Barndoor.ai

ENTERPRISE EVALUATION GUIDE 2026

A buyer's guide to evaluating MCP gateway solutions

Barndoor.ai

Table of Contents

Barndoor.ai

80%+ **ENTERPRISES PILOTING** AGENTIC AI IN 2026	**5** **RISK CATEGORIES** TO EVALUATE IN ANY SOLUTION	**15** **EVALUATION CATEGORIES** FOR RFP-READY PROCUREMENT

WHO THIS IS FOR

CISOs, security architects, and application security leaders evaluating how to safely deploy AI agents connecting to MCP tools and services.

HOW TO USE THIS GUIDE

Read the 5 key risks to align your team on the threat model. Use the 15-category checklist to compare vendors side by side. The mapping at the end shows how a leading platform addresses each risk.

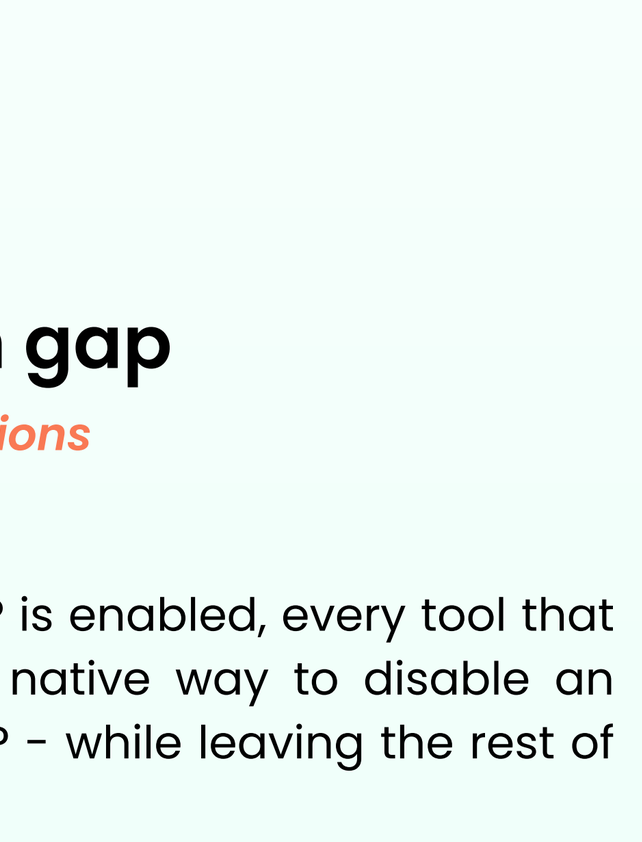
Why This Guide

MCP is now the default for connecting agents to enterprise tools

Adoption of Model Context Protocol (MCP) is accelerating faster than many integration standards. According to KPMG, by early 2026, the majority of enterprise AI initiatives include at least one MCP-connected tool, and analyst surveys put the share of enterprises with active agentic-AI pilots well above half.

However, the protocol was designed for developer convenience, not for enterprise security. Most production-ready MCP deployments today lack user-level controls, tool-level restrictions, audit logs, and fine grained access controls. If the vendor who hosts the MCP makes updates, including new destructive tools, those changes are invisible to admins.

This guide gives security and AI governance leaders a practical framework to evaluate MCP security solutions. It distills the five categories of risk that consistently surface in enterprise MCP deployments, translates each evaluation criteria, and provides a feature checklist based on 15 categories. These recommendations come from real enterprise procurement.



RISK 1 OF 5

The authorization gap

All-or-nothing tool permissions

The risk. MCP servers expose tools to an agent as a flat list. Once an MCP is enabled, every tool that the server provides is on for everyone who can reach it. There is no native way to disable an individual tool - say, the destructive delete_repo action on a GitHub MCP - while leaving the rest of the server enabled.

Why it matters. A single compromised, manipulated, or misused agent inherits the full surface of every MCP it can reach. This is also the layer where indirect prompt injection becomes dangerous: an attacker who steers an agent through poisoned content (a Jira ticket, a webpage, a support email) doesn't need new permissions, they just exercise the ones the agent already has. Without per-tool restriction, the blast radius is the entire server.

What to evaluate. Granular policy at the per tool and action level; the ability to disable destructive or sensitive tools without disabling the whole MCP; runtime authorization decisions that can incorporate context, not just a static credential; policy expressed as code so it's reviewable and version-controlled.

BARNDOOR APPROACH

Granular access control. Barndoor replaces server-grain permissions with per-tool and per-action policy. Destructive tools can be blocked, gated behind approval, or restricted to specific users while the rest of the server remains available. Even when an agent is steered by an injected prompt, blast radius is bounded by the least-privilege envelope assigned to the calling user.

The user permission gap

Once an MCP is enabled, every user has it

The risk. Today's MCP deployment model assumes that once it’s enabled, it’s “on” for everyone. When you turn on a Salesforce MCP for your Claude tenant, every Claude user in your org can use it. There is no native way to limit the Salesforce MCP to just the sales team, or have only senior engineers get the production database.

Why it matters. OAuth, your IdP, and the system's own permissions decide who can use that system. MCP conforms to those rules. For example, if a user can't log into Salesforce directly, they can't pull Salesforce data through an MCP either. But with LLMs, if an MCP is available, every user who can also log into the underlying system can use it that LLM.

That gap matters in two ways:

Short-term: *pilot or staged rollout.* You've just connected a new MCP and you want your sales ops team to pilot it for two weeks before opening it up to your BDRs because the rollout isn't ready for them yet.

Long-term: *permanent policy.* you may decide that entry-level business dev reps should never use Salesforce via Claude, even though they have legitimate Salesforce accounts and use it directly in the UI every day. Neither OAuth nor Claude's admin can address these two use cases. OAuth doesn't know who's a sales ops and who's a business dev rep. Claude's admin has only one switch: the MCP is on for everyone in the workspace, or off for everyone.

What to evaluate. Per-user and per-group MCP scoping driven by your enterprise IdP (Okta, Entra, any SAML/OIDC); end-to-end user identity propagation from the AI client through the MCP layer to the downstream tool, not a shared service-account credential; OAuth and token brokering instead of static API keys; automated de-provisioning when a user is offboarded in the IdP.

BARNDOOR APPROACH

IdP-driven, per-user MCP access. Barndoor scopes every MCP to specific users or IdP groups. Identity flows via OAuth and token brokering, replacing hardcoded API keys. When a user is offboarded in your IdP, their MCP access is also discontinued.

No MCP change management

Hosted MCP vendors can change the tool surface without telling you

The risk. The fastest-growing pattern in MCP today is using vendor-hosted MCP servers — the SaaS company you already use ships and operates the MCP for their product. That's convenient, but the vendor controls the tool surface unilaterally. If they ship a new tool, deprecate an existing one, change a tool's behavior, or — most importantly — add a new destructive action (delete_all_records, send_email_as_user), your agents inherit it the moment it's deployed. There is no native change notification, no diff, no review gate, and no rollback.

Why it matters. This is supply-chain risk. Your governance program assumes you've reviewed every privileged action your agents can take. The hosted MCP model breaks that assumption: tomorrow's tool list at vendor X is whatever vendor X decides it should be, and yesterday's risk review is stale the moment the vendor pushes a new release. For regulated industries this is also an audit problem.

What to evaluate. Change detection on hosted MCP tool catalogs (new tool added, tool description changed, schema changed); a pre-approvedworkflow that holds new or changed tools in a pending state until a human reviews them.

BARNDOOR APPROACH

Pre-approval workflow. Barndoor builds and maintains a catalog of approved MCPs and tools. When an MCP vendor changes their tool, the Barndoor platform will set them as OFF by default, meaning the policy engine returns a “deny” if an agent tries to call it until an admin explicitly enables it. This way, your agents never inherit a new destructive capability.

RISK 4 OF 5

Fragmented policies across AI

Permissions don't carry between Claude, ChatGPT, Cursor, and internal agents

The risk. Most enterprises don't have one AI client, they have several. Engineering uses Cursor and Claude Code; analysts use Claude or ChatGPT; product teams build internal agents. Each of these are connected to different MCPs, and each maintains its own settings for which MCPs are enabled, which credentials are used, and which users have access. The same user, with the same identity, gets a different MCP policy depending on which AI they're using, meaning your security team has to maintain complex sets of policies mapped to different users.

Why it matters. With policies living in different places, auditing and onboarding become disjointed - every new MCP has to be enabled, reviewed, with separate policies attached in every client.

What to evaluate. A single policy and audit plane that enforces whichever AI client the user is using; one place to onboard a new MCP, attach policy, and revoke it; one audit log that attributes every action to the underlying user identity regardless of which client they used.

RISK 5 OF 5

Sensitive data exposure to LLMs

Unfiltered tool data flowing into the model

The risk. By design, MCPs have no native filtering, redaction, or classification step between the source system and the LLM model. This means that sensitive data such as PII, credit card numbers and other financial identifiers, can flow into the model's chat window.

Why it matters. MCPs do not always enforce data protection controls of the underlying systems. Many APIs require developers to use specific DLP functions to check, but the LLM won't know this. Without an explicit policy layer between the tool and the model, every sensitive field a tool can return is likely exposed.

What to evaluate. A policy engine that sits in front of the LLM, lets you define what counts as sensitive for your organization, and can block the call before data reaches the model. Coverage of your specific regulated data types (PII, PHI, PCI, source code, secrets), data classification labels, per-tool and per-action granularity, and policy via an easy to use builder or built as code.

BARNDOOR APPROACH

Fine-grained access control via admin-authored data protection policies across both the MCP and LLM gateways. Policies are scoped by MCP server, tool, agent, or IdP group, and the same rules apply to LLM prompts and model responses. Detectors and classifiers determine which data is sensitive; transformations (tokenize, mask, redact, omit, etc) determine inbound and outbound behavior. The result: agents keep working, sensitive and regulated data never leaks, and indirect prompt injections never reach the model.

Barndoor.ai

The evaluation checklist

15 categories to evaluate in any MCP gateway. Use this checklist for vendor demos, RFPs, and internal architecture reviews. A credible vendor should be able to demonstrate coverage of these categories.

01 IDENTITY & AUTHENTICATION

- [ ] Integrates with enterprise IdPs (Okta, Entra, any SAML/OIDC)
- [ ] Full SSO across admin console and end-user UI Interactions
- [ ] Automated offboarding: IdP changes propagate to MCP permissions
- [ ] Admin roles driven by IdP groups, managed in IdP

02 ACCESS CONTROL (RBAC + ABAC)

- [ ] Customizable RBAC with granular permission assignment
- [ ] ABAC using IdP attributes (role, department, user type)
- [ ] Tool-level access control, not just per-server
- [ ] Policy management via UI, REST API, and infrastructure-as-code

03 POLICY ENFORCEMENT

- [ ] Real-time, pre-execution enforcement on every tool call
- [ ] Multi-dimensional scoping: per tool, per role, per user, agent, server, tool, action
- [ ] Policy lifecycle: draft → active → retired, with audit trails
- [ ] Validation, overlap detection, cloning, search, and live testing

04 GOVERNANCE & ADMINISTRATION

- [ ] Centralized admin console for MCPs, agents, and policies
- [ ] Pre-approval workflow for MCP servers and tools
- [ ] Catalog of allowed MCPs visible to admins and end users
- [ ] Clear separation between admin, security, and end-user roles

05 AGENT CONFIGURATION

- [ ] Per-agent registration with unique credentials and identity
- [ ] Multi-application type support: web, SPA, native, M2M
- [ ] Dynamic Client Registration (DCR) for new agent identities
- [ ] Allow/deny lists tied to agent identity, not just credentials

06 MCP GATEWAY

- [ ] Inline gateway architecture: every tool call passes through
- [ ] Data protection, indirect prompt injection detection
- [ ] Pre-execution policy enforcement at the gateway boundary
- [ ] Resilient under load with observable, scalable infrastructure

07 MCP SERVER MANAGEMENT

- [ ] Catalog of pre-configured connectors for common SaaS systems
- [ ] Support for custom and remotely hosted MCP servers
- [ ] Cloud-native secret management for OAuth tokens and API keys
- [ ] Multiple logical instances of one server (e.g. multi-tenant Jira)

08 TOOL INTELLIGENCE & OPTIMIZATION

- [ ] Intelligent tool routing to avoid context overload
- [ ] Tool filtering that respects access-control policies
- [ ] Semantic tool matching combined with policy-based filtering
- [ ] Token / cost optimization by filtering irrelevant tools

Barndoor.ai

09 DEPLOYMENT

- [ ] Flexible models: SaaS and private cloud
- [ ] Air-gapped deployment for regulated and high-security environments
- [ ] Native Kubernetes and container support
- [ ] Multi-cloud support across AWS, Azure, and GCP

10 AUDIT LOGGING

- [ ] Full attribution: user, agent, tool, server, timestamp on every call
- [ ] Configurable retention with near-real-time monitoring
- [ ] Tamper-proof / immutable log storage
- [ ] Native SIEM export (Splunk, Datadog, Elastic, etc.)

11 REPORTING & VISIBILITY

- [ ] Top-down dashboard of agentic MCP usage and policy decisions
- [ ] Usage analytics by server, agent, request type
- [ ] Single-pane-of-glass view across all AI applications and users
- [ ] Drill-down by agent, user, server, policy, and decision

12 AI PLATFORM INTEGRATION

- [ ] Works with major AI dev tools (Claude Code, Cursor, Windsurf, VS Code)
- [ ] Connectivity with major AI clients (Claude, ChatGPT, others)
- [ ] Support for command-line AI tools and agentic CLIs
- [ ] Compatibility with LLM gateways

13 BUSINESS SYSTEM INTEGRATION

- [ ] Pre-built connectors for Salesforce, Slack, Jira, Google Workspace, etc.
- [ ] Straightforward path to add custom MCP servers
- [ ] Full management API for programmatic configuration
- [ ] Webhook and event-driven integration with existing workflows

14 ONBOARDING & SUPPORT

- [ ] Structured, guided implementation from contract to production
- [ ] Comprehensive documentation, training, and knowledge base
- [ ] Clear SLAs and named contacts for enterprise support
- [ ] Reference architectures for common deployment patterns

15 SECURITY & COMPLIANCE

- [ ] SOC 2 Type II attested; policy library additionally aligned to ISO 27001:2022 and NIST 800-30 / 800-37 / 800-57
- [ ] CASA-aligned DAST assessment of the production application (TAC Security ESOF AppSec ADA)
- [ ] Phishing-resistant MFA for employee access, plus zero-trust internal architecture
- [ ] MCP request and response payloads are processed in memory only and never stored/persisted
- [ ] Dedicated egress IPs, TLS 1.2+, modern cert management
- [ ] No customer data used for model training, ever
- [ ] Identity propagation end-to-end — no shared service accounts
- [ ] Public Trust Center with audit reports and security posture
- [ ] Continuous compliance monitoring

Barndoor.ai

Key differentiators to probe

Beyond checklist coverage, these are the harder questions that separate mature platforms from early-stage tools. Use them in vendor demos to test claims.

Policy configuration: Code vs. forms

Ask: Can policies be defined and managed as code (JSON, YAML, or a policy language), or is the only option a UI form?

Form-only policy tools create hidden risk at scale. You cannot version-control a form submission, open a pull request against it, or run it through CI. When policies are code, they live in your existing security review workflows, and they can be audited, rolled back, and tested like any other control.

Probe: Ask the vendor to show you what a policy looks like in raw configuration. If they can only show you a web form, that is a governance gap.

Policy lifecycle management

Ask: What happens to a policy that is no longer needed? Can it be retired and archived, or only deleted?

Mature platforms support a full lifecycle: draft (in review) → active (enforced) → inactive (suspended) → archived (retained for audit). Tools that only support create-and-enforce leave you without the ability to build a historical compliance record, and deletion of policies should never destroy audit evidence.

Probe: Ask the vendor to walk you through retiring a policy and confirm that historical enforcement records remain accessible.

Policy testing before enforcement

Ask: Can we test a policy against real or simulated traffic directly in the platform?

The ability to test policies within the platform is a meaningful differentiator. It lets security teams validate intended behavior, catch conflicts with existing policies, and build confidence before enforcement. Platforms without this capability require testing in production or maintaining a separate staging environment.

Probe: Ask for a live demonstration of policy testing, not just a description of the feature.

Default-deny posture

Ask: What is the platform's behavior before the first policy is created? Are all tool calls allowed by default?

This is a critical and often overlooked question. Some platforms take a default-allow posture where every tool call is permitted until an explicit deny policy is written. In practice, this means there is a window, potentially a long one, where your MCP deployment is ungoverned. A default-deny platform enforces zero access until policies are explicitly granted.

Probe: Set up a test environment with no policies configured and attempt a tool call. The result will tell you everything about the platform's security philosophy.

Next Steps

Beyond checklist coverage, these are the harder questions that separate mature platforms from early-stage tools. Use them in vendor demos to test claims.

1. Inventory your MCP footprint, both sanctioned and unsanctioned. Document transport, credentials, and data sensitivity
2. Score your current state against the five risks and 15-category checklist.
3. Run vendor demos against the same threat model. Ask for live demonstrations, not just slides. Use the differentiator questions above.
4. Specifically test default-deny posture, policy testing capabilities, and policy lifecycle management.

Sources and further reading

KPMG, Enterprise AI Pilots: Survey on enterprise agentic-AI adoption and governance maturity.
kpmg.com/us/en/articles/2026/enterprise-ai-pilots.html

Model context protocol specification: Official MCP specification from Anthropic.
modelcontextprotocol.io

OWASP top 10 for LLM applications: PLLM07 (Insecure Plugin Design).
owasp.org

About Barndoor.ai

Founded in 2024, Barndoor is the AI governance platform built for the agentic enterprise. Barndoor's centralized control plane spans both an LLM gateway and MCP gateway with a shared data-protection layer, giving IT and security teams the visibility, fine-grained access control, and cost governance they need to govern AI usage across every model, agent, tool, team, and platform.

Learn more at **barndoor.ai** or book a demo at
barndoor.ai/get-a-demo/