MCP SERVER | NO CODE | CLOUD HOSTED

Cerbos MCP for AI Agents

Govern Policy-Driven Resource Access & Database Queries

Cerbos helps your AI agents manage complex, policy-driven resource access control. Connect this MCP to any client to evaluate permissions and generate optimized query plans instantly through natural language conversation.

A+ Quality Score 98.33/100

authorization | rbac | abac | policy-engine | access-control

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/cerbos

SCAN TO VIEW

The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the connectivity layer where AI connects to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

Cerbos MCP

6 tools available
Cloud-hosted on Vinkius

Authorization logic is usually the messiest part of an application. You write it once, but you spend hours debugging it across different services and user roles. This MCP lets your AI agent handle that complexity directly in your chat window. Instead of calling five separate endpoints to check if a resource is visible or editable by a certain role, you just ask. The system evaluates the policies instantly and tells you the outcome. You can even generate full query plans, so your downstream database calls are automatically filtered down to only what the user is authorized to see. Because this functionality handles core security logic, it's a perfect fit for Vinkius; you connect once from any compatible client and get access to robust policy management tools without writing boilerplate code.

Core Capabilities

01 — Verify specific permissions

Check if a user is allowed to perform an action on a given resource using `check_resources`.

02 — Generate database query filters

Produce detailed, optimized query plans that restrict results based on the principal's permissions using `plan_resources`.

03 — Process batch access requests

Evaluate multiple complex access policies at once using `authzen_evaluations` for standardized compliance checks.

04 — Execute single policy evaluations

Run a single, focused access check against the system model via `authzen_evaluation`.

05 — Inspect instance configuration

Retrieve vital metadata about your Cerbos setup and its current policies using `get_server_info` or `get_authzen_config`.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/cerbos** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your specific Cerbos instance base URL.

02 Your AI client connects, allowing you to interact with the policy engine through natural conversation.

03 You ask a question—for example, 'Can user X view resource Y?'—and the system returns a clear, definitive ALLOWED or DENIED result.

The bottom line is that your AI agent handles all the complicated API calls; you just talk to it like talking to a teammate.

Built For

Security Auditors and Software Engineers need this. If manually debugging complex access rules across multiple services slows down your development cycle, this MCP is for you. It lets you verify policy logic instantly without touching the underlying database or writing unit tests.

Software Engineer

Debugging a new feature's permissions by asking the agent to check resource access rules instead of running dozens of manual API calls.

Security Auditor

Verifying that compliance policies hold up across different roles and sensitive data attributes, ensuring no unintended access paths exist.

DevOps Engineer

Monitoring the health and configuration metadata of the Cerbos instance to keep sure the policy engine is running correctly in production.

What Changes When You Connect

01 Instantly verify permissions using `check_resources`. You no longer have to manually write API calls just to see if a user can edit a specific record.

02 `plan_resources` creates query plans that automatically filter database results. This means your application queries only pull data the user is actually allowed to see.

03 The batch evaluation tools, like `authzen_evaluations`, let you run full compliance checks across multiple policies at once—a huge time saver for security audits.

04 You get system visibility with simple calls like `get_server_info`. This lets your agent confirm the policy engine's version and build details on demand.

05 The standardized AuthZEN tools ensure your access requests meet industry compliance standards, reducing friction when building regulated applications.

Real-World Applications

A user needs to see all sensitive documents for a department

Instead of writing complex SQL with multiple `JOIN` statements and manual role checks, the agent runs `plan_resources`. It returns an optimized query plan that automatically filters results so only records matching the user's department attribute are visible.

A new feature needs to check permissions for 20 different actions

Manually calling a permission endpoint twenty times is painful. The agent uses `authzen_evaluations` to run all 20 checks in one go, giving you an immediate pass/fail report for the entire feature set.

The team needs to debug why a specific user can't access a resource

Instead of asking three different developers to check their policies, the agent runs `check_resources` with the principal and resource details. It immediately pinpoints if the policy itself is blocking the action.

Need quick confirmation on system health before deployment

The engineer uses `get_server_info` to confirm that the Cerbos instance is running the expected version, making sure the policies haven't been compromised by an outdated build.

Patterns to Avoid

Writing policy checks in application code

✗ AVOID

Embedding `if (user.role == 'admin') { return data; } else { throw Error('Forbidden'); }` logic into every service method makes the codebase messy and hard to update.

✓ INSTEAD

Let your AI agent handle it. Use `check_resources` to verify permission status first, keeping all access decisions centralized in the policy engine.

Running multiple sequential API calls

✗ AVOID

To check 5 different resource types for a single user, you might make five separate HTTP requests, slowing down the agent and adding complexity.

✓ INSTEAD

Use `authzen_evaluations` to evaluate all 5 access requests in one batch call. This is faster, cleaner, and more compliant.

Ignoring query constraints

✗ AVOID

Writing a broad database query that returns millions of records, forcing the application layer to filter out forbidden data.

✓ INSTEAD

Always use `plan_resources` first. This generates a highly optimized plan that restricts your database query *before* it runs, saving compute time and improving security.

The Right Fit

Use this MCP if access control is the most complex part of your application. If you routinely find yourself writing repetitive code blocks to check roles or permissions—that's a sign you need centralized policy enforcement. This tool excels when you need to debug policies interactively, using `check_resources` for single checks or `authzen_evaluations` for bulk auditing.

Don't use this if your access rules are simple (e.g., 'all users can read'). For simple cases, a basic database column check is fine. You need this MCP when you deal with attribute-based logic ('a user can only edit documents they created in department X on weekdays'). If you just need to retrieve data and don't care about the complex rules governing *who* gets access to that data, then your standard database API will suffice.

Cerbos MCP for AI Agents: Governing Resource Access Policies

Think about how you currently debug permissions. You open the dashboard, navigate through user roles, and manually run tests or use a series of separate API calls just to confirm if Principal A can perform Action B on Resource C. It's slow, it's tedious clicking across multiple screens, and it almost always requires writing temporary scripts that need cleaning up later.

With this MCP, you simply tell your AI agent what needs checking. The system instantly runs the policy evaluation. You don't write a single test script; you just get a clear answer: ALLOWED or DENIED. It changes debugging from an hours-long process into a quick chat exchange.

Cerbos MCP for AI Agents: Optimizing Database Query Plans

Before, when fetching data, developers often wrote queries that were too broad. They pulled

Now, you generate an AST-based query plan using `plan_resources`. This ensures your database

back everything and then relied on the application code to filter out the sensitive records based on policy—a massive security risk and a performance killer.

query is automatically constrained by the user's current policies. You get optimized data access right at the source.

6 Tools in the Cerbos MCP for Policy Evaluation

These tools allow you to check resources, plan queries, run batch evaluations, and retrieve system metadata directly from your AI agent chat interface.

#	TOOL	DESCRIPTION
01	`get_authzen_config`	Retrieves the specific endpoint URLs needed for using the AuthZEN APIs.
02	`authzen_evaluation`	Performs a single, focused evaluation of an access request using the standardized AuthZEN model.
03	`authzen_evaluations`	Handles batch evaluations for multiple access requests, supporting complex compliance rules like deny-on-first-deny.
04	`check_resources`	Performs a read-only check to determine if a principal has permission on a defined set of resources.
05	`plan_resources`	Generates an Abstract Syntax Tree (AST) query plan that filters database results based only on the user's allowed access.
06	`get_server_info`	Returns version and build details, helping you verify the status of your Cerbos instance.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U Hey, can I check if my marketing manager role can view expense reports for department 'Seattle'?

Access Check Result:

- **Principal:** Marketing Manager
- **Action:** View
- **Resource Filter:** Expense Reports
- **Outcome: ALLOWED**

The policy allows viewing, but only if the resource's 'department' attribute matches the user's assigned department.

U What query plan do I need to view all documents related to project Alpha?

Here is the optimized AST filter for your database query:

```json
{
"WHERE": "resource.project == 'Alpha' AND (user.role == 'Editor' OR user.department == resource.department)"
}
```

You can apply this exact condition to limit results, ensuring users only see data they are authorized for.

U Can I run a bulk check on 10 different roles against the 'invoice' resource?

Batch Evaluation Summary:

Role	Action	Resource	Status
Viewer	Read	Invoice	ALLOWED
Editor	Write	Invoice	DENIED
Admin	Delete	Invoice	ALLOWED

This confirms that the 'Editor' role cannot delete invoices, which matches our policy.

Frequently Asked Questions

01 **How does Cerbos MCP help me manage user permissions without writing complex code?**

It lets you talk to your AI agent and ask questions like 'Can this person do X on Y?' The system handles the entire policy evaluation, giving you a definitive answer. This cuts out hours of manual API scripting.

02 **Is Cerbos MCP better than just using database roles for access control?**

Yes. Database roles are static; this MCP allows dynamic, attribute-based checks. You can enforce policies based on things like a user's department or the resource's creation date, which is much more flexible.

03 **What if I need to see all resources for a given project? Does Cerbos MCP handle that?**

It does. You can use the query planning tools within the MCP. It generates an optimized filter you can apply directly to your database, so you only retrieve authorized records.

04 **Does connecting the Cerbos MCP affect my existing application logic?**

No. The MCP acts as a policy layer that your AI agent calls when needed. It enhances your current workflow by providing an external source of truth for all access decisions, so you don't have to change core business logic.

05 **Can I check compliance across many different users at once using Cerbos MCP?**

Absolutely. You can use the batch evaluation tools in the MCP. This lets you run large-scale audits, checking hundreds of potential access combinations with a single prompt.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"cerbos": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

Cerbos is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by Cerbos. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	Cerbos MCP
Server ID	`019e3875-b811-7120-ad7b-d6113ef92763`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/cerbos.