MCP SERVER | NO CODE | CLOUD HOSTED

CometAPI MCP for AI Agents

Building multimodal workflows and coordinating data streams

CometAPI connects your AI agent to hundreds of model types—from image generators and text models to speech processors. It handles complex, multimodal workflows by giving you a single layer to orchestrate different AI services, simplifying development for advanced applications.

A+ Quality Score 100/100

model-aggregation | generative-ai | multimodal | chat-completion | image-generation | speech-to-text

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

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.

CometAPI MCP

10 tools available
Cloud-hosted on Vinkius

Building an app that uses multiple kinds of AI is usually a mess. You're juggling API keys, managing rate limits across five different providers, and writing complex logic just to switch between text generation, image creation, and audio processing. CometAPI changes that. It gives your agent one place to access all those capabilities—it's an entire intelligence layer for multimodal workflows.

You don't have to write boilerplate code every time you want to add a new service or model type. Instead of toggling between dozens of separate dashboards, your AI client coordinates everything for you. You can generate text responses using `create_ai_chat_completion`, then immediately convert that text into an audio file with `convert_text_to_speech`. Need visuals? Just run the prompt through `generate_ai_image`. If you're building something complex, connecting CometAPI via Vinkius gives your agent access to thousands of tools and models right out of the box.

Core Capabilities

01 — Execute Multi-Model Text Generation

Generate text responses using any supported large language model through `create_ai_chat_completion`.

02 — Create and Manipulate Media Assets

Produce images from prompts with `generate_ai_image`, or convert existing audio files to readable text using `transcribe_audio_to_text`.

03 — Speech-to-Text and Text-to-Audio Conversion

Convert plain text into natural-sounding speech via `convert_text_to_speech`, or transcribe spoken audio files using `transcribe_audio_to_text`.

04 — Monitor API Usage and Costs

Check your account status, track credit consumption with `get_api_usage_statistics`, and retrieve current model pricing data via `get_pricing_information`.

05 — Manage Model Infrastructure

See exactly what's available by listing supported providers using `list_supported_ai_providers` or checking the full catalog with `list_available_ai_models`.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to CometAPI on Vinkius and get your API Key.

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

03 Your agent can then call any of the available tools, coordinating multiple services like text generation and image creation in a single prompt.

The bottom line is you get one consistent connection point for dozens of AI providers, eliminating the need to manage separate API keys or vendor dashboards.

Built For

This MCP is built for developers and product teams who aren't satisfied with single-model apps. If your application needs text *and* images *and* audio, this is the layer you need to connect everything.

AI Backend Engineer

Build robust pipelines that switch between different LLMs or media generation tools based on user input. You use it to manage all the underlying API calls.

ML Product Lead

Prototype complex, multimodal features quickly by testing multiple providers and model types without rewriting core logic.

Full-Stack Developer

Integrate advanced AI capabilities—like taking user audio input, transcribing it, summarizing it with text, and generating a visual asset—all in one workflow.

What Changes When You Connect

01 You can instantly switch between models. If one LLM performs poorly on a specific task, your agent uses `list_available_ai_models` to find and try an alternative provider.

02 Manage costs proactively. Instead of running into unexpected bills, use `get_api_usage_statistics` and `get_pricing_information` to track exactly how much every service call costs you.

03 Handle rich media natively. You can move from a user's spoken query (using `transcribe_audio_to_text`) straight into a text summary, then generate a supporting image with `generate_ai_image`, all within one agent conversation.

04 Reliability comes first. Periodically run `check_api_health` to ensure your application isn't failing because of an unstable endpoint. This keeps everything running smoothly.

05 Centralized control means less boilerplate code. You never have to worry about updating credentials or switching authentication logic when adding a new AI service.

Real-World Applications

Creating Interactive Training Materials

A company needs to generate training modules that include text, audio narration, and supporting diagrams. The agent first uses `create_ai_chat_completion` for the lesson summary, then `generate_ai_image` for concept art, and finally `convert_text_to_speech` to provide voice-over files, all without manual stitching.

Building Customer Support Bots

A support bot receives a user's audio complaint. The agent uses `transcribe_audio_to_text` to capture the message, passes it to an LLM via `create_ai_chat_completion` for a summary, and then sends the summarized text back to the client.

Developing Digital Art Tools

A user wants an art piece based on a detailed concept. The agent uses `generate_ai_image` with the prompt, but if the image isn't right, it can use `list_available_ai_models` to try a different style generator.

Running Proof-of-Concept AI Demos

A developer needs to demo an app using three different LLMs for comparison. Instead of writing three separate API calls, they simply use the agent's ability to access and select from `list_supported_ai_providers`.

Patterns to Avoid

Writing dedicated code for every AI model

✗ AVOID

If you need Claude, you write one block of code. If you switch to Gemini, you have to copy-paste and rewrite the entire connection logic and error handling.

✓ INSTEAD

Use this MCP's `create_ai_chat_completion` tool. It handles the provider switching for you, allowing your app to remain stable even if you update models or add new services.

Ignoring cost tracking until deployment

✗ AVOID

The prototype works great locally, but when it hits production volume, the unexpected $500 bill arrives because nobody tracked which model was costing the most.

✓ INSTEAD

Integrate `get_api_usage_statistics` into your dashboard. This lets you monitor real-time costs and pinpoint exactly where budget overruns are happening.

Hardcoding media formats

✗ AVOID

Your workflow expects a JPG output, but the service defaults to PNG, breaking the entire front end because the file type wasn't accounted for.

✓ INSTEAD

Use this MCP's orchestration layer. It manages multimodal outputs and provides visibility into model capabilities, ensuring your agent can handle varied media types.

The Right Fit

You should use CometAPI if your application requires more than one type of AI service to function—for example, a combination of text summarization, image generation, and audio processing. It's the single coordination layer you need for complex, multimodal applications. Don't use it if you only ever plan on using one specific LLM provider (e.g., only OpenAI). In that case, sticking to that provider's native SDK is simpler. If your main pain point is managing complexity and keeping a unified workflow across different AI services, this MCP is essential.

CometAPI: Solving Multimodal Workflow Headaches with API Coordination

Today, building an advanced app that handles user input from multiple sources—like a spoken request that requires both text summarization and a supporting diagram—is a painful, manual process. You write code to transcribe the audio, then another block to summarize the resulting text, followed by yet another call with the summary prompt to generate an image. Every step is disconnected.

With this MCP, your agent sees it as one task. It takes the audio input, transcribes it using `transcribe_audio_to_text`, passes that text into an LLM via `create_ai_chat_completion` for summarization, and then uses that summary to prompt the image generator with `generate_ai_image`. You get a cohesive output without stitching together dozens of separate API calls.

CometAPI: Simplifying AI Model Selection and Billing Oversight

The biggest headaches often aren't the calls themselves, but knowing which model to use and how much it costs. You spend time checking three different provider dashboards just to figure out if you should use GPT-4 Turbo or Claude 3 for a specific task, and then guessing your budget limits.

This MCP solves that by aggregating all capabilities. Your agent can check `list_supported_ai_providers` first, select the optimal model based on cost or performance metrics (using `get_pricing_information`), and execute the call—all through one single query.

10 Tools in the CometAPI MCP for Generative AI Pipelines

These tools let your agent perform everything from generating images and transcribing audio to monitoring account usage and listing supported models.

#	TOOL	DESCRIPTION
01	`create_ai_chat_completion`	Generates natural language text responses using various LLMs like GPT-4 or Claude.
02	`check_api_health`	Verifies the operational status of the connected API endpoints.
03	`convert_text_to_speech`	Converts written text into an audio file format.
04	`transcribe_audio_to_text`	Takes an uploaded audio file and returns the transcribed text content.
05	`generate_ai_image`	Creates a unique image based on a descriptive text prompt.
06	`get_current_user`	Retrieves the authenticated user's profile metadata.
07	`get_pricing_information`	Pulls detailed billing information for various AI models and services.
08	`get_api_usage_statistics`	Retrieves the account's current usage metrics, including credit consumption and costs.
09	`list_available_ai_models`	Lists every specific AI model supported by the service aggregator.
10	`list_supported_ai_providers`	Lists all major external providers integrated with CometAPI.

See It in Action

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

U Analyze this 5-minute recording of a sales meeting. Summarize the three main action items, list who is responsible for each, and generate a visual diagram showing the process flow.

Meeting Summary & Next Steps

- **Key Takeaway:** The team agreed to pivot focus from Product X to Service Y.
- **Action Items:**

1. [Alex]: Draft new pricing models by Friday.
2. [Sarah]: Schedule the follow-up demo with the client next week.
3. [You]: Update the internal wiki with the revised process flow.

A visual representation of the new Service Y workflow is attached here: [diagram_url]

U I need an educational graphic for my blog post about deep learning. It should show three overlapping concepts: RNNs, Transformers, and Attention mechanisms.

Visual Generation Complete!

Here is the high-resolution image illustrating how these concepts overlap in modern neural networks: [image_url]

Tip: The diagram shows that while RNNs were foundational, the Transformer architecture primarily improved parallelization by focusing on Attention mechanisms.

U What's my total credit usage this month and what are the costs associated with running multimodal tasks?

📊 **Account Usage Snapshot**

- **Current Credits Remaining:** 28.15 credits.
- **Average Daily Spend (Last 7 Days):** $0.95/day.
- **Cost Breakdown by Tool:**

* Text Generation: $45.00
* Image Generation: $32.50
* Speech Transcription: $18.90

You're tracking well! Keep an eye on the image generation costs, as they are currently the highest variable expense.

Frequently Asked Questions

01 **How do I find my CometAPI API Key?**

Log in to your account, navigate to the **API Dashboard**, and copy your secret key (sk-...).

02 **Can I generate images with different models?**

Yes! Use the `generate_ai_image` tool and specify the model ID (e.g., 'midjourney' or 'dall-e-3') for your creation.

03 **Does it support voice-to-text?**

Absolutely. The `transcribe_audio_to_text` tool allows you to convert any public audio URL into text using high-performance STT models.

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 `"cometapi": { "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

CometAPI 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 CometAPI. 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	CometAPI MCP
Server ID	`019dd0d4-fef7-7244-8f76-2c539a1efef3`
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/cometapi.