



The Guide to DevOps, Security, and AI Tool Consolidation

Too Many Tools, Not Enough Value

As AI-driven development accelerates, many organizations manage dozens of disjointed tools across the software development life cycle (SDLC). Dev and security teams each pick their own point solutions, and the resulting tool sprawl creates friction, visibility gaps, and runaway costs.

Tool consolidation means retiring redundant tools and moving multiple functions onto a single platform across DevOps, DevSecOps, DevGovOps, and AgentSecOps. Consolidating into a single source of truth can lower capital and operational costs, reduces vendor fatigue, and gives you broader visibility across the software and agentic software supply chain. It also removes overhead that holds back both human engineers and autonomous agents, giving agents broader system context.

The Industry Shift

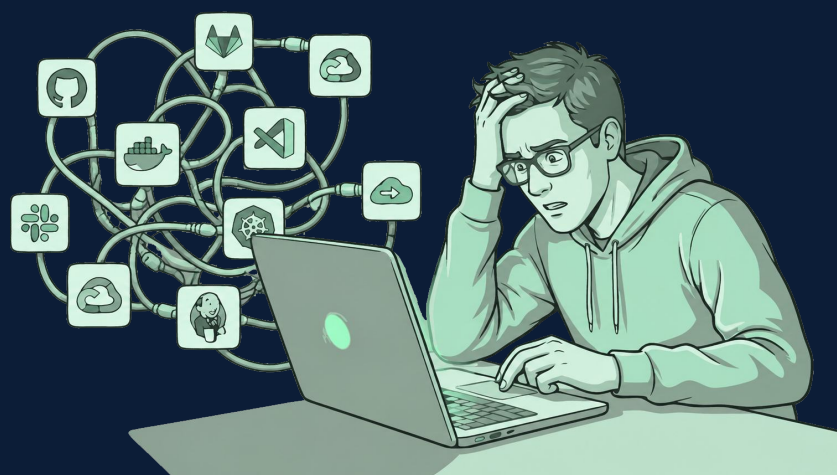
Data from the [JFrog 2026 Software Supply Chain Security State of the Union](#) report underscores why organizations are aggressively consolidating their tool stacks:

↘ Rapid Reduction in Point Tools

The share of surveyed organizations using 7 or more application security tools dropped sharply from 73% to 35% between JFrog's last two annual surveys.

📦 DevOps Platform Reliance

38% of surveyed organizations now rely primarily on their existing DevOps platform for application security.



The Hidden Cost of Tool Sprawl

Managing dozens of isolated point tools creates substantial financial and operational drag. Evaluating the true ROI of consolidation requires analyzing direct software costs alongside hidden developer labor lost to alert noise and manual compliance processes.

Measuring the hidden costs of tool sprawl

Evaluating ROI requires accounting for the hidden labor bottlenecks caused by fragmented point tools. These stats are taken from [The JFrog 2026 Software Supply Chain Security State of the Union](#):



Alert Fatigue & Overrated Severity

In JFrog Security Research's 2025 analysis of high-profile CVEs, JFrog rated 96% of NVD-rated "Critical" CVEs and 65% of "High" CVEs at a lower severity after assessing real-world exploitability. 66% of high-profile CVEs JFrog analyzed have low applicability (0–20%) in enterprise environments. Non-applicable alerts can pull engineering teams into chasing low-risk vulnerabilities.



Manual Audit Preparation

48% of surveyed organizations spend a week or more assembling compliance audit proof per application. In our view, build evidence and provenance data scattered across isolated tools adds to this burden.



AI Review Overhead

45% of DevSecOps teams cite reviewing and hardening AI-generated code as a top time burden.

In our view, a flexible, unified platform is more likely to yield a positive return on investment than point solutions that require ongoing custom integration maintenance.

Calculating the Cost

Use this formula to calculate the cost of tool sprawl:

License Cost + Infrastructure Cost + Maintenance Cost + Training Cost + Integration Cost

Areas for Tool Consolidation in Your DevOps Toolchain

When planning tool consolidation, consider all teams involved: DevOps, AppSec, Platform Engineering, and Data Science/ML teams. Unifying these domain areas helps reduce duplicate efforts and siloed data. Consider consolidating the following areas in your tech stack:

Area	What to Consolidate
Artifact & Binary Management	Replace fragmented package-specific repositories, S3 buckets, and Version Control System (VCS) releases with a universal single system of record for all software binaries, packages, container images, and AI models
Container & Package Registries	Centralize image management across development, staging, multi-cloud, and edge environments to help maintain consistency, immutability, and traceability.
Application Security & Supply Chain Defense	Replace multiple point security scanners (SAST, SCA, secrets detection, container scanning, etc.) with an end-to-end security solution that offers shift-left package curation, automated vulnerability detection, contextual analysis, and automated remediation
AIOps & AgentSecOps	Bring AI, engineering, and DevSecOps teams onto the same unified software supply chain platform to curate, store, secure, and deploy machine learning models, MCP servers, and AI-generated code alongside traditional software artifacts.
Infrastructure & Dynamic Deployment Technologies	Consolidate Infrastructure as Code (IaC) files, configuration scripts (Terraform, Ansible), and runtime artifacts in a centralized registry to streamline deployments across cloud, on-premises, and hybrid environments.
Runtime Security & Monitoring	Leverage unified platform capabilities to monitor application health, track Software Bill of Materials (SBOM) changes, and protect runtime environments from code to production.



A Five-Step Plan for Consolidating Your Software Delivery Tools

Managing tool adoption, deprecation, and integration requires a structured governance plan. Here is a five-step blueprint to guide your tool consolidation initiative:

01 Create a DevOps, security, and AI tools inventory

Start with a complete inventory of every tool your development, security, and AI teams use.



Catalog each tool

Record its version, active users, annual licensing cost, and role in the SDLC.



Flag overlap

Look for capabilities you already own but don't use, such as built-in container registries or integrated security scanning in your primary binary repository.



Keep it current

Update the inventory on a regular cadence, restrict who can edit it, and use scripts and API integrations to track active tools and license consumption. Usage data over time highlights redundant solutions.

02 Identify what matters most in your tooling

Analyze your tool inventory against your organizational strategy and operational priorities:



Feature Parity & Synergy

Identify feature overlaps across tools and determine where a single platform can replace multiple point solutions without sacrificing core functionality.



Maintenance & Operational Overhead

Evaluate vendor management overhead, support SLAs, SaaS vs. self-managed options, and infrastructure maintenance costs. Today, more than ever, ensuring you're on the latest versions of any tools is critical given frontier AI threats. This overhead should be taken into consideration when evaluating tools.



Total Cost of Ownership (TCO)

Conduct a thorough cost analysis of licenses, infrastructure, integration maintenance, and administrative effort. Balance direct financial savings against potential operational risks.

Tool consolidation is a strategic journey, not an overnight event. Legacy systems and specialized infrastructure often require phased migrations with sunset milestones.

03 Decide which focus areas to keep, combine, or phase out

Evaluate each functional domain across your SDLC to decide whether to centralize or phase out legacy systems:



Open-Source Package Curation

Block malicious and non-compliant open-source packages at the download stage before they enter developer environments.



Binary & Model Repository Management

Universal binary management serves as the anchor for consolidation, giving developers and AI pipelines a single system of record for all build components.



Software Supply Chain Security

Transition away from standalone security scanners toward integrated contextual analysis that helps filter out false positives and focus remediation on real exploitability.



Automated Testing & Quality Assurance

Eliminate redundant testing tools by integrating automated security, compliance, and regression checks directly into build pipelines.



Configuration & Infrastructure Automation

Consolidate IaC and configuration files into your central artifact repository to help maintain an auditable deployment software supply chain.



AI Coding Agents

Standardize on approved AI coding assistants and autonomous development frameworks to reduce tool sprawl, support consistent code quality, and maintain centralized governance over repository access.



Agent Platforms & Observability

Centralize agent orchestration and telemetry to monitor prompt execution, model performance, and autonomous decision-making across a unified observability plane rather than relying on fragmented LLM monitoring tools.

04 Choose a platform that unifies tools and accelerates delivery

Select a flexible, open platform that offers rich integrations and fills operational gaps across your entire software delivery lifecycle:

- ✓ Out-of-the-box support for multiple package ecosystems, agents, and AI model frameworks
- ✓ Robust MCP tools, Skills, REST APIs, and CLI tools for automation
- ✓ Multi-cloud, hybrid, and self-managed deployment flexibility
- ✓ High availability, disaster recovery, and global replication capabilities
- ✓ Deep integration with IDEs, CI/CD engines, and AI coding agents
- ✓ Enterprise SSO, RBAC, and granular access controls
- ✓ Automated SBOM management and continuous vulnerability scanning

Look for a platform that serves as a single source of truth across your software, security, and AI assets rather than attempting to stitch together isolated point tools.

05 Execute a structured migration plan

Create an actionable migration roadmap designed to move teams onto the consolidated platform with minimal disruption:

- 1 Set a realistic timeline, budget, and milestone schedule for tool deprecation and data migration.
- 2 Align security policies, compliance standards, and user access roles across all teams.
- 3 Execute phased migrations and continuously monitor pipeline health and release velocity.
- 4 Provide hands-on enablement and training for engineering, security, and data science teams.
- 5 Document the updated architecture and maintain it as part of core IT infrastructure governance.
- 6 Track metrics (such as MTTR, build times, vulnerability counts, and tool cost savings) to quantify business impact.

Engage Finance and Executive Leadership early to highlight both short-term migration costs and potential long-term operational savings.

Common Challenges When Consolidating DevSecOps Tools

While consolidating your DevOps and security tools can deliver significant operational value, navigating the organizational and technical hurdles requires a proactive approach.

Topic	Challenge	Solution
Getting buy-in from all stakeholders	Resistance to changing familiar workflows across engineering, security, and AI teams.	Involve team leads early in the selection process. Share real ROI data and demonstrate how platform consolidation can reduce friction, automate routine tasks, and accelerate software releases.
Consolidating and selecting the right tools	Navigating hundreds of vendor options and avoiding solutions that compromise core functionality.	Choose universal, vendor-agnostic platforms that integrate natively with your existing CI/CD, cloud, and AI ecosystem.
Getting the right tools to work together	Integration complexity between legacy systems and modern security or AIOps tools.	Select a platform built on open REST APIs, CLI tooling, and extensive out-of-the-box integrations (including with agent harnesses) so that your toolchain acts as a cohesive engine.
Training your team	Learning curves associated with adopting new platforms and processes.	Provide structured onboarding, utilize vendor enablement resources (such as vendor training programs and documentation), and establish internal champions to guide best practice adoption. Look for solutions that are AI native, so agents can onboard to the new tooling more easily.
Managing upfront expenses	Temporary overlap costs during migration phases when running legacy and new systems concurrently.	Present a clear financial model showing projected net savings and long-term TCO reduction to secure executive sponsorship.

Consolidate With JFrog

The JFrog Software Supply Chain Platform provides a natural point of consolidation for modern enterprise software development, security, and AI engineering teams. By unifying DevOps, DevSecOps, DevGovOps, and AIOps onto a single hybrid, multi-cloud platform, JFrog serves as a single source of truth for software packages, AI models, and agent assets across the software delivery lifecycle, all the way to production deployment and edge devices.

Quantifying the ROI of consolidating with the JFrog Platform

According to a commissioned [Forrester Consulting Total Economic Impact™ \(TEI\) study](#) in 2026, a composite organization using the JFrog Platform realized:

- \$136K in annual tool consolidation savings
- 282% ROI over three years
- 71% reduction in software development tool spend
- 65% reduction in critical vulnerabilities
- 80% reduction in remediation time

Core domain consolidation

Organizations choosing JFrog can eliminate redundant point tools across several key operational domains:



Universal Binary & Package Management (JFrog Artifactory)

Consolidate 60+ artifact types, including packages, container registries, build metadata, and AI model weights, into a single, high-availability system of record.



Shift-Left Open Source Defense (JFrog Curation & Catalog)

Help block malicious, vulnerable, or non-compliant OSS packages, AI models, and IDE extensions at download time before they enter developer workstations.



Advanced Application Security (JFrog Xray & Advanced Security)

Replace fragmented scanners with unified SCA, secrets detection, IaC scanning, and Contextual Analysis to reduce alert noise and prioritize real exploitability.



AI Security & Governance (JFrog AI Security)

Securely manage, version, and govern AI models, MCP servers, agent skills, and plugins alongside traditional software dependencies.

Conclusion: Manage Less, Achieve More

As organizations seek to optimize spending while protecting their software supply chain, tool consolidation is a strategic imperative. Consolidating onto the JFrog Platform can help your organization reduce operational complexity, elevate your security posture, and empower cross-functional teams to build and deploy software faster with confidence.



Explore tool consolidation with the JFrog Platform today.

[Free Trial](#)

[Learn More](#)

This ebook is provided for general informational purposes only. It is not legal, compliance, or security advice and does not guarantee any security, product, or financial outcome. It reflects information available as of the publication date shown below and may not be updated. Results depend on your environment, situation, and configuration. Product capabilities vary by edition and deployment model and are governed by your agreement with JFrog. JFrog is a registered trademark of JFrog Ltd. in the United States. All other trademarks referenced in this document are the property of their respective owners.