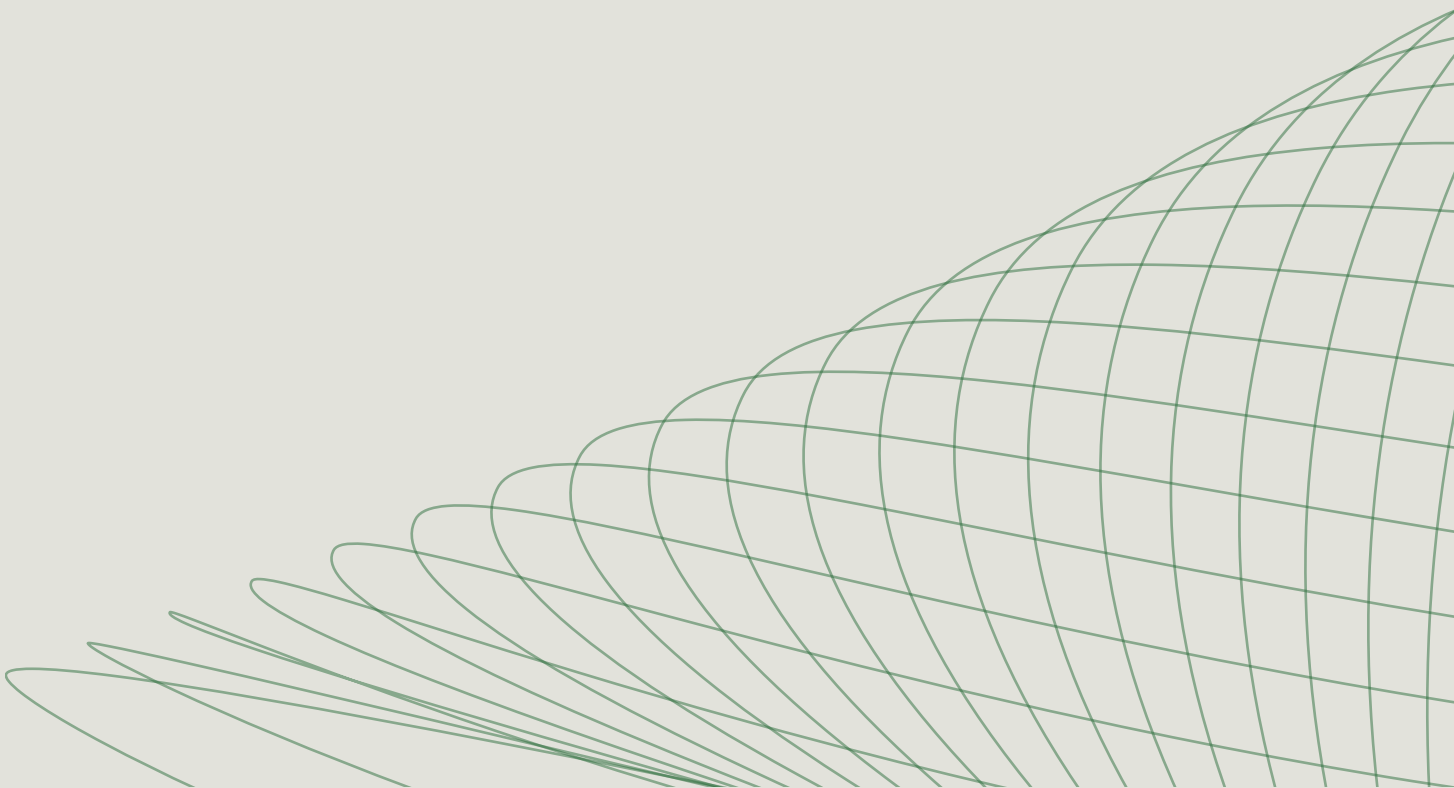
STACKLOK

CASE STUDY:

Global 2000 Software Category Leader



Contents

From Context Problem to Context Platform in Weeks

A Global 2000 software category leader had encountered a conundrum. Individual software developers in their organization had embraced Model Context Protocol (MCP) servers; they were using public MCP servers to greatly increase their productivity. However, there was no central visibility or control of those servers, and the potential for shadow AI introduced real security risks. So the developer experience team at this enterprise was tasked with ensuring developers could continue to use MCP servers, but with proper governance and security at increasing scale.

A few weeks later, the developer experience team had deployed a complete Enterprise MCP Platform. They had put an end to shadow AI, built-out a central registry of servers (upstream, public and internal) and given 500 employees a single endpoint to securely access the right context at the right moment.

Setting the Stage

In the spring of 2025, software developers at this Global 2000 software company started to experiment with MCP servers. At first, those MCP servers were installed in a developer's client of choice (Cursor, Claude Code, etc.) and run on their desktop. But increasingly, developers began using public, hosted servers. There were immediate benefits; developers accepted a higher percentage of AI coding assistants' completions. But there were also real risks of shadow AI becoming pervasive. The company had no standardized MCP server connection practices, no audit trail, nor visibility … and secrets were occasionally being stored in plain text configuration files.

Building in the Open

It was at this point that the developer experience team stepped in. They were tasked with simultaneously increasing both developer freedom and security controls. It wasn't immediately obvious where to start, and so the team began combing the landscape of MCP solutions. An early discovery was the open source project, ToolHive (developed by Stacklok and maintained by Stacklok and Red Hat), which the developer experience team deployed. As they dug in, the team encountered questions and generated ideas, and the natural outlet was Stacklok's discord channel. Here, the developer experience team encountered a group of experts who were quick to engage and support the team's efforts. This set the tone for a fast-forming collaboration.

The Goals

The developer experience team at this Global 2000 software category leader formalized a few clear goals for their initiative. First, they wanted to deploy an MCP platform for their organization. Rather than assemble point solutions, they sought something that was complete, so that it would be easy to maintain over time. They specifically wanted to operate a central, cloud-native model. Second, they knew they needed to address developer demand for MCP servers first, but eventually scale to all knowledge workers at the company. Third, they wanted to increase the efficiency of developer prompts to reduce token usage and costs.

After reviewing the landscape of solutions, it became clear that the Stacklok Enterprise MCP Platform was the only platform with proper security controls. And so the developer experience team set about working directly with Stacklok to deploy all four core elements of the platform in rapid succession: registry, runtime, gateway and portal.

Curated Registry

The developer experience team worked with Stacklok to deploy a central registry that included official upstream MCP servers, vetted public servers and internal servers. The emphasis on upstream compliant servers allowed the team to begin capturing MCP server metadata, helping them understand developer use cases and protect the integrity of their MCP estate.

The developer experience team shared with Stacklok their vision for a registry server that could manage the entire estate ... which Stacklok quickly built and rolled out both as part of their Enterprise MCP Platform and the open source ToolHive project.

Within a few weeks, the developer experience team was ready to share this centrally managed and curated registry with their developers. At this stage, the developer experience team could safely block use of any unauthorized MCP servers and put an end to shadow AI.

Hardened Runtime

As the developer experience team considered how to ensure consistency at scale, it became clear that they'd want to containerize every MCP server and centralize MCP deployments via Kubernetes. The Stacklok runtime includes a Kubernetes operator that the team began using to orchestrate access policies, network endpoints and other aspects of their growing MCP estate.

At the end of the day though, the most important factor for the developer experience team was running MCP servers in their private cloud; keeping their data in their environment was paramount to project goals.

Secure Gateway

In the words of the developer experience team, "identity is a respected aspect of our platform expression." While that might sound a little conceptual, the team was highly practical in their approach to a gateway; this was most evident in their use of Stacklok's simple integrations, an example of which was Okta, enabling SSO for their workforce.

Stacklok's platform also allowed the developer experience team to give every user a single endpoint through which to access MCP servers. This covered the obvious 1:1 connections between client and server; however, the team was also early to adopt Stacklok's concept of a virtual MCP server (vMCP), allowing them to group together multiple MCP servers (and their tools) and map them to a specific team or task.

The gateway also positioned the developer experience team to meet another of their project goals: reducing token usage and costs. As avid users of MCP servers, the concern was that context windows were being filled with low-value tools and their corresponding metadata. Stacklok's MCP Optimizer offered an elegant solution with a tool calling function that is reducing token usage by more than 85% and improving model performance (by reducing distractions).

Intuitive Portal

At time of writing, the developer experience team has rolled the platform out to more than 500 software developers at the company. In the near future, they will expand this rollout to all knowledge workers. And that's where the intuitive portal comes into play. Stacklok's Enterprise MCP Platform makes it easy for the team to permission MCP servers based on role, IP-address, project and other fine-grained criteria. And the team can preconfigure each MCP server, so that knowledge workers can discover and deploy the right MCP server(s) with a single click.

Differentiated Value

Across their interactions with Stacklok's team and technology, the developer experience team at this software category leader pointed to three drivers of differentiated value:

1

Platform sensibilities

As the developer experience explored the MCP landscape they largely ran into point solutions and/or platform promises that lacked production capabilities. The experience with Stacklok was different. In their words, "The Stacklok team is full of platform builders. Everything they architect is thoughtful and extensible. They understand what enterprises need to run in production and they've delivered a platform that meets our requirements."

2

Flexibility

Since this global 2000 software company intends to put MCP servers at the fingertips of all employees, Stacklok's ability to engage via CLI, UI, API and Kubernetes operator offered meaningful flexibility. Stacklok enabled the team to centrally curate upstream + public + internal servers, and even supported both local and cloud native deployments. Given the fast-changing MCP landscape, the developer experience team prized the flexibility embedded in Stacklok's platform.

3

Advanced security

Countless MCP solutions promise "enterprise-grade" security, but when put to the test, the developer experience team found that most just stuck to the MCP spec, which lacks real enterprise bells and whistles. Conversely with Stacklok, the developer experience team were early users of OTEL support, and have pushed beyond basic authentication with Okta integration and federated token exchange. As stated above, Stacklok has proven to understand what enterprises need to answer CISO questions and secure a context platform in production.

Business Impact

The developer experience team discovered ToolHive in the spring of 2025. Within 30 days they were experimenting on the open source solution. Weeks later, they had deployed all four core elements of the Stacklok Enterprise MCP Platform and rolled the platform out to 500 developers at the company. This represented rapid time-to-value, providing developers with simpler access to MCP servers. Next up is an expansion to all knowledge workers at the company as they seek competitive advantage from context and evolve into an AI-native enterprise.