MCP SERVER | NO CODE | CLOUD HOSTED

Level Time Estimator MCP

Know exactly how long the grind will take.

Level Time Estimator predicts how long specific RPG content will take you to complete. This MCP analyzes XP curves, comparing farming, questing, or dungeon runs to find your game's biggest time sinks and progression bottlenecks.

A+ Quality Score 100/100

rpg | progression | leveling | game-design | efficiency | xp-calculator

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/level-time-estimator

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.

Level Time Estimator MCP

3 tools available
Cloud-hosted on Vinkius

Need to figure out if that new raid is actually worth the grind? The Level Time Estimator helps designers and players predict exactly how long they'll spend reaching specific character levels. It pulls together data on XP requirements, activity efficiency, and overall player distribution. Instead of guessing, you get a clear timeline showing hours per level based on your chosen activities. When you connect this MCP through Vinkius, you can run complex scenarios—like comparing questing versus farming—to pinpoint exactly where the game slows down. It's pure data analysis for leveling up. You'll know if that late-game content is going to feel like a massive slog or if your character will actually keep progressing.

Core Capabilities

01 — Predict total progression time

Calculates the estimated hours required to move from one level range to another based on mixed activities.

02 — Compare activity efficiency rates

Determines which gameplay loop, like dungeons or quests, provides a higher rate of XP gain.

03 — Identify sudden difficulty spikes

Finds specific levels where the time needed to level up increases dramatically compared to the previous level.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/level-time-estimator** — connect your AI agent in three steps.

01 You feed the MCP your game's XP data, including activity rates and target level goals.

02 The system processes this information, analyzing how different activities affect overall XP gain and where difficulty curves steepen.

03 It returns a detailed timeline showing predicted hours per level range and flags any major progression breakpoints.

The bottom line is you get data-driven answers about your game's pacing, not just gut feelings.

Built For

Game designers or development teams who are tired of balancing based on feel. This MCP targets the pain point of 'the content feels slow.' It gives you hard numbers to justify changes in XP rates or required resources.

System Designer

Uses it to model different difficulty settings and predict if a new zone will frustrate players by having too few XP rewards.

Game Economy Specialist

Tests various activity mixes, like increasing dungeon loot or quest payouts, to maintain steady player engagement over time.

Level Designer

Determines the optimal curve for XP delivery, ensuring that progression feels rewarding and doesn't suddenly stall out at a specific level range.

What Changes When You Connect

01 Pinpoint bottlenecks: Use `find_progression_breakpoints` to immediately see which levels require disproportionately more effort, allowing you to smooth out the difficulty curve before release.

02 Optimize gameplay loops: Running a comparison with `evaluate_activity_impact` tells you instantly if farming gold is better for XP than running dungeons. It quantifies player choices.

03 Predict pacing accurately: The `calculate_timeline` tool gives you a concrete hour estimate from start to finish, replacing vague 'many hours' estimates with actionable data.

04 Balance content difficulty: Instead of guessing where the game slows down, use this MCP to model progression and ensure every level feels like a natural step forward.

05 Test complex scenarios: Input your desired mix of activities—say, 60% questing and 40% farming—and let the MCP predict the exact time investment.

Real-World Applications

The 'Late Game Grind' Problem

A designer needs to know if adding a new end-game raid will feel rewarding or like a massive stall. They use `calculate_timeline` to model the progression after the raid, finding that the time per level increases by 30% at level 65, indicating a severe bottleneck they need to address.

Comparing Activity Value

A QA tester wants to know if players will prefer questing or grinding loot. They run `evaluate_activity_impact` and discover that while quests provide high XP, dungeon runs actually offer a 25% higher rate, confirming the need to adjust quest rewards.

Fixing Predictable Stalls

A content manager notices player complaints about level 40. They run `find_progression_breakpoints` and confirm that this specific level requires a massive spike in XP, allowing them to adjust the required XP curve directly.

Patterns to Avoid

Treating it like simple math

✗ AVOID

Just plugging in total levels and expecting an average time. This ignores the non-linear nature of game difficulty.

✓ INSTEAD

You must use `calculate_timeline` to model activity mix, ensuring you account for the specific XP requirements at every stage of the progression curve.

Ignoring content variation

✗ AVOID

Assuming all activities (quests, dungeons, farming) contribute equally to leveling up.

✓ INSTEAD

Run `evaluate_activity_impact` first. This forces you to compare rates and understand which activity genuinely contributes the most XP per hour.

Fixing symptoms, not causes

✗ AVOID

Adding more rewards everywhere just because players complain about slowness.

✓ INSTEAD

Use `find_progression_breakpoints` to pinpoint the exact level or activity mix causing the slowdown. Fixing the source is better than adding a band-aid.

The Right Fit

Use this MCP if your core problem is pacing and time predictability. If you need to model how player behavior (like 'I spend 70% of my time questing') affects long-term retention, this tool is critical. It moves the discussion from subjective feeling ('It feels slow') to objective data points ('You hit a breakpoint at level X'). Don't use it if your only goal is simple arithmetic; for example, if you just need to know 'how many items do I need,' that's inventory management, not progression modeling. If you are building an economy model but don't care about the *rate* of leveling, you might be better off with a dedicated resource flow MCP.

The guesswork involved in balancing game difficulty is brutal.

Today, if a designer wants to know how players will feel at level 50, they usually rely on anecdotal evidence or rough spreadsheets. They manually track XP requirements across multiple tabs—quest rewards here, dungeon loot there, and farming rates in another file. Then, they have to guess how these variables interact when a player dedicates their time unevenly.

With this MCP, you stop guessing. You feed the system your entire progression structure, specify activity percentages, and it models the resulting timeline. It hands back clear data showing hours per level and precisely where that massive slowdown hits.

The Level Time Estimator gives you predictable progression curves.

You no longer have to manually compare the output of quests versus dungeons in a separate sheet. Running `evaluate_activity_impact` instantly shows which source provides the highest XP/hour, immediately directing your team where to focus balance efforts.

The outcome is simple: you get data that proves whether or not the game's pacing is sustainable for hundreds of hours of play.

Level Time Estimator: 3 Tools

These tools allow you to calculate total progression times, compare activity efficiency, and identify specific levels where the game's difficulty spikes.

#	TOOL	DESCRIPTION
01	`find_progression_breakpoints`	Identifies specific levels where the time required per level increases significantly.
02	`evaluate_activity_impact`	Compares different gameplay activities to determine which one yields the best XP rate.
03	`calculate_timeline`	Generates a detailed estimate of total time needed to reach a specific level goal.

See It in Action

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

U How long will it take me to go from level 1 to 50 if I spend 70% of my time questing and 30% farming?

Based on your activity mix, you will reach level 50 in approximately 142 hours, with the most significant time increase occurring at level 35.

U Which activity is more efficient for leveling: dungeons or quests?

Dungeons are currently providing 450 XP/hour, while quests provide 320 XP/hour, creating an efficiency gap of 130 XP/hour.

U Are there any major progression bottlenecks in my current setup?

Yes, a breakpoint was detected at level 40 where the time required per level increases by 25% compared to level 39.

Frequently Asked Questions

01 **How do I use Level Time Estimator MCP to find bottlenecks?**

You run `find_progression_breakpoints` and input your current progression data. It immediately identifies the specific levels where the time required jumps significantly, letting you know exactly what needs adjusting.

02 **Does Level Time Estimator MCP only work for quests?**

No. The tool handles mixed inputs. You can use it to model any combination of activities—dungeons, farming, or specific quest lines—to get a comprehensive time prediction.

03 Can I compare different activity types with Level Time Estimator MCP?

Yes. Use `evaluate_activity_impact` to directly compare the XP rates of distinct activities (e.g., 'dungeons' vs. 'quests') and see which one provides better returns.

04 What information does calculate_timeline require?

`calculate_timeline` requires your starting level, ending level, and the expected percentage mix of activities (e.g., 50% questing, 30% farming) to generate an estimate.

05 Is Level Time Estimator MCP better than using spreadsheets?

Yes. Spreadsheets are static; this MCP models dynamic interaction. It accounts for the non-linear way difficulty increases, something that is nearly impossible to model accurately in a simple spreadsheet.

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 `"level-time-estimator": { "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

Level Time Estimator 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 Level Time Estimator. 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	Level Time Estimator MCP
Server ID	`019efdb4-fd31-70a8-bf44-aeb692495c13`
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/level-time-estimator.