

MCP SERVER

NO CODE

CLOUD HOSTED

Docparser MCP for AI Agents

Automating structured data extraction from PDFs and scanned images

Docparser lets your AI agent automatically pull structured data from any document type, including PDFs, scans, and images. It manages parsing rules and tracks results so you never have to manually read an invoice or report again. You can check the status of documents in the queue and retrieve specific fields like order numbers and line items directly through conversation.

A+ Quality Score 100/100

data-extraction

ocr

pdf-parsing

automated-workflows

structured-data

data-processing



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

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.

Docparser MCP

10 tools available

Cloud-hosted on Vinkius

Manually pulling data from invoices, contracts, and reports is a massive time sink. Docparser connects your AI client to solve that problem by automating document extraction. It handles everything from complex PDFs to grainy scans, turning unstructured documents into clean, usable data points in seconds. You tell your agent what you need—like 'the total amount' or 'all line items'—and the system finds it across multiple uploaded files. The whole process is visible through conversational prompts: you can list all available parsing rules, check if a document failed extraction, and retrieve the structured results immediately. Since Vinkius hosts Docparser in its catalog, connecting your favorite AI client to this MCP gives you instant access to sophisticated data intelligence without managing any infrastructure.

Core Capabilities

01 — Retrieve account usage status

Check your current metadata and API rate limits for the Docparser platform.

03 — Audit overall system health

Pull a quick summary showing the activity levels and success rates across all your configured parsers.

05 — Review processed file lists

See a comprehensive list of all documents that have been successfully parsed by a particular rule set.

07 — Track recent results history

See a chronological feed of the most recently extracted data across every active parser.

02 — Inspect parser configurations

Get detailed settings and status information about any specific document parsing rule you've set up.

04 — Get structured data from documents

Retrieve the actual, processed data points—including complex tables and custom fields—from a specific document.

06 — Monitor extraction queue status

List files that are currently waiting in the system's processing pipeline to be analyzed.

One Click on Vinkius — From Prompt to Execution

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

- 01 Connect your preferred AI client to this MCP and authorize access using your Docparser API Key.
- 02 Tell your agent the task: 'Extract all order totals from PDFs in the 'Invoices' folder.'
- 03 The system runs the extraction, pulls the structured data, and returns the results directly into the conversation.

The bottom line is that you talk to your AI client about documents, and it handles the complicated reading and structuring of the raw data for you.

Built For

This MCP is essential for anyone whose job involves turning paper or digital files into actionable data. If you're spending time copying figures from PDFs, this saves your day. It targets roles that manage large volumes of varied documents and need reliable data intelligence.

Operations Manager

Uses the MCP to pull critical details like order numbers or shipment dates from incoming supplier invoices on demand.

Data Analyst

Gathers structured information —like quarterly sales figures or market metrics—from processed reports for immediate inclusion in a dashboard.

Finance Specialist

Automates the extraction of specific financial fields, such as tax IDs, payment terms, and subtotals, from complex billing statements.

What Changes When You Connect

- 01** Stop copy-pasting figures. By using the `get_document_extraction_results` tool, your agent pulls precise details like order numbers or contract values directly into the chat window.
- 02** Keep track of everything instantly. Instead of checking a dashboard, you can use `list_recent_extractions` to see the latest data pulled from all active parsers in one glance.
- 03** Never get lost in errors again. The ability to call `list_failed_document_extractions` means your agent finds broken scans or misformatted documents and tells you exactly why they failed.
- 04** See what's coming next. Use the queue tools, like `list_documents_awaiting_parsing`, so your agent can monitor incoming batches of documents before they even need processing.
- 05** Manage rules easily. You can call `list_document_parsers` to check all available parsing configurations and ensure your agents are using the correct extraction methods for different document types.

Real-World Applications

Processing a batch of vendor invoices

A finance specialist asks their agent: 'Get the total amount, due date, and tax rate from all invoices processed today.' The agent uses `'list_recent_extractions'` to pull this structured data for immediate reconciliation.

Auditing compliance documents

An operations manager asks: 'Show me the extraction results for every document labeled 'W-9' last month.' The agent searches and presents a clean list of necessary fields, confirming compliance data points using `'search_parsed_documents'`.

Troubleshooting failed reports

An analyst notices missing data and asks: 'What documents failed parsing today?' The agent uses ``list_failed_document_extractions``, identifies the bad file, and suggests checking its format for manual correction.

Checking parser status before a run

An automation lead wants to confirm readiness and asks: 'What's the health of our main contract parser?' The agent calls ``quick_parser_health_audit`` and confirms high success rates before initiating a large data pull.

Patterns to Avoid

Treating all documents the same way

X AVOID

The user asks the agent to extract 'the total' from every document they upload. The agent fails because it doesn't know if the total is on a PDF, an image, or a spreadsheet.

✓ INSTEAD

First, use ``list_document_parsers`` to check which specific parsers exist (e.g., 'Invoices', 'Orders'). Then, instruct your agent to only run extraction using the relevant tool, like calling ``get_parser_details`` for the right rule set.

Forgetting about processing queues

X AVOID

The user uploads a huge batch of documents and then waits. They have no idea if the agent is working on them or if they are simply waiting in line.

✓ INSTEAD

Always ask your agent to check ``list_documents_awaiting_parsing``. This confirms that the system has received the files and gives you an accurate estimate of when processing will start.

Asking for data without knowing what's processed

X AVOID

A user asks, 'Give me the customer name.' The agent replies with an error because it hasn't been told *which* document or parser to look at.

✓ INSTEAD

Before asking for specific data, always prompt your agent to use ``list_parsed_documents`` or ``get_document_extraction_results`` on a known file ID. This scopes the request and ensures accurate results.

The Right Fit

Use this MCP if you need reliable, structured data extraction from varied document types (PDFs, scans, images). It's perfect for automating tasks like invoice processing or report analysis where data points are buried in complex layouts. Don't use it if your goal is simple text summarization; for that, a general-purpose agent works

fine. However, if you need to *ensure* the extracted data fits a specific schema (e.g., Pydantic AI), another type of MCP might be better suited. If you only need to read text and don't care about table structures or field names, an OCR-only tool is enough. But when structure matters—when the total amount **MUST** be pulled out as a float—this Docparser MCP is what you need.

Docparser MCP for AI Agents: Automating Data Capture from Financial Documents

Today, handling vendor invoices or quarterly expense reports means opening ten different files. You're manually clicking through tabs, cross-referencing dates, and copy-pasting the total amount into a spreadsheet. This process is slow, tedious, and any single missed field costs time and money.

With Docparser connected to your agent, you simply ask: 'What was the total spent across these 12 invoices?' The system manages the parsing rules, extracts the specific fields—including line items and tax breakdowns—and gives you a clean list. You get ready-to-use data without ever touching a copy/paste function.

Docparser MCP for AI Agents: Monitoring Document Processing Status

Manual document pipelines require constant monitoring. Are the files queued up? Did the parser hit an error on the last batch? Checking status means juggling multiple dashboards and guessing which documents are stuck.

Now, your agent monitors it all. You can ask the MCP to run a health audit or list pending items. This immediate visibility into the document pipeline means you know exactly what's processing, what failed, and when to expect results.

Docparser 10 Tools for Document Data Extraction

Use these tools to check parser rules, track document status, and retrieve specific pieces of extracted data from various documents.

#	TOOL	DESCRIPTION
01	<code>get_docparser_account_metadata</code>	Retrieves usage limits and operational metadata for your Docparser account.
02	<code>get_parser_details</code>	Fetches the specific settings, rules, and status of a single document parser.
03	<code>quick_parser_health_audit</code>	Pulls a high-level summary showing overall activity levels and success rates across all parsers.
04	<code>get_document_extraction_results</code>	Retrieves the actual structured data points from a specific document file.
05	<code>list_parsed_documents</code>	Shows all documents that have been successfully processed by one of your defined parsers.
06	<code>list_failed_document_extractions</code>	Identifies and lists any documents that failed the parsing or extraction process, noting the error.
07	<code>list_document_parsers</code>	Lists every document parser rule set you have configured in your account.
08	<code>list_documents_awaiting_parsing</code>	Shows a list of documents currently waiting and queued for the parsing process to begin.
09	<code>list_recent_extractions</code>	Retrieves the most recent data extraction results across all active parsers in one view.
10	<code>search_parsed_documents</code>	Searches for previously parsed documents using a specific filename within a particular parser rule set.

See It in Action

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

U List all documents processed by my 'Client Contracts' parser in the last week.



Contract Parsing Report

I found 12 contracts processed this week. Here's a quick breakdown:

- `Contract_A_Q3.pdf` : Success. Key fields extracted.
- `ClientX_MSA_v2.pdf` : Success. Review required: Termination Date is missing.
- `BadScan_01.jpg` : Failure (Error: Unreadable). Needs rescan.

Do you want me to run a detailed extraction on the successful files, or pull error logs for the failed ones?

U Show me the extracted data from 'DOC-9988', specifically the order number and customer name.



Document Data Retrieval

Here are the structured details for `DOC-9988` :

- **Order Number:** ORD-456
- **Customer Name:** Tech Corp
- **Total Amount:** \$1,250.00
- **Status:** Paid

I can also pull the full JSON output if you need to pipe this data into another system.

U Are there any documents that failed extraction today?



🔥 Extraction Failures Found

I checked all active parsers and found 2 failures:

1. `BadScan_01.jpg` : Error: Unreadable (Possible cause: Poor lighting).
2. `UnknownFormat.pdf` : Error: Unsupported format.

Would you like me to pull the detailed error logs for these two files so we can figure out a workaround?

Frequently Asked Questions

01 How does Docparser MCP help with scanned images, not just PDFs?

It handles scanned images by converting them into usable text through advanced OCR. You don't have to worry about the quality of the scan; the system pulls structured data even if the original document is a picture.

02 Can Docparser MCP extract complex table data from reports?

Yes, it excels at this. It doesn't just read text; it identifies rows and columns in tables—like line items on an invoice—and returns that structured information for your agent to use.

03 What if my documents are from different sources? Does Docparser MCP handle them all?

The MCP is designed to manage various parsers, meaning it can apply specific rules whether the document came from a vendor portal, an internal scanner, or a cloud storage bucket.

04 Is Docparser MCP just for reading data? Can it track my workflow?

Beyond extraction, it gives you visibility. You can monitor documents waiting in the queue and check which files have already been processed, giving you a complete view of your data lifecycle.

05 What is the difference between listing parsed files and getting actual results?







Listing shows *that* a file was processed. Getting the results retrieves the specific, structured data—like just 'the total amount' or 'customer ID'—so your agent gets the useful payload immediately.

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 <code>"docparser": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

Docparser 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](https://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 Docparser. 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	Docparser MCP
Server ID	019d7587-4023-73bc-810f-cd98a421c815
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/docparser.