POSITION PAPER

The AI Security Maturity Model

A Deployment-Aware Framework for Securing AI

From Co-Pilots to Multi-Agent Systems

Jonathan Gill

AIRS Framework / airuntimesecurity.io

March 2026

Executive Summary

Existing AI maturity models fail to address a fundamental reality: the security requirements for AI vary dramatically depending on how an organisation deploys it. A company using Microsoft 365 Copilot faces a categorically different risk surface to one building autonomous agents that execute financial transactions. Current frameworks treat AI as a monolithic capability and prescribe a single maturity ladder. This produces either over-engineered controls for simple co-pilot deployments or dangerously inadequate controls for agentic systems.

This paper introduces a deployment-aware AI security maturity model that recognises five distinct AI deployment types, each with its own risk profile, control requirements, and governance overhead. Combined with five maturity stages from Reactive through Adaptive, this creates a matrix that serves both as a diagnostic tool and a prescriptive gating function. Organisations can identify exactly where they stand for each deployment type and, critically, understand what maturity level they must achieve before advancing to more complex AI deployments.

The model draws on the AI Runtime Behaviour Security (AIRS) framework and its Multi-Agent Security Operations (MASO) extension to provide operational specificity for the advanced deployment types where existing frameworks offer the least guidance.

The Problem with Current AI Maturity Models

The market offers no shortage of AI maturity frameworks. Gartner, Microsoft, CMMI, Accenture, and McKinsey have all published variations on a five-stage progression from experimentation to enterprise transformation. The NIST AI Risk Management Framework provides comprehensive risk taxonomy. ISO 42001 offers an auditable management system standard. Google's SAIF addresses secure AI architecture.

Each of these contributes something valuable. None of them solves the core problem.

The fundamental gap is that these frameworks assess AI maturity as a single dimension when the reality is multi-dimensional. An organisation can be highly mature in its deployment of embedded co-pilots while being completely unprepared for agentic AI. Existing models either cannot express this distinction or do not attempt to.

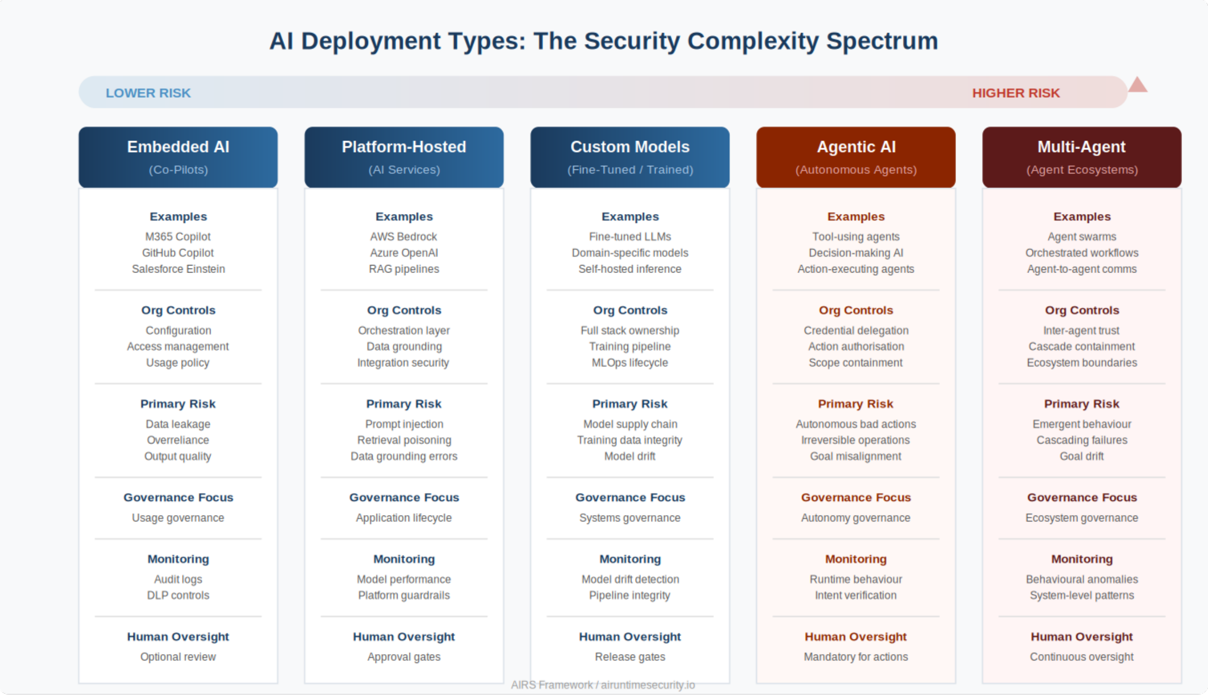
This produces two failure modes:

- **Over-restriction of low-risk deployments.** Organisations apply enterprise-grade governance to co-pilot tools, creating friction that drives shadow AI adoption. The controls are disproportionate to the risk and the business response is predictable: people route around them.
- **Under-restriction of high-risk deployments.** The same organisations then apply identical (or only marginally enhanced) controls to agentic systems, where the risk surface is fundamentally different. Static policies and manual approval gates cannot operate at the speed of autonomous agents.

The missing capability is a model that simultaneously assesses security posture and enablement friction, differentiated by deployment type. Security and usability must be treated as co-equal, interdependent maturity dimensions, not competing priorities.

AI Deployment Types

Rather than treating AI as a single capability, this model recognises five deployment types that represent progressively increasing organisational complexity, risk surface, and control requirements. Organisations will typically operate across multiple deployment types simultaneously, and the maturity model must accommodate this reality.



AI Deployment Types: The Security Complexity Spectrum

	Embedded AI (Co-Pilots)	Platform-Hosted (AI Services)	Custom Models (Fine-Tuned / Trained)	Agentic AI (Autonomous Agents)	Multi-Agent (Agent Ecosystems)
Examples	M365 Copilot GitHub Copilot Salesforce Einstein	AWS Bedrock Azure OpenAI RAG pipelines	Fine-tuned LLMs Domain-specific models Self-hosted inference	Tool-using agents Decision-making AI Action-executing agents	Agent swarms Orchestrated workflows Agent-to-agent comms
Org Controls	Configuration Access management Usage policy	Orchestration layer Data grounding Integration security	Full stack ownership Training pipeline MLOps lifecycle	Credential delegation Action authorisation Scope containment	Inter-agent trust Cascade containment Ecosystem boundaries
Primary Risk	Data leakage Overreliance Output quality	Prompt injection Retrieval poisoning Data grounding errors	Model supply chain Training data integrity Model drift	Autonomous bad actions Irreversible operations Goal misalignment	Emergent behaviour Cascading failures Goal drift
Governance Focus	Usage governance	Application lifecycle	Systems governance	Autonomy governance	Ecosystem governance
Monitoring	Audit logs DLP controls	Model performance Platform guardrails	Model drift detection Pipeline integrity	Runtime behaviour Intent verification	Behavioural anomalies System-level patterns
Human Oversight	Optional review	Approval gates	Release gates	Mandatory for actions	Continuous oversight

AIRS Framework / airuntimesecurity.io

Figure 1: AI Deployment Types and the Security Complexity Spectrum

Embedded AI (Co-Pilots)

This category covers AI capabilities embedded in third-party products: Microsoft 365 Copilot, GitHub Copilot, Salesforce Einstein, Adobe Firefly, and similar tools. The organisation does not control the model, the training data, or the runtime environment. The security perimeter is limited to configuration, access control, and usage policy.

The primary risks are data leakage through prompts, overreliance on AI-generated outputs, and output quality failures. These are real risks, but they are manageable through proportionate controls. Critically, the organisation is governing usage, not systems. The control overhead should reflect this.

Platform-Hosted AI (AI Services)

Organisations in this category are building AI applications on cloud platforms: AWS Bedrock, Azure OpenAI Service, Google Vertex AI, or Databricks. They control the orchestration layer, the data pipeline, and the integration architecture, but not the foundation model itself.

The risk surface expands significantly. Prompt injection, retrieval-augmented generation poisoning, data grounding accuracy, and integration vulnerabilities all enter scope. The organisation is now governing applications, not just usage. This requires application lifecycle governance, platform guardrails, and input/output monitoring that extends beyond simple audit logging.

Custom Models (Fine-Tuned and Trained)

This deployment type involves fine-tuning foundation models on proprietary data or training domain-specific models from scratch, with the organisation hosting its own inference infrastructure. The organisation now owns a significantly larger portion of the AI stack.

Model supply chain integrity, training data provenance, model drift, and the full MLOps lifecycle become primary concerns. The governance focus shifts to systems governance: version control, reproducibility, validation gates, and continuous monitoring of model behaviour against baseline performance.

Agentic AI (Autonomous Agents)

Agentic AI represents a qualitative shift in the risk profile. Single agents with tool access, autonomous decision-making capability, and the ability to execute real-world actions introduce risks that are categorically different from the previous deployment types. The concern is no longer limited to bad outputs; it extends to bad actions.

Credential delegation, action authorisation, scope containment, and irreversibility assessment become primary control requirements. Manual approval gates, which may be adequate for platform-hosted applications, cannot operate at the speed required by autonomous agents. This is where the transition from static governance to runtime behavioural controls becomes essential.

Multi-Agent Systems (Agent Ecosystems)

Multi-agent systems involve multiple AI agents interacting with each other, potentially across organisational boundaries, with emergent behaviours that cannot be predicted from the behaviour of individual agents. This is the frontier of AI deployment complexity.

Inter-agent trust, cascading failure containment, goal drift across agent networks, and emergent behaviour detection are the defining challenges. The security model cannot be perimeter-based because there is no single perimeter. Governance must operate at the ecosystem level, with continuous runtime monitoring capable of detecting system-level anomalies that individual agent monitoring would miss.

The Five Maturity Stages

The maturity stages describe an organisation's security and governance capability, assessed independently for each deployment type. The critical design principle is that maturity is measured by the relationship between security posture and enablement friction. The most mature stage is not the one with the most controls; it is the one where controls are proportionate, automated, and transparent to users.

Stage 1: Reactive

No formal AI security controls exist. AI tools may be in use without organisational awareness. There is no usage visibility, no data protection specific to AI interactions, and no governance framework. This is the default state for most organisations before AI security becomes a recognised function.

At this stage, the organisation has no basis for deploying any AI beyond embedded co-pilots, and even those carry unmanaged data leakage risk.

Stage 2: Restrictive

The organisation has recognised AI as a risk domain and has responded with controls, but those controls are blanket restrictions rather than proportionate measures. Common patterns include outright bans on AI tools, mandatory manual approval for every use case, and lengthy risk assessment processes that treat a customer service chatbot the same as an autonomous trading agent.

This stage creates a paradox: the controls exist to reduce risk, but the friction they create drives shadow AI adoption, which increases risk. The organisation believes it is governing AI when it is actually pushing AI usage underground. Blanket restriction is a governance failure, not a governance achievement.

Stage 3: Structured

Controls are risk-tiered and proportionate. The organisation has a use case evaluation framework that differentiates between deployment types and risk levels. Approval processes exist but are calibrated to complexity. Platform guardrails are in place for managed deployments. Governance is manual but deliberate.

This is the minimum acceptable maturity for platform-hosted AI deployments and custom model development. However, manual governance processes create a speed constraint that makes this stage insufficient for agentic AI, where decisions and actions occur faster than human review cycles can accommodate.

Stage 4: Integrated

Guardrails are automated and embedded in the deployment pipeline. Runtime monitoring is active. Risk-proportionate controls operate without manual intervention for standard use cases. Self-service AI deployment is possible within defined boundaries. The security function enables rather than obstructs AI adoption.

This is the minimum acceptable maturity for agentic AI deployment. The AIRS framework’s first two layers (Guardrails and LLM-as-Judge evaluation) provide the operational model for this stage. Action authorisation and scope containment are automated, with human oversight reserved for high-consequence decisions and edge cases.

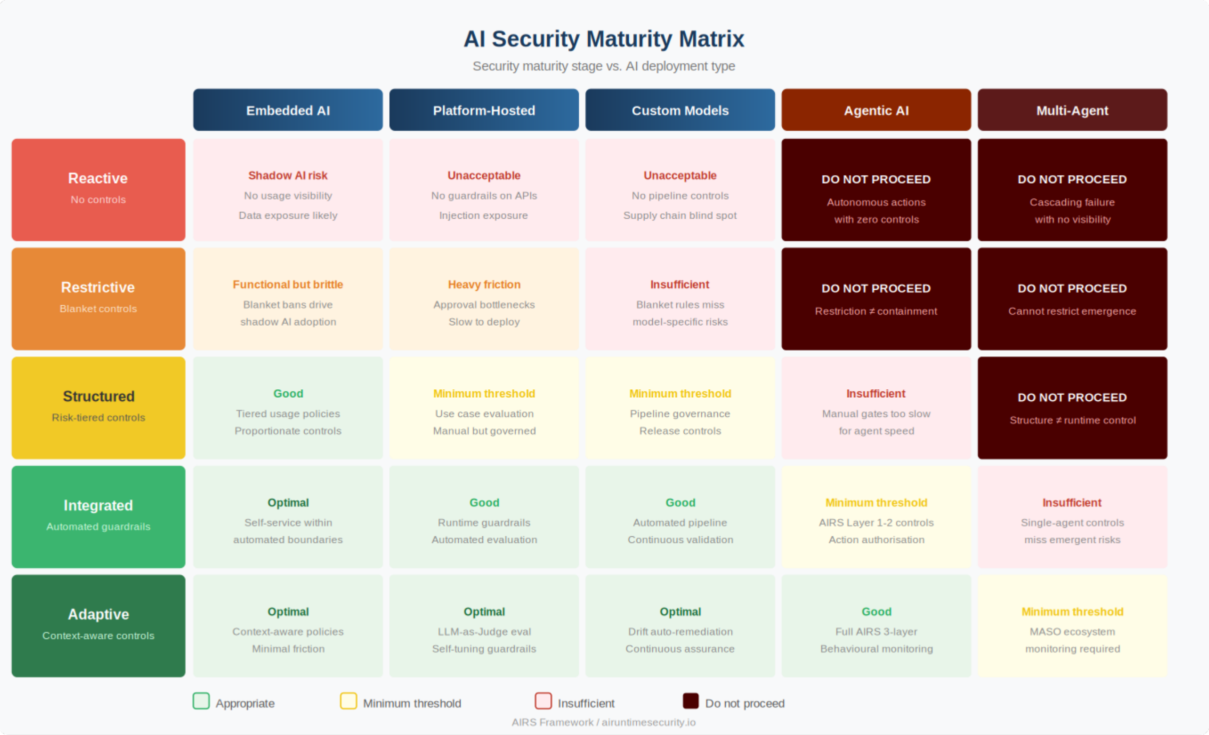
Stage 5: Adaptive

Controls are context-aware and self-adjusting. LLM-as-Judge evaluation provides continuous runtime assessment. Behavioural monitoring detects anomalies against established baselines. Guardrails adapt based on observed patterns, risk context, and operational environment. The full AIRS three-layer architecture (Guardrails, LLM-as-Judge, Human Oversight) is operationalised.

This is the minimum acceptable maturity for multi-agent systems. The MASO framework provides the ecosystem-level monitoring, inter-agent trust verification, and cascade containment capabilities required at this stage.

The AI Security Maturity Matrix

The core contribution of this model is the intersection of maturity stages with deployment types, creating a matrix that serves as both a diagnostic and a prescriptive tool. Each cell in the matrix represents a specific combination of organisational maturity and deployment complexity, with a clear assessment of whether that combination is appropriate, marginal, insufficient, or dangerous.



AI Security Maturity Matrix

Security maturity stage vs. AI deployment type

	Embedded AI	Platform-Hosted	Custom Models	Agentic AI	Multi-Agent
Reactive No controls	**Shadow AI risk** No usage visibility Data exposure likely	**Unacceptable** No guardrails on APIs Injection exposure	**Unacceptable** No pipeline controls Supply chain blind spot	**DO NOT PROCEED** Autonomous actions with zero controls	**DO NOT PROCEED** Cascading failure with no visibility
Restrictive Blanket controls	**Functional but brittle** Blanket bans drive shadow AI adoption	**Heavy friction** Approval bottlenecks Slow to deploy	**Insufficient** Blanket rules miss model-specific risks	**DO NOT PROCEED** Restriction ≠ containment	**DO NOT PROCEED** Cannot restrict emergence
Structured Risk-tiered controls	**Good** Tiered usage policies Proportionate controls	**Minimum threshold** Use case evaluation Manual but governed	**Minimum threshold** Pipeline governance Release controls	**Insufficient** Manual gates too slow for agent speed	**DO NOT PROCEED** Structure ≠ runtime control
Integrated Automated guardrails	**Optimal** Self-service within automated boundaries	**Good** Runtime guardrails Automated evaluation	**Good** Automated pipeline Continuous validation	**Minimum threshold** AIRS Layer 1-2 controls Action authorisation	**Insufficient** Single-agent controls miss emergent risks
Adaptive Context-aware controls	**Optimal** Context-aware policies Minimal friction	**Optimal** LLM-as-Judge eval Self-tuning guardrails	**Optimal** Drift auto-remediation Continuous assurance	**Good** Full AIRS 3-layer Behavioural monitoring	**Minimum threshold** MASO ecosystem monitoring required

🟩 Appropriate 🟨 Minimum threshold 🟥 Insufficient ■ Do not proceed

AIRS Framework / airuntimesecurity.io

Figure 2: The AI Security Maturity Matrix

The Gating Function

The matrix embeds a prescriptive gating function: an organisation’s security maturity for a given deployment type must meet a defined minimum threshold before deploying at that level. The minimum thresholds are:

- **Embedded AI (Co-Pilots):** Structured maturity (Stage 3) for governed deployment; Restrictive (Stage 2) is functional but creates shadow AI risk.
- **Platform-Hosted AI:** Structured maturity (Stage 3) minimum. Manual governance is acceptable if processes are calibrated and timely.
- **Custom Models:** Structured maturity (Stage 3) minimum. Pipeline governance and release controls must be in place.
- **Agentic AI:** Integrated maturity (Stage 4) minimum. Automated runtime controls are non-negotiable; manual gates cannot keep pace with agent execution speed.
- **Multi-Agent Systems:** Adaptive maturity (Stage 5) minimum. Ecosystem-level monitoring and cascade containment are required.

The "Do Not Proceed" cells in the matrix are not advisory. An organisation operating agentic AI at Reactive or Restrictive maturity has autonomous systems executing actions with no meaningful controls. This is not a risk to be managed; it is a risk to be eliminated.

The Maturity Inflection Point

The most important transition in the model is from Restrictive (Stage 2) to Structured (Stage 3). This is where organisations move from blanket controls to proportionate, risk-tiered governance. It is also where the relationship between security and usability inverts.

At Stages 1 and 2, increasing security capability reduces usability. Controls are added without calibration, creating friction. At Stage 3, the curve inflects. From this point forward, increasing maturity improves both security and usability simultaneously, because the controls become smarter and more targeted. The governed pathway becomes easier than the ungoverned alternative, which eliminates the incentive for shadow AI.

This inflection is the central insight of the model. Organisations that understand it will invest in reaching Stage 3 quickly for their most common deployment types, knowing that the return on that investment compounds as they advance. Organisations that do not understand it will remain trapped at Stage 2, fighting a losing battle against shadow AI while believing their controls are effective.

Mapping to the AIRS Framework

The AI Runtime Behaviour Security (AIRS) framework and its Multi-Agent Security Operations (MASO) extension provide the operational controls that underpin the advanced maturity stages for the more complex deployment types. The mapping is direct:

AIRS Three-Layer Architecture

- **Layer 1 (Guardrails):** Input/output filtering, content safety, and policy enforcement. Maps to the Integrated maturity stage. Provides the automated, always-on controls that replace manual approval gates.
- **Layer 2 (LLM-as-Judge):** Continuous evaluation of AI behaviour against defined criteria. Maps to the transition between Integrated and Adaptive maturity. Enables context-aware assessment that static rules cannot achieve.
- **Layer 3 (Human Oversight):** Escalation and review for high-consequence decisions and novel situations. Present at all maturity stages but changes in character. At lower maturity, human oversight is the primary control. At higher maturity, it is the exception handler.

PACE Resilience Methodology

The PACE methodology (Primary, Alternate, Contingency, Emergency) provides the resilience dimension that existing maturity models lack entirely. Each control layer has defined fallback positions, ensuring that security degrades gracefully rather than failing catastrophically when individual components are compromised or unavailable.

MASO for Multi-Agent Systems

MASO extends the AIRS control model to address the specific challenges of multi-agent ecosystems: inter-agent trust verification, behavioural anomaly detection at the system level, cascade containment, and emergent behaviour monitoring. These capabilities map directly to the Adaptive maturity stage for multi-agent deployments.

Using the Model

The model supports three primary use cases for organisations:

Diagnostic Assessment

For each AI deployment type in active use or under consideration, assess the organisation’s current maturity stage. This produces a maturity profile that immediately reveals mismatches between deployment complexity and governance capability. A profile showing Structured maturity for Embedded AI but Reactive maturity for Platform-Hosted AI, for example, identifies a clear governance gap before it produces an incident.

Investment Prioritisation

The matrix identifies where governance investment will produce the highest return. Moving from Restrictive to Structured for Embedded AI eliminates shadow AI risk and reduces operational friction. Moving from Structured to Integrated for Platform-Hosted AI automates manual governance overhead. The prescriptive thresholds also identify where investment is mandatory before deployment can proceed.

Deployment Gating

Before advancing to a new deployment type, the organisation verifies that its security maturity meets the minimum threshold for that type. This prevents the common pattern of technology adoption outpacing governance capability. The gating function is not a bureaucratic obstacle; it is a prerequisite for safe operation.

Conclusion

The AI security landscape requires a maturity model that reflects the diversity of AI deployment patterns, not one that treats all AI as equivalent. An organisation using co-pilots is solving a fundamentally different security problem to one deploying autonomous agents, and the governance model must acknowledge this.

This paper proposes a matrix model that assesses maturity across two dimensions: the organisation’s security and governance capability, and the complexity of its AI deployments. The model provides diagnostic, prescriptive, and gating functions that existing frameworks lack. It treats security and usability as interdependent rather than competing priorities, recognising that the most mature organisations are those where security enables rather than restricts AI adoption.

The AIRS framework and MASO extension provide the operational reference implementation for the advanced maturity stages and deployment types where the governance gap is most acute. As organisations progress from embedded co-pilots toward agentic and multi-agent systems, the need for runtime behavioural security controls will become the defining governance challenge. This model provides the roadmap for meeting it.

References and Further Reading

- AIRS Framework: airuntimesecurity.io
- NIST AI Risk Management Framework (AI RMF 1.0)
- ISO/IEC 42001:2023 Artificial Intelligence Management System
- Google Secure AI Framework (SAIF)
- Gartner AI Maturity Model
- Microsoft Responsible AI Maturity Model