

Please Stop Giving Your AI a Shell

CLI Tools, MCP, and the Governance You Think You're Buying

CTRL ALT Elite · Fortune 100-caliber infrastructure and operations leadership

Every AI governance debate eventually arrives at a deceptively simple question: should the agent call a vendor CLI like **glab**, **acli**, or the Figma CLI, or should it go through MCP? It is the wrong question. The governance property you actually care about does not come from the transport. It comes from whether the agent's capability surface is **bounded and mediated**. A CLI behind a shell hands the model general compute, which is the opposite of least privilege. A well-designed MCP server exposes a bounded, typed surface. A badly designed one that ships a "run arbitrary query" tool is just as dangerous. The protocol does not confer governance. Your tool design does.

GOVERNANCE DIMENSION	CLI TOOLS (glab, acli, Figma CLI)	MCP
Capability surface	Shell access is near-unbounded: arbitrary binaries, command chaining, curl-based exfiltration, and a direct prompt-injection-to-execution path.	Bounded and typed. The agent can invoke only the operations you explicitly expose, with defined schemas.
Least privilege	The token caps it, but the CLI does everything the API does. Constraining to "create issues, never delete repos" means fragile command-string interception.	Least privilege by construction. The exposed surface is the policy.
Auditability	Unstructured text output, messy to reconstruct. Vendor-side logs help but are not yours to standardize.	Uniform structured events (tool name, typed args, result) across every integration, in one telemetry format.
Credential exposure	Credentials live in the environment where the agent runs, widening the blast radius.	The server mediates. The model never touches raw credentials or a shell.
Central controls	Hard to place authz, rate limits, input validation, and DLP consistently across tools.	One place for authz, approval gates, validation, and output redaction: the server, or a gateway in front of it.
Coverage and cost	Full API on day one, vendor-maintained, zero build. This is the real advantage.	You build and maintain the servers, coverage gaps exist, and the ecosystem is still maturing.

The Bottom Line

MCP is the stronger foundation because it makes least privilege, structured audit, and approval gates first-class and centralizable, which is exactly what maps to a control framework your risk, audit, and legal stakeholders can attest to. But do not confuse the protocol for the control. The pattern mature organizations land on is a hybrid: **MCP as the governed front door**, with servers that may internally call the very same vendor CLIs or APIs. You keep the CLI's coverage and the vendor's

hardening. You keep the shell out of the model's hands. The layer you don't think you need, doing precisely what it is for.

CTRL ALT Elite

reboot@ctrlalteleite.io · 954-516-6909 · ctrlalteleite.io · Fort Lauderdale, FL

Disrupt · Innovate · Elevate

© CTRL ALT Elite. Shareable in full with attribution. The frameworks described are vendor-neutral; product names are illustrative.