MCP SERVER | NO CODE | CLOUD HOSTED

Miro MCP

Manage whiteboards and collaboration content conversationally.

Miro lets your AI client control complex visual collaboration boards through natural conversation. Create entirely new whiteboards, add sticky notes with specific content, track team feedback via comments, or review every widget on existing canvases without opening the Miro app. Manage board members and audit project structures instantly.

A+ Quality Score 98.33/100

whiteboarding | visual-collaboration | diagramming | team-workflow | sticky-notes | real-time-sync

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

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.

Miro MCP

14 tools available
Cloud-hosted on Vinkius

You can take full control of your visual workspace by connecting your account to any AI agent. Instead of clicking through menus in a separate application, you simply talk to your client about what needs fixing or adding to a board. You can tell it to generate an entire new whiteboard for a project kick-off, and it'll handle the setup. Need meeting notes? Ask your agent to drop a sticky note with specific text at coordinates (x:100, y:200). It handles that automatically. Your client can list all current board members or check if a certain widget exists on a canvas. When you see tools like this hosted in the Vinkius catalog, it means your AI agent is ready to treat complex visual boards like structured data. You get instant visibility into everything—from reviewing historical comments to listing every single card and shape currently placed on the board.

Core Capabilities

01 — Generate new whiteboards

Create an entirely new Miro board with a specified title or description.

02 — Add content widgets

Place specific items, like sticky notes or structured cards, onto any existing whiteboard at defined locations.

03 — Audit board contents

List every widget on a board—sticky notes, images, shapes, etc.—including their content and coordinates for review.

04 — Manage team access

See who is already on the board and add new users with specific roles like owner or editor.

05 — Track conversations

Read existing comments or post new feedback directly to a whiteboard for asynchronous discussion.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Miro Access Token (OAuth 2.0 or Personal Access Token).

02 Your AI client authenticates the connection, granting permission to manage your visual workspaces.

03 You prompt your agent with a natural language request—for instance, 'Add three sticky notes about Q3 goals'—and it executes the action directly in Miro.

The bottom line is you get direct, conversational control over your complex whiteboarding content without leaving your AI chat window.

Built For

Product Managers who hate manually checking board status; Designers needing to audit and manage permissions quickly; Team Leads tired of switching apps just to post a quick reminder.

Product Manager

Quickly review all contents on a board, add meeting notes as sticky notes, or check who has editor rights before a major review.

UX Designer

Audit the full list of widgets and cards on a board to ensure brand consistency, or manage membership for design sign-offs.

Team Lead / Scrum Master

Monitor team collaboration activity by listing recent comments or ensuring all necessary contributors are added as board members.

What Changes When You Connect

01 Stop context switching. Instead of opening the Miro app to add a quick sticky note, just ask your agent to create one for you instantly.

02 Audit board contents completely. You can use list_board_items to view every widget on a canvas—cards, shapes, notes—and understand what's actually there without manual clicking.

03 Maintain team hygiene effortlessly. Use list_board_members to check who needs access and then call add_board_member to grant or adjust roles like 'editor' or 'viewer.'

04 Keep project history organized. List comments retrieves all feedback, letting you review the discussion thread without scrolling through a messy UI.

05 Build new structures fast. If you need a starting point, use create_board to generate an entirely new board with specific naming and descriptions.

Real-World Applications

Need to compile feedback from multiple stakeholders.

A project lead asks their agent to list all comments on the 'V2 Launch' board. The agent returns a summary of recent discussions, allowing the lead to instantly see where key decisions were made without reading through hundreds of individual messages.

Launching a new initiative requires a dedicated workspace.

A Product Manager simply asks their client to create_board with the name 'Q4 Strategy' and description 'Focus on market penetration.' The board is generated immediately, ready for team input.

Need to update core documentation widgets.

A designer uses list_board_items to find all cards labeled 'API Spec' across the board. They then use create_card and provide updated text, ensuring all documentation widgets are current in one conversational step.

Onboarding a new team member who needs limited access.

A Team Lead uses list_board_members to verify the board permissions. They then call add_board_member, granting the new hire only 'commenter' status so they can participate without accidentally moving critical elements.

Patterns to Avoid

Trying to manually list all content.

✗ AVOID

An engineer opens Miro, clicks through tabs, and copies data from 50 different widgets just to paste it into a Jira ticket. This takes minutes of frustrating clicking.

✓ INSTEAD

Ask your agent to use list_board_items on the board ID. Your client returns all widget details—type, content, position—in clean text that you can copy and paste immediately.

Assuming a new board is ready for collaboration.

✗ AVOID

A PM creates a draft board but forgets to set up the proper roles or initial structure, leading to confusion about who can edit what.

✓ INSTEAD

First, use get_board to verify permissions. Then, call add_board_member to ensure every necessary user has the correct role (e.g., 'admin' for setup, 'editor' for content).

Editing board metadata in multiple places.

✗ AVOID

The team rebrands a project and needs to update the name across the board, but someone forgets to change it in one of the linked documents.

✓ INSTEAD

Use update_board. You provide the existing board ID and the new name or description text; your agent handles the metadata change instantly.

The Right Fit

Use this MCP if your workflow revolves around *visual* planning, brainstorming, or collaborative diagramming. If you need to read a sticky note's content, add a card widget, check user roles, or get an inventory of everything on the canvas, this is what you use. Don't use this if you just need to process structured data like updating customer records in a CRM or running financial reports. For those tasks, connect your agent to a dedicated database MCP; it handles row-level updates and complex queries better than a visual whiteboard tool can. If all you need is text generation based on an input file, use a document processing MCP instead of trying to write it onto a sticky note.

Managing whiteboards feels like clicking through ten different tabs.

Think about today's process. To check the status of a board, you open Miro, manually navigate to the 'Ideas' section, then scroll through dozens of sticky notes, copying specific dates or names into a spreadsheet just so your project manager can read it later. You repeat this for members and comments, jumping between boards and tabs.

With this MCP connection, you simply ask your agent: 'What are the key action items from the last week?' Your client doesn't open Miro; it runs list_board_items to inventory every widget. It pulls the data, summarizes the findings, and hands you a clean, actionable text summary.

Miro MCP gives your agent full control over board content.

Before this, updating a board required multiple steps: manually clicking to add a sticky note, then using the member panel to check roles, and finally opening the comment section. It’s slow, prone to human error, and requires constant context switching.

Now, you can issue a single command like 'Add three new cards detailing API endpoints' or 'Update the board name to Q3 Launch Plan.' The agent handles all those specific actions—create_card, update_board—in one flow. It’s that simple.

Miro MCP - 14 Tools

These tools let you perform specific actions in Miro, like creating new cards or listing board members, all controlled through your AI agent.

#	TOOL	DESCRIPTION
01	add_board_member	Adds a specific user to the whiteboard and lets you set their access role (owner, admin, editor, commenter, viewer).
02	create_board	Generates an entirely new Miro board with a given name and optional description.
03	create_card	Creates a structured card widget on the board, allowing you to set a title and description.
04	create_comment	Posts new feedback or comments directly onto the whiteboard, optionally replying to an existing comment.
05	create_sticky_note	Places a simple sticky note with custom text content on the board at specific coordinates.
06	delete_board_item	Removes any widget from the board using both the board ID and the item's unique ID. (Warning: this action is permanent.)
07	get_board	Retrieves all metadata for a specific Miro board, including its owner and permissions.
08	get_board_item	Fetches the detailed information for one single widget on a whiteboard using its ID.
09	get_user_context	Confirms your identity and access details, showing what Miro user the API calls will be made as.
10	list_board_items	Lists every widget on a board, allowing you to filter by type (sticky note, card, shape, etc.) and view its content and position.
11	list_board_members	Retrieves a list of all users who have access to the whiteboard, along with their roles and emails.
12	list_boards	Provides an overview of all Miro boards that your user account can currently see or access.
13	list_comments	Retrieves a list of comments on the board, including who wrote them and when they were posted.

#	TOOL	DESCRIPTION
14	update_board	Changes the name or description metadata of an existing Miro whiteboard.

See It in Action

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

U Show me all the sticky notes on my Sprint Planning board.

I found 12 sticky notes on your Sprint Planning board. Topics include: 'Implement OAuth flow', 'Fix checkout bug', 'Design new dashboard', 'Update API docs', 'Refactor auth middleware' and 7 more. Would you like the full content of any specific note?

U Create a new board called 'Q2 OKRs' with a description 'Quarterly objectives and key results'.

Done! I've created the board 'Q2 OKRs' with the description. The board is ready for collaboration. Board URL: https://miro.com/app/board/uXjV... You can now add sticky notes, cards and other items to it.

U Add a sticky note to my board saying 'Meeting notes: Discussed API versioning strategy' at position x:100, y:200.

Done! I've added the sticky note at position (100, 200) on your board with the text 'Meeting notes: Discussed API versioning strategy'. All board collaborators can see it immediately.

Frequently Asked Questions

01 **How do I use Miro MCP to list all boards?**

Use the list_boards tool to see an overview of every board accessible by your account. This lets you find the correct ID before running any other action, like get_board.

02 **Can Miro MCP help me add members with specific roles?**

Yes, use the add_board_member tool. You provide the board ID and the user ID, and you can specify exactly what role they need: owner, admin, editor, commenter, or viewer.

03 **What is the difference between list_board_items and get_board_item with Miro MCP?**

list_board_items returns a comprehensive inventory of every widget on the board. get_board_item only retrieves specific, detailed information for one single item you already know the ID of.

04 **How do I make sure my sticky notes are in the right spot using Miro MCP?**

When creating a note with create_sticky_note, you can optionally provide x and y coordinates. This ensures your agent places the content exactly where you want it on the canvas.

05 **Does Miro MCP allow me to read old comments?**

You can use list_comments to retrieve a history of conversations. The tool provides the comment text, author info, and date so you can track feedback over time.

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 `"miro": { "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

Miro 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 Miro. 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	Miro MCP
Server ID	`019d8459-bf54-73f7-82a8-1ca716086433`
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/miro.