

Overview

MCP governance

MCP servers give AI agents tools — file access, browsers, deploy keys. Sentilai governs them in real time on every request.

MCP Inventory

Every MCP server seen anywhere — discovered on developer machines by the Sentilai Endpoint, or observed in Gateway traffic — appears in **MCP Inventory** with its sources and last-seen time. You cannot govern what you cannot see; this is the seeing part.

Policy: default → per-server → per-tool

Your tenant sets a **default MCP action** (allow / warn / report / block). A **per-server rule** overrides the default for that server. A **per-tool rule** refines a single tool of a server — for example, block only `github/delete_repo` while the rest of the GitHub server stays allowed. Precedence: tool rule > server rule > tenant default; when multiple servers are involved, the most restrictive wins.

Enforcement is live at two points: the Gateway (requests that declare MCP tools) and the local shim (below) for stdio servers.

The local MCP shim

The Endpoint app wraps each local (stdio) MCP server in a thin shim. Every `tools/call` asks the Gateway for a policy decision first — a blocked tool is rejected with a clear message to the AI agent, while allowed sibling tools flow normally.

Rug-pull tripwire

MCP tools can silently change their descriptions or schemas after you've approved them (a classic poisoning vector, OWASP MCP03). Sentilai pins every tool's definition on first sight and raises a finding when it changes — including for **local** servers, which only the shim can see. Cross-tenant corroboration adds context when the same change hits multiple organizations.

Routed remote endpoints

Register a remote MCP server under an alias and point your tools at `https://<gateway>/mcp/<alias>` instead of the vendor URL — the Gateway proxies the traffic and applies your policy in-path.

Lethal trifecta warning

When your governed tools combine private-data access, exposure to untrusted content, and an exfiltration channel, the MCP Inventory shows a warning naming which tool contributes each leg — one prompt injection away from a leak, and the remedy is one per-tool rule.

Revision #1

Created 2026-08-01 13:06:23 UTC by Sentilai Docs

Updated 2026-08-01 13:06:23 UTC by Sentilai Docs