

MCP SERVER

NO CODE

CLOUD HOSTED

Tempo Tap Averager MCP for AI Agents

Deriving Accurate Musical Tempo and Rhythm Analysis from Raw Taps

Tempo Tap Averager calculates precise Beats Per Minute (BPM) and measures rhythmic stability using raw tap timestamps. It converts a simple sequence of taps into professional, measurable musical metrics, including suggested half/double time tempos and formal musical descriptors.

A+ Quality Score 100/100

bpm

rhythm

tempo

music-theory

audio



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.

Tempo Tap Averager MCP

3 tools available

Cloud-hosted on Vinkius

Dealing with raw audio data or tap inputs can be messy. You've got a list of millisecond timestamps, but what does that actually mean musically? The Tempo Tap Averager solves that. This MCP lets your AI client take any sequence of taps and instantly calculate the precise tempo (BPM). It doesn't just give you one number; it suggests how that rhythm would sound in half-time or double-time. Furthermore, if you need to know how steady the beat is—if there are noticeable fluctuations or outliers—it generates a stability score for you. If you're writing music theory pieces, it even maps those BPMs to formal musical labels like 'Vivace.' Connecting this MCP via Vinkius gives your agent access to these deep rhythmic insights without needing complex audio processing libraries. You just give the taps; the system returns actionable musical data.

Core Capabilities

01 — Derive Beat Per Minute (BPM) from Taps

It takes a list of tap timestamps and calculates the exact BPM, providing suggested tempos for both half-time and double-time variations.

03 — Translate BPM into Musical Terms

The system accepts a specific BPM and returns the corresponding formal musical descriptor, like 'Allegro' or 'Largo'.

02 — Assess Rhythmic Stability

You get an objective stability score (0 to 1) that tells you how consistent the tapped rhythm is, plus a count of detected timing outliers.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/mcp/tempo-tap-averager — connect your AI agent in three steps.

- 01** You feed your agent a list of tap timestamps (in milliseconds) that represent the rhythm you want analyzed.
- 02** The MCP processes these raw data points, first calculating the base BPM and checking for rhythmic consistency by looking for timing deviations. It also checks if there are specific musical descriptors associated with that tempo.
- 03** Your AI client receives a structured output containing the precise BPM, a stability score, outlier counts, and any relevant musical labels.

The bottom line is you turn raw tap data into quantified musical metrics usable for composition or analysis.

Built For

This MCP is essential for music technologists, rhythm game developers, and audio engineers. If your job involves taking recorded rhythmic input—like a sequence of taps or drum hits—and needing to quantify its musical value, this tool saves hours of manual analysis.

Music Technologist

Determining if raw user tap data is musically consistent enough for an interactive piece requires knowing the stability score and calculating the underlying BPM.

Audio Engineer

Analyzing recorded rhythm tracks to determine their natural tempo, checking for timing drift or unevenness with ``evaluate_rhythm_consistency`` before mixing.

Music Educator

Converting a student's physical tap performance into formal musical terms and quantifiable metrics like BPM to provide structured feedback.

What Changes When You Connect

-
- 01 Get immediate, quantifiable musical metrics. Instead of guessing the tempo, use `calculate_bpm` to get the exact BPM and suggested half/double-time tempos.

 - 02 Pinpoint rhythmic weaknesses instantly. The stability score from `evaluate_rhythm_consistency` tells you exactly how consistent a performance is, identifying specific timing outliers for correction.

 - 03 Add professional context to your data. Use `get_tempo_metadata` to map any calculated BPM (like 140) directly to its musical label (e.g., Allegro), making your results sound polished.

 - 04 Eliminate manual calculation time. Your agent handles the complex math of tempo variation, freeing you from tedious spreadsheet work and repetitive analysis.

 - 05 Understand rhythm variations deeper than basic timing. The system doesn't just count taps; it assesses overall pattern stability, which is crucial for composing interactive music.
-

Real-World Applications

Analyzing a Drummer's Practice Session

A user uploads raw tap data from a drum practice session. They ask their agent to check the rhythm's consistency, and the agent runs `evaluate_rhythm_consistency`, returning a score of 0.85 and flagging three major timing outliers that need correction.

Setting Tempo for an Interactive Game

A game developer inputs taps representing the desired beat pattern. The agent uses `calculate_bpm` to determine the precise tempo (120 BPM) and receives suggested tempos, allowing the developer to set difficulty levels accurately.

Annotating Musical Scores

A music student provides a sequence of beats. They ask what musical term applies, and the agent uses ``get_tempo_metadata`` on 140 BPM, instantly labeling it 'Allegro' for their coursework.

Comparing Performance Variations

A teacher wants to compare two different recorded performances. They run both through ``calculate_bpm`` to see the average tempo and then use ``evaluate_rhythm_consistency`` on each set of taps to grade their stability.

Patterns to Avoid

Treating Taps as Simple Counts

X AVOID

Assuming 10 taps means a perfect tempo. This ignores variations in timing and overall musical context, leading to inaccurate BPM estimations.

✓ INSTEAD

Don't just count the taps. Use ``calculate_bpm`` on the full set of millisecond timestamps for an accurate average. Then use ``evaluate_rhythm_consistency`` to check if those timings are actually stable.

Ignoring Tempo Context

X AVOID

Knowing a piece is 120 BPM, but not knowing how that relates to half-time or double-time. This limits the musical interpretation of the rhythm.

✓ INSTEAD

Always use ``calculate_bpm``. It doesn't just give you the primary tempo; it suggests both the half and double time versions, giving you a complete view of rhythmic possibilities.

Manual Lookups for Musical Labels

X AVOID

Having to cross-reference a calculated BPM (e.g., 140) in a separate dictionary or guide to find the correct musical term ('Allegro'). This is slow and prone to error.

✓ INSTEAD

Use ``get_tempo_metadata``. Give it your exact BPM, and it instantly returns the proper musical label like 'Vivace,' keeping your workflow fast and accurate.

The Right Fit

You should use Tempo Tap Averager if you need to quantify rhythm. If your goal is simply to count how many times a beat occurred, don't bother with this MCP; just use basic counting methods. However, if you need to know *how steady* those beats were or what the underlying musical context means, this is your tool. Specifically, if you are comparing multiple performances, run both sets of taps

through `evaluate_rhythm_consistency` first. If the performance is stable but you still need to confirm the tempo, use `calculate_bpm`. Only use `get_tempo_metadata` when you have already determined a final BPM and just need the corresponding musical label for documentation.

Tempo Tap Averager MCP: Solving Rhythmic Consistency in Music Theory

Before this MCP, analyzing raw tap data was a manual nightmare. You'd have to copy timestamps into spreadsheets, calculate the difference between adjacent points, and then manually spot deviations. Finding if a rhythm was truly steady required hours of subjective listening and tedious outlier flagging.

Now, you feed the agent the sequence, and it handles all that math for you. It instantly runs `evaluate_rhythm_consistency`, giving you a single stability score and listing exactly which taps deviated from the norm. You get definitive data points instead of guesswork.

Tempo Tap Averager MCP: Mapping BPM to Formal Musical Terms

When documenting musical pieces, you often need more than just a number; you need the formal name. If your calculated tempo was 140 BPM, you'd traditionally have to consult a music theory book or online chart to find out if that meant 'Allegro,' 'Vivace,' or something else entirely.

This MCP eliminates that lookup process. You calculate the exact BPM, and with `get_tempo_metadata`, your agent returns the correct musical term immediately. Your output is instantly ready for publication.

3 Tempo Tap Averager Tools for Advanced Rhythm Analysis

Use these tools to calculate tempo variations, check rhythmic consistency, and identify formal musical descriptors from any set of taps.

#	TOOL	DESCRIPTION
01	<code>calculate_bpm</code>	It calculates the Beats Per Minute (BPM) from tap timestamps, giving you standard tempo, suggested half-time, and double-time versions.
02	<code>evaluate_rhythm_consistency</code>	This tool analyzes a tapped rhythm, providing a stability score between 0 and 1 and identifying the total number of timing outliers.
03	<code>get_tempo_metadata</code>	You give it a BPM, and it returns the corresponding formal musical label or term associated with that tempo.

See It in Action

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

U I tapped out a beat sequence: [100, 350, 600, 850]. What's the tempo and how steady was it?



Rhythm Analysis Report

- **Calculated BPM:** 200.0 (This is a strong metric.)
- **Stability Score:** 0.98 (Excellent consistency! No major drift detected.)
- **Outliers Detected:** 0

Suggested Tempos: Half-time: 100 BPM | Double-time: 400 BPM

U What musical label should I use for a tempo of 165 BPM?



The musical term corresponding to 165 BPM is **Vivace**.

Definition: Vivace means 'lively' or 'fast.' This suggests the piece needs high energy and rapid movement in its performance.

This label helps set expectations for performers reading your score.

U Check this tap sequence: [0, 450, 1200, 1800]. Is it rhythmically stable? What's the BPM?



Tempo & Stability Metrics

- **Calculated BPM:** 133.33.
- **Stability Score:** 0.65 (Moderate stability; there are noticeable timing variations.)
- **Outliers Detected:** 2 (The gaps between the first and second taps, and the last two, show significant deviation from the average rate.)

Frequently Asked Questions

01 How can Tempo Tap Averager help me analyze raw drum recording data?

Tempo Tap Averager quantifies your rhythm. Instead of just listening, you get a stability score and precise BPM figures that pinpoint exactly where the timing is off or if the tempo drifts over time. This gives you objective feedback for editing.

02 Does Tempo Tap Averager tell me what musical section I'm in?

No, it doesn't identify sections, but it can give context to your BPM using ``get_tempo_metadata``. If you know the tempo is 140 BPM, it tells you that the appropriate musical term is 'Allegro,' which helps label and categorize the music.

03 Is Tempo Tap Averager better than just counting taps for setting a beat?

Yes. Simple counting ignores timing variance. This MCP uses your exact millisecond timestamps to calculate BPM, giving you an average rate that is far more accurate and professional than a simple count.

04 What if my rhythm has slight variations in tempo? Will Tempo Tap Averager catch it?

Absolutely. That's what the stability score does. It rates your rhythm's consistency (0 to 1) and counts specific outliers, letting you know exactly how much variation exists in the taps.

05 Can I use Tempo Tap Averager for music education purposes?







Yes. You can input a student's performance data, calculate its precise BPM with ``calculate_bpm``, and then give them objective feedback on stability using ``evaluate_rhythm_consistency``.

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>"tempo-tap-averager": { "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

Tempo Tap Averager 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 Tempo Tap Averager. 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	Tempo Tap Averager MCP
Server ID	019f1f35-b848-70ce-91ae-bec0717b8487
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/tempo-tap-averager.