MCP SERVER | NO CODE | CLOUD HOSTED

Fibery MCP

Query, update, and manage work records via conversation.

Fibery MCP connects your entire work management workspace to your AI agent. It lets you read schemas, query data across all custom databases, create new tasks, and add status updates—all through natural conversation. Stop navigating menus; start asking questions about your project data.

A+ Quality Score 100/100

task-management | workflow-automation | schema-management | collaboration | database-query

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

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.

Fibery MCP

11 tools available
Cloud-hosted on Vinkius

This MCP gives your agent direct access to the heart of your Fibery workspace. You don't have to jump into the portal and click around just to find a piece of information or update a status. Instead, you talk to your AI client, and it handles the complex database interactions for you.

Need to know what's going on with Product X? Your agent can pull structured data from specific databases, search across every app in your workspace, and even tell you exactly who needs to review something by listing all users. You can ask your agent to update a task status or add a comment directly to an entity record without ever leaving your chat window. If you're looking for robust ways to handle custom workflows from different clients, Vinkius hosts this MCP, ensuring deep compatibility with any AI agent.

This means software teams can sync development progress and product managers can gather insights simply by asking questions in plain English.

Core Capabilities

01 — Discover workspace structure

Retrieve a map of all your apps, databases, and custom fields so the agent knows where to look.

02 — Read specific record data

Pull details for any single item (entity) using its unique ID or by searching keywords across multiple databases.

03 — Write and change records

Create, modify, or delete entire records in custom databases based on your instructions.

04 — Manage team communication

Read the conversation history or add a new comment to any project record.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Fibery workspace name along with an API token.

02 Connect your preferred AI client (like Claude, Cursor, or Windsurf) through the Vinkius catalog.

03 Instruct your agent on what you need—for example, 'Find all tasks assigned to me in Marketing Operations that haven't been updated this week.'

The bottom line is, once connected, your AI agent treats Fibery like a single data source you can talk to.

Built For

Product Managers and Operations staff who spend too much time switching between the project management tool and their chat window. It's for anyone whose job involves querying complex, interconnected business data stored in custom databases.

Product Manager

Gathering status updates or finding all related tasks across different product spaces without having to manually navigate the portal.

Operations Specialist

Running complex data checks, like verifying if a required field is populated before closing out an account record, and logging that confirmation in a central task.

Software Developer / DevOps Engineer

Pulling specific entity details about bug reports or feature requests to update the sprint board status directly from their chat environment.

What Changes When You Connect

01 Stop navigating menus. Instead of opening multiple tabs to gather data from different project spaces, you simply ask your agent for the information. The agent handles the cross-database lookup automatically.

02 You can automate status changes instantly. If a bug is fixed and ready for QA, tell your agent instead of finding the card, clicking 'Status,' and selecting 'Ready.' Use `update_entity` to change it directly.

03 Keep team context alive. Need to know what was decided last week? Your agent can use `get_comments` to pull the conversation history attached to any record, so you never lose critical decisions in a thread.

04 Build custom reporting without SQL. Instead of exporting data and running joins in a spreadsheet, use `query_entities` to pull exactly the structured data set you need and get it back immediately for analysis.

05 Centralize knowledge. If you're unsure which databases exist or what fields they hold, let your agent run `get_schema`. It gives you a full map of everything available in your workspace.

Real-World Applications

The QA Team needs to check all related tickets.

Instead of opening the 'Software Development' space and manually cross-referencing multiple databases for bug reports, the agent uses `search_entities` with keywords like 'API failure' across all spaces. It compiles a single list of affected records instantly.

The Product Manager needs to check user permissions.

The PM can ask their agent for a list of users and then specifically request details on who has access, using `list_users` to verify team membership before onboarding new contractors.

A developer finished the feature and needs to signal completion.

The dev doesn't need to click through multiple dropdown menus. They simply ask their agent to 'Mark this entity as complete and assign it to QA.' The agent runs `update_entity` using the appropriate record ID.

A stakeholder needs a summary of all active projects.

The agent can run `list_apps` first, giving the stakeholder an overview of all functional areas. Then, it uses `query_entities` to pull key metrics (like 'Last Updated Date' and 'Status') from each app into one readable report.

Patterns to Avoid

Trying to search unstructured documents.

✗ AVOID

You can't just ask the agent, 'Find me the PDF attachment about Q3 planning.' The MCP is designed for structured work data (entities), not file storage.

✓ INSTEAD

Changing a record without knowing its ID.

✗ AVOID

A user tries to say, 'Change the status of the bug report to Done.' The agent won't know which one; it needs a specific identifier to run `update_entity`.

✓ INSTEAD

Ignoring the schema map.

✗ AVOID

Assuming data exists in a field name like 'Client ID,' when the actual database uses 'Customer UUID.' Running `query_entities` with the wrong field name fails immediately.

✓ INSTEAD

The Right Fit

Use this MCP if your core workflow involves managing structured records, tracking status changes, or querying cross-functional data (e.g., linking a bug report to a product space and an assigned user). You need reliable actions like `update_entity` and the ability to run complex queries using filters.

Don't use it if your main goal is simple document archiving, managing external files, or communicating with systems that don't treat information as structured 'entities.' If you only need to read a static list of names, simply asking for `list_users` might suffice. This MCP shines when the data needs manipulation: creating records (`create_entity`), updating statuses (`update_entity`), or comparing data points across databases.

The Pain of Manual Data Collection

Today, gathering a full project status report means opening the main work management platform. You then have to click into the 'Product' tab, check its key metrics. Next, you switch over to the 'Software' tab and run a separate query for bug counts. Then you might need to open an unrelated 'Marketing' space just to see who is assigned to review it. This process takes 15 minutes of clicking, copying names, and pasting into a summary spreadsheet.

With this MCP, your agent handles the entire sequence in seconds. You tell it: 'Pull the status metrics for Product, Software, and Marketing, and list any outstanding action items.' The agent uses multiple tools—like `query_entities` across different spaces—and gives you one coherent, ready-to-read answer.

Accessing Records with Fibery MCP

The tedious steps that vanish are the context switches and manual data pulls. You no longer have to ask a team member, 'What's the status of X?' Instead, your agent runs `get_entity` using the specific ID, providing you with the current details instantly.

It’s not about automating clicks; it’s about eliminating context loss entirely. Everything—the data, the comments, the history—is available in one conversational flow.

Fibery: 11 Tools for Enterprise Automation

These tools allow your AI agent to perform granular actions across every part of your Fibery workspace, from listing users to running advanced database queries.

#	TOOL	DESCRIPTION
01	`add_comment`	Posts a new comment onto an existing project record (entity).
02	`create_entity`	Builds and saves a brand-new item into any specified database.
03	`delete_entity`	Permanently removes an existing project record (entity).
04	`get_comments`	Retrieves all comments posted to a specific record, letting you see the full history.
05	`get_entity`	Fetches all data for one specific item using its unique ID.
06	`get_schema`	Retrieves a blueprint of your entire workspace, showing every available database and field type.
07	`list_apps`	Lists all the main functional areas (spaces) used in your Fibery account.
08	`list_users`	Shows a list of every user who has access to your workspace.
09	`query_entities`	Searches and pulls data from specific databases using detailed filters you provide.
10	`search_entities`	Looks for information across every single database in your workspace using keywords.
11	`update_entity`	Changes the data (like status or assignee) on an existing record.

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 spaces in my Fibery account.

I've retrieved your spaces. You have 'Product Management', 'Software Development', and 'Marketing Operations' active. Which one would you like to explore further?

U Show me the tasks assigned to me in the 'Software Development' space.

Inspecting entities... I found 5 tasks assigned to you, including 'Fix Bug #102' and 'Implement API caching'. Would you like the full details for any of these?

U Add a comment to task UUID-123 saying 'The client approved the design'.

Comment added successfully! Your message has been posted to the task and your team has been notified flawlessly.

Frequently Asked Questions

01 **How does Fibery MCP handle cross-database searches?**

The agent runs `search_entities`, which looks for keywords across every single database type (or 'app') in your workspace. It doesn't just check one area.

02 **Can I use Fibery MCP to update an entity status?**

Yes, you can. Using `update_entity`, the agent changes any field on a record, such as moving a task from 'In Progress' to 'Review.' You just need the specific item ID.

03 **What is the best way to get an overview of my workspace?**

Run `list_apps` first. This gives you a list of all major spaces, and then you can use `get_schema` for details on what data lives inside each one.

04 **Does Fibery MCP only work with tasks?**

No, it works with any structured 'entity.' You're not limited to task management; you can manage custom databases—like lists of vendors or required equipment—too.

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 `"fibery": { "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

Fibery 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 Fibery. 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	Fibery MCP
Server ID	`019d7598-7881-701b-93f4-378efe8cc1ab`
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/fibery.