



Governed MCP Stack

Enterprise Architecture Blueprint

How to use this blueprint

This is a working blueprint for exposing enterprise systems to AI agents through governed MCP servers. It is based on the MCP specification and public roadmap (modelcontextprotocol.io), the Agentic AI Foundation announcements, and Linux Foundation primary sources. It is an engineering instrument, not legal or vendor advice. Duplicate this file. Work through the four moves in order — each fits on one page. Use the checkboxes (■/□) to track status, and treat every “done when” as a gate, not a suggestion.

Why 2026 is the year to build

Milestone	What changed
Dec 9, 2025	Anthropic donated MCP to the Agentic AI Foundation under the Linux Foundation — co-founded by Anthropic, Block, and OpenAI, with AWS, Google, Microsoft, Cloudflare, and Bloomberg as platinum members. Vendor-neutral governance: the roadmap is no longer one company's.
Jul 28, 2026	The 2026-07-28 specification makes MCP stateless at the protocol layer — servers run behind standard load balancers on existing Kubernetes — and adds a feature lifecycle and deprecation policy enterprises can plan against.
Adoption	97M+ monthly SDK downloads and 10,000+ active public servers as of December 2025 (Anthropic); ~19,800 servers indexed by community registries by March 2026.

Key fact: The protocol standardizes the plug — it does not standardize the permissions. Per MCP's own 2026 roadmap, standardized audit trails and gateway patterns are still open work items. Identity, policy, approval, and audit are yours to design. That is exactly what this blueprint covers.

The Governed MCP Stack — reference architecture

Objective: One governed pathway between agents and systems of record. Expose each system once, govern it centrally: $N \times M$ integrations become $N+M$.

Layer	What it does	What you own
Agent clients	Claude, Copilot-class assistants, your own agents. They never talk to systems directly.	The client allowlist: which agents exist, and as whom they act.
MCP gateway / control plane	The single checkpoint. Identity (OAuth 2.1, no shared keys) · Registry (what exists, who may call it) · Policy (per-tool RBAC, quotas, rate limits) · Audit (every call logged).	All four responsibilities — whichever gateway product you run.
Per-system MCP servers	One thin server per system of record, exposing a curated toolset: the five tools your use cases need, not the two hundred the API has.	Tool curation, input/output schemas, server identity, a named owner.
Systems of record	ERP, WMS, CRM, data stores — unchanged. That is the point.	Source-system access policies stay authoritative.

Design rule: Servers stay thin; governance concentrates at the gateway. Policy enforced in twelve places is policy enforced in none.

- Every agent-to-system path goes through the gateway — no direct connections anywhere
- Each system of record has exactly one MCP server and a named owner
- The gateway's audit log is exportable to your SIEM or evidence store

Move 1 — Expose one system, read-only, as a spike

Objective: Prove that governed agent-tool calls produce value on one system before building the platform. Timebox it: 2–4 weeks, one system, 3–5 read-only tools, real data.

Pick the system: The one with the highest question-density — usually the ERP or the data warehouse.

Checklist:

- ☐ Target system chosen; server owner named
- ☐ 3–5 read-only tools defined, with input/output schemas written down
- ☐ Real or sanitized data — not hand-picked samples
- ☐ Remote HTTP server targeting the 2026-07-28 spec — not stdio on a laptop
- ☐ Authentication in front of it from day one, even in the spike
- ☐ Every call logged from the first call

Done when: A named agent answers real questions from the real system, through auth, with every call logged.

Plan your first server:

System	Tool (verb + object)	What it answers

Move 2 — Stand up identity before the second server

Why this order: Retrofitting auth onto MCP servers already in production is expensive, technically and organizationally. Identity comes before scale.

- OAuth 2.1 at the gateway; tokens scoped per agent and per tool
- Every agent has a distinct identity — no shared API keys anywhere
- The human principal is propagated: a call is “agent X for user Y,” never just “agent X”
- Done-when passes: “which agent called what, as whom, when” is answerable from logs alone

Move 3 — Registry and policy as servers multiply

Why this order: The moment two teams run servers, discovery and authorization become the problem. Solve them centrally.

- Central registry: every server listed with its owner, toolset, and allowed callers
- Unregistered servers are unreachable — enforced technically, not policy-on-paper
- Per-tool RBAC: read and write scopes assigned per agent
- Quotas and rate limits per agent and per tool — a runaway agent is a cost incident too
- Done-when passes: an unregistered server can’t be reached; a registered one can’t be over-called

Move 4 — Audit as a first-class product

Objective: Not access logs — decision evidence. A reviewer must be able to reconstruct any consequential agent action after the fact without asking an engineer.

Record field	Why it is in the record
Tool + arguments	What was actually done, with parameters.
Agent identity	Which agent acted.
Human principal	Who the agent acted for.
Result	What came back, or what changed.
Approval (if any)	Who approved a Propose-tier action, and when.
Timestamp + trace ID	Order and correlation across multi-step chains.

Key fact: For EU-exposed systems this is regulatory evidence, not hygiene. EU AI Act Article 26 deployer duties include human oversight, monitoring, and log retention of at least six months.

- The audit record carries all six fields above
- Retention configured — at least six months where Article 26 applies
- Logs export in a format your auditors accept
- A non-engineer reviewer has reconstructed one real action end-to-end — tested, not assumed

The Write-Access Ladder

Write access is earned per tool, per scope — not switched on. A REST call runs code a developer reviewed; an MCP tool call is a decision an agent makes at runtime. Same wire, different animal — govern it like one.

Tier	What the agent may do	Promotion requires
0 — Read	Scoped, curated, logged reads. Where every system starts.	—
1 — Propose	The agent drafts the action; a named human approves before execution; the approval lands in the audit trail.	Tier 0 stable · eval set in place · owner named.
2 — Scoped write	Direct writes for narrow, bounded, reversible actions with hard caps and allowlisted targets.	Red-teaming passed · monitored Propose period · kill switch tested.
3 — Autonomous	Broad autonomous write. For most organizations in 2026: not yet.	Not this quarter. The rung exists so the roadmap is honest.

Readiness score

Count your unchecked boxes across all pages:

Open items	Where you stand
0–3	Platform-grade. Close the gaps and put the quarterly review on the calendar.
4–8	Workable — in build order. Identity and audit first; write access waits.
9+	Exposure: your agents are ahead of your governance. Start at Move 1 this week — read-only.

Go deeper: This blueprint accompanies the SumatoSoft playbook “MCP Is Becoming Enterprise Plumbing: A Build-Your-Own-MCP-Server Playbook” and the guide “Top 7 Enterprise AI Adoption Risks in 2026.”



Thank you for your time!

Any questions? Drop us a line!

Headquarters

One Boston Place, Suite 2602
Boston, MA 02108, United States

Other ways to get in touch

info@sumatosoft.com
sumatosoft.com