CSAI Foundation

Cloud Security Alliance AI Safety Initiative

AI Infrastructure Monoculture: Foundation-Layer Concentration Risk

Security Implications of Converging AI Orchestration Dependencies

Unofficial AI-assisted Research

2026-04-06

Table of Contents

Executive Summary

Enterprise AI is converging faster than the security industry can assess the consequences. Three foundation model providers now account for 88% of enterprise large language model (LLM) API spending [1]. A single orchestration library — LangChain — has surpassed 130 million cumulative downloads and underlies more than 132,000 known production applications [2]. A gateway library called LiteLLM, present in roughly 36% of cloud environments, was the vector for a sophisticated supply chain attack in March 2026 that targeted cloud credentials across tens of thousands of affected systems [3]. When critical infrastructure converges on shared dependencies at this scale, isolated vulnerabilities become systemic events.

This whitepaper examines the structure of the AI infrastructure monoculture across three dependency layers: foundation models, orchestration frameworks, and integration protocols. It documents the vulnerability history that each layer has accumulated, analyzes the LiteLLM supply chain compromise as a case study in how shared AI dependencies amplify threat actor impact, and draws parallels to historical software monoculture failures including Log4Shell, SolarWinds, and the XZ Utils backdoor. It concludes with risk-tiered recommendations for security and platform engineering teams, and maps findings to applicable CSA frameworks.

The central finding is that AI infrastructure concentration risk is not theoretical. The industry has already experienced a supply chain attack against a universally embedded AI component, multiple critical-severity vulnerabilities in dominant orchestration frameworks, and protocol-level weaknesses in the fastest-spreading AI integration standard. The question for enterprise security teams is no longer whether monoculture risk will materialize, but how to reduce exposure before the next systemic event.

Introduction: The Convergence Phenomenon

Every major technology stack goes through a consolidation phase, where a fragmented field narrows to a small number of dominant components that become load-bearing infrastructure for countless downstream systems. Web frameworks, container runtimes, logging libraries, and cryptographic toolkits have all followed this arc. In each case, consolidation brought genuine benefits — shared maintenance effort, interoperability, community support — alongside a structural risk that the software security community calls monoculture: the condition where a flaw in a universally shared component becomes everyone's problem simultaneously.

The AI application stack is consolidating rapidly — aggregate open-source AI agent framework downloads grew 340% year-over-year through 2025 [4], a velocity that exceeds consolidation timelines seen in comparable infrastructure categories. Between 2023 and 2025, the open-source AI agent framework

ecosystem went from a fragmented collection of experimental libraries to a mature market with clear leaders, measured in aggregate downloads of 34.5 million in 2025 alone [4]. Alongside this consolidation in orchestration tooling, the foundation model market consolidated around a handful of frontier providers, and a new integration protocol — the Model Context Protocol (MCP) — emerged as a de facto standard for connecting AI models to external tools and data sources.

The convergence is occurring across three distinct layers simultaneously. At the foundation layer, frontier model providers serve most of the enterprise market. At the orchestration layer, one library dominates by adoption while a small set of alternatives handles most of the remaining market. At the integration layer, a new protocol championed by a leading provider is being adopted industry-wide at remarkable speed. This vertical convergence creates what is not merely a wide attack surface but a deep one — where compromising a component at any layer can cascade across the entire stack.

The Three Layers of Concentration

Foundation Model Concentration

The foundation model market, despite its relative youth, has already reached levels of provider concentration that would draw regulatory scrutiny in most established technology sectors. By the end of 2025, three providers — Anthropic, OpenAI, and Google — collectively accounted for an estimated 88% of enterprise LLM API usage, with Anthropic leading at approximately 40% of enterprise spend, OpenAI at 27%, and Google at 21% [1]. This represents a notable shift from 2023, when OpenAI held roughly 50% of the enterprise market; the concentration has not decreased as the market matured, but has instead shifted between providers while remaining at the same structural level [1][27].

Capital concentration reinforces market concentration. Foundation model companies collectively captured $80 billion in funding in 2025, representing 40% of all global AI investment. OpenAI and Anthropic alone absorbed an estimated 14% of all global venture capital across every sector — a figure drawn from industry investment tracking and warranting independent verification against primary data sources such as Pitchbook or CB Insights [5]. This capital asymmetry makes it difficult for alternative providers to achieve the scale necessary to serve as credible alternatives at the enterprise tier, reinforcing the concentration dynamic.

Vendor lock-in compounds the risk. Only 11% of enterprise AI builders switched providers in the first half of 2025; 66% upgraded within their existing provider relationship [1]. The technical and contractual costs of switching are significant: prompt engineering, fine-tuning, evaluation pipelines, and operational monitoring

are all calibrated to specific model families and API behaviors. When the majority of production AI workloads depend on a provider that also controls the underlying model, enterprises face a concentration risk that is both technical and commercial in nature.

AI workloads carry embedded assumptions about model behavior — including prompt engineering, fine-tuning artifacts, and evaluation criteria calibrated to specific model families — that create switching costs above and beyond those typical in service migration. The practical implication is clear: if a dominant foundation model provider experiences a prolonged outage, a significant security incident, or a model regression, the impact will ripple simultaneously through the large majority of enterprise AI deployments.

Orchestration Framework Concentration

Above the foundation model layer, the orchestration layer presents a similar concentration profile. LangChain and its graph-based extension LangGraph have emerged as the dominant frameworks for building production AI applications. By February 2025, LangChain had accumulated more than 28 million monthly PyPI downloads, over 99,000 GitHub stars, and 130 million total downloads across Python and JavaScript distributions — figures verifiable via PyPI download statistics — with more than 132,000 LLM applications documented as built on LangChain, according to the project's own reporting [2]. This figure is not independently audited and should be treated as a floor estimate from the project's own tracking. LangSmith, LangChain's observability product, reported 25,000 monthly active teams and 1 billion trace logs — figures that suggest LangChain-based systems are deeply embedded in production infrastructure, not merely development environments.

LlamaIndex, the dominant framework for retrieval-augmented generation (RAG) workflows, occupies a distinct but equally critical position in the stack: it handles the data integration and retrieval tier that feeds context into LLM calls. In this sense, the AI application stack exhibits a layered monoculture, with different dominant components at each functional tier feeding into each other.

At the enterprise tier, Microsoft's consolidation of AutoGen and Semantic Kernel into a unified Microsoft Agent Framework — announced in October 2025 — represents a different dimension of concentration: a large technology provider absorbing multiple popular open-source projects into a single strategic platform [6]. This move accelerates adoption in Microsoft-aligned enterprise environments while simultaneously increasing the dependencies that flow through Microsoft's AI platform choices.

The AI agent framework market has not maintained the diversity that characterized its early experimental phase. LangChain's dominance at the orchestration layer, combined with LlamaIndex's near-monopoly on RAG tooling and Microsoft's consolidation of AutoGen and Semantic Kernel, has concentrated the majority of production workloads in a small number of components — even as smaller alternatives continue to exist. A handful of dominant frameworks, some vendor-controlled and some community-driven, collectively serve most production AI deployments.

Gateway and Protocol Concentration

The third layer — and the one that has already produced a major security incident — is the AI gateway and integration protocol layer. LiteLLM, a Python library that provides a unified API translation layer for more than 100 LLM provider APIs, has become embedded infrastructure in the AI stack. It is used in MCP servers, agent frameworks, internal developer portals, and cloud AI platforms as a normalizing layer between applications and the diversity of underlying model providers. By 2026, it was present in an estimated 36% of cloud environments and downloaded approximately 3.4 million times per day [3].

The Model Context Protocol represents a different kind of concentration: a protocol standard rather than a library. MCP, developed by Anthropic and rapidly adopted across the industry, defines how AI models communicate with external tools — file systems, APIs, databases, code execution environments. Its adoption has been swift enough that security researchers assessing public MCP server deployments found that a severe SSRF vulnerability pattern identified in Microsoft's MarkItDown MCP server may be latent in approximately 36.7% of all publicly accessible MCP servers [7][28]. When a single protocol governs how AI agents interact with external systems, and that protocol has architectural security properties that are inconsistently implemented, the resulting risk is not isolated to individual servers but is structurally distributed across the ecosystem.

The Vulnerability Record

Critical Vulnerabilities in Dominant Frameworks

The monoculture problem would be concerning even if the dominant components had strong security track records. In practice, the major AI orchestration frameworks have accumulated a significant catalogue of critical vulnerabilities — several of which directly affect the production systems of organizations that rely on these tools.

LangChain, by virtue of its dominant adoption, has been a particularly active target for security research. CVE-2025-68664, published in December 2025 and assigned a CVSS score of 9.3 by the issuing CNA, exposes a serialization injection vulnerability in LangChain's `dumps()` and `dumpd()` functions. User-controlled fields — including metadata and response metadata fields that applications routinely pass through without sanitization — can be injected as LangChain object structures, enabling extraction of secrets from environment variables and, via Jinja2 template instantiation, potential arbitrary code execution [8]. CVE-2025-68665, a companion vulnerability with a CVSS of 8.6, extends the same mechanics to LangChain's JavaScript implementation [9]. Both were fixed in specific patch releases, but the window between disclosure and patching across a cumulative download base exceeding 130 million PyPI and npm

installations represents substantial dwell time for exploitation. The actual count of distinct production deployments is not publicly reported but is substantially smaller than the cumulative download figure; the 132,000 known production applications cited earlier represents the best available estimate of production exposure.

LangChain's vulnerability history extends beyond the 2025 disclosures. CVE-2024-36480 identified remote code execution resulting from unsafe `eval()` usage in custom tool definitions, with a CVSS of 9.0 [10]. CVE-2023-44467 documented a prompt injection pathway that allowed attackers to use Python's `__import__()` function to bypass AST sanitization and import the `subprocess` module – enabling RCE through what amounted to an AI framework that trusted model outputs as executable code [10].

LlamaIndex, the dominant RAG framework, contributed CVE-2025-1793 to the record: a CVSS 9.8 critical SQL injection vulnerability in multiple vector store integrations, where methods that directly handled LLM output used that output as input to SQL queries without sanitization. The affected integrations include ClickHouse, Couchbase, DeepLake, and several others, with vulnerable behavior present in versions through v0.12.21 [11]. The significance of this vulnerability class extends beyond its technical severity: it represents a category of flaw where an AI system's natural inputs – model outputs – become injection vectors for downstream data stores. Traditional SQL injection defenses are not designed for attack surfaces where SQL is constructed from language model responses.

CrewAI, one of the fastest-growing multi-agent frameworks, was the subject of security research in early 2026 that identified four exploitable vulnerabilities including RCE, SSRF, and local file read. The researchers demonstrated that these could be chained via prompt injection targeting the framework's Code Interpreter tool to escape the framework's execution sandbox and run code on the underlying host [12]. A peer-reviewed study presented at COLM 2025 provided quantitative context for these findings: in controlled adversarial testing designed to measure worst-case susceptibility, CrewAI running on GPT-4o was convinced by a specially crafted malicious file to exfiltrate private user data in 65% of trials. The Magentic-One orchestrator, developed by Microsoft, executed arbitrary malicious code in 97% of trials when presented with a malicious file [13]. Across all tested agent-model combinations, the overall refusal rate for malicious prompts was 41.5%, meaning more than half of adversarial prompts succeeded. These figures derive from controlled lab conditions designed to maximize exploit success rates; real-world outcomes against hardened production deployments will vary, but the underlying vulnerability classes are genuine.

Case Study: The LiteLLM Supply Chain Compromise

The most significant security event in the AI infrastructure stack to date is the LiteLLM supply chain attack of March 2026, which demonstrated how concentration in a gateway library translates directly into systemic exposure.

The attack vector was multi-stage and technically sophisticated. The threat actor, identified as "TeamPCP," did not attack LiteLLM's codebase directly. Instead, they targeted LiteLLM's continuous integration pipeline by first compromising Aqua Security's Trivy container scanning tool, which LiteLLM's CI/CD workflow pulled from an `apt` repository without version pinning. The poisoned CI/CD runner exfiltrated the `PYPI_PUBLISH` token used to authenticate LiteLLM releases to the Python Package Index [3].

Armed with legitimate publishing credentials, the attacker published two malicious versions – litellm==1.82.7 and litellm==1.82.8 – that contained a `.pth` file causing their payload to execute on every Python interpreter startup on any system where the package was installed. The malware collected environment variables, SSH keys, cloud platform credentials for AWS, Google Cloud, and Azure, Kubernetes secrets, database passwords, SSL and TLS private keys, and Git and CI/CD secrets. Collected credentials were encrypted with AES-256-CBC, wrapped with an RSA-4096 key, and posted to `https://models.litellm.cloud/` – a domain crafted to mimic legitimate LiteLLM infrastructure [3].

The Kubernetes-aware component of the malware was particularly consequential. Where a Kubernetes service account token was present, the malware read secrets from all namespaces in the cluster and attempted to deploy a privileged pod to every node in the `kube-system` namespace – converting an initial PyPI download into cluster-wide compromise.

The malicious packages remained available on PyPI for approximately 40 minutes before being quarantined. For environments running unpinned or latest-version dependencies, even a 40-minute window at LiteLLM's reported download velocity represented real and bounded exposure; the actual scope depended on update cadence and dependency pinning practices across affected deployments. The incident's documented impact – confirmed credential theft across real production systems, as reported by Sonatype [24] and Snyk [25] – does not require extrapolation from download rate alone to establish its severity. The incident illustrated that concentration in a library used as universal infrastructure – not as a primary application framework but as a plumbing component embedded inside other frameworks – creates a risk profile distinct from concentration in any single application stack. LiteLLM was not the application; it was the substrate.

Protocol-Level Risks in MCP

The Model Context Protocol's rapid adoption has outpaced the security community's ability to assess its deployment patterns. Several significant vulnerabilities have been documented across MCP server implementations. Anthropic's own MCP Inspector developer tool was found to allow unauthenticated remote code execution via its inspector-proxy component – an especially notable finding because developer tooling is often treated as outside the production security perimeter [14]. CVE-2025-68145 documented a path traversal vulnerability in the official Git MCP server through which administrative restrictions on repository access were not enforced [15]. A prompt injection attack against the official

GitHub MCP server allowed a malicious GitHub Issue to cause an AI assistant to exfiltrate data from private repositories, demonstrating that the attack surface for prompt injection expands substantially when an AI model is connected to live data sources through a standardized protocol [16].

The supply chain attack surface in the MCP ecosystem has also materialized. A backdoored npm package directed compromised MCP servers to blind-copy all outgoing email to attacker-controlled addresses — using the trusted position of an MCP server in an AI agent's toolchain as a covert exfiltration channel [16]. Red Hat's analysis of MCP security risks and controls documents additional architectural patterns that create systemic exposure across implementations [28]. These incidents collectively indicate that MCP is not merely an API standard but an attack surface, and that its standardization creates shared exposure rather than reducing it.

Historical Parallels

The Log4Shell Template

The Log4Shell vulnerability (CVE-2021-44228) is the canonical example of monoculture risk in software infrastructure. Apache Log4j, a logging library maintained by a small number of volunteers, had become so ubiquitously embedded in Java applications — with an estimated three billion devices running the Java runtime — that a single remote code execution vulnerability in the library translated immediately into critical exposure across an extraordinary breadth of systems. The U.S. Department of Homeland Security's Cybersecurity and Infrastructure Security Agency characterized the vulnerability as posing "severe risk" to the internet, and DHS's Undersecretary for Policy estimated in 2022 that organizations would be dealing with remediation challenges "for at least the next 10 years" [17].

The structural parallel to AI orchestration is direct. LangChain occupies a position in the AI application stack analogous to Log4j in the Java ecosystem: it is a foundational component documented in more than 132,000 known production applications, with indirect downstream dependencies likely extending that count further. The critical vulnerabilities documented in LangChain in 2023, 2024, and 2025 have not yet produced a Log4Shell-scale event, but the preconditions for one are present. A sufficiently severe vulnerability in LangChain core — particularly one affecting the deserialization or code execution pathways that CVE-2025-68664 demonstrates are accessible — could require coordinated remediation across applications that have no direct relationship with each other except their shared dependency.

A peer-reviewed analysis of three major supply chain attacks — SolarWinds, Log4Shell, and XZ Utils — mapped 114 unique attack techniques across all three incidents to 73 mitigation tasks derived from ten supply chain security frameworks [18]. The researchers found that three critical mitigation categories —

including sustainable open-source software funding and environmental scanning tooling — were absent from all ten frameworks. This gap is directly relevant to the AI orchestration ecosystem, where several dominant frameworks depend on community maintenance structures with limited security-specific investment.

The XZ Utils Dimension

The XZ Utils backdoor of 2024 introduced a more targeted form of monoculture risk: a nation-state-attributed threat actor who spent two years establishing trust as a legitimate open-source contributor before inserting a backdoor into a compression library present in most Linux distributions. The incident demonstrated that the relevant attack surface for monoculture risk is not only the technical interface of a library but its human governance structure — the trust relationships that determine who can commit code and publish releases.

The LiteLLM supply chain attack follows a similar logical structure, though with a different initial vector. Rather than impersonating a contributor to LiteLLM itself, the attacker worked upstream through LiteLLM's CI/CD toolchain, compromising a scanning tool to obtain the credentials needed to publish releases. The shared lesson is that the security of widely-deployed components cannot be evaluated solely by reviewing their code: the entire CI/CD pipeline, dependency chain, and credential management posture of a library's maintenance infrastructure constitute its effective security boundary.

Systemic Risk in the Financial Sector

While the software monoculture literature has historically focused on operational disruption and adversarial exploitation, a parallel analytical tradition has emerged in financial system risk assessment that addresses the consequences of shared AI model adoption. The European Systemic Risk Board's Advisory Scientific Committee published a formal report in December 2025 concluding that reliance on a small number of AI providers and similar model architectures could create correlated exposures across financial markets — the AI equivalent of herding behavior — where diverse institutions reach similar conclusions simultaneously, amplifying rather than dampening market volatility [19]. The M³ Framework analysis published as a preprint in March 2026 formalizes this concern through the concept of "systemic amplification events," where shared AI middleware and model architectures transform locally optimal decisions into correlated systemic failures [20]. Though this preprint has not yet undergone peer review, its conceptual framing is consistent with the ESRB's independently produced findings.

The relevance to enterprise security extends beyond the financial sector. Any domain in which multiple organizations simultaneously operate AI systems drawing on the same orchestration frameworks and foundation models faces the possibility that a single adversarial event — a model provider compromise, a critical framework vulnerability, a protocol-level attack — could affect all of them at once. The security

implications are not limited to the affected organization but include third-party dependencies, shared infrastructure providers, and the cumulative regulatory and reputational consequences of industry-wide simultaneous incidents.

Compounding Risk: When Layers Converge

The three concentration layers described above do not represent independent risk pools. They are vertically integrated: a typical enterprise AI deployment uses a foundation model from one of three providers, orchestrated by LangChain or a closely related framework, connected to external tools via MCP, with LiteLLM or an equivalent gateway providing the API translation layer. This architectural pattern means that a threat actor who successfully compromises any one of these components gains access to a system that is also processing credentials, context, and instructions from all the others.

A prompt injection attack delivered via an MCP server, for example, enters the system at the protocol layer but executes within the context of the orchestration framework, which may have access to environment variables, tool execution capabilities, and retrieval databases. If the framework contains a serialization vulnerability of the kind documented in LangChain, the injected instruction may be able to escalate from context manipulation to code execution. The gateway library — LiteLLM or equivalent — may then exfiltrate credentials extracted during execution to the foundation model provider's API logging infrastructure or to an attacker-controlled endpoint. This is not a theoretical chain; elements of it have been individually demonstrated in the vulnerability record.

The compounding effect also operates at the organizational level. When multiple enterprise AI deployments share the same orchestration framework, gateway library, and protocol implementation, a threat actor who maps the shared infrastructure can design exploits that are effective simultaneously across all of them. The economics favor the attacker: the cost of developing a single exploit against a universal component is amortized across every downstream deployment, while the cost of remediation falls on each organization individually. This asymmetry is the defining characteristic of monoculture risk, and it is why the AI infrastructure convergence warrants security attention disproportionate to what the current incident record alone might suggest.

Regulatory Landscape and Industry Response

NIST IR 8596: The Cyber AI Profile

In December 2025, NIST published the preliminary public draft of IR 8596, a Cybersecurity Framework Profile for AI — commonly called the Cyber AI Profile [21]. Structured around the six CSF 2.0 functions (Govern, Identify, Protect, Detect, Respond, Recover), the profile explicitly identifies unvetted third-party libraries and AI dependencies as a primary entry point for supply chain attacks. It addresses adversarial machine learning threats — data poisoning, model evasion, supply chain integrity risks in model development pipelines — in dedicated sections. The Cyber AI Profile is the most comprehensive NIST guidance to date on AI-specific security, and its treatment of third-party dependencies is directly applicable to the concentration risks described in this paper.

CISA and the AI Bill of Materials

The Cybersecurity and Infrastructure Security Agency has extended its software bill of materials (SBOM) work to cover AI-specific components through an AI Bill of Materials (AIBOM) initiative [22]. The AIBOM concept adds structured descriptions for AI models, training and testing datasets, and model provenance to traditional software component inventories. This is a necessary precondition for organizations seeking to understand their actual exposure to foundation model provider concentration or orchestration framework vulnerabilities: you cannot assess risk you cannot enumerate. CISA's updated SBOM Minimum Elements guide, released in August 2025, provides the foundational requirements that the AIBOM initiative builds upon.

EU AI Act: Systemic Risk Provisions

The European Union's AI Act imposed GPAI model obligations on providers of high-capability general-purpose AI models beginning August 2, 2025, with systemic risk assessment and mitigation requirements included in scope [23]. For organizations subject to the Act, supply chain vulnerabilities in third-party AI APIs and orchestration dependencies constitute systemic risks that require documented assessment and mitigation plans. Full high-risk AI system obligations become enforceable August 2, 2026, making the current period a critical window for organizations to build the compliance infrastructure that demonstrates adequate concentration risk management. This date should be re-verified at final publication given the history of EU AI Act implementation adjustments.

Conclusions and Recommendations

Summary of Risk Posture

The AI infrastructure monoculture presents a categorically different risk profile from typical software supply chain concerns. The combination of extreme concentration across three vertically integrated layers, a documented history of critical vulnerabilities in each layer, and at least one major supply chain incident that exploited this concentration constitutes a systemic risk condition. Organizations that have not explicitly assessed their exposure to this risk should treat it as a priority gap.

The risk is not uniformly distributed. Organizations with centralized AI platform teams, clear AI asset inventories, and active dependency management postures are materially better positioned than those whose AI deployments are fragmented across business units with ad-hoc toolchain choices. The recommendations below are organized by the investment required to implement them.

Immediate Actions

The first priority is visibility. Organizations should inventory all AI components in production environments — foundation model providers, orchestration frameworks, gateway libraries, and MCP server deployments — with the same rigor applied to other critical infrastructure dependencies. An AI Bill of Materials for each production AI system is the minimum baseline for assessing exposure to a specific vulnerability or supply chain event.

Organizations using LiteLLM, LangChain, or LlamaIndex should confirm that they are running versions patched for the critical vulnerabilities documented in this paper: CVE-2025-68664 (fixed in `langchain-core` v1.2.5 [8]) and CVE-2025-68665 (fixed in `langchain` v0.3.81 [9]), and CVE-2025-1793 (LlamaIndex, fixed in v0.12.28 [11]). Dependency pinning and automated vulnerability scanning across the Python and npm package ecosystems should be verified as operational.

For organizations deploying MCP servers, network segmentation and authentication requirements should be reviewed against the patterns documented in recent MCP vulnerabilities. MCP servers that have privileged access to internal systems should not be exposed to untrusted inputs — including outputs from AI models that have processed external data — without explicit trust boundaries enforced at the infrastructure layer.

Short-Term Mitigations (within 60–90 Days)

Organizations should evaluate their foundation model provider concentration and document a credible contingency plan for major provider unavailability or incident response. This does not require active multi-provider operation in steady state, but it does require that API abstraction layers, model evaluation pipelines, and prompt engineering artifacts be maintained in a state where provider switching is a realistic option within a defined recovery time objective.

Orchestration framework dependencies should be treated as security-sensitive components with dedicated patch management policies. The velocity of vulnerability disclosure in this ecosystem — multiple critical-severity findings in the dominant frameworks in 2023, 2024, and 2025 — suggests that passive dependency management is insufficient. Organizations should establish explicit patch windows and acceptance criteria for framework updates, and should test framework patches in isolated environments before production deployment.

Supply chain integrity controls should be extended to the AI component layer. LiteLLM's CI/CD pipeline was compromised through an unpinned upstream dependency — a class of vulnerability that software composition analysis tools can detect when properly configured. Package signing verification, hash pinning, and provenance attestation using standards such as SLSA (Supply-chain Levels for Software Artifacts) are applicable to AI dependencies and should be implemented with the same prioritization given to security-critical libraries in traditional software stacks.

Strategic Considerations

At the architectural level, organizations building new AI platforms should apply a diversity principle to component selection. This does not mean artificial fragmentation — using multiple orchestration frameworks within a single system creates complexity that outweighs the resilience benefit — but it does mean avoiding single-vendor dependencies at each layer. A system that uses LangChain for orchestration, Anthropic Claude as its primary model, and LiteLLM as its gateway presents a different concentration profile than a system where the same vendor controls two or three of those layers simultaneously.

The financial sector's emerging analysis of AI monoculture and correlated failure provides a useful framework for evaluating industry-wide risk beyond individual organizations. Security teams at organizations whose AI failures could have systemic sector effects — financial institutions, critical infrastructure operators, large cloud providers — should participate in sector-level AI incident coordination mechanisms as these develop under CISA guidance and analogous international frameworks.

Finally, the governance of open-source AI infrastructure warrants dedicated investment. Several of the dominant components in the AI orchestration ecosystem depend on maintainer teams whose security review capacity is not proportional to the production exposure they carry. The LiteLLM attack succeeded in

part because the library's CI/CD pipeline had an unpinned dependency on a security scanning tool — a configuration gap more common in community projects than in enterprise software with dedicated security engineering resources. Enterprise consumers of open-source AI frameworks have both interest and leverage in improving the security posture of these projects, and should consider channeling security engineering resources, vulnerability disclosure partnerships, and funding through mechanisms such as the Alpha-Omega Project toward the specific components that carry the most concentrated risk.

CSA Resource Alignment

MAESTRO (Agentic AI Threat Modeling)

The Cloud Security Alliance's MAESTRO framework provides a systematic approach to threat modeling for agentic AI systems. The concentration risks described in this paper are addressable within MAESTRO's supply chain threat modeling approach, particularly its treatment of AI supply chain threats and the trust boundary analysis relevant to multi-agent orchestration. Organizations using MAESTRO for agentic AI deployment reviews should extend their threat models to explicitly include framework dependency vulnerabilities, gateway library supply chain risk, and protocol-level attacks via MCP.

AI Organizational Responsibilities (AIOR) Framework

CSA's AIOR framework addresses organizational accountability for AI system behavior, including the accountability structures needed to respond to third-party AI component failures. The concentration risks described here create scenarios where an organization's AI system failure is triggered by a third-party event outside the organization's direct control. The AIOR framework's guidance on third-party AI accountability and incident response responsibilities is applicable to contingency planning for framework and provider incidents.

Zero Trust Architecture Guidance

CSA's Zero Trust guidance applies to the AI infrastructure layer in two important ways. First, the trust boundary analysis that underlies Zero Trust design is applicable to MCP server deployments, where the protocol's architecture creates implicit trust relationships between AI models and the external systems they can access. Second, the credential access patterns exploited in the LiteLLM supply chain attack — where environment variable extraction provided cloud platform credentials — represent a failure of Zero Trust credential management that CSA's guidance on secrets management and just-in-time privilege addresses.

Cloud Controls Matrix (CCM) and AICM

The CSA Cloud Controls Matrix, and its AI-extended counterpart AICM, provide control domains applicable to the concentration risks analyzed here. The Supply Chain Management and Transparency domain (STA) addresses third-party dependency risk in cloud environments and should be read as encompassing AI orchestration framework dependencies. The Cryptography, Encryption and Key Management domain (CEK) addresses the credential protection failures that made the LiteLLM attack's exfiltration component effective. Organizations conducting CCM or AICM gap assessments should include AI infrastructure dependencies as in-scope assets.

STAR (Security Trust Assurance and Risk)

The CSA STAR program's assurance mechanisms provide a pathway for organizations seeking independent validation of their AI supply chain risk posture. As the AI infrastructure ecosystem matures, CSA STAR assessment criteria should evolve to incorporate AI-specific supply chain controls — including AIBOM generation, framework dependency management policies, and provider concentration risk documentation — as assessed capabilities.

References

[1] Menlo Ventures. "2025 State of Generative AI in the Enterprise." Menlo Ventures, December 2025.

[2] Contrary Research. "LangChain Business Breakdown." Contrary Research, 2025.

[3] LiteLLM. "Security Update: March 2026 PyPI Supply Chain Incident." LiteLLM Official Blog, March 2026.

[4] Fordel Studios. "The State of AI Agent Frameworks in 2026." Fordel Studios Research, 2026. *Note: Methodology not independently disclosed; figure should be treated as an industry estimate.*

[5] AI Business Weekly. "AI Market Share and Investment Analysis 2026." AI Business Weekly, 2026. *Note: Industry newsletter; the 14% global VC share figure should be verified against primary sources such as PitchBook or CB Insights.*

[6] Microsoft Azure. "Introducing Microsoft Agent Framework." Microsoft Azure Blog, October 2025. Also: VentureBeat. "Microsoft Retires AutoGen and Debuts Agent Framework to Unify and Govern." VentureBeat, October 2025.

[7] Dark Reading. "Microsoft and Anthropic MCP Servers at Risk of Takeovers." Dark Reading, 2025.

[8] NIST National Vulnerability Database. "CVE-2025-68664." NVD, December 2025. Also: Cyata AI. "Lang Grinch: LangChain Core Serialization Injection (CVE-2025-68664)." Cyata, December 2025.

[9] NIST National Vulnerability Database. "CVE-2025-68665." NVD, December 2025.

[10] Palo Alto Networks Unit 42. "Vulnerabilities in LangChain Gen AI." Palo Alto Networks, 2024.

[11] NIST National Vulnerability Database. "CVE-2025-1793." NVD, June 2025. Also: Endor Labs. "Critical S QL Injection Vulnerability in LlamaIndex (CVE-2025-1793)." Endor Labs, 2025.

[12] SecurityWeek. "CrewAI Vulnerabilities Expose Devices to Hacking." SecurityWeek, 2026. Also: GBHackers. "CrewAI Hit by Critical Vulnerabilities Including RCE, SSRF, and File Read." GBHackers, 2026.

[13] Wfrei, et al. "Penetration Testing of Agentic AI Systems." Conference on Language Modeling (COLM) 2025, arXiv 2512.14860, 2025.

[14] eSentire. "Model Context Protocol Security: Critical Vulnerabilities Every CISO Should Address in 202 5." eSentire Threat Response Unit, 2025.

[15] Pillar Security. "The Security Risks of the Model Context Protocol (MCP)." Pillar Security, 2025.

[16] AuthZed. "Timeline of MCP Security Breaches and Vulnerabilities." AuthZed, 2025.

[17] National Library of Medicine (PMC). "Log Jam: Lessons From Log4Shell." Biomedical Instrumentation and Technology, 2023.

[18] Lamb, et al. "Closing the Chain: How to Reduce Your Risk of Being SolarWinds, Log4j, or XZ Utils." arXiv 2503.12192, North Carolina State University and Yahoo!, March 2025.

[19] European Systemic Risk Board Advisory Scientific Committee. "AI and Systemic Risk (ASC Report No. 16)." ESRB, December 2025.

[20] Kumar, et al. "Model Monoculture Risk: Systemic AI Convergence in Banking and Financial Markets." Preprints.org, March 2026. *Note: This is a preprint and has not yet undergone peer review.*

[21] NIST. "NIST IR 8596 (Initial Public Draft): Cybersecurity Framework Profile for AI (Cyber AI Profile)." NIST CSRC, December 2025.

[22] CISA. "2025 Minimum Elements for a Software Bill of Materials (SBOM)." CISA, August 2025.

[23] European Commission. "EU AI Act: Regulatory Framework for Artificial Intelligence." European Commission, 2024–2025.

[24] Sonatype. "Compromised litellm PyPI Package Delivers Multi-Stage Credential Stealer." Sonatype, March 2026.

[25] Snyk. "Poisoned Security Scanner: How LiteLLM Was Backdoored Through Its Own CI/CD Tools." Snyk, March 2026.

[26] Trend Micro. "Inside the LiteLLM Supply Chain Compromise." Trend Micro Research, March 2026.

[27] Menlo Ventures. "2025 Mid-Year LLM Market Update." Menlo Ventures, July 2025.

[28] Red Hat. "Model Context Protocol (MCP): Understanding Security Risks and Controls." Red Hat, 2025.