MCP SERVER | NO CODE | CLOUD HOSTED

Linear MCP

Manage issues, cycles & tasks with your agent.

Linear MCP connects your agent directly to your issue tracking workspace. List all tickets, get detailed metadata on any bug or feature request, and update status across projects using natural conversation. It lets you coordinate agile development cycles without ever leaving your chat window.

A+ Quality Score 100/100

issue-tracking | agile | sprint-planning | workflow-automation | task-management | software-delivery

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

Linear MCP

11 tools available
Cloud-hosted on Vinkius

Need to know the state of a project? This MCP gives your AI agent full control over your Linear workspace. Instead of jumping through multiple tabs to gather info on bugs or features, you talk to your agent. You can list all active tickets and pull detailed metadata instantly. Want to update status for a whole team? Your agent handles it. It's like having a dedicated project coordinator that lives inside your chat app. When you connect this MCP via Vinkius, your AI client becomes an agile architect for your entire organization. You manage everything from creating brand new issues to adding contextual comments and verifying the health of your API connection, all in plain conversation.

Core Capabilities

01 — Track project progress

List active projects, cycles, labels, and teams to understand your organizational structure.

02 — View issue details

Retrieve full metadata for a specific ticket, including its current status and history.

03 — Manage tickets

Create new issues or update the status of existing ones programmatically.

04 — Communicate contextually

Add specific comments to an issue, ensuring team members see the latest updates right where they belong.

05 — Check workspace health

Verify that your API connection is working correctly and check overall service status.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP in Vinkius. Then, grab your Personal API Key from the Linear settings page.

02 Connect that key to your preferred AI client (like Cursor or Claude).

03 Ask your agent a question like, 'List all active issues for the Frontend team,' and it executes the command.

The bottom line is you talk to your agent once; it does the manual work inside Linear and gives you the answer back.

Built For

This is for anyone whose job involves coordinating technical efforts. Product Managers who struggle with status reports, DevOps Leads needing quick access to team assignments, or Engineers who hate context switching.

Product Manager

Needs to monitor project progress and ensure feature delivery stays aligned with the roadmap without manually checking every ticket.

DevOps Lead

Must verify technical issue logs, check team member assignments, and optimize resource allocation using simple AI queries.

Software Engineer

Wants to instantly retrieve issue lists or update ticket statuses using natural language commands without leaving their development environment.

What Changes When You Connect

01 Stop manually updating statuses. You can tell your agent to update an existing issue or create a new one instantly, keeping all development records current without opening the app.

02 Get instant team directory insight. List workspace teams and members to understand exactly who is assigned what, saving time when coordinating cross-functional efforts.

03 Understand project scope quickly. By listing active projects or cycles, you get a clear overview of which goals are currently being worked on across different departments.

04 Maintain perfect context. Instead of emailing updates, use the MCP to add specific comments directly to an issue, keeping all historical notes tied to the ticket itself.

05 Coordinate complex tasks. Your agent can pull detailed information for any single issue and then allow you to update it in one go, dramatically reducing clicks.

Real-World Applications

Need a status report on three different features?

A Product Manager needs to know the current state of 'Dashboard UI', 'Auth Fix', and 'Billing Flow' before a stakeholder meeting. They ask their agent, and it executes get_linear_issue for all three tickets, compiling the detailed metadata into a single conversational summary.

A team needs a new bug ticket created immediately.

An Engineer finds a critical issue during testing. Instead of navigating to 'Create Issue,' they simply ask their agent to create_linear_issue with the title and description, ensuring it gets assigned to the right project.

The team structure changed, who owns this feature?

A DevOps Lead needs to know which department is responsible for a legacy system. They ask their agent to list_linear_teams and then use list_linear_users to pinpoint the right owner.

Need to update tickets after a sprint planning meeting.

The PM asks the agent to update_linear_issue for all high-priority bugs from the previous cycle, changing their status from 'Backlog' to 'Ready for Dev,' confirming everything in one go.

Patterns to Avoid

Using a generic API wrapper

✗ AVOID

Manually writing complex multi-step scripts or calling raw endpoints just to list all active projects and their associated labels.

✓ INSTEAD

Use the dedicated MCP tools. Just ask your agent to run list_linear_projects and then follow up with list_linear_labels for a full scope review.

Copying statuses into spreadsheets

✗ AVOID

The team spends hours at the end of the week manually copying ticket titles, current status, and assignee from Linear into Excel for reporting.

✓ INSTEAD

Let your agent retrieve detailed metadata using get_linear_issue. It pulls the precise data points you need directly into the chat, ready to paste or summarize.

Forgetting who is on which team

✗ AVOID

A manager can't find out if a specific person was moved from the 'Mobile' team to the 'Backend' team last month.

✓ INSTEAD

Use list_linear_users and list_linear_teams. Your agent quickly maps the organizational structure for you, confirming resource assignments.

The Right Fit

Use this MCP if your primary bottleneck is coordinating technical status updates across multiple projects or teams. You need a single conversational layer to manage tickets, cycles, and user data in Linear without context switching. Don't use it if you only need basic read-only access; for simple reading, other general API connectors might suffice. However, if you need to actively change things—like creating an issue, updating status, or adding a comment—this specific MCP is necessary because it exposes the write tools (create_linear_issue, update_linear_issue) that make your agent useful.

The constant cycle of manual status checks.

Every week, someone has to check Linear. They jump into the app, open the project board, filter by assignee, and then they have to click on three different tickets just to see if a feature is ready for testing. Then they copy that status ('In Review') into a Google Doc, which usually gets outdated before anyone can use it.

With this MCP, you tell your agent what you need —for example, 'List all issues in the billing cycle that are blocked.' Your agent runs the necessary checks and gives you the answer right away. No clicking through boards; you get a direct summary.

The Linear MCP lets you write tickets and update status with natural language.

Before, creating an issue meant navigating to 'New Issue,' filling out the title, description, assigning it a label, and selecting the project. Updating required opening the ticket, finding the status dropdown, changing it, and then maybe adding a comment for context.

Now, you simply tell your agent, 'Create a new issue about X for the Backend team.' It handles the entire process—creation, assignment, and initial metadata—in one step. The friction is gone.

Linear: 11 Tools for Issue Tracking

These tools let you execute every possible action in Linear—from creating new bugs to listing team members—all through your AI agent.

#	TOOL	DESCRIPTION
01	`create_linear_comment`	Adds a comment directly to any specific issue ticket for team context.
02	`create_linear_issue`	Creates an entirely new issue within the Linear workspace.
03	`get_linear_issue`	Retrieves comprehensive, detailed information about one specific ticket.
04	`list_linear_cycles`	Lists all active development cycles currently running in your organization.
05	`list_linear_issues`	Provides a broad list of issues across the workspace, showing their general status.
06	`list_linear_labels`	Retrieves a directory of all issue labels used by your teams.
07	`list_linear_projects`	Lists and shows the status of all active projects within Linear.
08	`list_linear_teams`	Lists all distinct teams or departments that belong to your workspace.
09	`list_linear_users`	Retrieves a directory listing of all individual members in the Linear workspace.
10	`check_linear_status`	Runs a check to confirm that your API connection is active and working correctly.
11	`update_linear_issue`	Changes the status or details of an existing ticket using natural language instructions.

See It in Action

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

U List all active issues and show their status.

I've retrieved your issues. You currently have 15 active high-fidelity tickets, including 'API Integration' (In Progress) and 'UI Bug' (Backlog). Would you like to see the detailed metadata for any of them?

U Create a new issue for the 'Frontend' team titled 'Implement Dashboard'.

Issue orchestrated! I've successfully created 'Implement Dashboard' (ID: FRONT-45) for the Frontend team. This high-fidelity update is now live in Linear. Shall I verify the team's current technical velocity?

U Check the team members in the 'Mobile' team.

Team directory orchestrated! The 'Mobile' team currently has 8 high-fidelity profiles assigned. Your API connection is healthy. Shall I retrieve the detailed resource utilization metadata for this team?

Frequently Asked Questions

01 **Can I use Linear MCP to list all projects?**

Yes, you can. Use the MCP's tools to run list_linear_projects. This gives your agent a directory of every active project in your workspace so you know where everything is.

02 **How do I update an issue status using Linear MCP?**

You use the update_linear_issue tool. You simply ask your agent to change the status for a specific ticket, and it handles updating the record in Linear automatically.

03 **Does Linear MCP help me track team members?**

Absolutely. The list_linear_users tool lets you retrieve a directory of all workspace members. This is critical when figuring out who owns which feature or bug.

04 **What if I need to add context to an issue? Is it Linear MCP?**

Yes, use the create_linear_comment tool. You can have your agent write and post a comment to any specific issue, keeping perfect communication history right where the work happens.

05 **Is Linear MCP just for reading data from my issues?**

No, it's fully bi-directional. Beyond listing or getting details, you can also create_linear_issue and update existing ones, making your agent an active participant in the workflow.

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 `"linear-alternative": { "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

Linear 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 Linear. 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	Linear MCP
Server ID	`019dd119-3e40-719a-90c9-c39f5130bea8`
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/linear-alternative.