MCP SERVER | NO CODE | CLOUD HOSTED

Deepgram MCP for AI Agents

Process Audio Streams and Manage Transcription Keys

Deepgram gives your AI agents full control over high-speed audio processing. It lets you transcribe remote audio streams (WAV, MP3) using the Nova-2 model and generate professional speech from raw text using Aura voices. Beyond transcription and synthesis, this MCP handles core account functions: managing API keys, tracking usage across projects, monitoring credit balances, and inviting team members.

A+ Quality Score 100/100

speech-to-text | text-to-speech | transcription | voice-ai | natural-language-processing | audio-processing

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

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.

Deepgram MCP

10 tools available
Cloud-hosted on Vinkius

Deepgram lets your AI agent handle complex audio workflows right inside your chat environment. You don't have to leave your development tool or dashboard just because you need to transcribe a recording or synthesize voiceover content. Instead of manually uploading files to an external portal and waiting for batch processing, your agent sends the request directly. It can pull transcriptions from remote URLs—supporting formats like WAV or MP3—using their fast Nova-2 model. For speech generation, it converts plain text into high-fidelity audio streams using Aura voices.

And since building these systems means managing access and costs, this MCP also lets you manage the underlying infrastructure. You can list project API keys, create new ones with specific scopes, or check current credit balances to ensure your pipelines never drop due to limits. Because Vinkius hosts and manages these connections, it brings all these critical audio AI functions—from transcription to key management—into one place for your agent to access.

Core Capabilities

01 — Transcribe Audio from URLs

Send automated requests to transcribe audio files hosted at a specific URL using the Nova-2 model.

02 — Generate Speech from Text

Convert raw text into high-quality, natural-sounding speech audio streams using Deepgram's Aura voices.

03 — Monitor API Usage and Limits

Analyze detailed usage statistics for a project, mapping transcription time and TTS byte consumption over custom date ranges.

04 — Manage Project Access Keys

List existing Deepgram access keys or create entirely new ones with specific scopes and expiration dates.

05 — Manage Team Membership

View all team members associated with a project or send invitations to expand the development team's access.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and input your Deepgram API Key, which you find in the Deepgram Console under Settings > API Keys.

02 Your AI client uses natural conversation to determine if it needs to transcribe audio from a URL or generate speech from text. It then executes the required tool calls against Deepgram's services.

03 The agent returns the requested output—either structured transcription data, binary audio streams, or usage reports—directly in your chat interface.

The bottom line is that you manage complex, multi-step audio AI tasks and billing oversight entirely through conversation.

Built For

This MCP serves developers building commercial voice applications, data engineers auditing large-scale media pipelines, and operations teams managing critical infrastructure costs. If your job involves turning spoken word or text into structured data, this is for you.

AI Developer

Uses the MCP to test STT/TTS models and manage project API keys without switching development environments.

Data Engineer

Audits transcription volumes across multiple projects and manages complex audio pipelines using natural language commands.

Product Manager

Monitors real-time audio AI usage data to verify accuracy and predict resource needs before launching new features.

What Changes When You Connect

01 Get instant transcriptions: Instead of uploading files, you simply provide a URL, and the agent uses `transcribe_url` to get text from audio streams.

02 Save time on content creation: Use `speak_text` to convert any block of copy into natural-sounding speech audio instantly for voiceovers or alerts.

03 Maintain security and control: You can use `list_keys`, `create_key`, and `delete_key` to manage access credentials directly within your workflow.

04 Keep costs under wraps: Check project limits and credit balances using `get_balances` and `get_usage` before running expensive pipelines.

05 Scale development teams: Use the team management tools, like listing members or sending invites via `send_invite`, so multiple people can work on the same audio projects.

Real-World Applications

Analyzing Customer Feedback Recordings

A product manager needs to analyze 50 hours of customer call recordings. Instead of manually processing each file, they ask their agent to use `transcribe_url` on a batch of recording links. The agent returns structured text, allowing the PM to instantly categorize pain points.

Building Voice-Activated Tutorials

A developer needs to create an internal training module with custom voice narration. They feed marketing copy into `speak_text` and generate MP3 assets on demand, which are then integrated into the application's help flow.

Auditing Multi-Tenant Billing

An operations team needs to check if a specific department exceeded its allocated audio budget last quarter. They use `get_usage` with precise date filters to map out exactly where the consumption occurred, preventing unexpected overages.

Onboarding New Development Staff

A lead engineer needs to grant a new team member access to the main audio project. They use `list_members` to confirm who is on board and then execute `send_invite` to securely onboard the newcomer.

Patterns to Avoid

Manually checking usage reports

✗ AVOID

Having to log into a separate dashboard, navigate through date pickers, and manually download CSV files just to see how much audio was processed last month.

✓ INSTEAD

Just ask your agent to run `get_usage` with the desired start and end dates. It pulls the data directly and reports it back in natural language.

Using default, root API keys

✗ AVOID

Giving every developer a single master key because 'it's easier.' If that key gets leaked, your entire system is compromised.

✓ INSTEAD

Use `create_key` to generate specific access keys with the minimum required scope and set an expiration date. This limits damage if the key leaks.

Ignoring team roles for projects

✗ AVOID

Adding every person who needs 'access' to a project, leading to unnecessary permissions and security risks.

✓ INSTEAD

First, use `list_members` to review current access. Then, only run `send_invite` when absolutely necessary.

The Right Fit

Use this MCP if your workflow involves both converting spoken audio into text *and* generating synthetic voice content, coupled with the need for robust, programmatic key and usage management. If you are only performing one task—say, just transcribing simple files without worrying about billing limits or team invites—you might find a simpler single-function tool adequate. However, if your system requires tracking costs (like using `get_usage`) while also managing the credentials (`create_key`), this MCP is built for that complexity. Don't use it if you only need simple file uploads; you need programmatic URL or stream handling.

Deepgram MCP: Managing Audio Transcription Pipelines with Deepgram

Right now, managing large-scale voice AI pipelines is a headache. You have to jump between multiple consoles—one for uploading audio files, another for checking API usage, and yet another just to generate the necessary access keys. This constant switching breaks your flow, introduces latency, and makes auditing costs a nightmare.

With this MCP, you talk to your agent, and it handles the whole pipeline. You can ask it to transcribe an audio stream from any URL using `transcribe_url`, get the text, and then immediately use that resulting text with `speak_text` for a voiceover—all in one turn. It just works.

Deepgram MCP: Controlling API Keys and Project Access

Keeping your audio infrastructure secure means managing keys, scopes, and who has access to what data. Manually listing all team members or having to create a new key every time someone leaves is slow and error-prone.

This MCP lets you enforce granular control instantly. Need an audit? Use `list_keys` to see everything active. New hire? Use `send_invite`. You maintain total security visibility without ever leaving your development environment.

Deepgram: 10 Tools for Comprehensive Audio Processing

These tools let you control every aspect of your audio workflow—from transcribing remote streams to managing API credentials and tracking detailed usage statistics.

#	TOOL	DESCRIPTION
01	`send_invite`	Sends an invitation to add a team member to a Deepgram project.
02	`list_keys`	Retrieves a list of all active API keys for the current project.
03	`get_balances`	Checks and reports the remaining credit balance for a specific Deepgram project.
04	`create_key`	Generates a new API key, allowing you to specify its comment, required scopes, or an optional expiration date.
05	`delete_key`	Permanently revokes and deletes an existing API key for security purposes.
06	`list_members`	Shows all current team members who have access to the project.
07	`list_projects`	Provides a comprehensive list of all Deepgram projects associated with your account.
08	`transcribe_url`	Transcribes audio content from any accessible URL, supporting formats like WAV, MP3, and FLAC using the Nova-2 model.
09	`speak_text`	Converts provided text into high-fidelity speech audio data using Deepgram's Aura voices.
10	`get_usage`	Retrieves detailed API usage statistics for a project, allowing filtering by date ranges.

See It in Action

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

U Can you transcribe this podcast audio and tell me what the main points were?

Transcription Report (Nova-2 Model)

- **Source:** https://podcast.com/ep14
- **Time Processed:** 35 minutes, 12 seconds
- **Key Takeaways:**

* AI is moving past chatbots into agentic workflows.
* The best data pipelines require constant usage monitoring.
* Focus on key management to secure your infrastructure.

(Confidence Score: 94%)

U I need a high-fidelity voiceover for this paragraph about our new product.

Audio Generation Successful!

- **Text Input:** "The future of AI is agentic and requires secure, scalable audio processing."
- **Voice Used:** Aura Asteria (English)
- **Output File:** `audio_v2.mp3` (Binary stream provided for download).
- **Notes:** The synthesized audio maintains a natural pace and high fidelity, ready for immediate use in your application's UI.

U What was my deepgram usage last month and do I have enough credit?

Checking Deepgram usage for the period 2026-05-01 to 2026-05-31...

Usage Summary:

- Total Transcription Time: 480 minutes (up from 390 last month).
- Total TTS Generation Bytes: 18 MB.
- Current Balance: **$45.50 credits remaining.**

Looks good; your current spending keeps you well within the project's overall limits.

Frequently Asked Questions

01 **How can I use the Deepgram MCP to transcribe audio from a website link?**

You just tell your agent the URL and the request. It handles connecting to that remote stream, transcribing the content using Nova-2, and returning clean text data directly into your chat window.

02 **I need synthetic voices for my app; what is the easiest way with Deepgram MCP?**

Simply provide the text you want spoken. The agent uses the `speak_text` tool to generate high-quality MP3 binary data using Aura voices, which you can then embed directly into your application.

03 **How do I manage billing and API keys when using Deepgram MCP?**

The MCP gives you full oversight. You can run `get_usage` to see exactly how much audio was processed over a date range, or use `create_key` to generate new access credentials while keeping the old ones secure.

04 **Does Deepgram MCP help me manage who on my team has access?**

Yes. You can list all current project members using `list_members`, and if you need to add someone, you just send an invitation using the `send_invite` tool.

05 **What if I run out of Deepgram credits? Can the MCP help me track that?**

Absolutely. You can use `get_balances` to check your current credit status and `get_usage` to see spending patterns, so you always know when you'll need a top-up.

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 `"deepgram": { "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

Deepgram 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 Deepgram. 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	Deepgram MCP
Server ID	`019d7583-5370-730b-bee8-0b0adf500b50`
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/deepgram.