From Shadow MCP to Sanctioned MCP

Building an Enterprise Agent Governance Program

Navin Pai | **Archana Rajkumar**

StackGen | SentinelOne

MCP Dev Summit Mumbai · June 15, 2026

YOU ALREADY HAVE AN MCP PROBLEM

~2,000 MCP servers found on the public internet,
every one sampled answered `tools/list` with **zero auth**

79% of IT leaders have found agents deployed outside IT

Only **~24%** of orgs know which agents talk to each other

lean startup → velocity is existential
public security company → scrutiny is existential
same question, very different bosses:

**”What are our agents allowed to do…
and how do we know?”**

2025–26: NOT HYPOTHETICAL ANYMORE

Jun '25 · Asana MCP leaks data across ~1,000 customer orgs

Jul '25 · mcp-remote RCE (CVE-2025-6514) · 437k downloads

Sep '25 · postmark-mcp: first malicious MCP server in the wild. One line of code BCC'd every email

Jan '26 · OpenClaw: 21,000+ agent instances exposed online

May '26 · the NSA publishes an MCP security advisory

Every single one of these passed auth.

AUTH + POLICY CHECKS: NECESSARY, NOT SUFFICIENT

- **Rug pulls**: tool definitions mutate after approval
- **Confused deputy**: valid token, wrong authority
- **Prompt injection**: LLMs can't tell instructions from data
- **Multi-agent chains**: nobody sees the whole graph

stop asking "Is this tool call allowed?"
start asking "Do we know what's running,
what it can do, and what it did?"

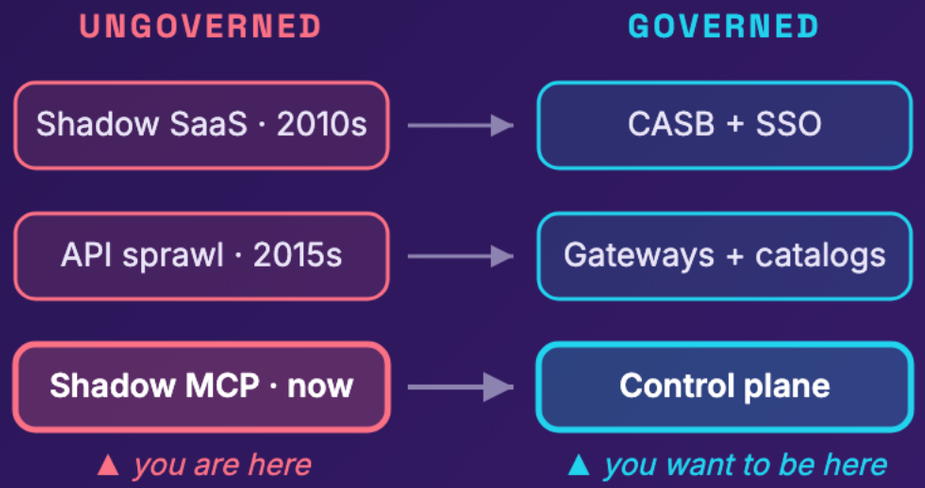
Governance is a program, not a checkpoint.

WE'VE SEEN THIS MOVIE BEFORE

Shadow MCP = **Shadow IT, replayed**
The playbook is known: discover → catalog → broker → monitor
API sprawl → gateways + catalogs. Same arc.

What's genuinely new:

- callers are non-deterministic
- capabilities mutate at runtime
- identity: neither user nor service
- the input is natural language...
  so is the attack



same playbook, new protocol

You don't need a new religion: adapt one you already practice.

A MATURITY MODEL FOR MCP GOVERNANCE

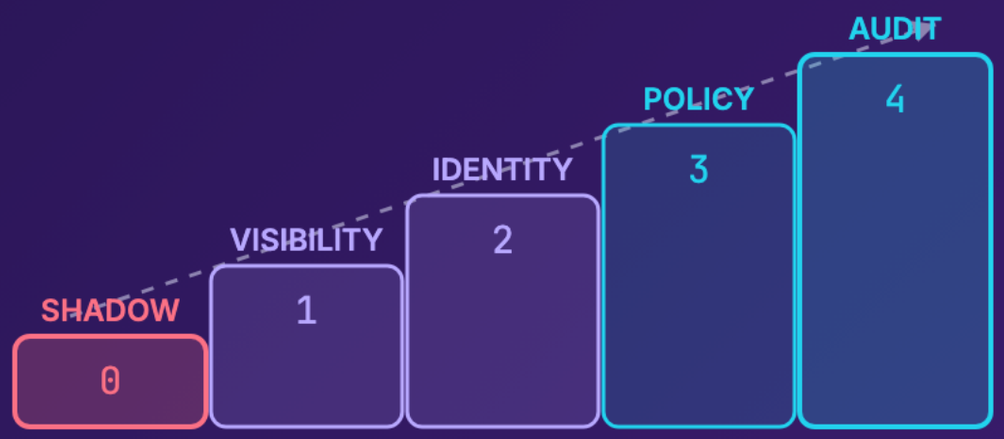
0 · **Shadow**: "We have no idea what's running."

1 · **Visibility**: "We know every agent, server & connection."

2 · **Identity**: "Every tool call is attributable."

3 · **Policy**: "Every tool call passes a deterministic check."

4 · **Audit**: "We can prove what any agent did...
and why it was allowed."



stages overlap: you never "finish" Stage 1

VIEW FROM THE STARTUP

Constraint: **zero security headcount · governance can't cost velocity**

Reality: **our customers are enterprises → governance IS the product**

”Sanctioned in 5 minutes, or developers route around it.”

- **Defaults over gates**: curated registry as the easy path
- **Policy defined once**: in config, enforced wherever agents run
- **Audit logs from day one**: retrofitting is the expensive part

VIEW FROM THE ENTERPRISE

Constraint: **customers, auditors, regulators: defender AND target**

Forrester ’26: a major breach will be attributed to an AI agent.

- **Tier agents by blast radius**: one-size-fits-all backfires
- **Breakglass paths**: governance must survive incidents
- **Watch the endpoint too**: agents are processes…
  EDR-style behavior detection backs the protocol boundary
- **Map to frameworks**: OWASP Agentic Top 10, NIST AI RMF →
  compliance comes for free

SPOILER:
WE CONVERGED ⚡

Two wildly different orgs. The same five primitives.

1 **One control plane**: centralized policy, distributed enforcement

2 **Deterministic checks** at the tool-call boundary: config, not prompts

3 **First-class agent identity**: short-lived, scoped, attributable

4 **Append-only audit**: tied to identity + decision

5 **Curated registry**: the paved road

CENTRALIZE POLICY, NOT TRAFFIC

One policy source of truth, **enforced at every boundary**:

- **Network gateway**: remote MCP servers
- **Managed client config**: IDE & desktop agents
- **Sandbox wrappers**: local stdio servers... no gateway will ever see them
- **Even the OS**: Windows 11 MCP → signed registry + trusted proxy

”We’ve all seen the ESB movie. The gateway that sees everything is the gateway that breaks everything.”

Microsoft Cloudflare Kong IBM PolicyLayer LiteLLM

A startup, a public security co, Microsoft & a dozen vendors built the same thing.
Not coincidence: **the shape of the problem.**

STILL UNSOLVED

- **No signing in the MCP spec**: registry verifies namespaces now, but artifact signing & attestation are still the frontier
- **No identity propagation**: "on whose behalf?" is lost one hop in. The '26 roadmap: enterprise auth via extensions, not core
- **No audit standard**: OTel traces ≠ regulator-grade, tamper-evident audit
- **Tool shadowing**: no namespacing, a rogue server hijacks a trusted one
- **Agent→agent chains** & sampling injection: sampling can now call tools
- **Policy can't judge semantics**: LLM judges, eBPF, honeypots
- **Standards catching up**: NIST agent overlays, India AI Guidelines, EU AI Act (high-risk slipping to Dec '27, provisionally agreed)

Buy or build: your model has to outlive your tools.

START MONDAY

THIS WEEK

Inventory

scan configs, count the surprises

THIS MONTH

Registry

make sanctioned the easy path

THIS QUARTER

Control plane

one policy, every boundary, agent identity

THIS YEAR

Provable audit

what any agent did, and why it was allowed

Shadow MCP is already in your org.

Go look.

Thank You!

From Shadow MCP to Sanctioned MCP: Building an Enterprise Agent Governance Program

Scan to share feedback

In the same boat? Come find us. 😊

navinpai.github.io/mcp-governance