MCP SERVER | NO CODE | CLOUD HOSTED

Song Structure Timer MCP

Map out your song's rhythm and timing perfectly.

Song Structure Timer calculates musical timing instantly. Input your Beats Per Minute (BPM) and time signature, and it tells you exactly how long every beat, bar, or section of your song lasts. Producers use this to build accurate reference tracks for recording, plan complex arrangements, or prepare files for mastering.

A+ Quality Score 100/100

bpm | music-production | timing | songwriting | audio

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/song-structure-timer

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.

Song Structure Timer MCP

3 tools available
Cloud-hosted on Vinkius

Need to know the precise timing for a chorus or verse? This MCP calculates exact musical timings for producers and musicians. You feed it the BPM and meter, and it instantly gives you the duration of individual beats and bars. Forget guessing; this tool accurately determines how long specific sections should run. You can calculate the length of any segment, then generate a complete, timestamped blueprint for your entire track. It's essential for anyone doing recording reference or arrangement planning. All these calculations are managed through Vinkius, making complex timing data accessible to your agent from anywhere.

Core Capabilities

01 — Determine rhythm metrics

Calculate the precise time in seconds for both individual beats and full bars based on BPM.

02 — Measure song sections

Figure out the exact length of any specific musical section, like a verse or bridge.

03 — Generate structural timeline

Create a full, organized breakdown of an entire song, complete with timestamps for every major part.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/song-structure-timer** — connect your AI agent in three steps.

01 Input your musical parameters, including the Beats Per Minute (BPM) and the time signature.

02 The MCP runs these metrics to determine beat durations, bar lengths, or total section timings.

03 You receive precise measurements—seconds per beat, seconds per bar, or a full timeline breakdown.

The bottom line is that you get immediate, mathematically accurate timing data for any part of your song, removing guesswork from the process.

Built For

This MCP solves problems for professional audio engineers and recording artists. It's for anyone whose job depends on precise rhythm—from initial demoing to final mastering prep. If you spend time measuring timings manually, this is for you.

Music Producer

Uses the MCP to plan complex arrangements and build structural timelines before recording any vocals or instruments.

Audio Engineer

Calculates exact bar durations for reference tracks, ensuring all recorded sections align perfectly with the intended song structure.

Songwriter/Composer

Determines optimal section lengths and rhythm metrics to ensure their lyrical ideas fit musically in a specific time signature.

What Changes When You Connect

01 Know the exact length of every piece. Instead of guessing, you use calculate_section_duration to know precisely how many seconds a bridge or verse needs.

02 Build professional reference tracks faster. Use get_bar_rhythm_metrics to get accurate beat and bar timing metrics instantly, so your tracking session starts on time.

03 Perfect arrangement planning. Generate a complete structural blueprint using generate_song_timeline. You'll see the whole song laid out with timestamps from intro to outro.

04 Eliminate manual math errors. This MCP handles complex BPM and meter calculations that used to take hours of spreadsheet work, giving you reliable data immediately.

05 Speed up mastering prep. By knowing precise timing metrics, you ensure your final stems are cut and labeled correctly for the mastering engineer.

Real-World Applications

The chorus is too long in the mix

A producer finds that their main chorus section is running 3 seconds over the intended beat. They use calculate_section_duration, inputting the BPM and desired bar count, and immediately know they need to trim exactly X milliseconds.

Starting a new demo track

A songwriter has lyrics for three distinct parts (Intro, Verse, Chorus). They run generate_song_timeline using the target BPM and bar counts. The output provides a clear timeline, showing exactly where each section starts and ends.

Preparing an instrumental loop

An audio engineer needs to make sure an 8-bar loop syncs perfectly with a master track that is in 4/4 time. They use get_bar_rhythm_metrics to confirm the bar duration, guaranteeing the loop locks into place.

Quickly verifying tempo changes

A composer writes an arrangement that shifts tempo twice. Instead of manually calculating the seconds lost or gained, they use generate_song_timeline, and the MCP handles all the complex timing transitions automatically.

Patterns to Avoid

Using generic measuring tools

✗ AVOID

Trying to calculate song length using a general word processor or basic spreadsheet formula that doesn't account for BPM changes or time signatures.

✓ INSTEAD

Use this MCP. Run generate_song_timeline; it handles the complex math of multiple tempo shifts and bar definitions in one step.

Manual beat-by-beat counting

✗ AVOID

Spending hours marking every single quarter note on a digital audio workstation just to figure out if the arrangement is too long or short.

✓ INSTEAD

Run calculate_section_duration. Feed it the section name and bar count, and you get the exact total time in seconds without touching the DAW grid.

Ignoring beat metrics

✗ AVOID

Setting up a loop based only on bar counts, which fails if the BPM changes or if the meter isn't clean 4/4.

✓ INSTEAD

Start with get_bar_rhythm_metrics. Confirm your beats per second before you build anything else. This establishes a rock-solid rhythmic foundation.

The Right Fit

Use this MCP if your problem is purely about *timing* and *structure*. You need to know: 'How many seconds long is X?' or 'When does Y start?'. If you are planning an arrangement, checking tempo shifts, or building reference tracks based on beats per minute, this tool is essential. Don't use it if your problem involves mixing effects (like reverb decay), frequency balancing, or analyzing the emotional tone of a section; for those tasks, you need dedicated audio analysis tools. You must have BPM and time signature data to get meaningful results from any of its functions.

The hassle of manually mapping out song timings is real.

Right now, when a producer wants to plan an arrangement, they're deep in spreadsheets or staring at DAW timelines. They have to constantly cross-reference BPM changes against time signatures, copy bar counts here and there, and manually calculate if that 16-bar verse will actually fit before the chorus hits. It’s tedious math that slows down creativity.

With this MCP, you feed in your parameters once. You don't have to babysit a spreadsheet of rhythmic formulas anymore. You simply ask it what the duration should be for any segment, and get back accurate numbers instantly. What you gain is time and perfect structural confidence.

Song Structure Timer gives you total timing certainty.

Forget clicking between different tempo calculators or struggling to reconcile beat counts across changing meters. You can use get_bar_rhythm_metrics and calculate_section_duration in one flow, checking the rhythm of an entire piece before a single note is played.

The biggest difference now is that your timing data is always mathematically certain. It's not an estimate; it's the precise number you need for professional audio work.

Song Structure Timer MCP: 3 Tools

These tools allow you to mathematically determine the duration of beats, bars, or entire musical sections based on BPM and time signature.

#	TOOL	DESCRIPTION
01	`get_bar_rhythm_metrics`	Calculates the duration of beats and bars in seconds based on your BPM.
02	`calculate_section_duration`	Determines the specific length, in seconds, for any defined section of a song.
03	`generate_song_timeline`	Builds a complete, sequenced structural breakdown of the entire piece from start to finish.

See It in Action

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

U What is the duration of one beat at 120 BPM in 4/4 time?

At 120 BPM, a single beat lasts exactly 0.5 seconds.

U Calculate the duration of a 16-bar verse if one bar is 2 seconds long.

A 16-bar verse with a 2-second bar duration will last 32 seconds.

U Generate a timeline for a song at 100 BPM, 4/4 time, with an Intro of 8 bars and a Chorus of 16 bars.

The timeline starts with the Intro at 0:00 (lasting 16 seconds), followed by the Chorus starting at 0:16 (lasting 32 seconds).

Frequently Asked Questions

01 **How do I use the Song Structure Timer to find my BPM?**

This MCP doesn't calculate BPM from sound, but if you know your target tempo (e.g., 120 BPM), you input it along with the time signature and let the tools calculate everything else for you.

02 **Can I use generate_song_timeline if my song changes tempo multiple times?**

Yes, generate_song_timeline is designed to handle complex arrangements. You simply specify the BPM change points and their corresponding durations, and it builds the timeline accurately.

03 **What kind of information does get_bar_rhythm_metrics provide?**

This tool gives you two key measurements: the number of seconds per individual beat and the total number of seconds for a full bar, based on your input BPM.

04 **Is this better than using my DAW's built-in measuring tools?**

This MCP offers external, standardized calculations that are useful for pre-production planning. It provides a clean, specific output without needing to open or interact with your actual Digital Audio Workstation.

05 **Does Song Structure Timer only work in 4/4 time?**

No. You can input various time signatures into the MCP, allowing you to calculate rhythms and durations for different meters beyond just standard 4/4 time.

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 `"song-structure-timer": { "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

Song Structure Timer 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 Song Structure Timer. 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	Song Structure Timer MCP
Server ID	`019efc58-9657-720c-ace4-61a455e4383b`
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/song-structure-timer.