MCP SERVER | NO CODE | CLOUD HOSTED

Groq MCP

Ultra-Fast Inference for Media and Logic.

Groq MCP delivers ultra-fast LLM inference by leveraging LPU hardware acceleration directly through your AI client. It lets you run chat completions on models like Llama 3 and Mixtral with blazing speed, while also handling complex media tasks. You can transcribe audio streams into text, translate non-English speech immediately to English, or force the output into rigid JSON formats for system integration.

A+ Quality Score 100/100

llm-inference | lpu-acceleration | ai-latency | audio-transcription | generative-ai | high-performance-computing

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

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.

Groq MCP

8 tools available
Cloud-hosted on Vinkius

Connect this MCP to your preferred AI client to gain full control over high-speed generative AI and multimodal workflows. Instead of waiting minutes for complex requests, you run everything—from simple text generation to audio processing—at hardware speed using Groq's LPU architecture. You can instruct the agent to transcribe an audio file, then immediately translate that resulting text into English. Need data for a database? Use structured output to force the AI response into perfect JSON format, eliminating messy parsing steps later on. Furthermore, you don't have to worry about model compatibility; you can use tools like list_models and get_model to check exactly what high-speed models are available before running your main chat completions or creating embeddings for context.

Core Capabilities

01 — Execute Ultra-Fast Conversational AI

Run text generation, using chat_completion, against accelerated hardware endpoints supporting Llama and Mixtral.

02 — Process Audio to Text

Transcribe audio files into accurate language transcripts using the transcribe_audio tool.

03 — Translate Spoken Language

Take non-English audio and retrieve immediate text translations exclusively in English via translate_audio.

04 — Generate Structured Data

Constrain AI inference to output only valid JSON format using structured_output, perfect for automating data pipelines.

05 — Embed Text Data

Create high-quality text embeddings using create_embedding for advanced retrieval and context building.

06 — Manage Model Instances

Check available models or retrieve detailed metadata about specific LLMs through list_models and get_model.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and enter your Groq API Key. You'll find the key in your Groq Cloud Dashboard under API Keys.

02 Next, connect it to your AI client—like Cursor or Claude—through Vinkius. Your agent now sees all available high-speed tools.

03 Finally, you prompt your agent with a complex request, and it executes the necessary actions (e.g., transcribe_audio, followed by translate_audio) using accelerated hardware.

The bottom line is that instead of managing separate APIs for speed, media, or structure, everything runs through one unified, blazing-fast connection point.

Built For

This MCP is built for developers and data scientists who hit a wall with standard API latency. If your workflow requires combining fast language generation with media processing or strict data typing, this is what you need. It’s for anyone whose job involves moving raw, messy data into clean, actionable formats instantly.

AI Developer

You're debugging complex LLM prompts and tool-calling logic; Groq helps you test these flows with sub-second latency.

Software Engineer

You need to take an audio recording, transcribe it, and then use the text results to populate a database schema in JSON format directly from your IDE.

Data Scientist

You're comparing different open-source model performances on specialized hardware without having to manage multiple cloud endpoints.

What Changes When You Connect

01 You get immediate results when generating text. Using chat_completion means you're not stuck waiting on slow endpoints; responses arrive almost instantly, letting you build real-time applications.

02 Your data pipelines become reliable. Instead of hoping the AI gives readable output, using structured_output forces it into perfect JSON, making post-processing trivial and bug-free.

03 You handle global content without friction. If you need to process audio from a non-English speaker, combine transcribing with translate_audio to get immediate English text.

04 Context retrieval is fast and accurate. By running create_embedding first, your agent can pull relevant knowledge from massive datasets quickly, ensuring the LLM responds with highly specific information.

05 Model management happens in context. You don't guess which model works best; you use list_models to check availability before initiating a complex workflow.

Real-World Applications

Analyzing international meeting transcripts

An operations team member records an audio meeting in Mandarin. They ask their agent to first transcribe the entire file using transcribe_audio, and then immediately run translate_audio on that transcript to get actionable English notes.

Building a structured knowledge base

A data scientist uploads 10 research papers. They use create_embedding to index the content. Later, they ask their agent a question and retrieve the answer using chat_completion, grounded by the indexed context.

Automating form submission data

A developer needs an agent to process user input text about a new product. They use structured_output to force the AI to return a clean JSON object containing specific fields like 'product name,' 'price,' and 'category' for immediate API insertion.

Testing model capabilities pre-launch

A product team needs to know if their new agent can handle different models. They use get_model to check the metadata and context window size of Mixtral before running a final, high-stakes chat completion test.

Patterns to Avoid

Handling structured data manually

✗ AVOID

Asking an agent general questions and then spending 20 minutes writing Python code to parse the resulting text block into keys and values.

✓ INSTEAD

Just use structured_output. Tell your agent exactly what JSON format you need, and it gives you clean, ready-to-use data every single time.

Ignoring audio source languages

✗ AVOID

Trying to run a simple transcription tool on Spanish audio and getting gibberish because the model wasn't designed for translation.

✓ INSTEAD

Use the specialized translate_audio tool. It handles both the initial transcription and the immediate cross-lingual conversion into English.

Relying on general purpose APIs

✗ AVOID

Using a standard, non-accelerated API for chat completions, resulting in noticeable delays that break the user's flow.

✓ INSTEAD

Connect this MCP. The LPU acceleration through chat_completion slashes latency down so fast it feels instantaneous.

The Right Fit

Use this MCP if your workflow involves combining multiple data types or demanding perfect structure. If you need to go from spoken word (audio) to text, and then translate that text into a specific language, you need the audio processing tools here. Similarly, if any piece of LLM output needs to feed directly into a database or another service, structured_output is non-negotiable because it guarantees clean JSON. Don't use this MCP if your only task is writing a simple email draft; then a basic text completion tool

will do fine. But if that 'simple email' requires you to first summarize an attached audio file and embed the result into a knowledge base, you need everything Groq offers.

Dealing with messy data streams today is brutal.

You record an international meeting. You export the raw MP3. Then, you have to upload it to one tool for transcription, download a massive text file, and finally copy-paste that whole thing into another service just to get an English summary. It's a painful loop of uploads, downloads, and manual copy/pasting across three different tabs.

With this MCP, the process collapses. Your agent takes the audio file once. It handles transcription with transcribe_audio and then immediately translates the text using translate_audio, giving you a clean English transcript in minutes, not hours.

Groq gives your agents perfect data structure.

Before this, when an LLM gave you information—say, about a product—you'd get paragraphs of text. You'd have to manually search for the price, the name, and the category, then copy those three pieces into your internal form.

Now, using structured_output, you ask for the data once. The agent responds with flawless JSON that is ready to be piped directly into your system. No parsing required.

Groq: 8 Powerful Tools for Accelerated Inference

These tools let you perform every step of a complex AI workflow. You can chat, transcribe media, generate embeddings, or force structured JSON output with simple commands.

#	TOOL	DESCRIPTION
01	`chat_completion`	Generates a response using Llama, Mixtral, or Gemma models with ultra-fast inference speed.
02	`list_models`	Retrieves a list of all available high-speed language models you can use.
03	`get_model`	Fetches specific metadata and details about any particular model.
04	`create_embedding`	Converts text into vector embeddings, which allows your AI agent to understand relationships between pieces of text.
05	`transcribe_audio`	Takes an audio file and converts the spoken word into a written transcript.
06	`translate_audio`	Converts non-English audio files into English text translations.
07	`moderate_content`	Checks any given content to determine if it violates safety guidelines.
08	`structured_output`	Forces the AI model to generate output that strictly adheres to a predefined JSON data structure.

See It in Action

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

U Ask llama3-70b: 'Write a python function to scrape a website.'

Inference complete! Llama 3 response: 'Here is a simple python function using BeautifulSoup and requests to scrape data...' [Blazing-fast response delivered via Groq LPU].

U Transcribe this audio meeting: https://example.com/meeting.mp3

Transcription started! I'm using Groq optimized Whisper large-v3 model to parse your meeting audio. I'll provide the full timestamped text for you in just a few seconds.

U Get model info for 'mixtral-8×7b-32768'

Retrieving model metadata... Mixtral-8×7b-32768 is a high-performance LLM with a context window of 32,768 tokens. It supports chat completions and tool-calling on Groq's LPU architecture.

Frequently Asked Questions

01 **Does Groq MCP support multiple file types?**

Yes, this MCP handles both text and audio files. You can use transcribe_audio on an MP3 or WAV file and then process the resulting text.

02 **How do I make sure the output is usable in my database using Groq?**

Use structured_output with the tool. By defining a rigid JSON schema, you guarantee that the AI response will match the exact format your database expects.

03 **Can Groq MCP handle audio translation and transcription together?**

Absolutely. You can chain these operations. First, transcribe_audio captures the speech, and then translate_audio takes that output to provide a clean English text file.

04 **Why should I use Groq MCP for embeddings instead of another service?**

Groq provides extremely fast context generation. Using create_embedding ensures your knowledge base is updated and searchable with minimal latency, keeping your agents responsive.

05 **What models can chat_completion access on Groq MCP?**

The chat_completion tool supports several high-performance open-source models, including Llama 3, Mixtral, and Gemma, all optimized for speed.

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 `"groq": { "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

Groq 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 Groq. 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	Groq MCP
Server ID	`019d75ab-f54d-7016-b10c-0ed40a186e8c`
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/groq.