MCP SERVER | NO CODE | CLOUD HOSTED

Figma MCP

Inspect assets and pull data from your design files.

Figma MCP connects your AI agent directly to Figma files and projects. It lets you inspect complex design structures, extract reusable design tokens, render layers as high-quality images (PNG/SVG/PDF), manage comments across teams, and track project history—all without ever opening the desktop app.

A+ Quality Score 98.33/100

ui-ux-design | prototyping | vector-graphics | design-systems | file-inspection | image-rendering

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

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

15 tools available
Cloud-hosted on Vinkius

Design assets are messy. They exist in a visual interface that's great for designers but terrible for developers or engineers who just need clean data. This MCP lets your AI agent bypass the UI entirely. You can ask it to read the full document tree, identify specific components across an entire design system, and even extract published styles and variables so they sync perfectly with your code. Need proof of what was designed two versions ago? Your agent accesses file version history instantly. If a product manager needs to review feedback on 50 screens, you can pull all comments into one place. It treats the Figma workspace like structured data, not just pretty pictures. With Vinkius in the mix, your agent gets access to this powerful connectivity alongside thousands of other services, turning design iteration from a manual nightmare into a conversational task.

Core Capabilities

01 — Inspect Design Structure

Retrieve the full document tree or specific layers within a file for detailed analysis.

02 — Render Visual Assets

Generate images, components, or entire frames from Figma files in PNG, SVG, or PDF formats.

03 — Manage Design Tokens

Extract published component metadata, styles, and local variables needed for coding.

04 — Coordinate Team Feedback

Read existing comments or post new feedback directly onto design files.

05 — Navigate Projects & Teams

List projects and team metadata to understand the full scope of a company's design assets.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and provide your Figma Personal Access Token. This authenticates your agent against your workspace.

02 Next, you tell your AI client exactly what you need: 'Get the component set metadata for X' or 'Render nodes Y and Z as SVG'.

03 Your agent executes the required action via the connection, pulling structured data (tokens, file trees) or generating the requested assets right back to your chat interface.

The bottom line is you tell the system what structural design information or asset type you need, and it gets it for you.

Built For

This MCP targets product designers who hate manual handoff processes, frontend developers tired of guessing token values, and product managers needing a centralized view of design feedback. It solves the problem of bridging the gap between visual design and structured code.

Frontend Developer

Uses this MCP to automatically extract local variables and published styles from Figma into JSON or YAML, eliminating manual copy-pasting for coding components.

Product Designer

Leverages the agent to list team projects or get file version history, allowing them to document design evolution and track assets without navigating multiple folders.

Product Manager

Directs the AI to retrieve all comments on a specific file key, compiling qualitative feedback from stakeholders into an actionable report.

What Changes When You Connect

01

02

03

04

05

Real-World Applications

The Hand-off Audit

A developer needs to know if the current design system has a primary button color token. Instead of asking a designer to screenshot it, they prompt their agent: 'Run `get_local_variables` and find all tokens related to buttons.' The agent immediately returns structured JSON data for implementation.

The Project Scope Review

A product manager needs to audit a new feature. They ask the agent to 'List all files in the Marketing Assets project and check their version history.' This gives them a complete overview of which design elements are stable versus those still under development.

The Missing Component

A designer realizes they can't find the correct logo variant. They ask the agent to 'List all published team components.' The system filters through hundreds of files and points them toward the exact component set key, saving hours of searching.

The Feedback Aggregator

During a sprint review, an engineer asks the agent: 'Get all comments on the Checkout flow file.' The AI compiles feedback from QA, marketing, and design into one readable list, allowing for immediate prioritization.

Patterns to Avoid

Treating Figma like a simple image library

✗ AVOID

Trying to manually download screenshots of multiple layers or components just to pass them off to development. This is slow, non-scalable, and loses metadata.

✓ INSTEAD

Use the agent's `get_images` tool to render nodes into structured formats like SVG or PNG automatically. For data extraction, always run `get_local_variables` instead of guessing.

Confusing 'file structure' with 'content'

✗ AVOID

Asking the AI just to 'list files,' which only gives names, but failing to ask it to inspect the internal nodes and pages. This leaves you blind to actual content.

✓ INSTEAD

To get full context, use `get_file` to pull the entire document tree or use `list_project_files` to see all contained assets. Never rely on file names alone.

Ignoring historical changes

✗ AVOID

Assuming that because a component exists now, it has always been used correctly, and skipping the check for old versions.

✓ INSTEAD

Always verify design evolution by running `get_file_versions`. This shows exactly when components changed, who approved them, and what the state was previously.

The Right Fit

Use this MCP if your workflow requires treating design assets as structured data. You need to extract tokens (via `get_local_variables`), read file metadata (via `list_project_files`), or render elements programmatically (via `get_images`). If the process involves moving raw, clean code variables out of a visual canvas and into an automated workflow, this is your tool.

Don't use it if you simply want to look at a design file. If you just need to eyeball the layout or quickly check one specific color value on screen, opening Figma directly is faster. Also, if your only goal is communication—like sending a simple "thumbs up"—you don't need this MCP; basic messaging tools suffice. But when the job requires deep inspection of *how* the design system works beneath the surface, use the full suite of tools here.

The painful process of design handoff today

Right now, moving a design from Figma to development is usually a mix of pain points. You have to ask designers for specific screenshots, copy-paste color hex codes into documentation,

With this MCP, you stop doing that legwork. Your agent handles the grunt work: it runs `get_local_variables` and spits out a perfect JSON object of every token and style. You get

and then manually hunt through files to find the correct component name or local variable ID. It's slow, error-prone work that takes up hours just trying to gather data.

structured, ready-to-use code variables instead of blurry, manual screenshots.

Figma MCP: Structured data from visual designs

The biggest time sink is manually verifying the scope and status of an asset. You waste time asking 'Which version?' or 'Where did I save that component?' because the information is buried across different project folders and historical drafts.

Now, you ask your agent to list team projects and get file versions in one query. The MCP treats the entire Figma workspace like a navigable database of assets, giving you immediate context on status, history, and location.

Figma: 15 Tools for Designers and Developers

Use these tools to read complex file structures, pull reusable variables, render assets, and manage all aspects of a Figma design project directly through your AI agent.

#	TOOL	DESCRIPTION
01	`delete_comment`	Removes a comment you previously posted or found on a specified Figma file.
02	`get_component_set`	Retrieves the detailed metadata for an entire published component set within your design system.
03	`get_team_info`	Fetches general organizational details and metadata about a specific Figma team.
04	`get_images`	Renders selected nodes or frames from the design file into output images.
05	`get_local_variables`	Lists and retrieves published design tokens, styles, and local variables used in a Figma file.
06	`list_components`	Retrieves a list of all published team components available across your workspace.
07	`get_me`	Provides basic details about the authenticated user connected to the Figma account.
08	`get_comments`	Reads and lists all existing comments left on a specified Figma file.
09	`get_file_nodes`	Retrieves the details of specific, targeted nodes within a Figma file structure.
10	`get_file`	Accesses and retrieves the complete content and node hierarchy of a specified Figma file.
11	`get_file_versions`	Lists every historical version available for a given Figma file key, allowing you to check history.
12	`list_project_files`	Retrieves all individual files contained within a larger design project folder.
13	`list_styles`	Gets metadata and names for published team styles (e.g., color palettes, typography).
14	`list_team_projects`	Lists all major projects housed under a specific Figma team.

#	TOOL	DESCRIPTION
15	post_comment	Allows you to create and post new comments directly onto a designated Figma file.

See It in Action

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

U List all projects in my Figma team ID 123456.

I've retrieved your team projects. You have 'Design System', 'Mobile App Redesign', and 'Marketing Assets'. Which project would you like to list the files for?

U Get the document tree for file key abcDEF123.

Retrieving file tree... I've loaded the 'E-commerce App' file. It contains 5 pages: 'Onboarding', 'Checkout', 'Profile', 'Inventory', and 'Settings'. Would you like to inspect specific nodes on any page?

U Render nodes 1:2 and 1:5 as PNG images.

Generating images... I've rendered the requested nodes. You can view them here: [Link to Node 1:2], [Link to Node 1:5]. These links are valid for 30 days flawlessly.

Frequently Asked Questions

01 **How do I use the Figma MCP to extract design tokens?**

You use the `get_local_variables` tool. Just ask your agent to list all published variables for a specific file key, and it will return them in a structured format.

02 **Can I render entire frames using the Figma MCP?**

Yes, you can use `get_images`. You just need to specify the node IDs or frame keys, and the agent handles rendering those selected areas into PNG, SVG, or PDF.

03 **Does the Figma MCP allow me to check old versions?**

Absolutely. Run `get_file_versions` to see a list of every saved version for any file. You can then ask the agent to inspect specific nodes from that historical state.

04 **What is the difference between listing components and styles?**

Use `list_components` when you want metadata on reusable UI elements (like buttons or cards). Use `list_styles` when you are looking for global design properties like color palettes or text sizes.

05 **How does the Figma MCP help with team collaboration?**

The agent can read existing feedback using `get_comments`, compiling all comments on a file into one summary. It can also post new feedback for you via `post_comment`.

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": { "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	`019d7598-f918-734e-97ef-7511d7c1fef0`
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.