Sample MCP Gateway Readiness Audit

An illustrative, anonymized sample of the report you receive — based on a real, read-only assessment of a public production LLM/agent gateway (a mid-size SaaS platform running multiple MCP servers across multiple teams). Names, paths, and identifiers have been genericized.

Sample report · 2026-06-09 ✓ Production-Ready with caveats ● 4 Green ● 3 Yellow ● 0 Red

01 Executive Summary

Verdict: Production-Ready with caveats

For the gateway as written, assuming an operator who turns on the controls the platform provides.

This gateway is **architecturally sound on the dimensions that matter most for safety.** Authorization is enforced at the gateway on caller identity — resolved as a strict intersection of key, team, end-user, agent, and org permissions — and **the model never decides what it may call.** That single property is what defeats prompt-injection-to-tool-call. Secrets are referenced, never inlined. Identity is JWT/OIDC-rooted on the call path and propagates end-to-end. Cost is genuinely bounded — budget overruns raise an enforced exception rather than merely alerting.

The central risk is **not a missing control but a defaulting one.** Per-tool least-privilege and third-party-server pinning are opt-in, and one top-level authorization resolver path fails *open* on an unexpected exception. The single highest-leverage move is to flip those defaults to fail-closed and required-at-onboarding.

THE TWO HIGHEST-LEVERAGE MOVES

1. **Change the one fail-open authorization line**
   Return "no servers" (not "all allow-all servers") on an unexpected error, and add a regression test. *Effort: small.*
2. **Pin and allowlist the third-party MCP catalog**
   By version + digest, closing the floating-tag supply-chain exposure. *Effort: medium.*

02 Scope & Methodology

We assess **7 dimensions** of MCP gateway readiness through **read-only inspection** — source code, configuration, CI workflows, and deploy artifacts at a pinned commit. **Every finding cites a specific artifact** (file and line) captured in an evidence index.

Read-only inspection

Source code, config, CI workflows, and deploy artifacts at a pinned commit. No write access, no system modification.

Evidence-backed findings

Every finding cites a specific artifact (file and line). Every gap-matrix color traces to at least one evidence-index row.

Stated limitations

Live checks that could not run are marked static-only. We report the limitation rather than implying coverage we don't have.

03 Scored Gap Matrix

All 7 dimensions at a glance. Status colors trace directly to the per-dimension findings in Section 4.

Scored Gap Matrix — Sample Assessment

Production-grade · Partial · Absent / unsafe

#	DIMENSION	STATUS	FINDING	SEVERITY
01	Tool-access governance & RBAC	Partial	Strong gateway-enforced intersection RBAC — but per-tool least-privilege is opt-in.	High
02	Fail-close vs fail-open	Partial	Per-level authz failures fail closed; one top-level resolver path fails open to allow-all servers.	Critical
03	MCP / agent onboarding flow	Partial	Dual source-of-truth (declarative config and runtime DB); third-party servers unpinned; no tool-PR CI gate.	High
04	Observability & tracing	Pass	First-class OpenTelemetry + GenAI semconv; W3C trace-context extracted (propagation operator-configurable).	Medium
05	Multi-LLM routing & cost controls	Pass	Declarative routing; budget caps enforced (not alert-only); rate limits per key/model/MCP server.	Low
06	Security, secrets & identity (IDP)	Pass	Zero inline secrets; JWT/OIDC on call path; end-user identity propagates; OAuth token-exchange with audience+scope.	Low
07	Production-readiness gaps	Partial	Block-by-default guardrail, rate-limit 429s, alerting present — but no single tested global kill-switch or enforced canary.	Medium

Legend: Production-grade = control exists, is enforced, and is verifiable. Partial = intent exists but has gaps. Absent/unsafe = no effective control, or the control fails open.

04 Per-Dimension Findings

01 Dimension 01 — Tool-Access Governance & RBAC

Partial

WHAT WE LOOKED AT	Where the authorization decision lives, whether the model ever participates in it, and how granular the grants are.
WHAT WE FOUND	Authorization is enforced at the gateway on caller identity, resolved as a strict intersection of key → team → end-user → agent → org permissions. Deny-by-default holds for unmapped callers. Per-tool granularity exists, but it is opt-in per server: with no allowlist configured, the tool-permission check returns "allow."
WHY IT MATTERS	The hard part — keeping the model out of the authorization decision — is done correctly. The residual risk is operator misconfiguration: a write or external tool on a server with no tool allowlist is callable by anyone with server access.

02 Dimension 02 — Fail-Close vs Fail-Open

Partial

WHAT WE LOOKED AT	Every exception handler on the authorization and call path — does a degraded check deny or allow?
WHAT WE FOUND	Every per-level permission resolver fails closed. One exception: the top-level policy resolver returns the set of allow-all servers on an unexpected error — a partial fail-open, bounded to servers an operator already marked public.
WHY IT MATTERS	Fail-open in an authorization resolver is the single highest-risk class — it's how a degraded check silently becomes "allow." Here it is bounded, but it is still the one line that errs toward exposure. It is also a one-line fix.

03 Dimension 03 — MCP / Agent Onboarding Flow

Partial

WHAT WE LOOKED AT	How servers and tools are registered, whether the running tool set is reconstructable from source control, and whether onboarding enforces governance.
WHAT WE FOUND	A declarative config block exists, but MCP servers can also be created at runtime via an authenticated REST endpoint that writes to a database — a dual, mutable source of truth. The curated third-party catalog references servers via unpinned floating-tag commands with no version, digest, or checksum.
WHY IT MATTERS	You cannot fully reconstruct the live tool set from source control when servers can be added via the API, and an unpinned third-party server is a tampered-package away from a supply-chain incident (OWASP LLM05).

04 Dimension 04 — Observability & Tracing

Pass

WHAT WE LOOKED AT	Whether per-model/token/cost attribution and end-to-end request reconstruction are possible, and whether trace context survives hops.
WHAT WE FOUND	OpenTelemetry is a first-class integration with dedicated GenAI semantic-convention mapping (operation name, token usage, cache metrics), multiple exporters, and inbound W3C traceparent extraction.
WHY IT MATTERS	The building blocks for end-to-end reconstruction and per-team cost attribution are present and standards-aligned. Caveat: context propagation into every MCP hop is operator-configurable (not guaranteed by default).

05 Dimension 05 — Multi-LLM Routing & Cost Controls

Pass

WHAT WE LOOKED AT	Whether routing is a declarative policy and whether cost is enforced or merely alerted.
WHAT WE FOUND	Routing is a central declarative mapping from virtual model names to physical deployments, with router-level retries and timeouts. Budget caps are enforced — overruns raise a budget-exceeded exception, not just an alert.
WHY IT MATTERS	This is the dimension the platform is purpose-built for, and it shows — there is a real, enforced path against bill-shock and cost-based DoS (OWASP LLM10).

06 Dimension 06 — Security, Secrets & Identity (IDP)

Pass

WHAT WE LOOKED AT	Whether secrets are inlined, where identity is rooted, and whether the MCP token model matches the spec.
WHAT WE FOUND	No inline secret values. Identity is JWT/OIDC-rooted on the gateway call path, and the end-user identity propagates through to MCP handling and spend logs. MCP tokens use OAuth token-exchange (RFC 8693) with audience and scope binding.
WHY IT MATTERS	This is the strongest dimension — the secure defaults are in the code, and the token model is the one the MCP spec asks for. Who-can-do-what is answerable and revocable at the IDP.

07 Dimension 07 — Production-Readiness Gaps

Partial

WHAT WE LOOKED AT	Guardrails, alerting, the ability to stop a misbehaving tool, and staged-rollout discipline.
WHAT WE FOUND	A dedicated MCP security guardrail defaults to block (not alert). Slack/email/Prometheus alerting ships. Tools and servers can be disabled via config without a binary redeploy. Helm chart, hardened compose, and Terraform ship for staged deploy.
WHY IT MATTERS	The operational levers exist, so an incident is recoverable — but with friction. There is no single tested global kill-switch surfaced in code, no enforced canary/blue-green, and the red-team guardrails are available but operator-wired, not a standing CI gate.

05 Verdict & Top Risks

VERDICT

Production-Ready with caveats. No dimension is red. The three safety-critical dimensions — RBAC (the model is kept out of the authorization path), fail-close (correct everywhere except one bounded line), and identity/secrets (green) — are all solid. The yellows are *defaulting* and *operational* gaps, not absent controls: exactly the profile of a capable platform that needs config hardening, not re-architecture.

TOP RISKS — RANKED BY IMPACT

#	RISK	DIM	LIKELIHOOD	IMPACT
1	Authorization resolver returns the allow-all set on an unexpected error — a bounded fail-open path	2	Low	High
2	Unpinned floating-tag third-party MCP servers — supply-chain exposure (OWASP LLM05)	3	Medium	High
3	Per-tool least-privilege off by default — a write/external tool callable by any key with server access	1	Medium	Medium
4	Running tool set can drift from source control via runtime DB writes	3	Medium	Medium
5	No single tested global kill-switch / no enforced staged rollout	7	Medium	Medium

06 90-Day Roadmap

A sequenced fix plan ordered by risk-reduction-per-effort. Each item traces to a yellow finding and its cited evidence.

PHASE 1 — Weeks 0–2

Launch-blocking / highest-leverage

- **Change the one fail-open resolver line** — S

  Return an empty set (deny) instead of the allow-all set on an unexpected exception; add a regression test that asserts deny-on-error.

- **Pin the third-party MCP catalog** — M

  Replace every floating-tag command with a version + digest pin and allowlist the pinned set. No floating-tag MCP server can resolve.

PHASE 2 — Weeks 2–6

Default-on the controls

- **Make per-tool least-privilege default-on** — M

  Require a per-server tool allowlist so the permission check denies absent an explicit grant; gate the "allow all keys" foot-gun behind an audited override.

- **Add a tool-PR CI validation gate** — M

  Fail any tool/server PR unless tool allowlist, RBAC grant, rate limit, and pinned source are present; resolve the dual source-of-truth.

PHASE 3 — Weeks 6–12

Operationalize & prove

- **Ship a single tested global kill-switch** — M/L

  Disable a tool, a server, or the whole MCP gateway without a redeploy, plus an enforced staged rollout (canary/blue-green) with a tested one-command rollback.

- **Wire red-team / injection guardrails into a standing CI eval gate** — M

  Add a refusal-rate threshold on every tool/server PR. Injection, jailbreak, and tool-misuse regressions are caught before merge.

Net effect: Phase 1 removes the only fail-open path and the floating supply chain. Phase 2 makes least-privilege and governance default-on. Phase 3 makes readiness durable with a tested stop lever and a standing safety gate.

07 What You Receive

A fixed-scope, read-only engagement delivering four artifacts designed to be read together, plus a live walkthrough.

1 Readiness Report

The full narrative: executive summary, scope, methodology, per-dimension findings, verdict, top risks, and prioritized recommendations.

2 Scored Gap Matrix

All 7 dimensions with status, one-line finding, severity, effort, and a pointer to the roadmap item that closes each gap.

3 90-Day Roadmap

The sequenced fix plan, phased by risk-reduction-per-effort, with owners, effort, expected outcomes, and dependencies.

4 Evidence Index

Every finding cited to a specific artifact (file and line), with the observed behavior, the rubric line it matches, its color, and its severity.

Plus a live review session

A walkthrough of the findings and roadmap with your engineering and security stakeholders, so the report lands as a shared plan, not a PDF on a shelf.
