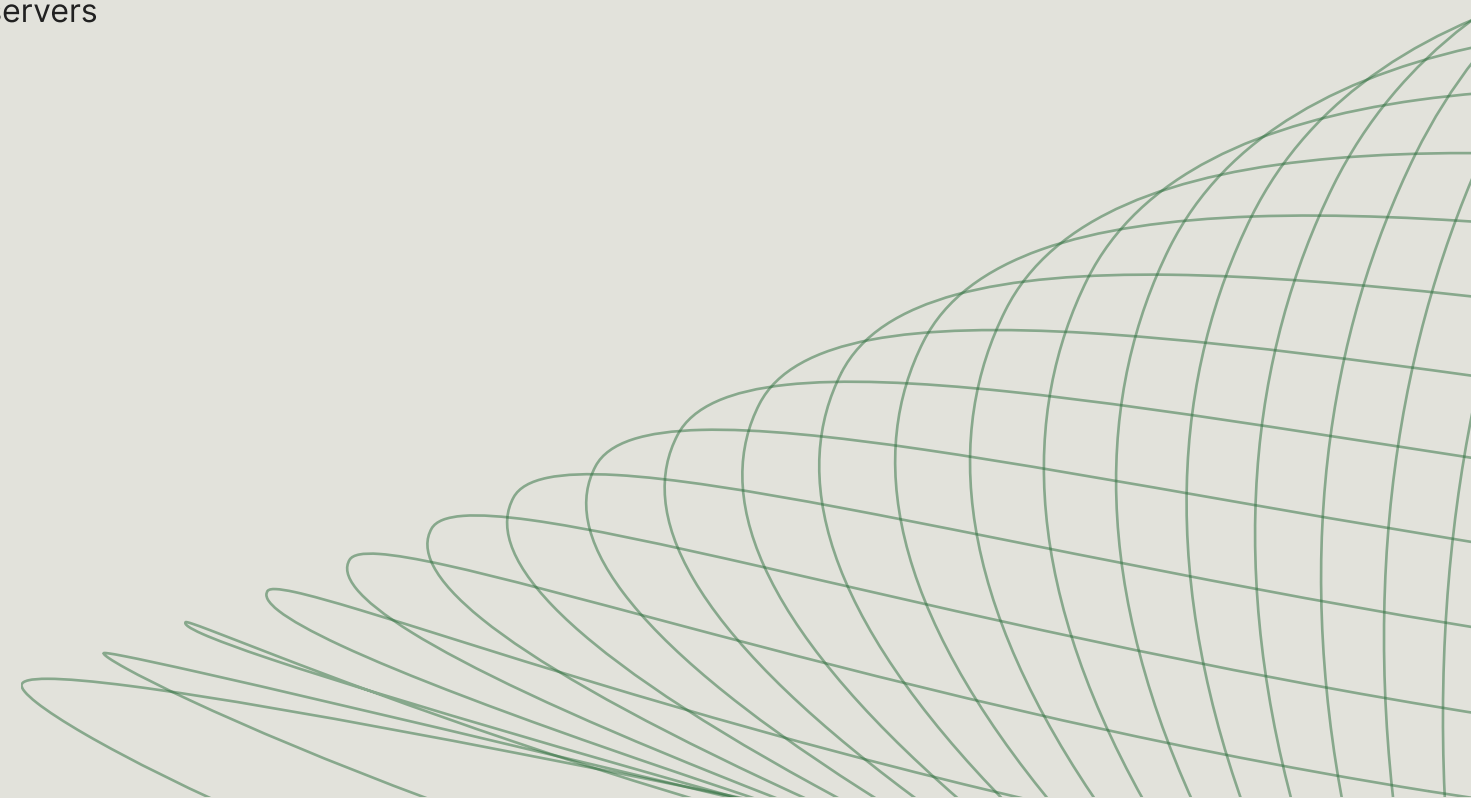
STACKLOK

MCP Maturity Model for Financial Services (2026)

Identify where your firm is today, and drive deeper adoption of AI agents and MCP servers



Contents

Introduction

The Model Context Protocol (MCP) represents a fundamental shift in how AI systems integrate with enterprise tools and data sources. Introduced by Anthropic in November 2024, MCP standardizes how AI systems connect to external services and data, extending the capabilities of AI models, agents and assistants.

MCP has evolved rapidly. Tens of thousands of MCP servers have been published, from individual developers through to some of the world's largest enterprises. MCP has evolved from proposal to fully governed protocol with a formal structure, release cadence and roadmap.

While it is still early days, technology leaders in banking, wealth management, and insurance are embracing MCP now because they recognize the opportunity for competitive advantage. In an industry defined by intense regulatory scrutiny, fiduciary obligation, and sensitive data, they are multiplying the value of their AI agents and assistants, and moving pilots into production to drive measurable outcomes.

The MCP Imperative for Financial Services

There is another reason for financial services leaders to embrace MCP: your employees are already using MCP servers, and that is creating risk for your institution. Inside most banks, asset managers, and insurers, MCP adoption is not a controlled, top-down initiative. With thousands of MCP servers now publicly available, it is easier than ever for a developer, quant, or analyst to wire an AI agent directly into core banking systems, trading and risk platforms, policy administration systems, or customer data. This creates a growing operational, security, and regulatory gap.

This is the shadow AI problem: unsanctioned MCP servers running on local machines, and unknown tools reaching into systems that hold PII, account data, transaction histories, material non-public information, and regulated records. Shadow AI emerges as people reach for whatever helps them move faster. Unless you provide a safe, governed alternative, this usage continues invisibly, without oversight, and outside the controls your regulators expect you to maintain.

You can change the equation by standardizing the way your teams (and their AI agents and assistants) access context behind your firewall. Rather than blocking MCP adoption (an approach that inevitably fails), institutions must offer a governed MCP platform that delivers the same speed and flexibility users want, but with the security, auditability, and control that satisfy both your security function and your regulators. The imperative is clear: adopt MCP intentionally, or inherit an ungoverned landscape of ad-hoc integrations that put customer data, regulatory standing, and operational resilience at risk.

MCP Before and After: A Financial Services View

A common question we receive from institutions at the earliest stages of the maturity model is, "What will we be able to do with MCP that we cannot do without it?"

Today, most AI agents and assistants show up like an intern. They are capable, but they know nothing about your institution (your customers, your products, your policies, your regulatory obligations) and so they require constant supervision to produce useful work. That is why roughly 46% of AI pilots are currently failing, and why so many leave end users and internal champions frustrated.

Now consider what it would be like to operate as an MCP-native institution. Some concrete examples of what becomes possible:

- **Relationship managers and advisors** will have assistants that pull a client's full position across accounts, products, and prior interactions, surface suitable next-best actions within compliance guardrails, and draft client communications and meeting prep, with every action logged for audit and supervision.
- **Operations and risk teams** will have agents that reconcile transactions across systems, flag exceptions and potential breaches before they escalate, summarize exposure across portfolios in real time, and update case-management systems automatically, freeing analysts to focus on judgment rather than data gathering.
- **Contact-center and servicing teams** will know a customer's entire history with the institution (every account, claim, transaction, and prior contact) and resolve queries with fully personalized, fully compliant solutions, increasing first-contact resolution while maintaining a complete audit trail.
- **Developers and engineering teams** will ship faster with agents that raise issues, summarize context from internal systems, and propose changes for approval, all while consuming only the data and tools their role and entitlements permit.
- **All of the above** happens under your watch and within the full control of your private cloud or on-premises environment. Every interaction aligns with your governance, data residency, and regulatory obligations, with the identity passthrough and audit trail an examiner would expect.

In short, the promise of AI in financial services (personalization within compliance, faster and better-governed decisions, automated operations) has a massive dependency on context acquisition and optimization, and therefore, on MCP.

MCP FAQ

Before we get to the Maturity Model, let's address a few common questions. We assume you are reading this because you are interested in MCP, but we will not assume you have pushed in all your chips.

Why should I bet on MCP rather than other protocols like A2A (Google) or ACP (IBM)?

At present, MCP is the only protocol with real momentum. As of December 2025, more than 30,000 MCP servers had been published. Major providers that financial institutions rely on, including Salesforce, Databricks, ServiceNow, GitHub, Atlassian, and others have introduced MCP servers but have not invested in competing protocols. MCP is the only real game in town.

Can't we avoid MCP by tuning agent .md files or enabling Anthropic's "skills"?

Tuning agents via .md files and skills are low-friction steps towards better performance, but neither gives an agent real, valuable context. A .md file can instruct an agent on your coding or content style, but it does not connect it to the account, customer, and transaction data behind your firewall, nor does it enforce the entitlements your regulators require. Skills provide specific executables, but they are not portable across agents or discoverable across your organization, so they serve individual users, not the enterprise.

What are the security risks of adopting MCP, given reports of tool poisoning and other attacks?

Security is a paramount consideration when you are connecting AI systems to core banking, trading, risk, and customer systems. When the MCP spec was first introduced, the security approach was overly simplistic. Since then the spec has been revised to introduce OAuth, and enterprise MCP platforms have come to market with the advanced security and identity capabilities a regulated institution requires.

If you have other questions, please ask. We have an active Discord community at https://discord.gg/stacklok, or email enterprise@stacklok.com and we will respond quickly.

The Financial Services MCP Maturity Model

This Maturity Model gives banks, wealth managers, and insurers a framework to assess their current MCP adoption level and chart a path toward becoming an MCP-native institution. We start with a summary of the four stages, with both technical and process markers for each:

MCP STAGE	Experiment & Prepare	Build Pilots & Capabilities	Scale MCP Infrastructure	MCP-Native Institution
DEPLOYMENT	Local UI / CLI with containers	Local UI / CLI with containers	Local and K8s Operator	Service mesh
FOCUS	Discovery & learning	Pilots & deployment	Platform & governance	Innovation & resilience
PROCESS MARKERS	Individual dev / quant / analyst experiments Ungoverned use Off-the-shelf integrations with current agents	Team-led pilots (ops, risk, servicing, dev) AI pilot success metrics defined Use-case-specific pilots	Enterprise-led deployments across LOBs / entities Governance & model-risk frameworks Context acquisition strategy	Enterprise-wide context infrastructure AI-embedded client & advisor experiences Asynchronous, supervised agentic workflows
TECHNICAL MARKERS	MCP servers on personal laptops No standardized connection practices No containerization, RBAC or audit trail Secrets stored locally / in plain text	MCP servers deployed in containers Standardized authentication (OAuth/OIDC) Basic monitoring and logging Secrets managed via Vault / 1Password	MCP servers deployed via K8s Consistent RBAC across entities / namespaces Network policies & egress controls enforced Observability stack & SIEM integrated	Autonomous, supervised orchestration Hybrid registries: public, private, SaaS Identity passthrough & token exchange Policy engine evaluates every action

On the following pages, we examine each stage in more detail, including the risks of stalling, the metrics that signal readiness to advance, and the roles you will need in place.

Stage 1: Experiment & Prepare

Institutions at Stage 1 are just beginning to explore what MCP can do. Early adopters are usually developers, quants, or analysts experimenting with public MCP servers or simple local integrations to accelerate their work. Adoption is organic, uncoordinated, and largely invisible to IT, security, and your second line of defense, creating the first signs of shadow AI risk. This stage is defined by the realization that AI agents become dramatically more useful once connected to real institutional systems and data.

Process Markers

- Individual developers or analysts run MCP servers locally against systems like internal wikis, ticketing, market data, or account data, without visibility
- Multiple AI tools are used inconsistently across engineering, operations, risk, and front-office teams
- No standard for authentication, secrets, or connectivity to core and customer systems
- Security, compliance, and model-risk functions learn about AI usage ad hoc, often after a data-handling or control concern is raised
- Early wins exist (a faster reconciliation, a useful research summary) but are isolated and cannot be repeated across teams

Technical Markers

- MCP servers run on laptops or personal sandboxes
- Basic filesystem or public MCP servers used for experimentation
- No containerization, network isolation, RBAC, or audit trail
- Credentials to institutional systems (CRM, core banking, market data) stored locally instead of an enterprise vault
- No standardized connection patterns for AI tools

Risks if You Stay in This Stage

- Shadow AI usage expands faster than IT and the second line can understand or govern
- PII, account, transaction, or material non-public information is accessed or moved without visibility, creating GDPR, GLBA, and supervisory exposure
- Teams duplicate work because no shared registry or patterns exist (the same core-banking or Salesforce integration rebuilt repeatedly)
- Early AI wins stall because they cannot pass security or model-risk review to reach production

Success Metrics (to complete Stage 1)

- Inventory of all known MCP usage across the institution
- At least 2–3 high-value MCP-assisted workflows identified (e.g. reconciliation, research summarization, client-history lookups)
- Baseline security, data-handling, and model-risk requirements agreed with InfoSec, compliance, and the model-risk function
- Decision made on deployment path (local + containerization vs. centralized)

Required Roles

- Developer / quant / analyst experimenting with MCP
- Security, compliance, and model-risk partners providing initial review guidance
- Program sponsor or AI / innovation lead

Stage 2: Build Pilots & Capabilities

In Stage 2, institutions move beyond experimentation and begin deploying MCP servers for real workflows and early production use cases, such as a servicing assistant, an operations reconciliation copilot, or an internal research agent. Teams start building custom MCP servers, standardizing connection patterns, and establishing security and supervisory controls. Pilots demonstrate clear productivity gains, but each team is still operating semi-independently. This stage marks the transition from isolated experiments to repeatable, policy-aligned capabilities.

Process Markers

- Teams across operations, risk, servicing, and engineering are running MCP-based pilots for real workflows
- Early internal champions are emerging and sharing best practices
- Security and compliance have begun formalizing policies for AI access to customer and regulated data, but lack automation
- Leadership recognizes MCP's potential and expects validated outcomes that can withstand audit and model-risk review

Technical Markers

- MCP servers deployed in containers (Docker, Podman), not on laptops
- Standardized authentication established (OAuth/OIDC) against your IdP (Okta, Entra ID, Ping)
- Reusable MCP configurations for dev/test environments
- Basic monitoring and logging instrumented (stdout, container logs)
- Secrets to institutional systems managed through Vault, 1Password, or Kubernetes Secrets

Risks if You Stay in This Stage

- Pilots remain siloed by line of business or legal entity and fail to converge into a platform strategy
- Security and compliance fatigue grows due to inconsistent policy implementation across teams handling regulated data
- Operational burden increases as more pilot servers are added
- Fragmentation makes it harder to migrate later to centralized governance, especially across entities and jurisdictions

Success Metrics (to complete Stage 2)

- ≥ 5 pilot-grade MCP servers deployed with consistent container standards
- Reusable connection templates and permission profiles created (e.g. a read-only client-lookup profile)
- MCP-assisted workflows producing measurable productivity or risk-reduction impact
- Ability to deploy a new MCP server in < 1 day (including configuration + testing)

Required Roles

- Platform engineer / DevOps practitioner
- Security architect (MCP permission profiles, secrets and data-handling policies)
- Compliance / model-risk partner reviewing pilot controls
- Pilot workflow owners (operations, risk, servicing, engineering)
- AI program manager tracking reuse and outcomes

Stage 3: Scale MCP Infrastructure

Stage 3 represents the shift from team-level pilots to an enterprise-wide platform. MCP servers are centrally orchestrated, governed, and deployed at scale through Kubernetes, CRDs, and automated policies. Multiple lines of business and legal entities now rely on MCP-powered workflows, and the platform team is focused on reliability, observability, and security enforcement, including the audit and identity controls examiners expect. MCP becomes a shared, supervised organizational capability rather than a niche tool.

Process Markers

- Multiple lines of business and legal entities are requesting access to MCP-powered workflows
- Pressure increases to provide a governed, reliable, enterprise-wide platform that satisfies the three lines of defense
- AI assistants need MCP toolchains to support cross-system workflows (positions + transactions + customer)
- Teams begin expecting production-grade SLAs and resilience commitments for MCP availability

Technical Markers

- MCP servers orchestrated via Kubernetes Operator or similar automation
- RBAC applied consistently across namespaces, teams, and legal entities
- Network policies enforced; egress to customer, payment, and trading systems is controlled
- Centralized registry or catalog of MCP servers introduced
- Observability stack deployed (OpenTelemetry, Prometheus, log aggregation) and feeding your SIEM
- GitOps used for MCP server CRDs and configuration updates; end-to-end identity passthrough in place

Required Roles

- Platform engineering team (2–5 FTEs depending on scale)
- Security engineering (policy automation, anomaly detection on regulated-data access)
- SRE / observability engineer
- AI enablement lead (user training & workflow design)
- Governance committee spanning the three lines of defense (security, platform, compliance, model risk, and business stakeholders)

Stage 4: MCP-Native Institution

At Stage 4, MCP becomes part of the institution's core fabric. AI assistants orchestrate complex workflows across front, middle, and back office, using governed context to automate tasks that once required human coordination, always within supervisory and entitlement boundaries. Business teams build and adapt their own assistants, while security, compliance, and platform controls ensure safe, auditable execution. MCP is no longer perceived as AI tooling; it is embedded infrastructure enabling continuous innovation, faster decisions, and operational resilience.

Organizational Symptoms

- AI-driven workflows are embedded in daily operations across front, middle, and back office
- Multi-MCP assistants automate recurring tasks with appropriate human-in-the-loop supervision
- MCP usage is no longer considered "AI"; it is standard, governed organizational plumbing
- AI governance and platform operations run as a mature, continuous program aligned to your model-risk framework

Technical Markers

- Autonomous, supervised orchestration across MCP servers (agent-led or platform-led)
- Hybrid registries combining public, private, and SaaS MCP servers (Salesforce, Databricks, ServiceNow, and internal)
- Dynamic permissioning and token exchange with least privilege, scoped per user, per entitlement, and per task
- Cost, performance, and context optimization tooling integrated
- Behavioral guards and policy engines evaluate every AI action before execution, critical when agents touch positions, payments, and customer data
- Multi-model routing for agents across LLM providers, with full audit of model and data lineage

Risks if You Stay in This Stage (plateau risk)

- Innovation slows if the platform is not continuously updated as the MCP ecosystem evolves
- Governance drifts if AI oversight is not maintained as a continuous, examinable program
- Competitors who reached MCP-native status compound their advantage in client experience and operating efficiency
- Technical debt and unmanaged agent sprawl accumulate, increasing operational and audit risk

Success Metrics (steady-state indicators)

- Majority of mission-critical workflows rely on supervised MCP orchestration
- Lines of business deploy or modify their own assistants within enforced guardrails
- ≥ 95% audit log coverage across all MCP interactions, with identity passthrough end to end
- Time to onboard a new team with MCP-enabled workflows: < 1 week
- Demonstrable revenue, efficiency, or risk-reduction gains attributable to MCP (advisor productivity, reconciliation throughput, loss avoidance)

Required Roles

- Platform & SRE team operating MCP as core infrastructure
- AI governance board ensuring responsible use of customer and regulated data
- Business workflow architects building and sharing assistant templates across lines of business
- Security and compliance analysts monitoring AI-enabled interactions and policies

Enterprise MCP Platforms

By now it is apparent that to advance through the stages of the Maturity Model, your institution will need an MCP platform. Today, an MCP platform must offer at least four core components:

1 Registry

The foundation of a platform is a catalog of trusted MCP servers, including official upstream servers, public vetted servers, and internally developed servers connecting to your core systems. All servers should eventually be stored and managed centrally for full control, visibility, and supervisory oversight. The better registries let administrators permission and pre-configure every MCP server.

2 Runtime

Institutions must insist on running MCP servers in their private cloud or on-premises environment; customer and regulated data must stay within your boundary, in line with data-residency obligations. A hardened runtime lets you containerize and deploy MCP servers to a Kubernetes cluster, using Kubernetes to orchestrate access policies, network endpoints, and more.

3 Gateway

The gateway ensures the integrity of your MCP estate. Most gateways manage security with a basic implementation of MCP-spec'd OAuth. Enterprise-worthy solutions push further and include federated token exchange, full IdP integration, end-to-end identity passthrough, and OTEL-driven analytics. Every user and AI agent has a single, secure, audited endpoint to access the context they need. In financial services, this is also where you enforce entitlements and contain the blast radius of any single integration touching customer or trading systems.

4 Portal

For enterprise-wide adoption, it has to be dead simple for end users (especially non-technical knowledge workers like advisors and operations staff) to discover the context they need and install MCP servers with a single click. An intuitive portal is a key element of any MCP platform.

Now that we have defined a platform, let's consider the decision to build vs. buy, and, if buying, the criteria to weigh.

Build or Buy an MCP Platform

If you have decided you need an MCP platform, the next decision is whether to build your own (proprietary or assembled point solutions) or buy a complete platform. Here are the criteria to consider:

Time-to-value	If you face limited internal pressure to stop shadow AI and centralize control (a 6+ month horizon), you can consider building. But in financial services, where shadow AI is already touching regulated data and examiners expect demonstrable controls, the pressure to deploy a governed solution quickly usually makes buying a proven solution the obvious choice.
Competency	If your engineering organization has spare capacity and genuine platform-building sensibilities, building is an option. If your teams are competing for resources or skew more towards business applications and services than infrastructure platforms, as most institutions do, you should likely buy.
Ongoing capacity	It is early days for MCP; both the protocol and the ecosystem are evolving fast. Even with resources at the outset, you will need ongoing capacity and an appetite for constant change to maintain a homegrown platform, and to keep its controls current with evolving regulatory expectations. If that is unlikely, a self-built platform will struggle, and you will end up buying anyway.
Operational complexity	If you expect a small number of MCP servers for a defined set of departments, building is feasible. If you anticipate pursuing MCP-native status (deploying hundreds of servers across lines of business and entities, automating lifecycle management, and meeting resilience and audit requirements) it is time to buy a leading solution.

Criteria for Buying an MCP Platform

Most institutions will choose to buy, given the pressure to minimize time-to-value and maximize control. The MCP market is noisy and many platforms sound the same. Here are the criteria we have seen matter in practice:

Track record

Many entrants in the MCP ecosystem are vibe-coded gateways from first-time founders. A few providers have solutions running in production with large, regulated enterprises, led by people who have long served demanding institutions. They understand enterprise requirements, audit expectations, and complexity. Look for pedigree and ask for references, ideally from other financial institutions.

Completeness

An MCP platform includes four core components: registry, runtime, gateway, and end-user portal. A provider offering only one or two will force integration work or proprietary effort onto your team, extending time-to-value. At present, only a small number of providers offer all four components at an enterprise standard.

Security

The MCP ecosystem faces both internal (shadow AI) and external (malicious actor) threats. For an institution connecting agents to customer PII, account and transaction data, and trading systems, a strong security posture is the most critical capability of all. Minimum viable offerings address authentication via OAuth. Insist on more: federated token exchange, end-to-end identity passthrough, SSO and IdP integration, OTEL analytics and SIEM export, secrets management, and cryptographic server provenance checks.

Openness

Ensure any solution you buy has a sustainable future. Many MCP providers are startups; you can weigh investor strength and runway, but more confidence comes from an established open source core. An underlying open source project offers access to innovation, a path forward, and the ability to evaluate before committing, which de-risks the procurement conversation. Look specifically for projects with external maintainers as an indicator of momentum, relevance, and longevity.

Stacklok Enterprise MCP Platform

Stacklok offers a complete Enterprise MCP Platform that simplifies the deployment and management of MCP servers. Fortune 500 enterprises, including regulated financial institutions, are using this platform in production to move through the Maturity Model and deliver more return on their AI investments.

The Stacklok Enterprise MCP Platform includes all four core components referenced above. These components are tightly integrated, and integrations extend to your existing authentication, observability, secrets management, and other solutions.

Deployment Modes

Stacklok's platform is available in three modes to suit different maturity stages:

UI

Desktop application for individual developers and analysts to discover, deploy, and manage MCP servers locally with a user-friendly interface.

CLI

Command line interface for quick deployment with advanced features like custom permissions and telemetry.

Kubernetes Operator

Enterprise-grade operator for teams to run MCP servers in multi-user, multi-entity environments with centralized management and security controls.

Key Enterprise Features

Secure by Default

Every server runs in isolated containers with only necessary permissions. Secrets managed securely, never in plaintext, which is critical when agents reach customer and trading systems.

Declarative Management

MCP Server Custom Resource Definitions enable GitOps workflows and infrastructure-as-code practices.

Observability

OpenTelemetry and Prometheus metrics bridge the monitoring gap in MCP deployments, with usage attributable by user, team, and entity, and export to your SIEM.

Enterprise Authentication

OAuth/OIDC SSO integration, end-to-end identity passthrough, secure token exchange, and audit logging that meets GDPR, GLBA, and SOX requirements.

Multi-Namespace Support

Cluster-wide or namespace-scoped deployment modes following the principle of least privilege, well suited to multi-entity institutions.

Kubernetes-native

A Kubernetes operator lets you use familiar patterns and declarative practices as you scale your MCP footprint.

For more information about the Stacklok Enterprise MCP Platform, visit stacklok.com, email **enterprise@stacklok.com**, or engage us via Discord at **https://discord.gg/stacklok**.

To try the ToolHive open source project, check out our GitHub repo: **https://github.com/stacklok/toolhive.**