

MCP SERVER

NO CODE

CLOUD HOSTED

Delay Time Compensator MCP for AI Agents

Achieving perfect beat synchronization in studio recordings

The Delay Time Compensator MCP calculates precise audio offsets, letting you account for system latency and BPM. Use it to keep rhythmic elements perfectly aligned in your music production or live performance setup.

A+ Quality Score 100/100

bpm

latency

audio-engineering

rhythm

delay-compensation



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

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.

Delay Time Compensator MCP

3 tools available

Cloud-hosted on Vinkius

System latency is a major headache for any producer or engineer. When the time it takes for sound to travel from output back to input (round-trip latency) gets too high, timing falls apart, ruining the groove. This MCP solves that problem by calculating exactly how much delay you need to apply to keep everything perfectly synchronized with your track's tempo.

Instead of manually guessing or fiddling in multiple plugins, you let your agent handle the math. It figures out the precise millisecond adjustment needed for flawless timing. If latency forces an awkward rhythmic division, it warns you and finds the next shortest playable note that still works. You can even analyze how the delay is messing with the *feel* of a beat—is your 16th note actually sounding like a 32nd note? All this data is available through Vinkius, letting you get perfect timing offsets directly into your workflow without leaving your main DAW.

Core Capabilities

01 — Calculate the necessary delay offset

Determine the precise millisecond adjustment required to cancel out hardware or system latency for any given BPM.

02 — Analyze perceived rhythmic shifts

Get a human-readable description of how latency is altering the natural rhythm and feel of a note.

03 — Identify playable rhythmic divisions

Find the shortest available musical division that can accommodate current system latency without causing timing errors.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/mcp/delay-time-compensator — connect your AI agent in three steps.

- 01** You feed your agent three inputs: your track's BPM, the rhythmic value you are working with (e.g., eighth notes), and your measured hardware latency in milliseconds.
- 02** The MCP runs these values through its core calculations to determine the exact time adjustment needed to maintain alignment.
- 03** Your agent returns a single number—the required delay compensation in milliseconds—that you apply directly in your audio plugin.

The bottom line is, it takes complex timing math and gives you one simple, accurate number for your DAW settings.

Built For

This MCP is built for professional musicians, recording studio engineers, and live performance artists. If your job requires perfect rhythmic accuracy across multiple hardware sources, this tool saves you hours of frustrating manual timing adjustments.

Audio Engineer

Needs to ensure that external gear (like effects processors or specialized mics) doesn't introduce latency that throws off the main mix tempo.

Music Producer

Requires precise timing offsets when incorporating complex, time-sensitive elements from different sources into a single track.

Live DJ/Performer

Must instantly compensate for fluctuating hardware latency during live sets to maintain perfect beat synchronization across multiple decks and effects units.

What Changes When You Connect

-
- 01 Eliminate timing drift: Use `calculate_delay_adjustment` to get the exact millisecond setting needed, so your rhythm always stays locked to tempo.

 - 02 Diagnose rhythmic problems: Run `analyze_rhythmic_shift` when things feel 'off' and get a clear explanation of how latency is affecting the groove.

 - 03 Avoid unplayable rhythms: If latency is too high, use `find_next_valid_division` to know what the shortest workable note division actually is.

 - 04 Save setup time: Never manually calculate delay offsets again. Your agent handles all the complex timing math instantly.

 - 05 Improve mix quality: By guaranteeing perfect rhythmic alignment, your final recordings sound tighter and more professional.
-

Real-World Applications

My external reverb is introducing too much lag.

A producer notices that when they run their main snare through an outboard reverb unit, the impact feels sloppy. They ask their agent to calculate the necessary delay adjustment based on BPM and latency, getting a precise millisecond number to input into the unit's compensation dial.

I need to know what my DAW's timing limitations are.

An audio engineer wants to know if their current latency exceeds the playable range for quarter notes. They query `find_next_valid_division` and get confirmation of the shortest acceptable rhythmic division, allowing them to adjust gear accordingly.

The beat is sounding too rushed because of gear lag.

A live performer realizes that high system latency is making their sixteenth notes sound like thirty-second notes. They ask the MCP to analyze the rhythmic shift, which confirms the distortion and suggests a safer minimum note division they can use.

Patterns to Avoid

Treating latency as a simple constant**X AVOID**

Assuming that because your DAW is fast, you can just set the delay manually without accounting for the specific round-trip time of all connected hardware.

✓ INSTEAD

Always use `calculate_delay_adjustment` first. This tool factors in the precise latency measurement to give you a true compensation value.

Ignoring rhythmic constraints**X AVOID**

Forcing a complex rhythm (like a dotted eighth note) when latency is high, resulting in an unmusical and jarring timing failure.

✓ INSTEAD

If your intended rhythm fails, check `find_next_valid_division`. This tells you the actual shortest playable division that respects your current hardware limitations.

Ignoring perceived differences**X AVOID**

Not noticing that even if the timing is technically 'correct,' it still sounds unnatural or rushed to the human ear.

✓ INSTEAD

Use `analyze_rhythmic_shift` to get a description of how latency is affecting your *perceived* groove, not just the raw math.

The Right Fit

You should use this MCP if you deal with physical audio chains where latency compensation is non-negotiable. Specifically, if you're integrating external hardware (outboard effects, specialized mics) or working in a live setting where timing must be rock solid against the BPM. Don't use it if your problem is purely compositional—for instance, if you just need to change the *tempo* of a loop; that requires a time-stretching tool. Use this MCP when the issue is specifically about fixing the time difference between signals getting out and coming back in.

Fixing Beat Synchronization with Delay Time Compensator for Audio Engineers

Manually calculating delay offsets across multiple plugins or hardware units is a nightmare. You're constantly toggling between meters, reading spec sheets, and running complex math in your head just to keep the kick drum locked perfectly with the snare.

With this MCP, you feed your agent the simple variables—BPM, latency, rhythm—and it spits out one number: the exact delay compensation value. You spend zero time calculating; you spend all your time making music.

Delay Time Compensator for Analyzing Rhythmic Integrity in Production

Without specialized tools, you only know the timing is 'off.' You can't pinpoint *how* it sounds off—is it too fast, or does it sound like a different note entirely? The manual process leaves you guessing.

This MCP allows you to diagnose the problem. It doesn't just give a number; it tells you what the latency is doing to your beat's actual feel, letting you fix the root cause instead of masking symptoms.

3 Tools for Delay Time Compensation and Rhythmic Analysis

These tools let you calculate necessary time adjustments, diagnose timing issues, and find safe rhythmic divisions for flawless music production.

#	TOOL	DESCRIPTION
01	<code>analyze_rhythmic_shift</code>	Provides a textual breakdown of how system delay is making the perceived rhythm sound different from its intended note value.
02	<code>calculate_delay_adjustment</code>	Calculates the precise time adjustment needed in milliseconds to offset measured hardware latency at a specific tempo.
03	<code>find_next_valid_division</code>	Identifies the shortest musical note division that is still playable and larger than your current system latency.

See It in Action

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

U My BPM is 120, I'm playing eighth notes, and my latency is 50ms. What should my delay setting be?



Compensation Calculation:

Based on your input (BPM: 120, Rhythm: Eighth Notes, Latency: 50ms), the required adjustment is:

- **Required Delay Adjustment:** 250ms

Please set your effect unit delay to this value for perfect alignment.

U At 140 BPM, how does a 30ms latency affect my sixteenth notes?



Analysis of Rhythmic Shift

The measured 30ms latency at 140 BPM significantly alters the perceived rhythm. Your intended sixteenth note will actually sound like an extended thirty-second note. This shift can pull the overall groove out of time with your main track.

U I have 100ms of latency at 180 BPM. What is the shortest note division I can use?



Rhythmic Constraint Analysis

Given a 100ms latency and 180 BPM, using a quarter note is safe.

- **Shortest Valid Division:** Quarter Note
- **Minimum Safe Latency:** < 100ms

Trying to use an eighth note would result in audible timing errors.

Frequently Asked Questions

01 What is the Delay Time Compensator MCP for AI Agents actually used for?

It fixes timing issues caused by hardware or system delay. If your gear makes sound lag, this MCP calculates the exact millisecond offset you need to apply to keep your rhythm perfectly synced.

02 Can I use the Delay Time Compensator MCP when working with multiple audio sources?

Yes. It handles complex scenarios by calculating offsets relative to a master BPM, ensuring every source remains in sync even if they pass through different effects chains.

03 If my latency is too high, what does the Delay Time Compensator MCP tell me?

It identifies the shortest playable rhythmic division. This prevents you from trying to use a note value that your system can't accurately reproduce due to lag.

04 Is the delay adjustment calculation specific to my BPM or does it work generally?

The calculation is always tied to the current tempo (BPM) you provide. Changing the tempo means recalculating the necessary offset for accurate timing.

05 How do I find out if latency is changing the 'feel' of my music with this MCP?







You run the rhythmic shift analysis tool. It gives a plain-language description, telling you if your 16th note sounds stretched or compressed due to the timing lag.

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 <code>"delay-time-compensator": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

Delay Time Compensator 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](https://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 Delay Time Compensator. 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	July 2026
MCP Server	Delay Time Compensator MCP
Server ID	019f1f33-b2fa-706f-a48c-ea64d0da069d
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/delay-time-compensator.