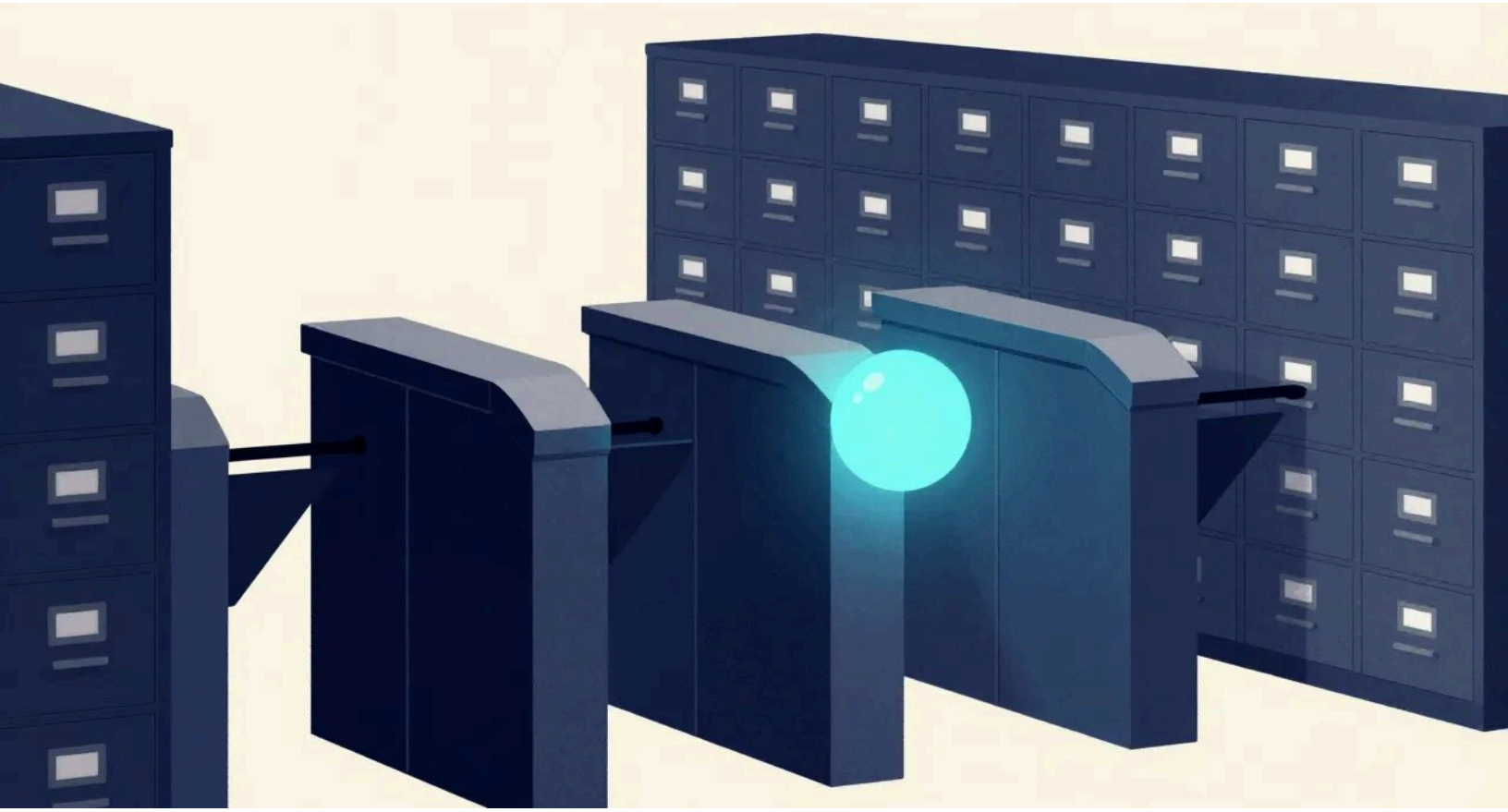


EXECUTIVE PAPER · AI

MCP Servers Give AI Governed Access

Every AI assistant and agent in the company now wants a connection to email, files, tickets and databases. Executives should decide once how those connections are built, approved and watched, before each team wires its own.

AUTHOR

Chris Watkins

SERVICE

AI

FOR

CEO, CTO, CISO · 12 min read

AT A GLANCE

The problem. The Model Context Protocol (MCP) became the common way to connect AI tools to business systems in about a year. Anthropic counted more than 10,000 active public MCP servers by December 2025. Internet scans in 2025 and 2026 found thousands of them reachable by anyone, In 2025 researchers showed a prompt injection pulling private code through an AI connector, a major software vendor exposed customer data across tenants, and the first malicious MCP package appeared.

Why it persists. An MCP connection takes minutes to set up on a laptop, and the protocol leaves authorization optional. Each team picks its own servers from public package registries and hands them whatever credentials are at hand. Nobody owns the list.

The recommendation. Treat AI integrations as company infrastructure. Run a small catalog of approved MCP servers behind one gateway, each with its own identity, the narrowest permissions that do the job, complete logs, and a named owner. Block everything else, and review the catalog every quarter.

The Decision in Front of You

AI assistants are only useful when they can reach company data. A chatbot that cannot read the ticket queue, the contract folder or the customer database answers from guesswork. So every AI product now ships with connectors, and most of them speak the same protocol.

The decision is who builds and controls those connectors. One path lets each team, vendor and developer connect AI tools to systems as they see fit. The other runs every AI connection through a governed layer the company owns: a catalog of approved MCP servers, a gateway in front of them, and the same identity, logging and ownership rules that apply to any other system with access to sensitive data.

The first path is faster for about a quarter. After that the company has dozens of unlisted integrations holding live credentials, written by people who may have left, and no log of what any AI tool read or changed. A companion paper covers how to limit what a single agent may do. This paper is about the plumbing every agent shares.

What's Happening

One protocol won, fast

Anthropic released MCP as an open standard on November 25, 2024, to replace one-off integrations between AI assistants and the systems where data lives [1]. OpenAI added support in March 2025, starting with its Agents SDK [2]. By December 2025 Anthropic reported more than 10,000 active public MCP servers, 97 million monthly downloads of the software kits used to build them, and support in ChatGPT, Cursor, Gemini, Microsoft Copilot and Visual Studio Code [3]. That month Anthropic gave the protocol to the Agentic AI Foundation, a fund under the Linux Foundation co-founded with Block and OpenAI [3].

A shared standard helps buyers: one integration serves many AI tools, and security teams have one protocol to control. The specification is also clear about where its protection stops. Authorization is optional. Servers that run on a user's own machine are told to take credentials from the local environment, which in practice means whatever tokens and keys that user's machine holds [4]. Where authorization is used, servers must accept only tokens issued for them and must never pass a client's token through to another service [4]. Those rules help only when someone checks that each server follows them.

Microsoft reached the same conclusion when it built MCP into Windows. Its May 2025 security design names tool poisoning, credential leakage, command injection and an unvetted public registry as threats. Its answer is a registry that admits only code-signed servers with fixed tool definitions and declared privileges, with every call passing through a proxy that enforces policy and keeps an audit record [5].

Exposure is measurable

Researchers at Knostic scanned the internet in July 2025 and found 1,862 MCP servers exposed. They checked 119 by hand, and every one let an anonymous visitor list its tools without logging in [6]. Censys counted 12,520 internet-reachable MCP services on 8,758 addresses in April 2026. The largest group offered data and knowledge tools, including direct database queries, and about 90 advertised tools with names such as run_command or shell_exec [7]. Censys notes some of these are test systems or decoys, so the count is an upper bound on real exposure.

Five incidents from 2025 and 2026

GitHub MCP server, May 2025. Researchers at Invariant Labs showed that a malicious issue posted to a public repository could instruct an AI agent to copy data from the user's private repositories into public view [8]. The agent was doing what it was built to do with the access it had been given. The researchers called it an architectural problem with no simple patch, and recommended limiting each agent session to one repository and using narrowly scoped tokens [8].

Asana, May to June 2025. Asana launched an MCP server on May 1, 2025. On June 4 it found a logic flaw that let users see project data belonging to other organizations, within their own access scope [9]. About 1,000 customers were affected, the exposure lasted over a month, and the server stayed offline until June 17 [9]. No attacker was involved; a tenant-isolation bug in a new integration layer did it.

postmark-mcp, September 2025. An npm package posing as an official MCP server for the Postmark email service ran cleanly through 15 versions. Version 1.0.16, published September 17, added one line that blind-copied every email sent through it to an outside address [10]. It was downloaded 1,643 times before removal. The fix advice was to remove it, rotate any credentials ever sent by email, and search mail logs [10].

mcp-remote and OpenCode, 2025 and 2026. A widely used connector, mcp-remote, let a malicious server run code on the user's computer during the login step. JFrog rated it 9.6 out of 10 in July 2025 [11]. In October 2026 Pillar Security disclosed the same class of flaw in the OpenCode coding agent on Windows, with a proposed score of 9.8. No CVE was issued, so scanners may miss old versions [12].

Grafana and Dolt MCP servers, August 2026. The security vendor Pillar Security reported that both servers would run tools for callers with no valid login, using the server's own service credentials [13]. Both were fixed.

Why the Usual Approach Fails

Most companies are handling MCP the way they handled early cloud accounts and browser extensions. Adoption runs ahead of policy, and policy arrives as a document that nothing enforces.

The local install is the default. A developer or analyst adds a server to an AI client with one line of configuration, pulled from a public registry. It runs with that person's credentials and network access. Security never sees it, which is why OWASP's draft MCP Top 10 lists "shadow MCP servers" as its own risk [14].

Credentials are borrowed. The GitHub case turned on a token that could read every repository the user could reach [8]. The specification allows local servers to take whatever credentials sit in the environment [4]. An AI tool with a person's full access inherits every permission that person collected over years.

Trust is placed in the package. postmark-mcp behaved for 15 releases before it turned [10]. Reviewing a server once at install time misses the update that matters. Microsoft's registry requires tool definitions that cannot change at runtime for this reason [5].

The vendor's server is assumed safe. Asana is a mature company and its flaw sat in its own hosted server [9]. A vendor-run MCP endpoint is a new product with new bugs. The company still needs its own record of what flowed through it.

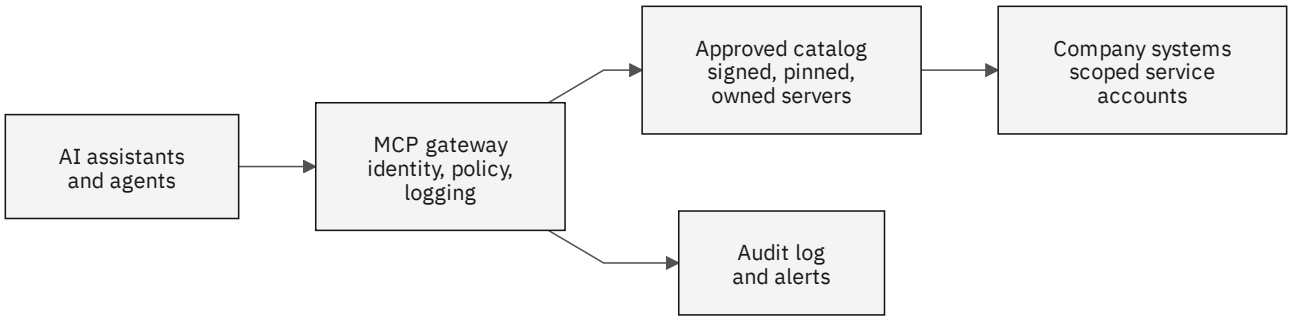
Nothing is logged in one place. When an AI tool reads a contract or edits a ticket, the record sits in the AI client, the MCP server, the target system, or nowhere. OWASP lists missing audit and telemetry as a top-ten risk [14]. Without that record, the Asana advice to "review data accessed through MCP" cannot be followed [9].

NIST's AI Risk Management Framework already asks for the missing pieces. GOVERN 1.6 calls for an inventory of AI systems. GOVERN 6.1 calls for policies covering third-party software and data, and MAP 4 calls for mapping the risks of every component, third-party parts included [15]. MCP servers are those components.

The Framework: One Door for AI

The model is simple to state. AI tools reach company systems through one door, and the company owns the door.

Exhibit 1. One Door: every AI connection passes through an owned, logged gateway



Source: author's framework, informed by Microsoft's Windows MCP security design [5] and the MCP authorization specification [4].

Catalog. Keep a short list of approved MCP servers, ideally fewer than 20 at first. Each entry records the owner, the business purpose, the systems reached, the permissions granted, the data classification, and the pinned version. Build or fork servers for core systems so the code is yours. For vendor-hosted servers, record the vendor's security terms and incident contact.

Gateway. All AI clients connect through one gateway the company runs. It authenticates the person or agent with the company's identity provider, checks the request against policy, and forwards it to the approved server. Clients are configured to reach nothing else. The gateway is where you add rate limits, block a server in minutes, and cut off one user without touching the others.

Identity and scope. Each server gets its own service identity with the narrowest access that serves its purpose. A server that summarizes tickets gets read access to tickets. Tokens are issued for that server alone, as the specification requires [4]. Destructive and financial actions route to a person for approval, as the companion paper on agents describes.

Supply chain. Install servers only from the catalog. Pin versions, verify signatures where they exist, and review the code of every update before it reaches the catalog. Treat the tool descriptions a server sends to the model as code, because the model reads and follows them.

Logging. The gateway records who asked, which tool ran, with what arguments, and what data came back, in a store the AI tools cannot alter. Alerts fire on new tools appearing, unusual data volumes, and calls to tools marked sensitive.

Owner and review. Every server has a named owner who answers for it. Each quarter the owners and security review usage, remove servers nobody uses, and re-approve the rest.

Exhibit 2. What the governed layer would have changed

Risk (OWASP MCP Top 10 [14])	Public case	Control in the One Door model
MCP09 Shadow MCP servers	Thousands of servers open to anonymous callers [6][7]	Clients may reach only the gateway; unknown servers are blocked
MCP02 Privilege escalation via scope creep	GitHub agent read every private repository [8]	Per-server service identity scoped to one purpose
MCP04 Supply chain attacks	postmark-mcp turned malicious in version 1.0.16 [10]	Catalog-only installs, pinned versions, code review of every update
MCP07 Insufficient authentication	Grafana and Dolt servers ran tools for anyone [13]	Gateway authenticates every call before any server sees it
MCP08 Lack of audit and telemetry	Asana customers asked to review data accessed through MCP [9]	Central log of every call, query and response
MCP05 Command injection	mcp-remote and OpenCode login flaws [11][12]	Remote servers reached only through the gateway; client versions managed centrally

Source: OWASP MCP Top 10 (beta) [14]; cases as cited; mapping is the author's analysis.

No control in this table stops a determined prompt injection on its own. The layer limits what a hijacked agent can reach and records what it did, which turns a breach of everything the user could access into a contained, traceable event.

Costs, Risks, and Tradeoffs

Exhibit 3. Tradeoffs of a governed integration layer

Area	Cost or risk	Mitigation
Speed	Teams wait for a server to be approved instead of installing one today	Publish a two-week approval target; pre-approve read-only servers for common tools
Engineering effort	Someone must run the gateway, maintain internal servers and review updates	Start with five to ten servers that cover most use; reuse vendor servers where terms allow
Single point of failure	A gateway outage stops every AI integration at once	Run it with the same availability design as other identity infrastructure
Vendor features	Some AI products only support their own hosted connectors	Contract for audit logs and admin controls; restrict scope where you cannot route through the gateway
False assurance	A gateway does not stop prompt injection inside an approved server	Keep scope narrow and require human approval for destructive or financial actions
When this is wrong	A small firm using one AI product with a few vendor-built connectors gets little from running its own gateway	Use the vendor's admin controls, restrict connectors to an allowlist, and turn on whatever logging the vendor offers

Source: author's analysis.

The standard is still moving. The specification's current revision dates from July 2026, and OWASP's list is still in beta [4][14]. A gateway makes those changes easier to absorb, because there is one place to update instead of dozens.

How to Act

Exhibit 4. A one-year roadmap

Window	Actions	Owner	Milestone
First 30 days	Inventory every MCP server and AI connector in use: AI client configurations on managed devices, vendor admin consoles, and network logs. Freeze new installs outside an interim allowlist.	CISO with IT operations	Complete list of servers, each with an owner or a removal date
First 30 days	Remove any server running with a person's full credentials against a production system.	Security lead with system owners	Zero servers using personal tokens on production systems
First 90 days	Stand up the gateway with company single sign-on. Move the most used servers behind it with their own service identities. Log every call centrally.	Platform engineering lead	80% of AI tool calls pass through the gateway
First 90 days	Publish the catalog and the approval process, with a two-week target. Map controls to NIST AI RMF GOVERN 1.6 and 6.1 [15].	CTO with risk or compliance	Catalog live; first approvals completed within target
First 365 days	Block direct client-to-server connections on managed devices. Add alerts for new tools, unusual data volume and sensitive actions. Run a prompt-injection test against each catalog server.	CISO	100% of AI tool calls through the gateway; quarterly review held four times
First 365 days	Add MCP terms to vendor contracts: audit log access, tenant isolation testing, breach notice.	General counsel with procurement	Terms in every renewal involving an AI connector

Source: author's recommendation.

Questions to Ask Your Team

1. How many MCP servers are running in our company today, and who installed each one?
2. Which of them hold a person's own credentials, and what could those credentials delete or send?
3. If a vendor told us tomorrow that its MCP server leaked our data to another customer, could we list what our AI tools sent and received through it?
4. Where does a new MCP server come from before someone installs it, and who reviews its updates?
5. Can we block one AI connector across the company within an hour?
6. Which AI tools in use can only connect through their vendor's hosted connectors, and what logs does that vendor give us?

7. Who owns the decision to approve a new connection, and how long does approval take?

Closing

AI tools will keep asking for more access, because access is what makes them useful. Each connection is a new path to company data, and in 2025 alone those paths mixed customer records between companies and copied email to strangers. Decide now that every AI connection goes through one door the company owns, with a catalog, a gateway, scoped identities, a complete log and a named owner for each server. Teams keep their integrations, and leadership can say what the AI tools reach and show what they did.

Sources

[1] Anthropic. "Introducing the Model Context Protocol." November 25, 2024. https://www.anthropic.com/news/model-context-protocol.

[2] TechCrunch. "OpenAI adopts rival Anthropic's standard for connecting AI models to data." March 26, 2025. https://techcrunch.com/2025/03/26/openai-adopts-rival-anthropics-standard-for-connecting-ai-models-to-data/.

[3] Anthropic. "Donating the Model Context Protocol and establishing the Agentic AI Foundation." December 9, 2025. https://anthropic.com/news/donating-the-model-context-protocol-and-establishing-of-the-agentic-ai-foundation.

[4] Model Context Protocol. "Authorization." Specification, revision 2026-07-28. 2026. https://modelcontextprotocol.io/specification/2026-07-28/basic/authorization.

[5] Weston, David, Microsoft. "Securing the Model Context Protocol: Building a safer agentic future on Windows." Windows Experience Blog. May 19, 2025. https://blogs.windows.com/windowsexperience/2025/05/19/securing-the-model-context-protocol-building-a-safer-agentic-future-on-windows/.

[6] Knostic. Research blog post on locating internet-exposed MCP servers with Shodan. July 17, 2025. https://www.knostic.ai/blog/find-mcp-server-shodan.

[7] Ellzey, Mark, Censys ARC. "MCP Servers on the Internet." May 27, 2026. https://censys.com/blog/mcp-servers-on-the-internet/.

[8] DevClass. "Researchers warn of prompt injection vulnerability in GitHub MCP with no obvious fix." Reporting Invariant Labs research. May 27, 2025. https://devclass.com/2025/05/27/researchers-warn-of-prompt-injection-vulnerability-in-github-mcp-with-no-obvious-fix/.

[9] BleepingComputer. "Asana warns MCP AI feature exposed customer data to other orgs." June 2025. https://bleepingcomputer.com/news/security/asana-warns-mcp-ai-feature-exposed-customer-data-to-other-orgs/.

[10] The Hacker News. "First Malicious MCP Server Found Stealing Emails." Reporting Koi Security research. September 2025. https://thehackernews.com/2025/09/first-malicious-mcp-server-found.html.

[11] JFrog. Press release on critical remote code execution vulnerability in mcp-remote, CVE-2025-6514. July 9, 2025. https://jfrog.com/press-room/jfrog-security-research-team-discovers-critical-remote-code-execution-vulnerability-hijacking-mcp-remote-clients/.

[12] The Hacker News. "OpenCode Flaw Lets Malicious MCP Servers Execute Code During Sign-In on Windows and WSL." Reporting Pillar Security research. October 2026. https://thehackernews.com/2026/10/opencode-flaw-lets-malicious-mcp.html.

[13] Pillar Security. "MCP Security: Risks, Real Incidents and Best Practices (2026)." 2026. https://www.pillar.security/blog/the-security-risks-of-model-context-protocol-mcp.

[14] OWASP Foundation. "OWASP MCP Top 10." 2025 list, beta release. https://owasp.org/www-project-mcp-top-10/.

[15] National Institute of Standards and Technology. "Artificial Intelligence Risk Management Framework (AI RMF 1.0)." NIST AI 100-1. January 2023. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf.

The views in this paper are the author's own and do not represent any employer. Scenarios are hypothetical. This paper is not legal, financial, or compliance advice.