MCP SERVER | NO CODE | CLOUD HOSTED

Email (.eml) File Parser MCP for AI Agents

Extracting Structured Data from Raw Email Archives

Email (.eml) File Parser takes raw, messy email exports—the kind filled with HTML noise and encoded attachments—and strips it down. It delivers a crystal-clear text payload containing only the pure message body, sender details, and metadata that your AI client can read instantly. Stop wasting tokens on garbage data.

A+ Quality Score 100/100

email-parsing | mime-decoding | data-extraction | text-summarization | context-optimization | raw-data-processing

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/email-eml-file-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.

Email (.eml) File Parser MCP

1 tools available
Cloud-hosted on Vinkius

Ever tried feeding an old, raw email archive into your AI agent? You know the drill. Those .eml files are technical messes—they're riddled with complex MIME boundaries, base64 encoding, and dense HTML that confuses any language model. Your agent wastes tokens trying to parse garbage just to read the first sentence.

This MCP solves that problem entirely. It operates locally, acting as a dedicated email distillation engine. It doesn't care about the messy bits; it strips away all the HTML noise and heavy attachments, leaving you with pure text, structured metadata (who sent it, who got it, when), and the clean body of the message. The result is a pristine JSON object that your AI client can digest immediately.

It’s like having a personal secretary for your inbox data, turning bloated files into tiny, actionable insights. When you connect this MCP via the Vinkius catalog, you're giving your agent guaranteed clean context, letting it focus on answering questions or drafting replies instead of fighting corrupted code.

Core Capabilities

01 — Extracting Core Metadata

Gathers fundamental email details like the sender, recipient list, date, and subject line into a structured format.

02 — Purging HTML Noise

Strips away all complex HTML structure and unnecessary formatting found in raw email files.

03 — Cleaning Binary Attachments

Removes heavy, unreadable base64-encoded binary attachments so the AI only processes text.

04 — Generating Structured Data

Outputs all extracted information into a clean JSON object that any compatible agent can read and interpret reliably.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/email-eml-file-parser** — connect your AI agent in three steps.

01 You provide the MCP with the absolute file path to your raw, difficult .eml email archive.

02 The MCP processes the file locally, stripping away all HTML noise, MIME boundaries, and non-text attachments.

03 It returns a clean JSON payload containing only the pure text body, sender details, and subject metadata.

The bottom line is: it turns messy, unreadable email data into simple, structured context that your agent can use immediately.

Built For

This MCP is essential for knowledge workers who deal with large volumes of raw email exports. Think legal analysts wrestling with case files, project managers summarizing stakeholder threads, or support engineers reviewing long ticket chains. If your job involves reading emails that look like web pages, this saves you hours.

Legal Analyst

Reviews archived email correspondence for key facts and action items, needing guaranteed clean data without the noise of formatting.

Project Manager

Summarizes multi-day threads from different stakeholders to identify urgent next steps or conflicting deadlines.

Support Engineer

Analyzes long, complex support ticket chains to understand the full scope of an issue and draft comprehensive responses.

What Changes When You Connect

01	Saves context window tokens. Instead of feeding a 5MB bloated file, you feed a tiny, clean payload. This means your agent can run more queries on fewer resources.
02	Eliminates hallucination risk. Because the MCP guarantees that the AI only sees pure text and defined metadata, your agent always knows who sent what, when, and why.
03	Saves time in data preparation. You skip the painful manual step of copy-pasting email threads into a simple text editor just to clean up formatting.
04	Supports complex tasks like drafting replies. Your AI client can draft professional responses or summarize dozens of emails instantly using the structured data provided by parse_eml_file.
05	Guarantees privacy. The entire process runs locally, meaning your confidential business communications never leave your machine.

Real-World Applications

Summarizing a lengthy client thread

A project manager has a 30-email chain from a client detailing scope changes. Instead of reading everything, they run the archive through the parser and ask their agent to 'list all required action items.' The agent provides an immediate, prioritized bullet list.

Extracting key personnel from meeting notes

A legal analyst receives a massive raw email export of meeting minutes. They use the parser and prompt their agent to 'list everyone CC'd along with their roles.' The agent instantly pulls the structured list without reading through boilerplate text.

Drafting an official response to a deadline change

A team lead receives an email confirming a new project deadline. They feed the raw file into the MCP and ask their agent to 'draft a polite confirmation accepting the new date.' The output is ready to send.

Analyzing multi-party correspondence

A support engineer needs to understand who was responsible for which part of an issue. They process the email thread and ask their agent to 'identify all parties and summarize their specific contribution.' The result clearly separates responsibilities.

Patterns to Avoid

Pasting raw emails into the chat

✗ AVOID

The user copies a long email chain from Outlook or Gmail and pastes it directly. This includes messy HTML tags, unreadable base64 segments, and excessive formatting.

✓ INSTEAD

Use the parse_eml_file tool instead. Provide the absolute file path to the raw .eml archive. The MCP cleans up all that noise first, giving your agent pristine text.

Relying on generic summarization tools

✗ AVOID

The user feeds a mixed data dump (some email, some document) into an AI client, hoping it isolates the message body. The tool often fails because of mixed file types.

✓ INSTEAD

Use this MCP specifically for .eml files. It is built to handle the unique MIME and encoding structure of emails, ensuring only relevant text is passed along.

Ignoring metadata extraction

✗ AVOID

The user asks their agent to summarize an email but forgets that key details like 'who was CC'd' are needed for context. The summary might be incomplete.

✓ INSTEAD

Always include a prompt asking the MCP to extract all available structured data, such as recipients and CC list, before summarizing content.

The Right Fit

Use this MCP if your source material is in the .eml format and contains formatting noise, base64 attachments, or complex MIME boundaries. You need reliable extraction of pure text and structured metadata (sender, recipient). Don't use it if you already have a clean, plain-text document; simply feeding that file to your agent works fine. Also, don't rely on this for live API feeds—it processes static local files only. If the data is streaming or coming from an active mailbox connection, look into other specialized connectors.

Email (.eml) File Parser MCP: Decoding Raw EML Archives

Today, when you work with large email archives, the process is brutal. You have to copy-paste threads from Outlook or Thunderbird into a text editor just to get rid of the weird HTML tags and broken formatting. Then, if your agent needs that data, you're hoping it doesn't get confused by embedded images or corrupted encoding.

With this MCP, you simply point your agent at the .eml file. It automatically handles the messy backend decoding, stripping away all the noise so you are left with a perfectly clean JSON object ready for deep analysis.

Email (.eml) File Parser MCP: Structured Data Extraction from Correspondence

Manually reviewing an email thread means jumping between the sender, the subject line, and scrolling through body text to piece together who said what. It’s time-consuming, and you often miss key metadata like secondary recipients or specific dates.

This MCP lets your agent access structured data points—sender, recipient, date—separately from the message content. You get a complete picture of the communication flow, not just a summary.

Email (.eml) File Parser: 1 Tool for Data Extraction

Use the parse_eml_file tool to reliably convert local .eml files into structured JSON, making complex email data accessible to your agent.

#	TOOL	DESCRIPTION
01	`parse_eml_file`	Reads a local .eml email file path and returns a clean JSON object with the sender, recipient, date, subject, and pure text body.

See It in Action

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

U I have this client thread archive. Can you give me a bullet-point list of the 3 most urgent action items?

Here are the three most critical tasks identified in the correspondence:

- **Q3 Invoice:** Send by Friday, as per John's note.
- **Proposal Review:** Tech team must review the attached proposal before Monday.
- **Follow-up Call:** Schedule a dedicated call with Sarah next week to confirm scope.

U Read these meeting notes and draft a polite reply accepting the new deadline.

Dear Team,

Thanks for the update. I confirm we can meet the revised deadline of October 15th. We'll adjust our internal timeline accordingly.

Best regards,

U Analyze this long email chain and list everyone who was CC'd with their emails.

The following people were copied on the thread, ensuring they stayed in the loop:

- John Doe: john@example.com
- Tech Support: support@acme.inc
- Legal Dept: legal@firmname.com

Frequently Asked Questions

01 **Are my confidential emails sent to the cloud?**

Absolutely not. The `.eml` parsing happens 100% locally on your computer. Only the cleaned text is passed to the AI for analysis, ensuring maximum privacy for business operations.

02 **Does this tool extract attachments?**

No, it intentionally strips out all attachments (like PDFs and images) to save token space. It focuses purely on extracting the conversational text and metadata (Sender, CC, Date).

03 **Can it read Outlook and Gmail exports?**

Yes. The `.eml` format is the universal standard for email exports. It works perfectly with files generated by Outlook, Apple Mail, Gmail, and Thunderbird.

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 `"email-eml-file-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

Email (.eml) File 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 Email (.eml) File 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	Email (.eml) File Parser MCP
Server ID	`019e3890-767a-72ca-94c2-554e31c98a67`
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/email-eml-file-parser.