MCP SERVER | NO CODE | CLOUD HOSTED

Figma MCP

Inspect nodes, track versions, and export assets via chat.

Figma MCP lets your AI agent access all your design files, components, comments, and version history. You can inspect nodes, render assets as PNG or SVG, manage team projects, and track every change without leaving your chat window.

A+ Quality Score 98.33/100

ui-ux-design | design-systems | prototyping | asset-export | file-inspection | version-history

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/figma-alternative

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.

Figma MCP

16 tools available
Cloud-hosted on Vinkius

Connect your Figma account to any compatible client and treat it like a dedicated design operations assistant. Instead of jumping between tabs just to check feedback or grab an image, you talk to your agent. It instantly finds the full document tree for deep inspection, letting you see every node, style, and component property. Need assets? You can tell it which part of the file needs exporting—PNG, JPG, SVG, or PDF—and get a direct link. Plus, it keeps track of everything that's happened: from tracking team projects to reviewing who changed what in the past using version history. When you connect through Vinkius, your AI client gets access to thousands of tools, making this Figma connection just one part of a massive workflow upgrade.

Core Capabilities

01 — Review Design Feedback

Pull all comments from a file, see who reacted using emojis, and read the entire discussion thread.

02 — Audit Component Libraries

Get a list of published components or full component sets across your team's design system.

03 — Inspect File Structure

Retrieve the complete hierarchy and properties for any specific node, frame, or component within a file.

04 — Export Visual Assets

Render specific nodes as high-resolution images (PNG, SVG, JPG, PDF) without opening the original design program.

05 — Track Design History

Browse all previous versions of a file, including labels and who created them, so you never lose context.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and provide your Figma Personal Access Token.

02 Connect your AI client (Claude, Cursor, etc.) to the Vinkius catalog using that token.

03 Ask your agent a natural language question, like 'What were the changes in version 3?' or 'Export the hero section as an SVG'.

The bottom line is you talk to your AI client, and it uses this MCP to pull live data from Figma and give you actionable answers.

Built For

This is for the designer drowning in comments, the developer who wastes time trying to find component specs, or the design ops person who has to audit dozens of files weekly. If your job involves moving data out of Figma and into documentation or code, you need this.

UX/UI Designer

They use it to quickly find recent files, review comments pinned on the canvas, or export a specific frame for client review without opening the desktop app.

Frontend Developer

They inspect node properties and component details directly through conversation. They can also render images needed for implementation reference.

Design Operations (Design Ops)

They audit the entire team's design system, checking published components or listing all project files to ensure consistency and version control.

What Changes When You Connect

01 Stop opening the Figma app just to see who commented. Use `get_comments` or `get_comment_reactions` to pull all design feedback right into your conversation thread for immediate context.

02 You never have to guess which version is correct again. By using `get_file_versions`, you can instantly compare design states and track exactly when a change happened, complete with creator details.

03 Getting assets used to mean manual exporting. Now, use `get_images` to render any node—like the hero section or an icon set—as a precise PNG, SVG, or PDF without touching the Figma UI.

04 Design systems are massive. Instead of clicking through dozens of folders, you can run `get_team_components` or `get_team_component_sets` to audit and list published components across your entire team in one go.

05 Need to know what's inside a complex file? Use `get_file_nodes` to inspect specific elements, pulling out node IDs and properties without having to download the entire design structure.

Real-World Applications

Reviewing Design Feedback for a Client Presentation

A designer needs to summarize client feedback. They ask their agent to run `get_comments` on the file, then use `get_comment_reactions` to see which comments sparked debate (lots of '🔥' or '🤔'). The agent compiles this into bullet points for a quick presentation slide.

Developer Reference Check

A developer needs the exact specs for the primary button component. They ask their agent to run `get_component` on the key, and the response gives them the owning file details and node ID needed for implementation reference.

Design System Audit

A Design Ops lead needs to know if all departments are using the correct color palette. They use `get_team_component_sets` and then cross-reference those against the file structure provided by `get_file` to flag inconsistencies.

Capturing a Specific Asset

A marketing team member needs an icon for a blog post but can't access Figma. They ask their agent to run `get_images` on the specific component node ID, and the agent returns a high-res SVG link immediately.

Patterns to Avoid

Copying IDs manually

✗ AVOID

The user opens the Figma file in their browser, right-clicks an element to copy its ID, and then pastes that ID into a separate API tool endpoint. This process is slow and prone to human error.

✓ INSTEAD

Instead of manual copying, ask your agent to run `get_file_nodes` on the desired area. The agent handles the node identification and property retrieval conversationally.

Checking history via UI

✗ AVOID

The user has to click into the 'History' tab in Figma, scroll through labels, and manually note down which version fixed a specific bug—a tedious process that takes minutes.

✓ INSTEAD

Just ask your agent to run `get_file_versions`. It pulls the label, date, creator, and description instantly for you.

Finding assets in multiple places

✗ AVOID

The user has five different files and needs to export a specific hero banner from each one. They have to open all five files and manually hit 'Export' on each instance.

✓ INSTEAD

Use `get_images` and provide the node IDs for all five instances in one prompt; the agent handles batch rendering requests.

The Right Fit

Use this MCP if your job requires extracting structured design data, managing team assets, or exporting visuals *without* opening the Figma application. You should use it when you need to know: 'What did they change?' (using `get_file_versions`), 'Is this component published and correct?' (`get_team_components`), or 'Send me a PNG of X' (`get_images`). Don't use this if your goal is to *edit* the design

itself, move layers around, or create new shapes. For those tasks, you still need to open Figma. This MCP is purely for reading, auditing, and extracting data.

The Design Feedback Loop Is a Documentation Nightmare

Today, reviewing design feedback means navigating multiple screens: the main canvas, the comments sidebar, and then sometimes cross-referencing a separate spreadsheet to see who owns that component. You copy IDs here, paste them there, jump between file versions just to figure out if the requested change was implemented.

With this MCP, you simply ask your agent about the feedback. It automatically runs `get_comments` and synthesizes the entire discussion thread for you. Your AI client doesn't just read comments; it compiles actionable summaries of design discussions.

Get Component Specs with get_component

Before, if a developer needed the exact details—like the component name, owning file, and node ID—they had to hunt through the sidebar or ask a peer who was already busy. It was a point-and-click scavenger hunt.

Now, you just tell your agent: 'What are the specs for the primary button?' The tool runs `get_component`, returning the precise metadata instantly. That’s it.

Figma Alternative: 16 Tools for Design Workflow Management

These tools let your agent perform precise actions on Figma files—from reading every comment to rendering high-res assets and auditing team components.

#	TOOL	DESCRIPTION
01	`get_comment_reactions`	Retrieves who reacted to a specific Figma comment and what emoji they used.
02	`get_comments`	Fetches all comments on a file, including the message, author, date, and position on the canvas.
03	`get_component`	Looks up details for a published component using its unique key.
04	`get_file`	Retrieves the full document tree, including nodes, styles, and components, for an entire Figma file by its key.
05	`get_file_nodes`	Gets the structure and properties of specific, requested nodes without loading the whole design file.
06	`get_file_versions`	Provides version history details for a file, listing labels, descriptions, creators, and dates.
07	`get_image_fills`	Lists all the image URLs used throughout a Figma design file.
08	`get_images`	Renders specific nodes as images in formats like PNG, JPG, SVG, or PDF by providing node IDs and scale.
09	`get_user`	Verifies the authenticated user's ID, email, and account type for token validation.
10	`get_project_files`	Lists all files belonging to a specific Figma team project.
11	`get_recent_files`	Finds and lists the keys, names, and last modified dates of your most recently accessed files.
12	`get_team_component_sets`	Retrieves published component sets used across the entire Figma team's design system libraries.
13	`get_team_components`	Gets a list of all published components for auditing purposes within a Figma team.
14	`get_team_projects`	Lists all projects managed by the Figma team, along with their creation dates and IDs.

#	TOOL	DESCRIPTION
15	`post_comment`	Adds a new comment to a file, optionally pinning it to specific coordinates or replying to an existing thread.
16	`post_comment_reaction`	Applies an emoji reaction to an existing Figma comment.

See It in Action

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

U Show me my recently accessed Figma files.

Here are your 5 most recent files: 'Mobile App v3' (modified 2h ago), 'Design System - Components' (modified yesterday), 'Landing Page Redesign' (modified 3 days ago), 'Dashboard Analytics' (modified last week) and 'Onboarding Flow' (modified 2 weeks ago).

U Show me all comments on my Mobile App v3 design file.

I found 12 comments on the file. The most recent is from Sarah: 'The button contrast doesn't meet WCAG AA standards' (2h ago, pinned to frame 47:12). There are also 3 unresolved comments from the design review meeting. Would you like the full thread for any of these?

U Export the hero section frame from my Landing Page design as a PNG at 2x scale.

Done! I've rendered node '12:34' (Hero Section) as a PNG at 2x scale. The image URL is https://figma-alpha-api.s3.us-west-2.amazonaws.com/images/... You can download it directly from this link.

Frequently Asked Questions

01 **How do I get a list of all files in my Figma project using Figma MCP?**

You run `get_project_files`. This tool lists every file within the team project, giving you keys and names so you know exactly what's available for inspection.

02 **Can I export a single frame from my design using Figma MCP?**

Yes. You use `get_images`, providing the specific node ID for that frame, along with your desired format (PNG, SVG, etc.) and scale setting.

03 **Does Figma MCP help me track changes in a design file?**

Absolutely. Use `get_file_versions` to pull the entire version history. You get labels, descriptions, who created it, and when it happened, making audits easy.

04 **What is the difference between getting components and getting component sets with Figma MCP?**

Use `get_team_components` to list individual published elements. Use `get_team_component_sets` if you need to audit an entire organized group of related components, like all card variations.

05 **Can I reply to a comment using Figma MCP?**

Yes. You can use the `post_comment` tool and include the existing comment ID in your prompt so that your agent replies directly within the thread.

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 `"figma-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

Figma 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 Figma. 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	Figma MCP
Server ID	`019d843a-e49c-7065-bf31-fc6ca746e4e7`
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/figma-alternative.