MCP SERVER | NO CODE | CLOUD HOSTED

Skill Tree Optimization MCP

Stop Guessing. Find Your Optimal Build Path.

Skill Tree Optimization Engine helps you figure out the absolute best skill combinations for any game system. Stop guessing and start building optimal character builds, whether your goal is maximum damage output (DPS), balancing survival stats, or maximizing utility points with limited resources.

A+ Quality Score 100/100

skill-tree | pathfinding | resource-allocation | game-design | dps-optimization | strategy

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/skill-tree-optimization-engine

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.

Skill Tree Optimization Engine MCP

3 tools available
Cloud-hosted on Vinkius

Building a character in an RPG can feel like guesswork. You've got points to spend, but every node depends on others, making it impossible to know where the best returns lie. This MCP handles that complexity for you. By analyzing how skills cost points and how much they boost specific attributes, your AI client calculates the most efficient path forward.

Need to focus purely on dealing maximum damage? Or maybe you need a balanced build that can survive a boss fight while still hitting hard? The engine runs complex resource allocation models to find the perfect mix of nodes for any objective. It’s powerful enough to handle intricate dependency chains, so you don't have to manually cross-reference dozens of skill guides or spend hours in test builds.

When you connect this MCP via Vinkius, your agent gets instant access to sophisticated build calculators and milestone trackers. You can find out exactly how many points you need before hitting a major power spike, or compare which upgrades give you the most bang for your buck, all without leaving your chat interface.

Core Capabilities

01 — Calculate optimal builds

Finds the ideal combination of skill nodes to maximize one specific attribute—like damage or survivability—given a set amount of points.

02 — Compare upgrade value

Compares various skills by determining how much boost they provide relative to their cost in points.

03 — Track skill milestones

Identifies the exact total point count needed to unlock major, high-tier nodes or significant power clusters.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/skill-tree-optimization-engine** — connect your AI agent in three steps.

01 Tell your AI client what you're working with: how many points you have and what your primary goal is (e.g., 'I need a max DPS build').

02 The MCP analyzes the skill tree structure, dependencies, and costs to run an optimization model.

03 It returns a list of recommended nodes and stats, showing exactly how those skills use up your points and what final attributes you'll have.

The bottom line is that instead of guessing, you get a precise, mathematically proven blueprint for the strongest build possible with your current resources.

Built For

This MCP is essential for high-level RPG players, game content creators, and indie game designers who spend hours optimizing character mechanics. If you're tired of spending weekends building spreadsheets just to track skill dependencies, this tool saves your time.

Competitive Player / Theorycrafter

Uses the engine to test and prove optimal builds for competitive play or raid content before dedicating hours in-game.

Game Designer

Tests balancing assumptions, ensuring that specific skill nodes don't give too much power relative to their cost point value.

Content Creator / Streamer

Quickly generates and validates complex optimal builds for video guides or live stream demonstrations, saving manual calculation time.

What Changes When You Connect

01 You instantly know where your points should go. Instead of manually testing builds, the `calculate_optimal_build` tool gives you a mathematically proven blueprint for maximizing any stat.

02 Never waste a point again. The engine lets you compare upgrades using `evaluate_node_efficiency`, showing you which skills give the most value for the lowest cost per point.

03 Set clear goals with total visibility. Use `get_unlock_breakpoints` to see exactly how many more points are needed before hitting that massive, game-changing high-tier node.

04 Build faster, better guides. Content creators can generate complex optimal builds and analysis reports instantly, eliminating days of manual spreadsheet work.

05 Compare diverse playstyles easily. You don't have to choose between DPS or survivability; the tool calculates the best balance for your specific needs.

Real-World Applications

The 'Max Damage' Build Dilemma

A player is stuck on a difficult boss fight and isn't sure if they should spend their last 20 points on raw damage or defensive utility. They ask the agent, and it runs `calculate_optimal_build`, returning two distinct optimized builds: one for maximum DPS and another for tanking survivability, allowing them to pick the right path immediately.

Balancing a New Game System

A game designer is updating an RPG and needs to make sure a new 'Iron Skin' node doesn't break the economy. They run `evaluate_node_efficiency` on dozens of nodes, getting instant comparative data that proves the upgrade is cost-effective but not overpowered.

Hitting the Next Power Spike

A player wants to know when they'll reach the 'Dragon Slayer' mastery node. Instead of clicking through 10 different prerequisite levels, they use `get_unlock_breakpoints` and get a single number: 150 cumulative points needed.

Comparing Build Philosophies

A streamer is debating the best build for viewers. They ask the agent to run three separate calculations using `calculate_optimal_build`—once for DPS, once for utility, and once for pure tanking—to show the viewer exactly what each philosophy achieves.

Patterns to Avoid

Guessing point distribution

✗ AVOID

A user spends hours manually tracking points on paper or using a basic spreadsheet that can't account for node dependencies, leading to builds that are suboptimal or impossible.

✓ INSTEAD

Instead of guessing, use `calculate_optimal_build` with your desired attribute focus (like MAX_DPS). This tool handles the complex math and shows you the absolute best combination of nodes.

Treating all skills equally

✗ AVOID

A user spends points on a skill that provides only marginal returns, thinking it's necessary because it looks cool, when in reality better options exist.

✓ INSTEAD

Always run `evaluate_node_efficiency` before committing points. This tells you the true value per point for every upgrade and ensures your resources are spent optimally.

Getting lost in prerequisites

✗ AVOID

A user is chasing a high-tier node but doesn't know if they have enough total points, leading to frustration and wasted time checking multiple skill guides.

✓ INSTEAD

Use `get_unlock_breakpoints`. It gives you one definitive number—the exact cumulative point total needed to hit any major milestone or mastery cluster.

The Right Fit

Use this MCP if your primary goal is mathematical optimization of resource allocation within a complex, multi-tiered system. Specifically, if you need to know the *best* way to spend points toward a measurable goal (like 120 DPS). Don't use it if you simply need to track inventory or manage basic character stats; those simple tracking tasks are better handled by standard database

connectors. However, if your problem is determining value—if you have two skills and can't tell which one is more cost-effective per point spent—then this MCP, using `evaluate_node_efficiency`, is the right call.

The Point Allocation Nightmare

Right now, figuring out a perfect build means opening up five different wikis and cross-referencing dozens of spreadsheets. You spend hours just trying to map dependencies: 'If I take Skill A, I need 5 points, but that only gives me 3 DPS, and if I skip it, I can't get the bonus from Skill D.' It’s exhausting.

With this MCP, you tell your agent your goals. You don't have to track dependencies or manually calculate ratios. Your AI client processes the entire graph and hands back a single list of recommendations—the exact nodes you need and why they're mathematically superior.

Optimize Builds with `calculate_optimal_build`

The manual steps that disappear are the endless 'what if' scenarios, the comparative stat sheets, and the headache of balancing DPS against survivability. You eliminate the guesswork entirely.

What you get is certainty. Your agent delivers an optimized build list—a proven path to peak performance based on your current point budget.

Skill Tree Optimization Engine (3 Tools)

These tools allow your agent to calculate optimum build paths, compare upgrade value, and track milestones across any complex skill system.

#	TOOL	DESCRIPTION
01	`calculate_optimal_build`	Prioritizes nodes with highest value for chosen objective relative to cost. Finds the best possible combination of skill nodes to maximize a specific attribute weight given a fixed amount of points
02	`evaluate_node_efficiency`	Compares how cost-effective a specific node is based on the immediate boost it provides to all attributes
03	`get_unlock_breakpoints`	Identifies the specific point totals required to unlock significant milestones or high-tier skill clusters

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 best skill build for 50 points focusing on maximum damage?

With 50 points, your optimal build includes nodes: 'basic_strike', 'power_surge', and 'elemental_mastery'. This configuration provides a total of 120 DPS, 30 Survivability, and 10 Utility, leaving you with 0 remaining points.

U How many points do I need to unlock the 'Dragon_Slayer' node?

To unlock 'Dragon_Slayer', you will need a total of 150 cumulative points, accounting for all its prerequisite nodes.

U Is the 'Iron_Skin' node efficient to upgrade?

The efficiency score for 'Iron_Skin' is 2.5, with a value per point ratio of 0.8 DPS, 1.2 Survivability, and 0.5 Utility.

Frequently Asked Questions

01 **How does Skill Tree Optimization Engine work for DPS?**

It focuses purely on maximizing damage output by using the `calculate_optimal_build` tool. It finds combinations that prioritize nodes with high raw attack or elemental bonus weights.

02 **Can I use calculate_optimal_build for a hybrid build?**

Yes, you tell it your objective is 'HYBRID' or specify the weighted balance (e.g., 50% DPS, 30% Utility). The tool then finds the optimal node mix that meets those specific ratios.

03 What if I don’t know how many points I need for a high-tier skill?

Use `get_unlock_breakpoints`. This tool tells you the precise cumulative point total required to unlock any major milestone, so you always know your next goal.

04 Is evaluate_node_efficiency better than just checking stats?

Absolutely. Instead of just seeing a stat boost (e.g., +5 DPS), this tool calculates the value *per point* spent, giving you a much clearer picture of the upgrade's true worth.

05 Does Skill Tree Optimization Engine support custom points?

Yes, you specify your current available pool size when running any of the calculation tools. It respects that fixed point budget for every recommendation.

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 `"skill-tree-optimization-engine": { "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

Skill Tree Optimization Engine 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 Skill Tree Optimization Engine. 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	Skill Tree Optimization Engine MCP
Server ID	`019efdb6-f2a8-714d-bd02-98b71b816bf2`
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/skill-tree-optimization-engine.