to how applications are built, deployed, and operated. When Gartner introduced the concept of Cloud-Native Application Protection Platforms, or CNAPP, the category gave organizations a more unified way to understand cloud risk across infrastructure, identities, workloads, data paths, and software supply chains. It was a needed shift. Cloud environments have become too dynamic, too interconnected, and too fast-moving for security teams to manage through isolated point products alone. That pattern is now playing out while organizations bring AI into production.

As enterprises move from experimentation to production AI workloads, the challenge is no longer limited to securing cloud infrastructure around a model. Organizations now must secure the full AI application stack: training data, prompts, model supply chains, retrieval pipelines, non-human identities, APIs, agentic workflows, runtime policies, and the operational systems that monitor and respond when something drifts or fails.

Just as importantly, AI workloads inherit the compute, networking, identity, and data dependencies of the cloud environments underneath them. Weak cloud foundations around data classification, access governance, encryption, and visibility carry directly into the AI layer, where they usually widen.

In other words, AI security builds on cloud security and carries it into an application layer with its own failure conditions, attack paths, and very little room for error.

This whitepaper explores why AI application security has become the next frontier in the evolution of CNAPP, the patterns organizations should expect as they operationalize AI, and the practices that help translate scattered controls into a durable security capability. It also outlines how AHEAD helps clients move from early AI experimentation to secure, scalable execution.

In short: organizations that once discovered that cloud security had to be addressed as an operating discipline are finding the same is true of AI. Securing AI demands coordinated decisions across architecture, governance, runtime, and the operating model that ties them together.

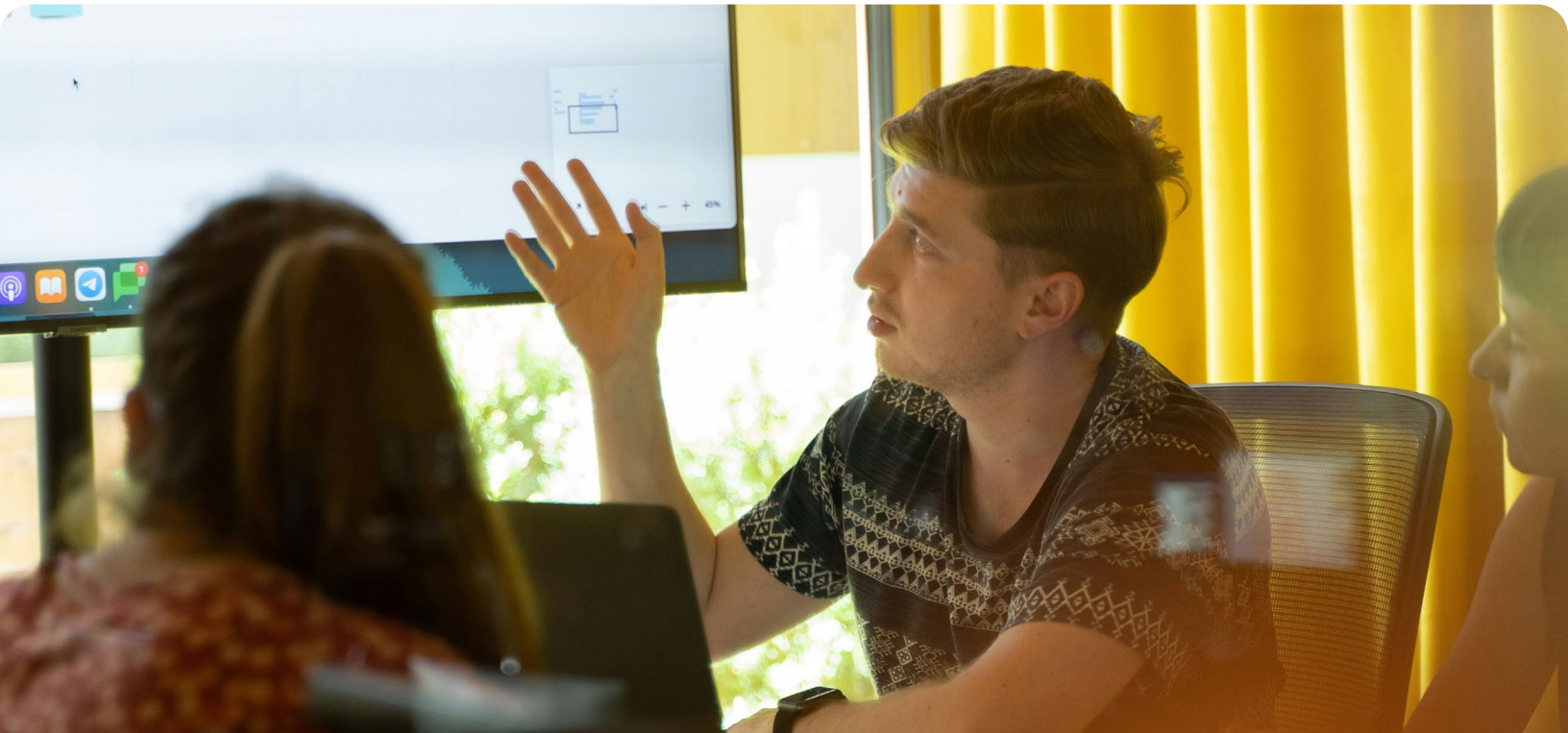
In this context, frameworks, maturity models, and best practices help define what good looks like. The operating model defines how the organization actually makes that real through roles, decision rights, workflows, tooling, and enforcement.

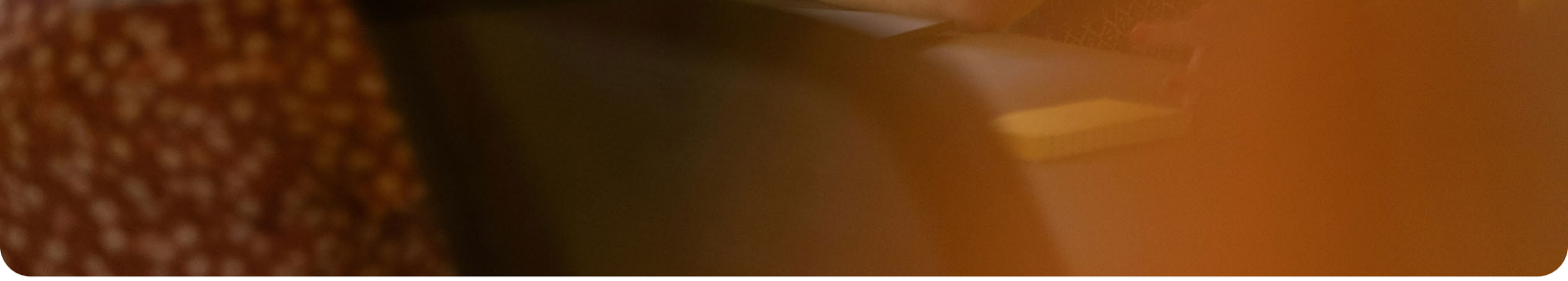
Why This Matters at the Initiative Level

Organizations that place AI adoption as one aspect of a broader modernization initiative are better-positioned to secure their AI deployments than organizations that view AI security as an isolated control stack. There are several reasons for this:

1. **AI transformation demands secure patterns for data, agents, runtimes, orchestration, and oversight.** Organizations that want to move from scattered AI pilots to governed, repeatable, enterprise-scale AI capabilities must be able to employ these patterns regardless of whether the applications use AI or not.
2. **AI applications require secure and resilient architecture.** As AI becomes embedded into business workflows and customer-facing applications, security has to be part of the design itself, woven into the platform, identity model, delivery pipelines, and operational response.
3. **Safe use of AI requires trusted data enablement.** AI applications only become trustworthy when the data behind them is governed, discoverable, access-controlled, and resilient enough to support production operations.

Organizations moving fastest tend to manage these priorities together, which is why AI application security quickly becomes a business concern as opposed to a topic owned solely by the security team.





The Next Inflection Point

CNAPP emerged because cloud risk doesn't neatly fit into the traditional silos of on-prem architectures. Identity, posture, vulnerability management, network reachability, data access, and infrastructure drift all intersect in the cloud, and this demands a more connected, unified view.

AI is creating that same kind of inflection point, in which practicing security requires correlating signals from many contexts to tell a risk story. In many organizations, this is following the same maturity curve cloud security followed. When teams never fully established a cloud operating model, they often discover that they also lack an AI operating model.

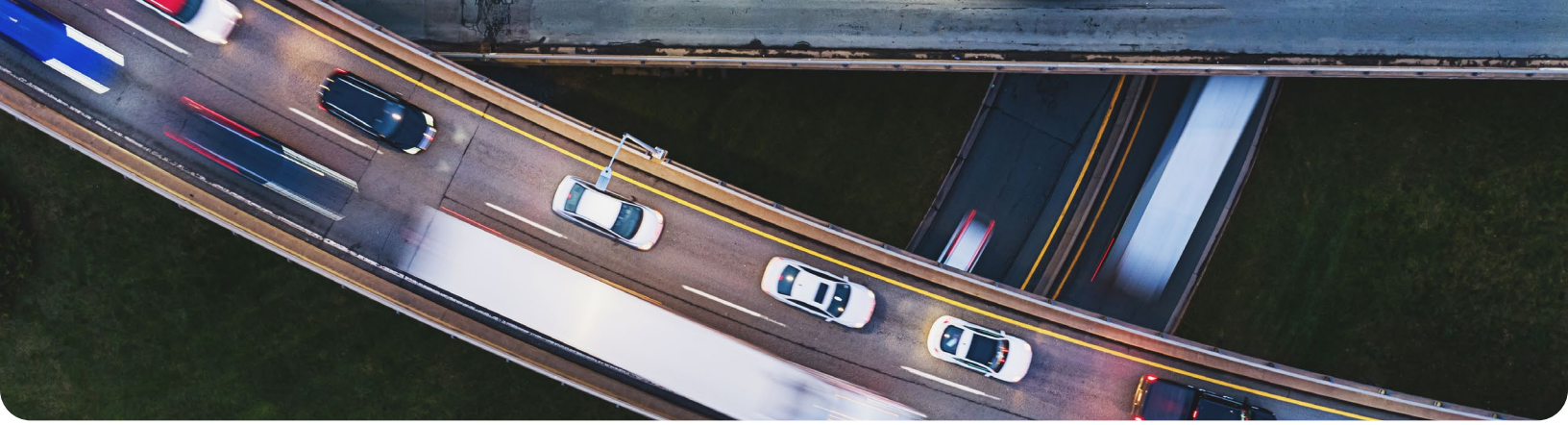
That gap rarely stems from a lack of guidance. More often than not, organizations have principles, controls, or reference architectures in place, but have not yet decided who owns key decisions, how teams run reviews, how they handle exceptions, or how they enforce controls in practice.

A modern AI application is a composite system that may include one or more models, a prompt framework, retrieval layers, orchestration logic, external tools, API integrations, data sources, agent memory, identity tokens, observability pipelines, and the cloud services beneath them. Even when each piece is secured on its own, exposure often shows up in the handoffs and dependencies between them.

As such, many organizations are starting to discover a new deployment trap:

First, a team might quickly stand up an AI assistant, a retrieval workflow, or an agentic automation. The team wants to move fast. The organization's security policies and risk management efforts will apply reasonable controls around the edges, such as turning on app logging and monitor access, introducing a provider for model review and approval, or adding a few guardrails.

Then, the environment starts to change. More teams begin experimenting, users introduce new tools, datasets, and prompts, and non-human identities gradually pick up additional permissions. Before long, the system has become a living application surface rather than a single AI feature. This is usually the point at which AI development has moved beyond the reach of traditional security operations.



Why Traditional Controls Are Necessary but Incomplete

To be clear, cloud and application security operations are still important to secure AI deployments. For example, secure IAM, network segmentation, secrets management, vulnerability management, SDLC governance, and machine data analysis remain essential; however, AI deployments introduce conditions that make those controls incomplete if they are not extended.

1. **AI systems are highly dependent on data quality, provenance, and access patterns.** An application can be fully hardened from a code and infrastructure perspective and still produce unsafe outcomes because it is grounded on overexposed, poisoned, low-integrity, or poorly governed data.
2. **AI systems are often non-deterministic.** The same input does not always produce the same output, and the same model may behave differently as prompts, context windows, tools, or orchestration logic evolve. That makes testing and assurance more difficult than in deterministic software.
3. **AI applications increasingly act through delegated authority.** Agents search, summarize, make recommendations, call tools, trigger workflows, and in some cases, execute commands. This shifts security for simple user access to a broader set of questions: what can this AI-enabled system do, under what conditions, with those permissions, and with what oversight?
4. **AI applications create new pathways for abuse that do not map cleanly to traditional software categories.** Prompt injection, context poisoning, insecure output handling, data leakage through grounded responses, unsafe tool invocation, and agent-to-agent trust abuse all require controls that sit closer to the AI runtime itself.

The next step after CNAPP is a broader AI-secure architecture that carries core cloud security disciplines into the full AI lifecycle.

The Patterns You Should Expect

So far, we've discussed the security challenges businesses are experiencing with AI adoption: business requirements, innovation practices, and the limits of thinking of security as isolated control stacks. What we see is that organizations do not typically fail at AI application security because they ignore security completely; more often, they struggle because their AI usage outpaces their application security model.

These patterns tend to show up with surprising consistency:

Shadow AI Becomes Architecture Before Anyone Calls It Architecture

Many AI risks start as decisions that are convenient. A team experiments with a public model, a developer connects a lightweight retrieval layer to an internal data source, a business unit adopts a vendor-hosted copilot, or an automation team gives an agent access to downstream tools. None of those decisions seems especially consequential on their own, yet together they create an unofficial architecture with real security and business implications. In practice, the issue also extends into cost governance and the operating model, since unapproved AI use can drain budget, expand the tool footprint, and establish habits before anyone has formally signed off.

Identity & Privilege Sprawl Expands Quietly

AI systems introduce more than human users do. Service principals, API keys, agent identities, connectors, plugins, and delegated permissions all expand the trust boundary. If these identities are not tightly scoped, reviewed, monitored, and secured, an AI application can inherit far more authority and risk than its designers intended. This is also where the familiar idea that "identity is the new perimeter" starts to break down. With non-human agents, security teams must evaluate not only who or what has access, but how that agent behaves once it has it. A legitimate identity can still act anomalously, exceed intended bounds, or produce unsafe actions.

Data Grounding Outpaces Data Governance

Retrieval-augmented generation and enterprise search integrations create business value quickly, but they also expose an old problem in a new way. Organizations often do not fully understand the sensitivity, ownership, quality, or access boundaries of the data they are grounding into AI experiences. AI can turn latent oversharing into immediate oversharing.

Supply Chain Risk Becomes Multi-layered

In traditional software, supply chain discussions often center on libraries, packages, and build pipelines. In AI, the supply chain also includes pre-trained models, embeddings, datasets, prompt templates, third-party tools, evaluation artifacts, and orchestration frameworks. The result is a wider trust chain and more places for integrity issues to enter.

Runtime Controls Arrive Late

Many organizations spend significant energy selecting models and platforms, then treat runtime enforcement as a secondary concern. But AI applications need active controls in production, such as prompt protections, rate controls, content handling rules, output filtering, model behavior monitoring, tool-use restrictions, and session-aware policy enforcement. This is also where many teams discover a visibility gap. Agentless approaches can provide fast, broad visibility, but they often capture only periodic snapshots. Real-time understanding of behavior, drift, and misuse typically requires deeper runtime telemetry through sensors or other workload-level instrumentation.

Monitoring is Fragmented Across Too Many Teams

Cloud teams hold the infrastructure telemetry, security teams the SIEM coverage, data teams the model-quality signals, and platforms teams the API metrics. Without a way to connect those views, early warning signs slip past, and AI incidents rarely stay contained within a single domain.

The Shift from AI Tooling to an AI Security Operating Model

Due to the various contexts we've discussed, a business will need to mature their operating model to secure AI for production workloads by moving from isolated controls to a lifecycle.

To be clear, an operating model is not a framework or a list of recommended controls. A framework describes the capabilities and practices the organization should adopt; the operating model defines how teams execute those capabilities across people, processes, and platforms.

In many enterprises, the most practical place to begin is with the intake and review disciplines already used for cloud adoption, extending them into AI use cases, agentic workflows, and model-enabled business processes. Requests, reviews, security engagement, escalation, and controlled change still matter; teams simply have to apply them to probabilistic systems and delegated actions.

Teams need to define who approves use cases, who owns runtime oversight, how they review data access and agent permissions, what evidence they require before deployment, how they escalate incidents, and which controls they automate versus review manually.

The strongest operating models pair guardrails with approved lanes and golden paths, making sanctioned use the easiest way to move.

Figure 1 is shown below and illustrates that operating-model flow from discovery through response.

Figure 1: *AI Security Operating Model spanning discovery, prioritization, governance, runtime monitoring, and response*

When teams stay disciplined about this flow, AI security becomes an ongoing method of reducing risk rather than a series of disconnected reviews.

Where Controls Need to Evolve

Security for AI

Security for AI is the work of protecting the AI system itself, spanning governance, secure development, and runtime behavior.

AI GOVERNANCE

AI governance cannot remain a high-level policy document detached from implementation. It needs to define who can approve AI use cases, which risk tiers require deeper review, how data usage is governed, what evidence is required before deployment, and how accountability is maintained after launch.

The most effective governance models are business-aligned rather than purely theoretical. They connect security, privacy, compliance, data governance, and risk management into one decision structure, while still leaving room for teams to move quickly within approved patterns. Governance sets decision rules and accountability, while the operating model carries those decisions into day-to-day execution through workflows, service ownership, tooling, and response processes.

A useful way to communicate this is through a three-layer assurance model:

- External assurance inherited from providers and third parties
- Internal assurance through policy, continuous compliance, and formal review
- Technical implementation through data protection, segmentation, identity control, and workload enforcement

That model also helps surface oversight gaps when affiliates, subsidiaries, or newly absorbed entities sit outside the standard intake path.

SECURE AI SYSTEM DEVELOPMENT LIFECYCLE

The AI SDLC should extend familiar secure development practices while accounting for AI-specific failure modes. That includes validating model and dataset provenance, reviewing prompt and orchestration logic, applying threat modeling to retrieval and tool-use flows, scanning dependencies, testing for misuse scenarios, and defining approval gates before release.

Teams must weave AI security testing throughout the lifecycle so that risky patterns surface before they harden into runtime behavior.

AI RUNTIME SECURITY

Runtime is where many of the most consequential AI failures appear. This is where prompts are manipulated, tools are misused, sensitive data is returned unexpectedly, and agent behavior diverges from design intent.

A practical runtime control set often includes:

- Strong identity and least-privilege enforcement for users, agents, connectors, and service principals
- Prompt and input protections to reduce injection and abuse
- Output handling controls to limit unsafe, unapproved, or overly sensitive responses
- Guardrails around tool invocation, code execution, and workflow automation
- Telemetry that captures meaningful events without creating unmanageable noise
- Monitoring for drift, anomalous behavior, and policy violations over time

AI for Security

AI for Security is the inverse discipline that uses AI to make the security function itself faster and more precise.

AI-OPTIMIZED SECURITY OPERATIONS

Once AI is in production, organizations need a clear operating model for AI-related incidents that extends beyond generic alerting.

That includes standardized telemetry, AI-specific incident taxonomy, playbooks for prompt injection and data leakage scenarios, workflows for triaging model or agent misuse, and response patterns that keep humans in the loop for higher-risk decisions. In mature environments, it also means blending deterministic and non-deterministic response patterns, allowing agents to contribute triage, summarization, and evidence assembly inside a playbook while humans retain decision authority and remain accountable for the intended outcome and assurance of materially risky actions.

The same mindset applies to exposure management, where AI can help move teams from thousands of findings toward a smaller set of validated, prioritized issues that are actionable. The goal is to evolve security operations so they can recognize and respond to AI-native threats as a matter of routine.

This section shows the operating model at work: not just which controls should exist, but who uses them, how incidents move, where decision authority sits, and how teams apply human oversight at runtime.

A Practical Maturity Path

A useful way to frame AI application security is as a maturity progression. This progression works best as a planning and assessment lens for the capabilities the organization should build overtime. It does not by itself define the operating model.

Foundational

At this stage, organizations have early AI usage, fragmented policies, and limited visibility. Security efforts focus on identifying where AI is being used, establishing basic acceptable-use rules, clarifying data access boundaries, and documenting the major AI platforms and applications in use.

Standardized

The organization starts establishing repeatable controls, whether through formal governance bodies or lighter review mechanisms. Identity patterns, secure API practices, approved architectures, and baseline runtime controls grow more consistent, and AI use cases begin to be tiered by risk.

Operationalized

Security controls become part of the AI SDLC and runtime environment. Monitoring gets stronger, exceptions are tracked, reviews rely more on evidence, and teams start correcting the architectural patterns that create repeated exposure instead of treating every issue as a one-off.

Optimized

At the highest maturity, AI security becomes adaptive and measurable. Policies are codified wherever it makes sense, runtime telemetry supports repeatable detection and response, human oversight is applied deliberately to high-impact actions, and governance, architecture, development, and operations reinforce one another.

This lens is useful because clients do not all need the same answer at the outset. The objective is to move from the current state toward defensible scale in a way that matches the organization's reality. The operating model should be designed to support movement through these stages by defining ownership, workflows, automation, escalation, and control enforcement in a way that fits the organization.

What an AI-Secure Architecture Looks Like

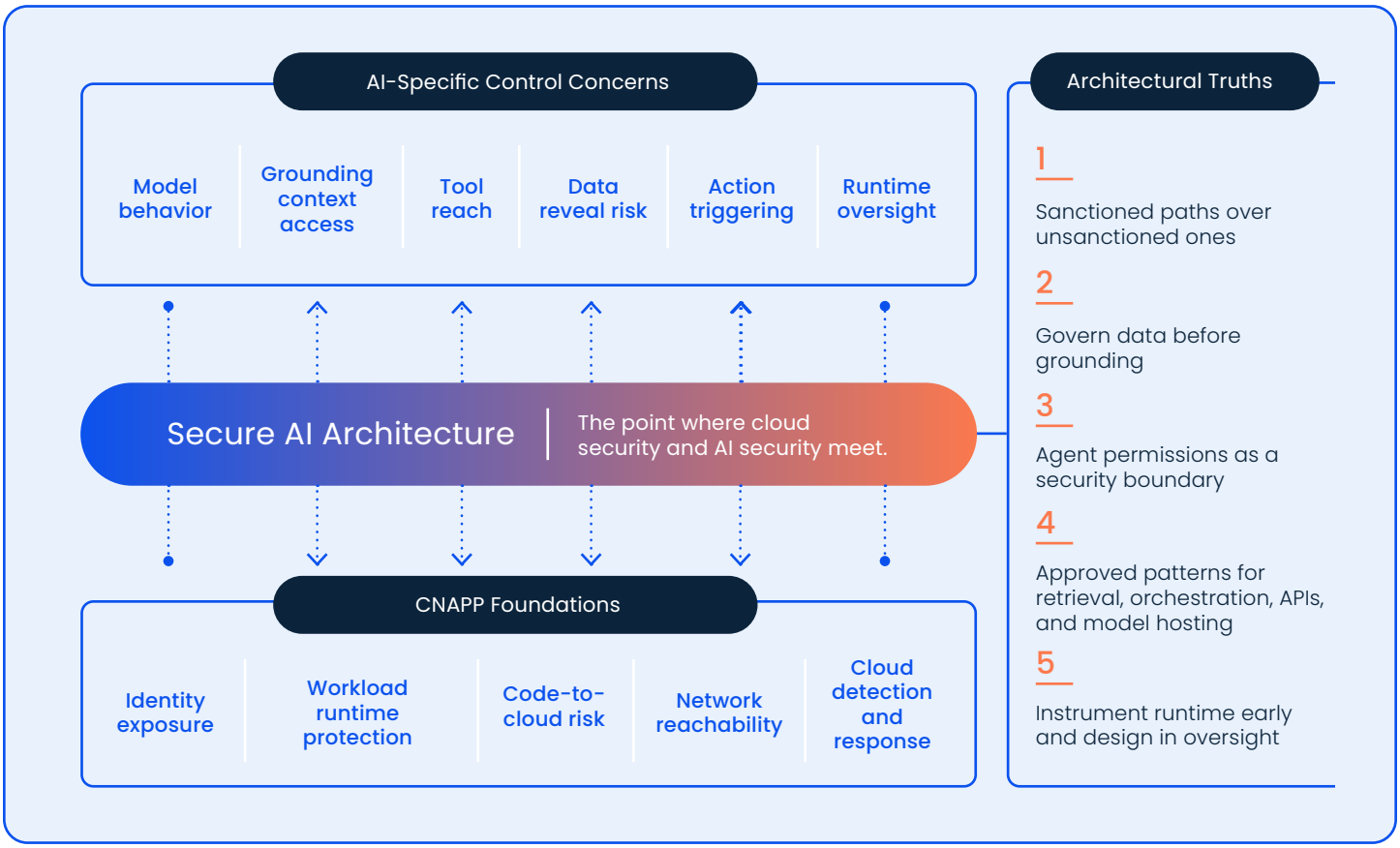


***Figure 2**: AI-Secure Architecture showing CNAPP foundations beneath AI-specific runtime and control-layer concerns*

The most structurally sound enterprise AI security programs tend to converge around a few architectural truths.

They make sanctioned paths easier to follow, govern access to data before it becomes grounding context, treat agent permissions as a first-class security boundary, define approved patterns for retrieval, orchestration, APIs, and model hosting, instrument runtime behaviors early, and build oversight into the process so the organization can show that controls are in place and functioning.

This is also where the relationship between cloud security and AI security becomes clearer. CNAPP disciplines are still necessary, and identity exposure, workload weakness, network reachability, and misconfiguration remain part of the picture. However, they now sit beneath a second set of questions about model behavior, contextual access, tool reach, data disclosure, and triggered actions, with secure AI architecture marking the point where those layers meet.

AHEAD's Perspective

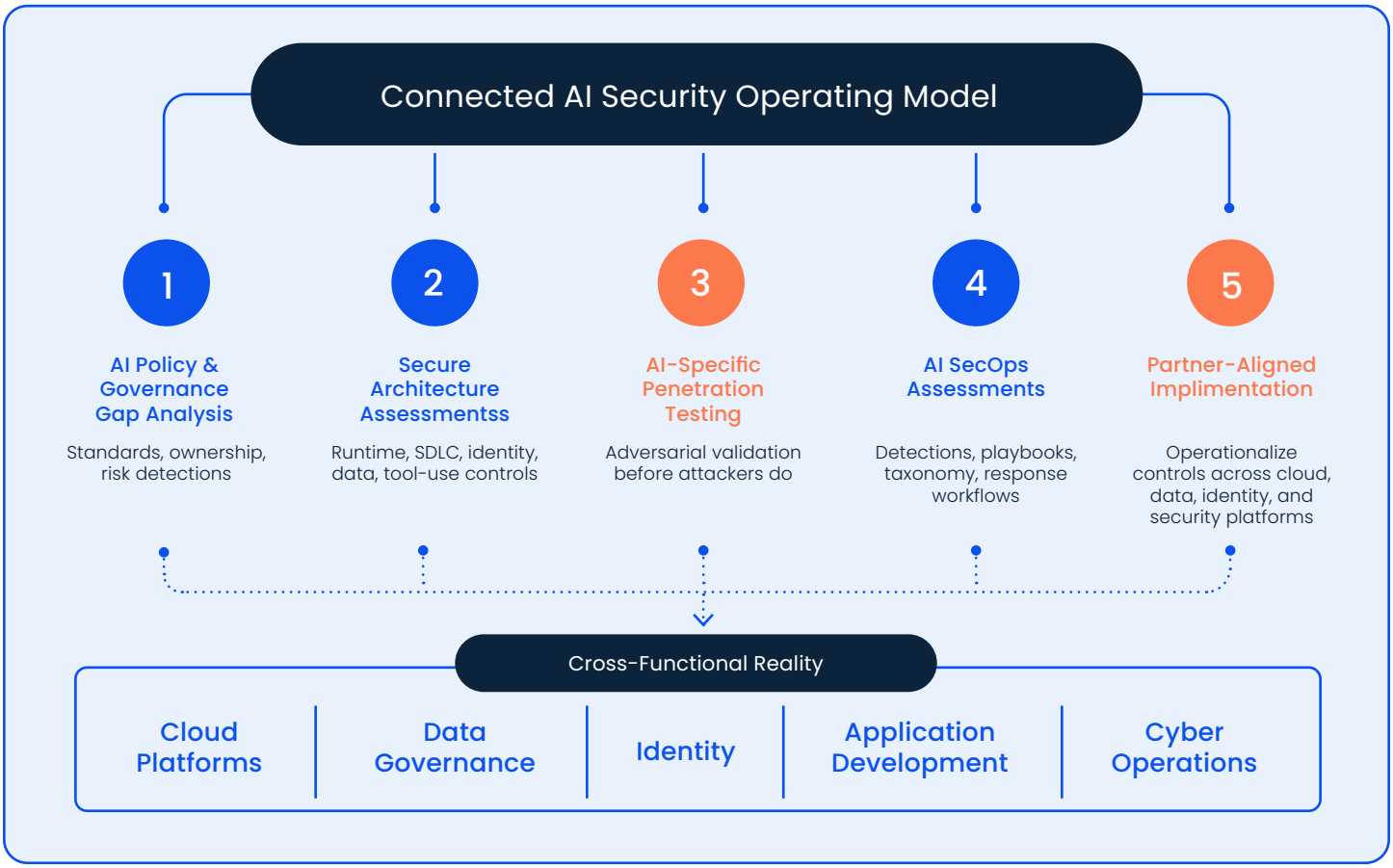


Figure 3: *AHEAD's connected AI security operating model across governance, architecture, testing, SecOps, and implementation*

AHEAD's work across cloud security, AI governance, architecture, data protection, and security operations points to a consistent conclusion: organizations need a practical way to turn AI security into something they can design, implement, operate, and improve.

That need separates framework alignment from operating-model execution. Many organizations can identify relevant controls or reference architectures; fewer can say who owns them, where they fit into delivery processes, how teams monitor them at runtime, and how the organization will sustain them across functions.

That is why AHEAD approaches AI application security as a connected discipline spanning governance, architecture, runtime, and operations.

In practice, that often includes:

- AI policy and governance gap analysis to clarify standards, ownership, and risk decisions

- Secure architecture assessments focused on runtime, SDLC, identity, data, and tool-use controls
- AI-specific penetration testing and adversarial validation to expose weaknesses before attackers do
- AI SecOps assessments to improve detections, playbooks, taxonomy, and response workflows
- Partner-aligned implementation that helps clients operationalize controls across their existing cloud, data, identity, and security platforms

AHEAD's broad experience is particularly valuable here because AI security almost never sits neatly with one team. It cuts across cloud platforms, data governance, identity, application development, and cyber operations, so the approach has to reflect that shared reality.

What Organizations Should Do Next

For many organizations, the first useful step is to build an understanding of where AI is already reshaping the application landscape and where the existing control model has started to fall short.

A pragmatic sequence often looks like this:

1. Identify current AI applications, assistants, agents, and sanctioned or unsanctioned usage patterns
2. Assess how data access, identity delegation, and tool connectivity are currently being governed
3. Evaluate the AI SDLC for gaps in threat modeling, testing, approval, and deployment controls
4. Review runtime protections, telemetry, and incident response readiness
5. Define a target operating model that fits the organization's size, industry, and maturity
6. Prioritize a small number of architectural and operational changes that reduce the most risk first

Final Thoughts

CNAPP helped the market understand that modern cloud risk is cumulative, interconnected, and operational. AI application security is forcing the next realization: when models, data, tools, and autonomous workflows are woven into production systems, security has to extend beyond posture into behavior. Organizations will still need strong identity, configuration, and workload controls, but with non-human agents in the loop, they must also prove that delegated actions remain bounded, observable, and governable over time.

Those finding success in this next phase will be the ones that build governance, secure architecture, runtime controls, and operational discipline into AI systems from the outset, before those systems sprawl into production on their own terms.

AI application security marks the next stage in the cloud journey's maturation. And just as CNAPP pushed organizations from fragmented visibility toward a more connected model, AI application security is pushing them toward a more complete definition of what a secure application really is.

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.

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