This side-by-side reference guide compares security controls for internal versus external MCP server deployments, with four priority ratings. The document includes shared concerns that apply to both models, quick validation questions for security assessments and a decision framework for choosing between self-hosted and vendor-hosted approaches. This guide is deal for security teams evaluating MCP architecture options or auditing existing implementations.

Table 1: Priority Legend

Priority	Level	Description
P1	Severe	Must have before production. Fundamental security controls.
P2	High	Should have for production. Important defense-in-depth.
P3	Medium	Plan for implementation. Governance and operational maturity.
P4	Recommended	Nice to have. Advanced capabilities for mature programs.

Shared Concerns

The concerns in Table 2 are essential regardless of whether you run internal or external MCP servers.

Table 2: Shared Concerns

Priority	Area	Control	Key Question
P1	**Kill switches**	• Ability to quickly disable tools, endpoints or entire MCP connections	• Can you disable any agent/tool within five minutes?
P1	**Authentication**	• Strong identity verification for all callers (humans, agents, services)	• Can an unauthenticated request ever reach the MCP?
P1	**Logging**	• Capture prompts, tool calls and responses with appropriate redaction	• Can you reconstruct what an agent did and why?
P1	**Prompt injection defense**	• Input sanitization • Injection detection • Output validation	• What happens if malicious content is in retrieved documents?
P2	**Human-in-the-**	• Require approval for high-risk, **loop** irreversible or financial actions	• Which actions can an agent take without human approval?
P2	**Data classification**	• Label and control data by sensitivity before it reaches MCP	• Do you know what data sensitivity levels the MCP can access?

P2	**Tool minimization**	• Only enable tools necessary for the use case; treat each tool as a risk	• Is there a tool enabled that isn't actively needed?
P3	**Lifecycle governance**	• Catalog, risk-tier and manage MCP servers/connections through lifecycle	• Do you have a single source of truth for all MCP connections?
P3	**Incident response**	• AI-specific playbooks for containment, evidence preservation and communication	• Do you have an incident response playbook specifically for AI/agent incidents?
P3	**Red-teaming**	• Test for prompt injection, data exfiltration, tool abuse, jailbreaks	• When did you last red-team your MCP integrations?

Table 3: Internal vs. External MCP Servers

Priority	Area	Internal MCP Server (Self-hosted)	External MCP Server (Vendor-hosted)
P1	**Scope and risk**	• Classify by data sensitivity, integration **tier** depth, agent capability (read-only → supervised → autonomous) • Use 4-tier priority model	• Classify by what data you'll send and what actions the vendor can perform • Start restrictive, expand only with justification
P1	**Network**	• No public IPs • Private subnets only • Access via VPN/ExpressRoute • Inbound only from gateway/bastion	• All traffic via controlled egress (API management/gateway) • Allowlist vendor fully qualified domain names (FQDNs) • Use private link if available
P1	**Identity and auth**	• Entra ID + OAuth 2.1 • Validate JSON Web Token (JWT) issuer/audience exactly • Short-lived tokens • Token binding for replay prevention	• Prefer your identity provider (Entra) over vendor accounts • Separate keys per environment • Verify vendor token handling • Confirm no sub-processor access
P1	**Authorization**	• Enforce at gateway AND backend • Object-level authz • Map Entra claims to MCP roles (admin/owner/user/read-only)	• Map internal roles to vendor permissions • Avoid broad admin scopes • Require approval workflows for high-risk actions
P1	**Secrets and non-identities**	• Unique service principal per **human** MCP/agent • Managed identities preferred • Vault for secrets • Rotation + expiration alerts	• Keys only in vault, injected at gateway • Never expose keys to users/agents. • Rotate keys on schedule and staff departure
P1	**Input validation**	• Direct and indirect prompt injection defense • Input sanitization • Output validation • Schema validation • Encoding checks	• DLP/redaction at gateway • Prompt shielding • Strip secrets before sending • Context length limits

P2	**Tool execution**	- Container/VM isolation - Resource limits (CPU/memory/time) - Network segmentation - Filesystem isolation - Recursive call limits	- Vendor tools are black boxes. Broker via your APIs. - No direct write to core systems. - Monitor for capability ~~drift.~~
P2	**Data protection**	- Classify data - Segment sensitive indexes - Access checks at retrieval - Data lineage - Unlearning/deletion - Poisoning detection	- Define allowed data classes - Start with non-sensitive data - Synthetic data for testing - Minimal vendor logging - Differential privacy
P2	**Session/memory**	- Encrypt context at rest - Session timeouts - Cross-session isolation - Bound memory size - Automatic clearing	- Verify vendor session isolation - Confirm no cross-tenant leakage - Test for context persistence across sessions
P2	**Gateway layer**	- Centralized gateway for all MCP access - JWT validation at edge - Rate limiting - Schema validation - Request signing	- Force all traffic through your gateway - Inject vendor keys at gateway - Log everything - Rate limit per user/agent
P2	**Supply chain**	- Assess model dependencies, libraries, base images - Monitor for vulnerabilities in tool integrations	- Map vendor's model providers and subprocessors - Understand inference location - Require notification of upstream changes
P2	**Logging and monitoring**	- Full tracing with redaction - Log integrity (WORM) - Oversight agents - Metrics - Correlation IDs - Alert on anomalies	- Log everything on your side - Ingest vendor signals - Correlation IDs - Behavioral baselines - Response integrity monitoring
P2	**Availability/SLAs**	- Define RTO/RPO - Backup configs and indices - Geographic redundancy - Disaster recovery testing	- Negotiate SLAs (uptime, latency) - Degradation plans - Failover options - Monitor vendor status
P3	**Governance**	- Catalog MCP servers/agents - Risk tier → capabilities - Change management - Promotion gates (shadow → autonomous)	- AI-flavored vendor risk assessment - SOC 2/ISO - Risk tier per vendor - Concentration risk - Financial health
P3	**Portability**	- Version control configs - Documented deployment - Reproducible infrastructure	- Assess lock-in - API compatibility with alternatives - Abstraction layers - Migration runbooks

			- Data export
P3	**Integrity verification**	- Request signing - Replay prevention - TLS everywhere, including internal	- Response authenticity - Cert pinning - Detect response modifications - Monitor for vendor-side injection
P3	**Legal/compliance**	- Data security posture management (DSPM) for data residency - Regulatory mapping. Multi-tenancy controls if applicable.	- Data processing agreement (DPA)/business associates agreement (BAA) - AI clauses (no training, intellectual property ownership) - Cyber insurance - Liability terms - Regulatory fit
P3	**Testing and validation**	- Threat model - Red team - Config review - Continuous security testing in CI/CD - Resource exhaustion tests	- Sandbox with synthetic data - Red-team integration - Continuous evaluation - Canary deployments - A/B testing - Cross-tenant tests
P4	**Multi-tenancy**	- Tenant isolation at MCP - Scoped indices - Per-tenant keys - Per-tenant rate limits and audit logs	- Verify vendor tenant isolation - Test for cross-tenant access - Contractual isolation guarantees

Security Validation Questions

Use these questions to quickly assess MCP server security posture.

Table 4: MCP Server Security Assessment Questions

Internal MCP Servers	External MCP Servers
Can requests reach MCP without going through the gateway?	Can any user/agent call the vendor directly (bypassing your gateway)?
What happens if a token is stolen?	Where does the vendor store your API keys and for how long?
Can one user access another user's data by manipulating IDs?	What data is the vendor allowed to log and retain?
What tools can an agent call without human approval?	Which of the vendor's tools can write to your systems?

How quickly can you disable a misbehaving agent?	How quickly can you cut all traffic to this vendor?
Where are secrets for this MCP server stored?	What happens to your data if the vendor is acquired?
Can you reconstruct what an agent did last week?	Does the vendor train models on your prompts/responses?
What's in your backup for this MCP and when was it tested?	What's your plan if this vendor has a major outage?

Table 5: When to Use Internal vs. External MCP Servers

Favor Internal MCP Servers When...	External MCP Servers May be Acceptable When...
Processing PII, PHI, PCI or highly confidential data	Working with public data or low-sensitivity internal data
Regulatory requirements mandate data residency control	Vendor meets all compliance requirements with attestation
Need full control over model behavior and guardrails	Vendor guardrails are sufficient for use case
Actions have high financial or operational impact	Actions are read-only or easily reversible
Require deep integration with internal systems	Integration is limited to specific, well-bounded use cases
Long-term strategic capability requiring investment	Rapid experimentation or time-to-value is critical