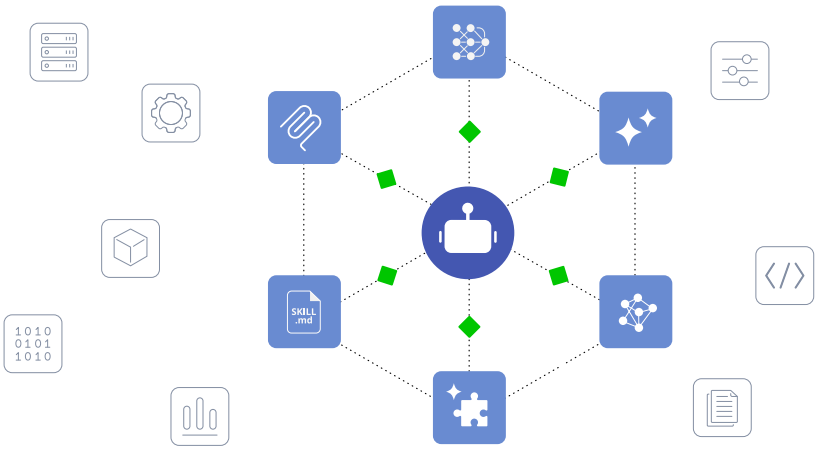
White Paper

The Evolution of the Supply Chain: From Software to Agentic AI

How to govern AI agents, MCP servers, and models with the same rigor you apply to your software supply chain.

Table of **Contents**

Executive Summary



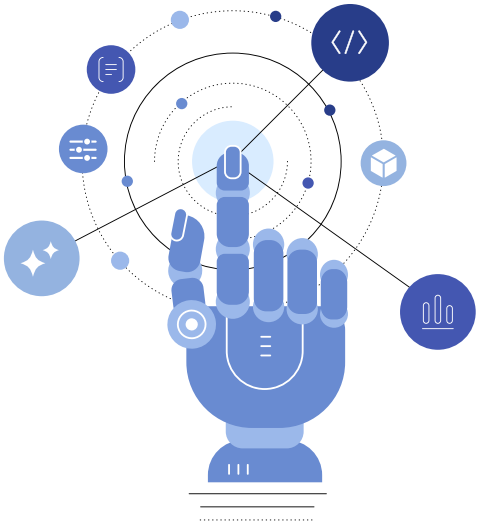
For the past decade, the software supply chain has been the battleground for enterprise security. Organizations have invested heavily in scanning binaries, curating open-source packages, and enforcing policy at ingestion. That discipline was hard-won, and it is already becoming insufficient.

AI is no longer a passive tool. Powered by the Model Context Protocol (MCP), Agent Skills, and Plugins, AI agents now reach into file systems, query databases, execute shell commands, and trigger CI/CD pipelines autonomously. By often inheriting full user permissions, these agents gain deep access to user systems. This elevated access is precisely what allows them to frequently bypass standard configurations and established governance controls. The developer who once manually reviewed every dependency is now deploying agents that pull and execute tools at machine speed, often from unvetted sources.

This shift demands a new security framework: the AI supply chain. This paper defines what the AI supply chain is, maps its unique threat surface, and outlines how organizations can govern it without sacrificing the development velocity that makes AI adoption worthwhile.

The Supply Chain Has a **New Participant**

Generative AI has evolved from a passive reasoning engine into an active agent capable of taking autonomous actions within your infrastructure.



From passive tool to active agent

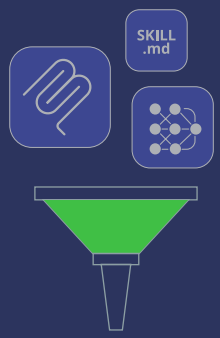
The traditional software supply chain was designed around human-authored artifacts: source code, compiled binaries, container images, and open-source dependencies. Security teams built controls around these artifacts: scanning them, curating them, and governing their movement from development to production.

That model assumed that humans remained in the loop at every meaningful decision point. A developer chose which library to pull. An engineer decided which container image to deploy. A security policy blocked a vulnerable package before it crossed the ingestion boundary.

That assumption no longer holds. The Model Context Protocol is the emerging standard that gives AI agents the ability to take action in the real world: reading and writing files, executing code, querying APIs, and committing to repositories. Where generative AI was once a reasoning engine answering questions, it is now a participant in your supply chain.

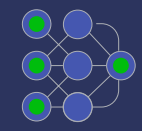
Key Insight

When you give AI 'hands,' you must apply the same governance you apply to any other participant in your pipeline. An unvetted MCP server is not a productivity tool; it is an open ingestion point into your most sensitive infrastructure.



Defining the AI software supply chain

The AI supply chain extends the traditional software supply chain to encompass the new classes of artifacts that AI-powered development introduces:

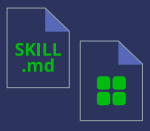


AI Models (open-weight and proprietary)

Pulled from registries such as Hugging Face, which IDC data shows are now favored by 61% of enterprises for generative AI use cases.

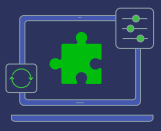
MCP Servers

Packages that expose tools (file access, API calls, shell execution) to AI agents. A single MCP server may contain both safe and dangerous capabilities.



Agent Skills and AI Catalogs

Structured registries of AI capabilities, analogous to npm or Maven for code. This pattern is already emerging in the broader ecosystem through platforms like LangChain Hub, LlamaIndex's LlamaHub, and various enterprise plugin stores. Without governance, these become shadow repositories.



Plugins

Specialized extensions integrated directly into AI agents to expand their contextual understanding and operational scope.

Training Datasets and Model Weights

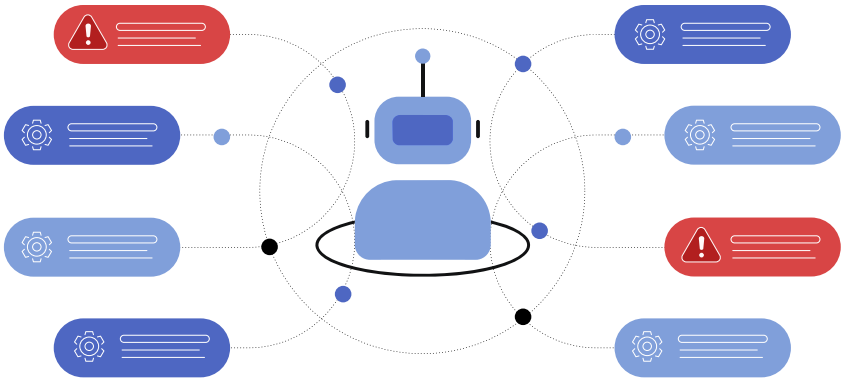
Artifacts that must be versioned, traced, and audited just as application binaries are today.

The AI ecosystem is evolving at an unprecedented pace. The industry continuously introduces novel AI assets engineered to augment agent capabilities and streamline autonomous workflows. This rapid introduction of dynamic dependencies creates entirely new vectors for risk.

The critical difference from the traditional software supply chain is not the technology; it is the blast radius. A poisoned npm package reaches developers who install it. A poisoned MCP server reaches every agent that invokes it, across every workflow it participates in, at speeds no human reviewer can match. Ultimately, managing this expanding blast radius mandates the same rigorous oversight and secure, centralized management provided by an enterprise-grade registry.

The Expanded **Attack Surface**

The AI supply chain introduces threat vectors that existing DevSecOps tooling was not designed to address. Understanding them is the first step toward mitigating them.



The shadow AI supply chain

Developers move fast. Faced with a deadline and a new AI agent skill, CLI extension, or MCP server that promises to automate a tedious workflow, most developers will install it without consulting security. This is not negligence; it is the same behavior that drove the adoption of unvetted npm packages a decade ago, and it carries the same consequences.

The JFrog Security Research team has documented how this new attack surface is already being actively exploited. Attackers have moved upstream into the very tools developers use to write code. In early 2026, JFrog extended its scanning to AI agent skill registries and identified 969 malicious AI agent skills in the ClawHub and Skills.sh repositories. These skills carried critical-impact payloads, confirming that the agentic attack surface has expanded beyond traditional package registries into the specific tools that power autonomous AI workflows.

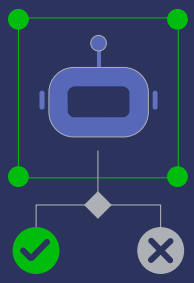
Additionally, the research team identified 495 malicious models on Hugging Face carrying live payloads, which included reverse shells, credential harvesting, and system command execution. Whether an organization adopts MCP servers, independent agent skills, or CLI utilities, these new dynamic dependencies represent a massive vulnerability class operating at an unprecedented level of trust.

Full access, zero oversight

Most MCP servers are built for helpfulness, not security. The core risk stems from how privileges are assigned within these new workflows. By default, it is the AI agent itself that inherits the full permissions of the developer who invokes it. Because these various AI supply chain items are installed directly into the agent, they immediately gain the same level of access the agent possesses. This operational model grants unvetted tools unrestricted reach into highly sensitive areas, such as access to source code, cloud credentials, CI/CD systems, and production secrets. An AI agent that hallucinates, is manipulated via prompt injection, or calls a compromised MCP server, does not just make a bad decision: it executes that decision with administrator-level permissions.

Key Insight

An AI agent does not have intuition. It will not hesitate before executing a dangerous command, notice something feels off, or ask for clarification. It will simply do what it was given permission to do. Permissions are the policy.

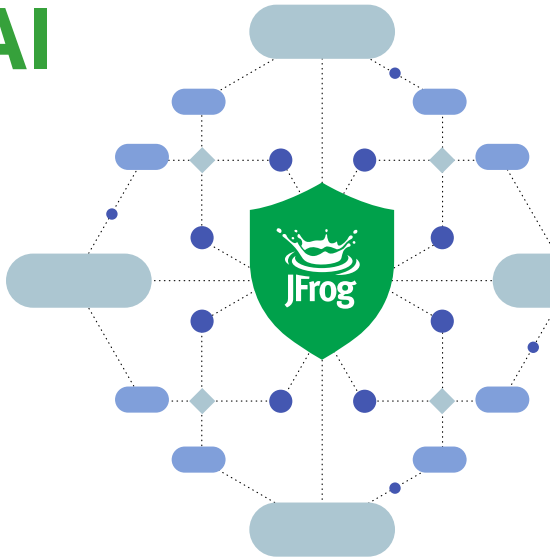


Poisoned models and automated leakage

Just as malicious packages compromise traditional code at installation time, poisoned public models can compromise the surrounding environment and exfiltrate sensitive intellectual property, such as training data, proprietary logic, and internal secrets, at machine speed and without human intervention. Unlike a developer who notices suspicious behavior, an agent running an inference pipeline in the early hours of the morning has no intuition to trigger.

The IDC forecast of over one billion AI agents by 2029 makes this threat surface not a future concern, but a present one. The question is not whether your organization will have AI agents operating autonomously; it is whether those agents are operating under governance.

Building the Governed AI Supply Chain with JFrog



Securing this new frontier requires extending your existing DevSecOps discipline to manage AI assets as first-class artifacts within a unified system of record.

Start with a unified system of record

The central lesson of traditional software supply chain security is straightforward: you cannot govern what you cannot see. Fragmented tooling, such as disconnected scanners, separate repositories, and inconsistent policies across teams, doesn't just create inefficiency; it creates the blind spots that attackers exploit. The organizations that weathered high-profile supply chain events most effectively were those that had already consolidated artifact management and security policy into a single control point, so that when a threat emerged, containment was a policy change, not a manhunt.

The AI software supply chain requires the same foundation. JFrog Artifactory extends its role as the enterprise binary repository to treat AI artifacts as first-class citizens alongside Docker images, Maven packages, and npm modules:

- AI Models and weights are stored, versioned, and traced alongside the code that uses them.
- MCP Servers are managed as governed packages with full dependency metadata.
- Training datasets are tracked with provenance linked to the models they produce.
- Agent skills, plugins, and emerging technologies such as APM are managed centrally within the JFrog Platform, providing a structured, searchable repository with a critical governance layer to ensure secure and compliant enterprise adoption.

The strategic value of this unification is not organizational convenience; it is security. When every model in production can be traced back to its training version, dataset, and build parameters, the blast radius of a supply chain compromise becomes bounded and auditable. When AI tools are managed separately from software, that traceability disappears.

Key Insight

Most security tools treat every vulnerability as equally urgent. 66% of CVEs analyzed by the JFrog Security Research team had a low applicability rate, and only 12% were found to be highly exploitable in real enterprise environments. Scanning everything is not the same as fixing what matters.

Choose proactive security over reactive measures

The math on reactive remediation does not work. An attacker operating at machine speed will always outpace a team hunting vulnerabilities after the fact. The JFrog Security Research Team's analysis of hijacking attacks found that most attacks are discovered within 48 hours, with more sophisticated attacks typically discovered within 14 days.

This creates the attacker's window of opportunity.

The JFrog Platform eliminates that window by blocking risky artifacts at ingestion, before they enter the developer environment. For AI artifacts, this means:

Immature Package Policies

Blocking unvetted or newly published MCP servers that have not yet been reviewed.

Deep Binary Scanning

Scanning MCP server binaries for known CVEs and malicious code post-download.

Contextual CVE Prioritization

Applying Contextual Analysis to cut through alert noise by determining whether a vulnerability is actually exploitable in your specific environment.

AIBOM Readiness

Establishing foundational tracking to support emerging AIBOM requirements. This comprehensive inventory catalogs the models in production alongside the specific agents deployed, including the underlying assets, tools, and behavioral instructions that define how those agents operate.

Granular governance without the friction

There is an honest tension at the center of this conversation that most security vendors avoid naming. For the engineering teams reading this, every new governance layer has historically meant slower builds, more tickets, and frustrated developers routing around controls they see as obstacles rather than guardrails. That experience is real, and it is the reason shadow IT exists in the first place.

The AI software supply chain does not make that tension disappear. It raises the stakes on both sides. The speed advantage of agentic AI is only realized if developers can actually adopt it and trust its output, which means security cannot be the department that says no.
But an ungoverned AI supply chain is not a fast one; it is a liability waiting to surface at the worst possible moment. The goal is not to choose between velocity and security. It is to make security invisible enough that developers do not have to choose.

Governing access without blocking it

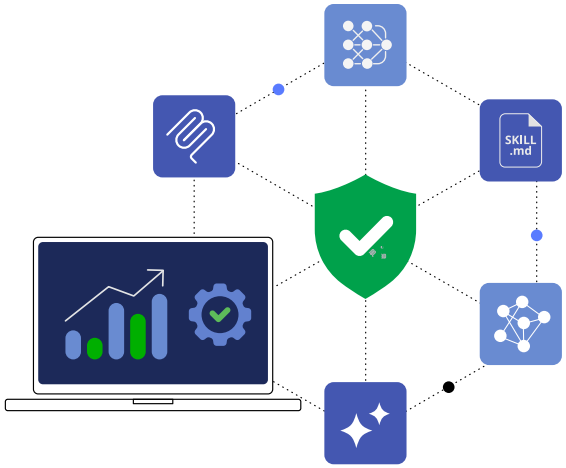
The instinct to simply ban all AI tools is understandable and counterproductive. When security teams impose blanket restrictions, developers route around them. The result is a shadow AI supply chain that is more dangerous, not less, because it is invisible.

The JFrog Platform provides the alternative: granular, tool-level access control. For example, an MCP server is a package that may contain multiple capabilities — some safe, some dangerous. To manage this risk, the JFrog Platform allows organizations to apply permissions at the individual tool level rather than broadly at the server level.

- Senior architects may be granted access to write and delete operations.
- Junior developers may be restricted to read-only capabilities on the same server.
- CI/CD pipelines may be permitted to invoke specific tools while being blocked from others.

This model reflects how enterprises already govern human developer access, and it extends that governance to the AI agents acting on their behalf. The goal is not to slow adoption, but to ensure that every agent in your environment is operating within a defined, auditable trust boundary.

Conclusion: Govern AI for Enterprise Success



The emergence of the AI software supply chain is not a cause to slow down. Developers using AI coding assistants report productivity gains of 20% or more, and the organizations that harness agentic AI effectively will have a significant competitive advantage. The choice organizations are now faced with is not whether to adopt AI; it is whether to govern it or remain in the risky "Wild West".

Without governance, the AI supply chain is fast, powerful, and exposed. Every unvetted MCP server is a potential ingestion point. Every over-privileged agent is a systematic vulnerability. Every ungoverned model is a potential exfiltration channel.

The organizations best positioned for this transition are the ones that extend their existing software supply chain discipline to cover AI artifacts, storing models and MCP servers in the same governed repository as their binaries. They are applying the same curation policies to AI tools as they apply to open-source packages, and enforcing the same access governance on agents as they enforce on human developers.

The supply chain didn't change its rules. It just added new players. Govern them accordingly.

Ready to govern the hands of your AI?

Start a free trial or schedule a demo and see how the JFrog Platform provides the enterprise-grade infrastructure needed to bridge traditional software and autonomous AI securely and at velocity.

JFrog empowers thousands of DevSecOps organizations globally to build, secure, distribute, and connect any software artifact to any environment using the universal, hybrid, multi-cloud JFrog Platform.