MCP SERVER | NO CODE | CLOUD HOSTED

Extracta MCP

Turn messy documents into clean, structured data.

Extracta uses AI to automate data extraction and document classification from PDFs, images, and other files. It lets you define exactly what data you need—like dates, amounts, or vendor names—and then processes entire batches of documents into clean, structured JSON formats using your agent.

A+ Quality Score 100/100

ocr | data-extraction | document-classification | json-parsing | automated-data-entry | unstructured-data

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/extracta

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.

Extracta MCP

10 tools available
Cloud-hosted on Vinkius

Imagine getting mountains of invoices, receipts, and contracts that all need to be logged into a database. Doing this manually is a nightmare. Extracta changes the game by connecting directly to your AI client, letting you handle complex data extraction through natural conversation. You don't just read; you build the process itself. You define custom JSON schemas—telling the system exactly which fields matter (like invoice dates or total amounts). Then, simply give it a URL for any document, and it handles the rest. It doesn't just pull text; it classifies documents first, telling you if that file is an 'Invoice' or a 'Receipt,' and then extracts the necessary data into structured JSON. If you're building out your toolset on Vinkius, this MCP gives you enterprise-grade document processing without needing to write custom API calls every time.

Core Capabilities

01 — Define Extraction Schemas

You create and configure data extraction processes by defining precise JSON schemas for the fields you need from documents.

02 — Process File URLs

Submit publicly accessible file links (PDF, JPG, PNG) to trigger a background workflow that returns structured JSON data later.

03 — Classify Document Type

Set up rules that automatically sort incoming documents into predefined types, like invoices or contracts, based on AI analysis.

04 — Audit Historical Results

Retrieve status and structured data for specific documents, including confidence scores and predicted categories.

05 — Manage Configurations

Update existing extraction settings or view the full configuration of an active document process without creating new endpoints.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/extracta** — connect your AI agent in three steps.

01 First, you define your data needs by setting up a specific extraction process and detailing the required JSON schemas.

02 Next, you submit one or more publicly accessible document URLs to kick off an asynchronous processing job.

03 Finally, you poll for results, receiving structured JSON containing the extracted data, its confidence score, and classification details.

The bottom line is that your agent handles the entire pipeline, from schema definition to final data output, so you get clean JSON ready for analysis.

Built For

Operations managers who are drowning in physical or digital paperwork; data analysts trying to build pipelines that ingest complex documents; and developers needing reliable extraction components. If your job involves converting unstructured files into usable data, this is for you.

Finance Manager

Processing batches of vendor invoices by ensuring every required field—like the total amount or payment date—is accurately extracted and logged.

Data Analyst

Converting a repository of scanned receipts from various sources into standardized JSON formats to calculate quarterly spending trends.

Operations Engineer

Building an automated system that ingests incoming client contracts, classifying them immediately and extracting key dates and parties for follow-up workflows.

What Changes When You Connect

01 Stop manually defining schemas. You tell the system exactly what fields you need—like invoice dates or product totals—and it handles the rest through the `create_extraction` tool.

02 You don't wait for manual file uploads. Just give it a URL using `upload_file_url`, and the background process does the heavy lifting, giving you structured JSON later on.

03 Classification is built-in. Before extracting data, the system uses document type rules (via `create_classification`) to ensure you know if the file is an invoice or a contract.

04 You never lose history. Use `get_batch_results` to pull records from hundreds of processed documents at once for audit purposes.

05 Need a quick change? You can use `update_extraction` to tweak mapping rules on a live process instead of having to build an entirely new setup.

Real-World Applications

Processing Vendor Payments

A finance manager needs to pay vendors using scanned invoices. They ask their agent to use `create_extraction` first, defining fields like 'vendor name' and 'total amount.' Then, they submit 50 URLs via `upload_file_url`, getting back structured JSON data ready for payment processing.

Building a Document Library

A legal team receives thousands of client agreements. They use the MCP to define document types using `create_classification`. The agent processes them, automatically identifying and grouping everything as 'Contract' or 'NDA,' allowing quick auditing.

Tracking Data Changes Over Time

An operations team needs to monitor how many receipts they process each month. They use the `get_batch_results` tool to fetch a paginated list of all processed documents and associated data payloads for historical review.

Validating New Data Pipelines

A developer needs to test if their new extraction schema works on live files. They use `view_extraction` to check the configuration, then submit a single URL using `upload_file_url`, and poll with `get_results` until they get structured JSON.

Patterns to Avoid

Expecting instant results

✗ AVOID

The user submits an invoice URL via `upload_file_url` and then immediately tries to read the data using a general command, assuming the AI can retrieve it right away.

✓ INSTEAD

Remember that processing runs in the background. After running `upload_file_url`, you must wait and then use `get_results` or `get_classification_results` to check if the job is finished before attempting to read the data.

Skipping schema definition

✗ AVOID

The user tries to extract amounts from a document without first running `create_extraction` and defining what 'amount' means in JSON format.

✓ INSTEAD

Always define your fields first. Start with `create_extraction` to establish the rules, then upload documents for processing.

Overwriting necessary settings

✗ AVOID

The user gets frustrated and attempts to manually re-enter every setting they configured when a small change is needed.

✓ INSTEAD

Don't recreate things. Use `update_extraction` to modify the mapping rules or field definitions on your existing process, saving time.

The Right Fit

Use this MCP if your core problem involves taking files—like PDFs or scanned images—and converting their *content* into structured data that a computer can use. You need classification (Is it an invoice?) and extraction (What's the date?). Don't use this if you just need to read simple text from a document; for that, a basic OCR tool will suffice. If your goal is purely workflow automation—like sending emails or setting up calendars—you should look at messaging or calendar integration MCPs instead. This is specialized for transforming messy, unstructured documents into clean JSON.

Copy-pasting data from receipts and invoices is a full-time job.

Today, logging expense reports means opening dozens of PDFs. You click into the total amount field in your spreadsheet, manually copy the date from one corner, and then paste it into another tab. If you're processing 50 documents, that's 200 individual data points moved, copied, and pasted by hand.

With this MCP, the process shifts to a conversation with your agent. You simply tell it: 'I need to extract all dates, amounts, and vendors from these files.' The system handles defining those fields, processing the URLs in the background, and giving you clean JSON data—no copy-pasting required.

Extracta gives you structured document knowledge.

The manual steps that disappear are opening individual documents, figuring out which field is which (is this 'Invoice Date' or 'Payment Due?'), and then cross-referencing data across multiple sheets to ensure accuracy. This takes hours of tedious human review.

Now, you define the schema once and get reliable, auditable results every time. You don't just read text; your agent processes it into usable, structured JSON format.

Extracta with 10 Tools

These tools let you manage the entire document workflow: defining schemas, uploading files, checking results, and auditing history.

#	TOOL	DESCRIPTION
01	`create_classification`	Sets up a new document classification model by defining the categories you want to sort documents into (e.g., invoice, receipt).
02	`view_classification`	Shows the specific details and settings of an existing document classification process.
03	`get_batch_results`	Retrieves historical results for a large number of documents processed through an extraction workflow.
04	`get_classification_results`	Provides the AI's predicted category and confidence score for a specific document.
05	`create_extraction`	Initializes an entire data extraction process, allowing you to specify required fields like dates or total amounts.
06	`delete_extraction`	Removes an existing document extraction configuration; this stops all future processing for that setup ID.
07	`get_results`	Checks the current status of a document's extraction job, indicating if it’s still running or complete.
08	`update_extraction`	Modifies mapping rules and field definitions for an already created extraction process without starting over.
09	`upload_file_url`	Submits a link to a document file, kicking off the background job necessary for data processing.
10	`view_extraction`	Displays all settings and current parameters of an existing extraction process configuration.

See It in Action

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

U Create an extraction process for invoices with fields: date, vendor, total

Extraction setup created! I've initialized a new process (ID: ext_abc) configured to extract 'date', 'vendor', and 'total' fields. You can now start uploading invoice URLs to this process.

U Extract data from this receipt URL: https://example.com/receipt.pdf

Processing document... I've submitted the receipt URL for extraction. Document ID: doc_789. I'll poll for the results now... [Agent retrieves structured JSON with date, total, and items].

U What type of document is doc_789 according to my classification rules?

Checking classification for doc_789... The AI has classified this document as 'Invoice' with a 98% confidence score. It fits within your defined document categories flawlessly.

Frequently Asked Questions

01 **How do I start using Extracta with my documents?**

You first need to run `create_extraction` to define what data you want. Then, use the `upload_file_url` tool to submit your files for processing.

02 **Can Extracta tell me if a document is an invoice or something else?**

Yes. You set up rules using `create_classification`, and then you can use `get_classification_results` to check the predicted type of any uploaded document.

03 **What happens if I change my extraction requirements after setting it up?**

You don't need to start over. Use the `update_extraction` tool to modify your existing configuration and mapping rules on the fly.

04 **Does Extracta handle large batches of documents?**

Yes, you use the `get_batch_results` tool to retrieve historical data from multiple processed files in bulk.

05 **What is the difference between `create_extraction` and `view_extraction`?**

`create_extraction` sets up a brand new process with defined schemas. `view_extraction` just shows you all the current settings for an extraction process that already exists.

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 `"extracta": { "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

Extracta 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 Extracta. 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	Extracta MCP
Server ID	`019d7595-3046-730b-8679-4ad1f8eb7998`
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/extracta.