MCP SERVER | NO CODE | CLOUD HOSTED

Regex High-Perf Parser MCP

Extract data patterns with guaranteed accuracy.

Regex High-Perf Parser runs pure V8 Regular Expressions against massive text blocks, guaranteeing 100% accurate entity extraction every time. Stop relying on Large Language Models to count specific IPs, order IDs, or email addresses from huge log files; they drop results and hallucinate. This MCP executes standard regex patterns locally, returning a complete, deterministic JSON array of every single match found.

A+ Quality Score 100/100

regex | v8-engine | high-performance | data-parsing | entity-extraction | log-analysis

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/regex-high-perf-parser

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.

Regex High-Perf Parser MCP

1 tools available
Cloud-hosted on Vinkius

When you're dealing with multi-gigabyte server logs or compliance reports, LLMs often fail at one critical step: counting everything accurately. They hit context limits or just miss the pattern when things get complex. This MCP fixes that. It executes standard V8 Regular Expressions strictly on a local runtime environment, giving you an exact array of every match possible. You provide the text and the specific pattern, and it returns clean JSON data—zero dropped entities, zero hallucinations. Instead of hoping your AI agent remembers to capture everything, you force the computation using this tool. It’s reliable extraction for complex data sets. Through Vinkius, you connect this specialized capability directly into any compatible workflow, letting your agent handle the heavy lifting without breaking on context limits.

Core Capabilities

01 — Extracting specific patterns

The MCP takes a large body of text and a pattern string, then returns an exact array containing every piece of data that matches that pattern.

02 — Guaranteed determinism

Results are always mathematically precise because the tool runs standard V8 RegExp outside of the AI's context window.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/regex-high-perf-parser** — connect your AI agent in three steps.

01 You provide the text block you need analyzed and the specific regular expression pattern you want to find.

02 The MCP executes the standard V8 Regex engine against that data, running a pure, deterministic computation.

03 It returns a single JSON array containing every exact string match found in the entire text.

The bottom line is: it gives you an exhaustive list of matches without any chance of LLM hallucination or context-related failure.

Built For

This MCP is for the SRE, DevOps Engineer, and Data Analyst who wakes up at 2 a.m. to sift through terabytes of error logs or network traffic dumps. You need guaranteed data counts that an LLM can't provide.

Site Reliability Engineer (SRE)

Using this MCP, they extract every unique IP address, transaction ID, and service endpoint from massive log files to pinpoint the exact source of a system failure.

Security Analyst

They use it to scan raw network packet captures for specific patterns, like all observed SSH connection attempts or suspicious API key formats, ensuring nothing is missed.

Data Quality Engineer

This MCP lets them verify data integrity by extracting every required field, such as structured order IDs and dates, across thousands of unstructured support transcripts.

What Changes When You Connect

01 Stop losing results in huge logs. Use the `extract_regex_matches` tool to count every single IPv4 address or UUID from a 10,000-line file, getting an exact JSON array every time.

02 Bypass context window limits. Because this MCP runs V8 Regex locally, your agent can process massive data dumps without dropping matches, which is critical for compliance checks.

03 Get deterministic results. You don't get a summarized list; you get the full, verifiable source of truth—an array containing every single match found by the pattern.

04 Identify patterns fast. Whether it's email addresses or proprietary order IDs like `ORD-[A-Z0-9]{8}`, this MCP reliably pulls out structured data from messy, unstructured text blocks.

05 Integrate deep analysis into any workflow. By connecting this through Vinkius, you make guaranteed pattern extraction a standard function of your agent's capabilities.

Real-World Applications

Investigating network breaches

A security analyst needs to find every single unique internal IP address mentioned across several gigabytes of firewall logs. They feed the log block and the IPv4 pattern into this MCP, receiving a complete JSON array that confirms all endpoints involved.

Processing customer support transcripts

A data quality engineer needs to count every instance of an order ID following a specific format (`ORD-[A-Z0-9]{8}`) across 50 different chat logs. They use this MCP, which returns the complete list of IDs, allowing them to validate that no records were missed.

Extracting data from machine-generated reports

A compliance officer must extract all email addresses and associated usernames from a lengthy text document. By using this MCP, they guarantee the resulting list is comprehensive and strictly formatted as JSON for downstream processing.

Analyzing application crash logs

An SRE needs to find every single unique error code (e.g., `E_CONN_FAIL`) from a massive, messy log file. This MCP executes the regex against the entire text and spits out an accurate array of all codes for immediate triage.

Patterns to Avoid

Asking an LLM to find everything

✗ AVOID

Prompting your agent: 'Find me all IPs and order IDs from this 10,000-line log.' The AI responds with a summary or drops about half the results because it hits context limits.

✓ INSTEAD

Instead, use the `extract_regex_matches` tool. You feed the text and the specific IPv4 pattern into the tool, guaranteeing every single match is returned in a strict JSON array.

Over-relying on generic extraction

✗ AVOID

Using a general purpose agent to pull out all structured data fields. The agent might mix up different types of IDs or fail when the format slightly changes.

✓ INSTEAD

Use this MCP with highly specific patterns, like `ORD-[A-Z0-9]{8}`. This forces absolute precision and type validation on the output.

Trying to parse data in a single prompt

✗ AVOID

Asking your agent to read 5 different logs and extract emails, IDs, *and* IPs simultaneously. The complexity overwhelms the model.

✓ INSTEAD

Break it down. Run separate calls using `extract_regex_matches` for each pattern type (one call for email, one call for IP, etc.).

The Right Fit

Use this MCP if your task requires extracting a high volume of structured data points from unstructured text—think logs, large reports, or complex transcripts. If you need to guarantee 100% accuracy and the sheer volume risks overwhelming an AI client's context window, this is your tool.

Don't use it if: A) You only need a summary (e.g., 'How many IPs are

there?'). Use standard natural language processing for that count. B) The text is small (under 50 lines). C) You don't know the exact pattern; this tool requires you to define the precise regex pattern upfront using `extract_regex_matches`.

If your goal is raw, verifiable data extraction at scale, use this MCP.

The Problem with Asking an AI Agent for Logs

Right now, when a system fails or you get a compliance report, the first thing you do is copy-paste thousands of lines into your agent. You ask it to find every unique IP address, every order ID, and all the emails mentioned. The agent reads it, summarizes it, and then—disaster. It misses half the IPs because the log was too long, or it hallucinates a few IDs that weren't there.

With this MCP, you treat the AI client like a smart orchestrator, not the data parser itself. You hand the raw text to your agent, but when the job comes down to extraction, your agent calls `extract_regex_matches`. This offloads the heavy lifting to a deterministic engine that guarantees every single match is returned in clean JSON.

Extracting Data with Regex High-Perf Parser

You no longer have to manually count matches or copy data from an LLM's sometimes incomplete response. The agent sends the text and the pattern, runs `extract_regex_matches`, and you get a definitive list back—no guesswork involved.

The difference is reliability. You move from accepting 'the best guess based on context' to receiving mathematically verifiable truth, allowing your workflow to proceed with absolute confidence.

Regex High-Perf Parser: 1 Tool Available

Use this MCP's tools to programmatically extract specific patterns and entities from large text blocks with guaranteed accuracy.

#	TOOL	DESCRIPTION
01	`extract_regex_matches`	Inputs a text and a regex pattern to return an exact JSON array containing all matched strings from the provided text.

See It in Action

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

U Use the regex parser with the pattern `\b\d{1,3}(\.\d{1,3}){3}\b` to extract every single IPv4 address from this massive server log.

The computation has been executed with mathematical precision. All results are exact and ready for review.

U Find all email addresses in this text block using regex and return them as a strict JSON array.

The computation has been executed with mathematical precision. All results are exact and ready for review.

U Extract all order IDs matching the pattern `ORD-[A-Z0-9]{8}` from this customer support transcript.

The computation has been executed with mathematical precision. All results are exact and ready for review.

Frequently Asked Questions

01 **How does Regex High-Perf Parser avoid hallucination?**

It runs standard V8 Regular Expressions in a dedicated local runtime environment. This means the extraction process is mathematical and deterministic, bypassing the generative nature of large language models entirely.

02 **Can I use Regex High-Perf Parser with very long log files?**

Yes. The tool is designed to handle massive text blocks by executing regex on a local runtime, which avoids the context window limitations that typically limit LLMs when parsing huge logs.

03 **What kind of data can I extract using extract_regex_matches?**

You can extract anything you define with a pattern: IPv4 addresses, email addresses, specific order IDs, GUIDs, or any unique string format found in your text.

04 **Is the output of Regex High-Perf Parser usable in other workflows?**

Absolutely. The tool returns results as a complete JSON array, which is immediately usable by subsequent steps, databases, and other components in any agent workflow.

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 `"regex-high-perf-parser": { "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

Regex High-Perf Parser 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 Regex High-Perf Parser. 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	Regex High-Perf Parser MCP
Server ID	`019e383c-bdf6-7140-beee-c5bdff0df40e`
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/regex-high-perf-parser.