MCP SERVER | NO CODE | CLOUD HOSTED

Prime Number Engine MCP for AI Agents

Perform precise primality testing and number theory analysis.

Prime Number Engine is a high-performance math tool for number theory. It lets your AI agent handle primality testing, integer factorization, and prime distribution analysis with precision. Use it for complex math problems that usually require heavy lifting or specialized algorithms like Pollard's rho.

A+ Quality Score 100/100

prime | factorization | mathematics | algorithms | number-theory

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JULY 2026
vinkius.com/mcp/prime-number-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.

Prime Number Engine MCP

4 tools available
Cloud-hosted on Vinkius

The Prime Number Engine gives your AI client the ability to perform rigorous number theory calculations with perfect accuracy. Instead of hoping your agent can do the math in its head, you can have it perform primality tests and factor large integers correctly. It's built for tasks where precision is non-negotiable, like checking if a massive number is prime or breaking down a large integer into its components. You'll find it's a huge step up when you're doing proximity analysis or need to grab statistical data on prime gaps. By adding this to your setup via Vinkius, you turn your agent into a specialized math engine that knows how to handle the nuances of prime distribution. You won't have to waste time writing custom Python scripts or using online calculators every time you need to verify a large number for a research paper. When you're deep in a project and need to know the distribution of primes in a specific range, you can just ask your agent to pull that data for you. It handles the heavy lifting of the math so you can stay focused on the actual analysis. Whether you're working on cryptography, academic research, or just exploring complex number theory, this tool ensures the math is right the first time. It's about getting reliable, high-performance results without the manual overhead of trial division or Pollard's rho algorithms. It handles everything from basic primality checks to complex decomposition of large integers. You get a consistent experience where the AI doesn't make mistakes on the factors of a 10-digit number. Instead, it calls the engine, gets the exact result, and feeds it back to you. This removes the guesswork from your workflow and makes your AI a much more capable partner for technical math tasks.

Core Capabilities

01 — Verify primality

Check if a large integer is a prime number with high precision.

02 — Factorize integers

Break down large integers into their prime components.

03 — Find nearby primes

Locate the closest prime numbers above or below a target value.

04 — Get distribution stats

Retrieve statistical data on prime density and gap sizes.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/prime-number-engine** — connect your AI agent in three steps.

01 Connect the Prime Number Engine to your preferred AI client.

02 Ask your agent to perform a specific math operation like factorization.

03 Get back accurate results based on deterministic or probabilistic algorithms.

The bottom line is your AI gets a reliable calculator for complex number theory.

Built For

This is for developers building cryptographic systems, math researchers, and students who need to handle large-scale number theory without manual calculation.

Cryptography Engineer

Verifying prime properties for security protocols and key generation logic.

Math Researcher

Analyzing prime gaps and distribution patterns for academic papers.

CS Student

Solving complex algorithm problems involving integer factorization.

What Changes When You Connect

01 Get accurate primality tests using `test_primality`.

02 Factorize large integers quickly using `factorize_integer`.

03 Get real density stats for prime distribution using `get_prime_density_info`.

04 Find nearby primes instantly with `find_adjacent_primes`.

05 Reduce math errors by offloading calculations to a specialized engine.

Real-World Applications

RSA Key Research

An engineer checks if a 50-digit number is prime for a theoretical key generation study.

Math Homework

A student asks for the prime factors of a massive integer to complete a complex assignment.

Data Analysis

A researcher looks for prime gaps in a specific range to identify distribution patterns.

Algorithm Dev

A coder finds the next prime after a specific threshold for a custom sorting algorithm.

Patterns to Avoid

Guessing primes

✗ AVOID

Asking the agent 'Is this a prime?' without using a tool.

✓ INSTEAD

Use `test_primality` to get a definitive answer based on Miller-Rabin or Trial Division.

Manual factoring

✗ AVOID

Trying to list factors of a 10-digit number in a chat prompt.

✓ INSTEAD

Use `factorize_integer` to decompose the number into its components using Pollard's rho.

Estimating density

✗ AVOID

Asking for prime gaps from memory.

✓ INSTEAD

Use `get_prime_density_info` to get real statistical data on prime distribution.

The Right Fit

Use this if you need rigorous math for number theory, cryptography, or academic research. It's perfect for tasks involving primality tests or factorization. Don't use it for basic arithmetic like addition or multiplication, as your agent can handle those fine. If you just need to find a random number, this isn't the tool for you.

Prime Number Engine for Cryptography and Primality Testing

Trying to verify primes or factor large numbers manually is a nightmare. You have to write out long division or implement complex algorithms like Pollard's rho yourself, and one small slip-up ruins the whole result. It's tedious and honestly pretty frustrating when you just want the answer.

The Prime Number Engine changes that by letting your agent do the heavy lifting. You just ask for the factors or a primality check, and it gives you the exact result. You get reliable math without the headache of manual calculation.

Prime Number Engine for Number Theory Research

Researching prime distribution usually requires digging through tables or running custom scripts to find gaps and density stats. It's a lot of context switching between your research and your code.

This MCP lets you pull that data instantly. You can query density info or find adjacent primes in one go. It keeps your workflow in your chat interface.

Prime Number Engine: 4 Tools for Number Theory

Use these tools to test primality, factorize integers, and analyze prime distribution stats.

#	TOOL	DESCRIPTION
01	`find_adjacent_primes`	Locates the nearest prime numbers above or below a specified value.
02	`get_prime_density_info`	Provides statistical insights into prime distribution.
03	`test_primality`	Determines whether a given integer is prime.
04	`factorize_integer`	Finds all prime factors of an integer.

See It in Action

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

U Is 104729 a prime number?

Yes, 104729 is a prime number.

U What are the prime factors of 123456789?

The prime factors of 123456789 are **3, 3, 3607, and 3803**.

FACTOR	VALUE
Factor 1	3
Factor 2	3
Factor 3	3607
Factor 4	3803

U Find the next prime after 1000.

The next prime number after 1000 is **1009**.

Frequently Asked Questions

01 **Can the Prime Number Engine check if a huge number is prime?**

Yes, it uses methods like Miller-Rabin to verify large integers quickly. This is great for anyone needing high-precision primality testing.

02 **How does Prime Number Engine handle prime factorization?**

It uses algorithms like Pollard's rho to break numbers into their components. This allows your agent to handle large integers that would be impossible to factor manually.

03 **Can I use Prime Number Engine for cryptography research?**

Yes, it is a perfect fit for cryptography research because it can verify prime properties and handle the complex number theory required for security protocols.

04 **Does Prime Number Engine provide stats on prime gaps?**

Yes, it provides density information and gap sizes for specific ranges, making it a useful tool for analyzing prime distribution.

05 **Is Prime Number Engine good for math homework?**

It is excellent for solving complex math problems. It can quickly find factors or adjacent primes, saving you from doing long division by hand.

06 **Will Prime Number Engine help with number theory research?**

Yes, it provides the statistical insights and primality verification needed for academic research in number theory.

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 `"prime-number-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

Prime Number 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 Prime Number 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	July 2026
MCP Server	Prime Number Engine MCP
Server ID	`019f3109-0d92-72bd-be72-dda6033af1b0`
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/prime-number-engine.