MCP SERVER | NO CODE | CLOUD HOSTED

Groove Quantize Calculator MCP

Absolute Millisecond Timing for Music Producers

Groove Quantize Calculator converts musical tempo (BPM) into precise millisecond grids for DAWs. It calculates exact note durations—like dotted eighths or triplets—and figures out swing offsets and latency needs, giving producers absolute time control over their rhythm tracks.

A+ Quality Score 100/100

bpm | quantization | swing | daw | latency | rhythm

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/groove-quantize-calculator

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.

Groove Quantize Calculator MCP

4 tools available
Cloud-hosted on Vinkius

Timing in music production isn't about feel; it's about milliseconds. This MCP solves the headache of translating musical tempo into absolute digital time. If you need to know exactly how long a dotted quarter note lasts at 130 BPM, this calculator handles it instantly. You can use `calculate_note_durations` to nail down precise timings for any standard or triplet subdivision. Need to adjust the groove? Use `calculate_swing_offset` to find the exact millisecond displacement required for specific swing percentages. The tool also gives you a full view of your rhythmic grid with `get_complete_rhythmic_map`, which is huge for complex arrangements. Plus, it helps manage buffer size by providing latency advice via `get_latency_recommendation`. Vinkius makes connecting this level of timing precision to your existing workflow simple; just connect your preferred AI client and start calculating.

Core Capabilities

01 — Determine note durations

It calculates the exact millisecond length for standard, dotted, or triplet rhythmic values based on BPM.

02 — Adjust swing timing

The tool computes the specific millisecond offset needed to apply a desired amount of rhythmic swing.

03 — View full rhythm map

It generates a complete, comprehensive view of all possible time divisions within a given tempo range.

04 — Estimate latency needs

You get a recommended buffer size based on the project's current BPM, helping prevent audio dropouts.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/groove-quantize-calculator** — connect your AI agent in three steps.

01 Tell your AI client the base tempo (BPM) and what specific rhythmic structure you need analyzed.

02 The MCP runs calculations across its tools to process the tempo against musical timing rules, returning precise millisecond values.

03 You receive structured data showing exact durations for notes, swing adjustments, or a full time map.

The bottom line is that you get absolute time measurements for complex rhythmic ideas without manually calculating fractions and offsets.

Built For

This MCP is essential for professional audio engineers, music producers, and sound designers. If your job involves aligning complex MIDI patterns or fixing timing issues across multiple tracks, you need this. Stop guessing at groove values; start working with absolute numbers.

Music Producer

Uses the calculator to nail down exact timings for drum programming and rhythmic pattern creation.

Audio Engineer

Checks latency recommendations before mixing high-tempo projects to ensure smooth playback.

Sound Designer

Determines precise millisecond offsets for complex sound effects that need to hit a specific, calculated rhythm.

What Changes When You Connect

01 Stop relying on visual estimations. Use `calculate_note_durations` to get the precise millisecond length of any note value, eliminating timing guesswork.

02 Achieve perfect groove consistency by using `calculate_swing_offset`. This gives you the exact delay needed for a specific swing percentage, every time.

03 Plan complex arrangements knowing exactly what's coming. The `get_complete_rhythmic_map` tool generates a full view of timing options at any given tempo.

04 Avoid audio dropouts in fast tracks. Use `get_latency_recommendation` to check buffer sizes based on your BPM, keeping playback smooth.

05 Bypass manual math entirely. Instead of opening multiple reference guides, let the MCP handle all calculations from one prompt.

Real-World Applications

Need a custom triplet feel at 150 BPM?

A producer needs to know the precise timing for triplets because their DAW's built-in quantizer feels too loose. They ask their agent, and it uses `calculate_note_durations` to return exact millisecond values, allowing them to program perfect timings into their track.

Building a highly swung drum loop.

An audio engineer wants the beat to swing exactly 65%. They use `calculate_swing_offset`, getting the precise milliseconds needed for that deviation. This level of accuracy is impossible with just visual reference.

Reviewing a complex, time-signature piece.

A composer needs to see every possible timing option across an entire measure at 128 BPM. They run `get_complete_rhythmic_map` and get a full breakdown of all divisions (1/4, 1/8, 1/16, etc.) in one go.

Preparing a high-tempo mix for mastering.

The engineer is worried about buffer issues. They prompt the system to run `get_latency_recommendation`, immediately getting advice on what buffer size they need before even starting the final mix.

Patterns to Avoid

Assuming timing calculations are simple

✗ AVOID

A user tries to calculate complex rhythm timings by manually looking up BPM charts or using a basic online calculator that only gives rough fractions.

✓ INSTEAD

Don't guess. Use `calculate_note_durations` for precise subdivision lengths, and then use `get_complete_rhythmic_map` if you need the whole picture at once.

Mixing multiple tools manually

✗ AVOID

A user has to switch between a swing calculator, a note duration tool, and a latency checker to finish one task.

✓ INSTEAD

Connect this MCP via Vinkius. Let your AI client orchestrate the process; it handles calling `calculate_swing_offset` and then using that result with other tools automatically.

Ignoring project technical limits

✗ AVOID

A producer starts a fast, complex track without checking hardware limitations or buffer sizes.

✓ INSTEAD

Always run `get_latency_recommendation` first. It tells you what your system needs to maintain the desired tempo reliably.

The Right Fit

Use this MCP if timing is a hard science for your project. If you are dealing with anything that requires absolute millisecond precision—like advanced drum programming, complex rhythmic sound design, or ensuring reliable DAW playback in high-tempo tracks—this tool is non-negotiable. It gives you the raw numbers to make creative decisions. Don't use it if you just want a general suggestion for 'a good groove'; you need specific calculations like those provided by `calculate_swing_offset` for that. If your only goal is basic tempo conversion, a simpler unit converter might suffice. But when you need to understand the relationship between BPM, note subdivision (via `calculate_note_durations`), and hardware limitations (`get_latency_recommendation`), this MCP is what you require.

The headache of timing math in post-production

You know the process: You set a tempo, but when your idea requires a dotted eighth note or a triplet subdivision, you have to stop everything. Then you open three separate browser tabs—one for BPM conversion, one for rhythmic notation, and another for swing offsets—just to figure out if that groove will actually sound right in your DAW.

With this MCP, the process collapses into a single prompt. You describe the rhythm, the tempo, or the feeling you want. The system instantly runs all necessary calculations, giving you precise millisecond values so you can program with absolute confidence.

Get Absolute Timing Control with Groove Quantize Calculator

Manual timing work means juggling multiple variables: the base tempo, the note fraction (like 1/16), and whether you're adding swing. You spend time copying values from one calculator to another, risking human error every single time.

Now, all these calculations are centralized. Whether you need a full rhythmic breakdown using `get_complete_rhythmic_map` or just the exact timing for one specific note, the answer is immediate and verifiable.

Groove Quantize Calculator: 4 Tools

These tools let you calculate exact rhythmic timings by converting BPM into absolute milliseconds, giving total control over your musical grid.

#	TOOL	DESCRIPTION
01	`get_latency_recommendation`	Calculates the recommended buffer size in milliseconds based on your project's BPM.
02	`calculate_note_durations`	Determines the exact millisecond duration for standard, dotted, or triplet note subdivisions (e.g., 1/4, 1/8).
03	`get_complete_rhythmic_map`	Generates a full grid map showing time values for all major rhythmic divisions at the current BPM.
04	`calculate_swing_offset`	Provides the specific millisecond displacement value needed to apply a calculated swing effect.

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 a dotted 1/8th note at 140 BPM?

At 140 BPM, a 1/8th note lasts approximately 214.29 ms, and its dotted version is approximately 321.43 ms.

U Calculate the swing offset for 70% swing at 90 BPM.

For a 90 BPM project with a 70% swing target, the off-beat should be delayed by approximately 34.29 ms.

U Show me a full rhythmic map for 128 BPM with 50% swing.

The complete rhythmic grid at 128 BPM (straight) includes: 1/4: 468.75ms, 1/8: 234.375ms, 1/16: 117.1875ms, etc.

Frequently Asked Questions

01 **How does the Groove Quantize Calculator handle swing offsets?**

It uses `calculate_swing_offset` to give you a precise millisecond displacement value. You don't just get 'more swung'; you get the exact number needed for your DAW.

02 **Can I calculate durations for odd note values using Groove Quantize Calculator?**

Yes, `calculate_note_durations` handles standard and dotted subdivisions. You input the fraction (like '1/8') and it returns the accurate millisecond timing.

03 **What is the best way to check for latency using Groove Quantize Calculator?**

Use the `get_latency_recommendation` tool. You provide the BPM, and the MCP tells you what buffer size your system should use to prevent audio dropouts.

04 **Does Groove Quantize Calculator give me all possible note timings?**

Yes, `get_complete_rhythmic_map` generates a full grid map showing time values for every major division at the current tempo. It's perfect for quick reference.

05 **Is this better than using my DAW's built-in timing tools?**

It's different. Your DAW quantizes based on settings; this MCP calculates the underlying mathematical truth of the rhythm, giving you verifiable data before you even open your project.

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 `"groove-quantize-calculator": { "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

Groove Quantize Calculator 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 Groove Quantize Calculator. 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	Groove Quantize Calculator MCP
Server ID	`019eff92-632f-724b-a6ea-1221632d64b1`
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/groove-quantize-calculator.