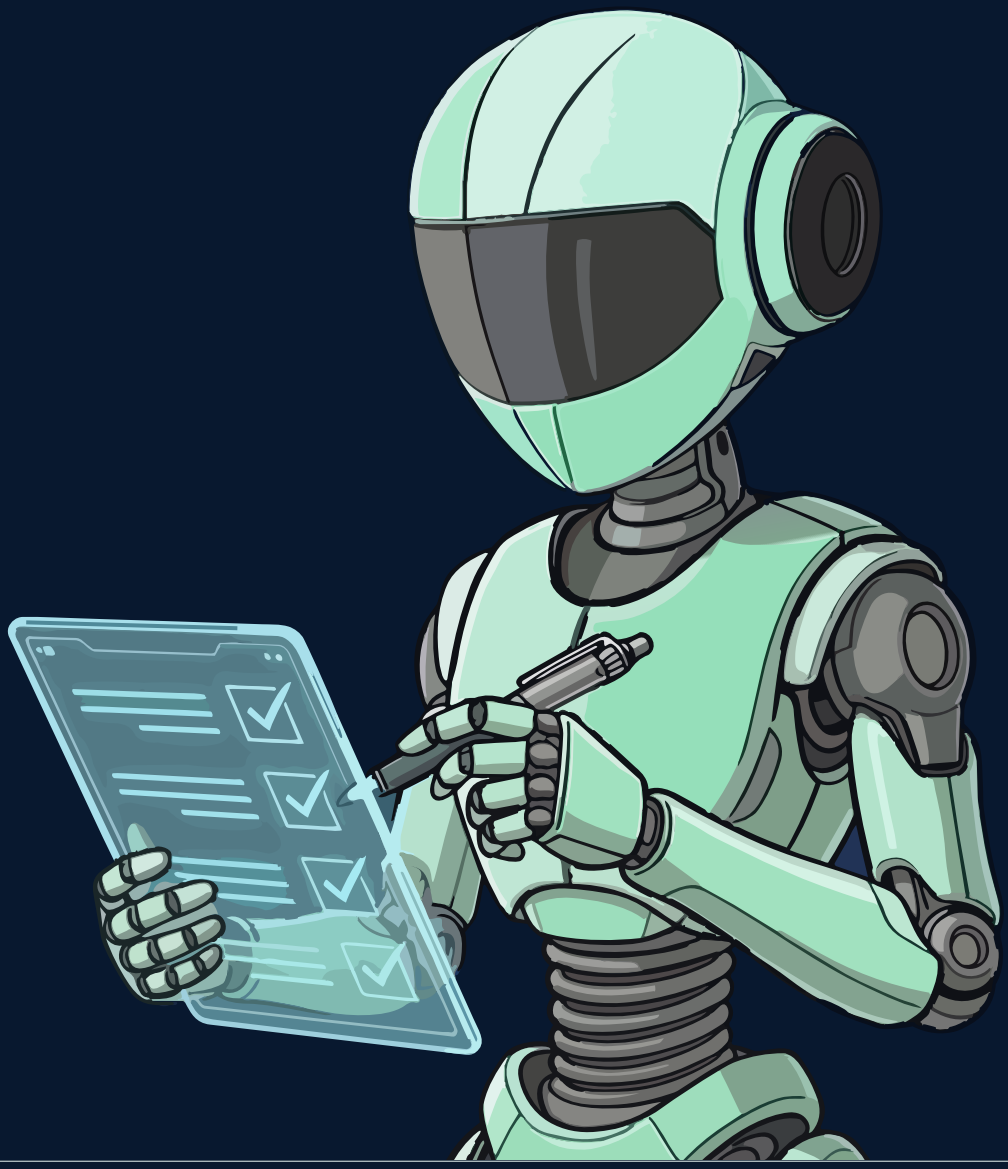
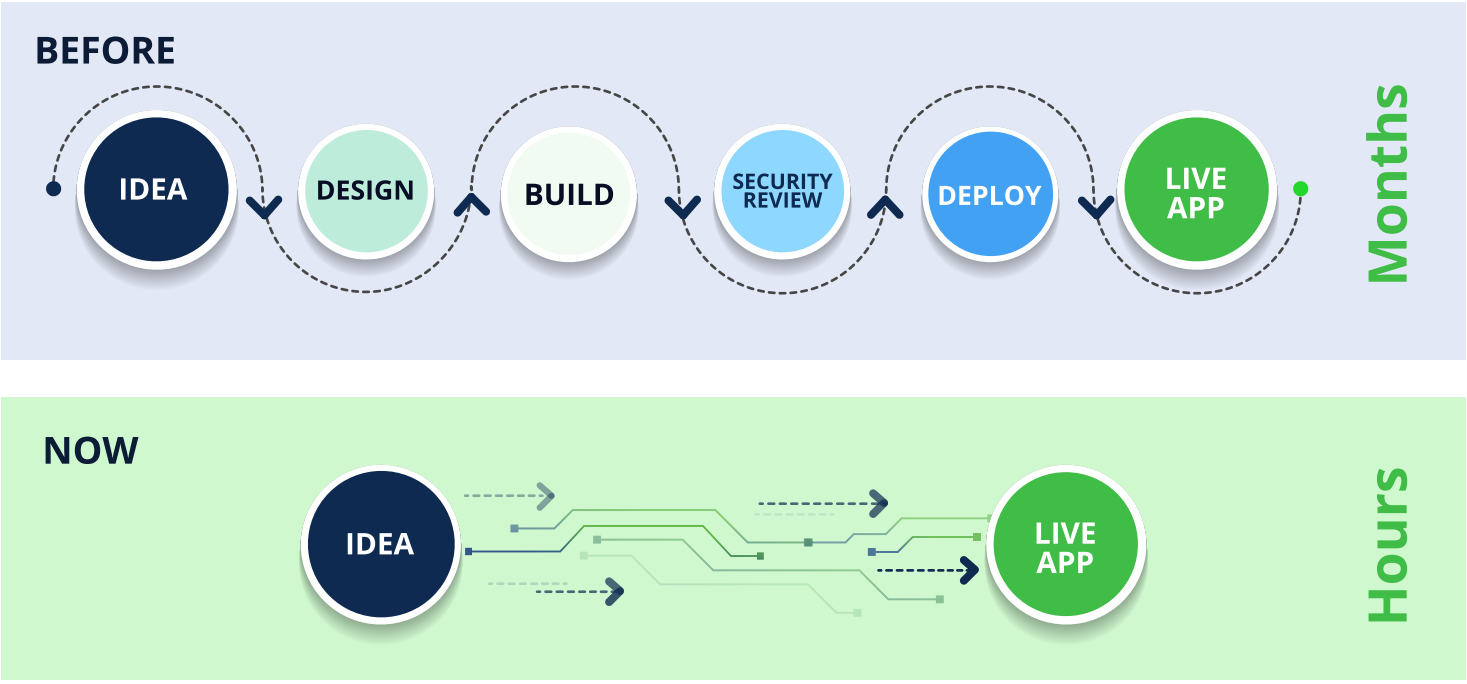
The CISO's Checklist for the Agentic Supply Chain



AI agents, MCP servers, and agent skills are rapidly becoming core components of software supply chain architecture, whether they are vetted or not, and whether your organization is ready or not.

But the bigger shift isn't just what is being built, it's who is building it. The developer population is exploding beyond engineering with non-technical teams (e.g., product, marketing, HR, legal, sales) now using AI tools to generate code, automate workflows, and deploy functionality with little to no development experience. These new "developers" aren't thinking about security; they're thinking about speed, output, and getting it live.



This is the reality of the Agentic Software Supply Chain: any employee's idea can become a live application in hours, not months.

For CISOs, this creates a two-fold problem: Beyond securing code written by trained developers, leadership must now establish security, governance, and trust standards. This requires providing education and guidance for a new wave of "developers" who are building and deploying software without traditional training.

This eBook provides an actionable checklist to help CISOs apply enterprise security standards to the Agentic Software Supply Chain. By securing the AI assets that grant agents the power to act, you can automate policy guardrails, enforce granular tool-level permissions, and prevent unauthorized access to your production environment, regardless of who is building the next app or agentic workflow.

The Four Pillars of Agentic Security

The following sections detail a four-phase strategy to secure your agentic development workflow:

Phase 1: Discovery & Visibility SEVERITY: HIGH

Create the processes and leverage the right tools to build an inventory of all AI models, agent skills, and MCP servers (whether on-device, remote, or custom-built). Additionally, map exactly where AI is being used to build or enable intelligent applications so you can effectively mitigate unmanaged risk vectors.

Phase 2: Supply Chain Integrity SEVERITY: CRITICAL

Establish automated security gates to proactively detect, warn, or block unvetted AI components, and apply the same unified scanning used for any other software artifact, Docker image, and application at runtime.

Phase 3: Granular Governance SEVERITY: CRITICAL

Move beyond binary blocking to enforce tool-specific permissions. This empowers security teams to govern MCP and agent skill usage, ensuring agents maintain least-privilege access to sensitive data.

Phase 4: Unified Platform Management SEVERITY: MEDIUM

Standardize on open protocols and integrate AI tools into your existing software supply chain to define a 'golden path' to production that automates manual reviews and eliminates operational and data silos.

Use the considerations listed under each phase to help you assess your organization's maturity in AI asset discovery and visibility, supply chain integrity, governance, and management.

SEVERITY SCORECARD

LOW

Governed & Contained. AI assets are somewhat vetted, centrally managed within a registry, and restricted by strictly enforced, least-privilege policies.

MEDIUM

Systemic Governance Gap. Fragmented governance controls and a lack of a unified system of record. Guarantees future compliance violations.

HIGH

Active Governance Failure. Ungoverned Shadow AI assets are actively bypassing security controls. Requires immediate visibility into AI usage.

CRITICAL

Immediate Exploitation Risk. Active threat of over-privileged agent, tool poisoning, or data exfiltration. Requires immediate remediation.

Agentic supply chain risk

Phase 1: Discovery & Visibility

SEVERITY: HIGH

You can't protect yourself from the Shadow AI assets you can't see.

Today, users (both developers and agents) run unvetted MCP tools and skills that have bypassed the organization's security perimeter directly on developer machines. You can't govern or secure what you can't see, and you don't want to get caught off guard trying to block malicious or non-compliant MCPs and agent skills that have already entered your environment.

- [ ] **Inventory the Spectrum:** Do you have a central registry that tracks all types of AI assets?
  - MCP servers:
    - On-Device / Desktop (IDE-processes)
    - Remote (cloud/API)
    - Custom-built (internal tools)
  - Agent skills:
    - Local (package-based)
    - Remote (intelligence-based)
    - Custom-built / Agent-built (context-specific)
- [ ] **Identify the Entry Points:** Are you monitoring where developers pull AI assets from (e.g., public GitHub repos, open-source libraries, community lists, or unverified Docker containers)?
- [ ] **Map the Connectivity:** Can you identify which coding agents (e.g., Cursor, Claude Code) are actively connecting to internal systems via MCPs?
- [ ] **Audit Functional Capabilities:** Do you have visibility into the actions those connections allow (e.g., whether the agent's skill is limited to "read-only" documentation access, or if it has "write" permissions to production databases or the ability to execute shell commands)?
- [ ] **Track the Usage:** Are control points in place to approve and track the use of MCP binaries and skills across different teams and projects?

Phase 2: Supply Chain Integrity

SEVERITY: CRITICAL

If an attacker manages to run the code on a developer's machine or a server, they've already breached your perimeter.

Like any software artifact, MCP servers and agent skills can hide malicious functionality or bypass expected security controls. They carry the same risks as any other dependency, like CVEs and malicious code.

- [ ] **Establish a Digital Border:** Have you implemented a "gatekeeper" that intercepts and proactively blocks unvetted or high-risk AI assets before they land on an employee's laptop?
- [ ] **Apply Unified Scanning:** Are you scanning your MCP servers and agent skills with the same security engines used for your production Docker images and NPM packages, prior to making them available within your enterprise?
- [ ] **Verify Provenance:** Can you confirm the "Golden Copy" of every MCP server or .md file is immutable, versioned, and stored in a trusted system of record?

Phase 3: Granular Governance

SEVERITY: CRITICAL

Least privilege principle: even vetted agents can make mistakes, so you want to limit the potential blast radius.

Governance isn't black and white. You shouldn't have to block an entire capability or asset (e.g., ban use of MCP servers) just to disable one dangerous capability.

- [ ] **Enforce Granular Tool Permissions:** Are your agents inheriting full developer credentials by default, or are you enforcing tool- or pattern-specific permissions based on the project scope?
- [ ] **Secure the Gateway:** Is every IDE-to-server connection routed through a secure, authenticated bridge rather than direct internet connections?

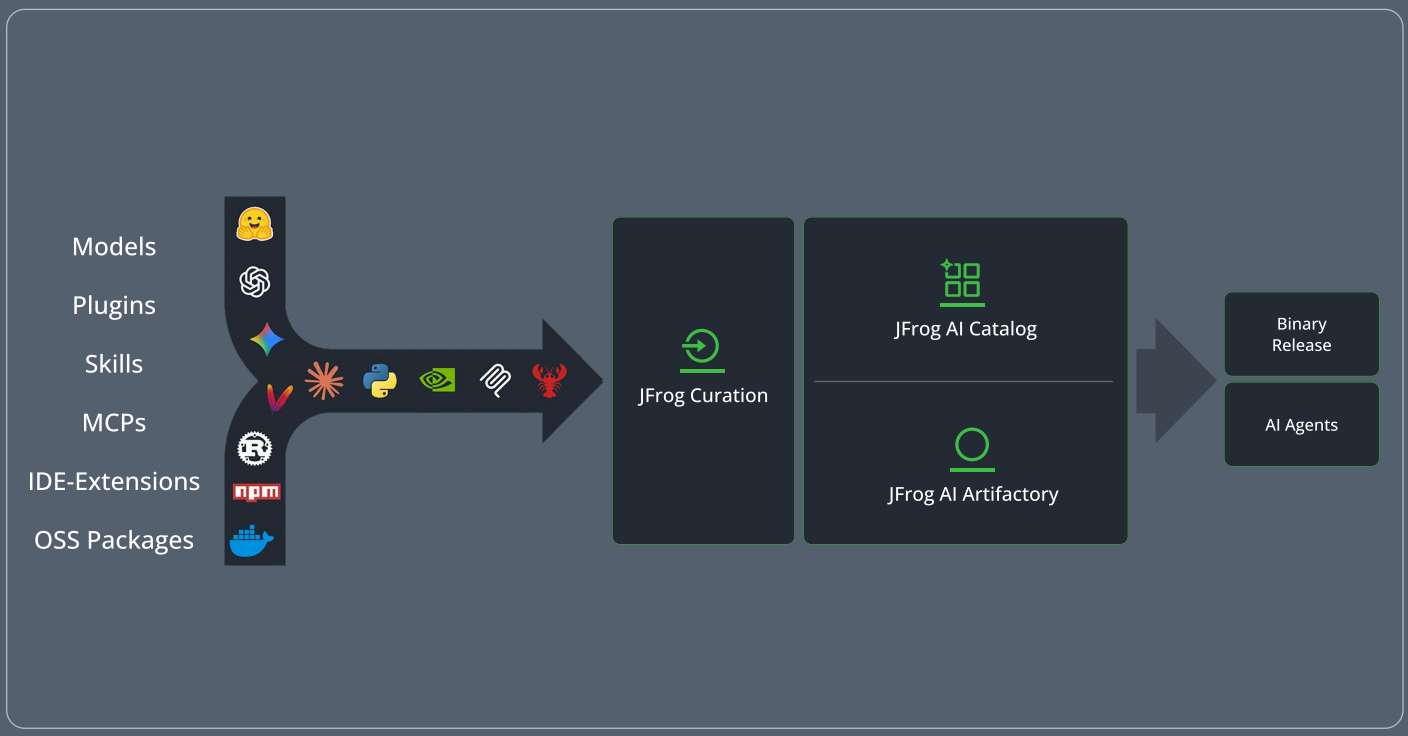
Phase 4: Unified Platform Management

SEVERITY: MEDIUM

To ensure long-term protection, avoid a fractured approach; without unified governance, you face uncontrollable risk and a total loss of business-level visibility.

Managing AI in a silo creates security double standards. AI tools must be managed alongside all other software assets and adhere to pre-existing controls.

- [ ] **Eliminate Operational Silos:** Are your AI assets' registries (e.g., models, MCPs, skills) integrated into your existing software supply chain platform (e.g., JFrog Artifactory) to ensure consistent auditability?
- [ ] **Standardize the Unified Artifact Package:** Can you push a single, secure release package that includes the application code, AI model, and vetted MCP server or agent skill?
- [ ] **Enforce a "Golden Path" for AI Agents:** Have you communicated a "Golden Path" that defines expected security controls and usage standards for anyone developing or working with AI agents?



Why This Matters Now

Current tools only monitor what the agent does, leaving a critical security and governance gap and unmanaged risk. The MCP Registry and Agent Skills Registry of the AI Catalog are the only enterprise-grade solutions that focus on the artifact—MCP servers and agent skills—and secure the code that defines an agent's ability to act freely. By treating these components as governed software packages, organizations can move beyond observation to proactive protection.

Furthermore, you can future-proof your architecture by defining clear usage guidelines and leveraging open standards like MCP. Proactively embracing these capabilities ensures you maintain enterprise control without stifling innovation. Without this enablement, teams will inevitably bypass legacy security controls to use these tools anyway. Denying or delaying this shift is no longer an option.

Bottom Line

As software supply chains become increasingly AI-driven, proactive security is your only path to success. Don't let AI operate unchecked. Regain control of your production environment by automating policy gates, enforcing granular permissions, and embedding security directly into coding agents like Claude Code or Cursor. Ultimately, true success means maintaining a secure, compliant supply chain—no matter how much AI is involved—without ever compromising delivery speed.

Secure your Agentic Supply Chain today.

Get Started with AI Catalog