

M C P S E R V E R

N O C O D E

C L O U D H O S T E D

Documentation Request Plan MCP

Automate the entire lifecycle of claim document gathering.

Documentation Request Plan MCP handles the heavy lifting of claims management by mapping document requirements to the right owners. It builds follow-up schedules, drafts the necessary messages, and suggests alternative evidence paths if primary records are missing. You can connect it to your AI client to manage document workflows without manual tracking.

A+ Quality Score 100/100

claims

documentation

orchestration

workflow

automation



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.

Documentation Request Plan MCP

4 tools available

Cloud-hosted on Vinkius

Managing claims requires constant chasing of missing paperwork. This MCP acts as an orchestration engine to keep your documentation process moving. Instead of manually checking who has what, you use this MCP to map specific claim requirements to the people or entities that hold those documents. Once the owners are identified, it builds a structured schedule for follow-ups so nothing falls through the cracks.

When it's time to reach out, the MCP drafts the actual request messages for you. If a primary source becomes unavailable, it doesn't just stop; it identifies contingencies by suggesting alternative ways to get the evidence you need. It turns a chaotic pile of missing files into a structured, automated workflow that your AI client can manage directly.

Core Capabilities

01 — Requirement Mapping

Your agent uses this to link specific claim needs to the right people.

02 — Automated Scheduling

The MCP builds a timeline of when to send requests and reminders.

03 — Message Drafting

Your AI client writes the text for the requests sent to holders.

04 — Contingency Planning

The tool suggests new ways to get evidence if a primary source is missing.

05 — Workflow Orchestration

It manages the entire lifecycle of gathering documents for a claim.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/en/ai-agent-connect/documentation-request-plan — connect your AI agent in three steps.

- 01 Connect the MCP to your preferred client like Claude or Cursor via Vinkius.
- 02 Provide the claim requirements to your agent.
- 03 The MCP maps these requirements to the correct document owners.
- 04 A follow-up schedule and message drafts are generated.
- 05 The agent executes the plan or suggests alternatives if documents are missing.

The MCP integrates with your AI client to turn claim requirements into an actionable plan.

Built For

This MCP is built for professionals managing high volumes of documentation in insurance or legal environments.

Claims Adjusters

They hand off the task of tracking missing evidence and scheduling follow-ups to the MCP.

Legal Assistants

They use the MCP to organize document requests and identify alternative evidence paths.

Insurance Operations Managers

They use the MCP to oversee the automated document gathering lifecycle.

What Changes When You Connect

- 01 Reduces manual tracking by automating follow-up schedules.

-
- 02 Speeds up document gathering by mapping requirements to owners immediately.
 - 03 Prevents delays by suggesting alternative evidence when primary sources are unavailable.
 - 04 Standardizes communications through automated message drafting.
-

Real-World Applications

Medical Claim Processing

Map required medical records to specific hospitals and clinics to start the gathering process.

Police Report Follow-ups

Create a schedule of reminders to ensure a missing police report is obtained within a set timeframe.

Missing Invoice Requests

Draft and schedule messages to claimants to collect necessary billing documentation.

Evidence Contingency

Find alternative ways to prove a claim when a primary document holder is unreachable.

Patterns to Avoid

Manual Follow-up Tracking

❌ AVOID

You try to keep a spreadsheet of every person you need to contact and manually check it every morning to see who hasn't replied.

✓ INSTEAD

Let the MCP handle the timeline. Use `generate\_request\_plan` to build a structured schedule of communications and follow-up dates automatically.

Guessing Document Owners

❌ AVOID

You assign a medical record request to a clinic without confirming if they actually hold that specific patient file.

✓ INSTEAD

Use `assign\_document\_owners` to map specific claim requirements to the most appropriate document holders first.

Hitting Dead Ends

❌ AVOID

A primary hospital refuses to release a file, so you just leave the claim sitting idle while you wait for a callback.

✓ INSTEAD

Don't let the process stall. Use ``identify_unavailability_contingencies`` to find alternative evidence paths when a holder is unavailable.

The Right Fit

Insurance adjusters and legal teams managing high volumes of paperwork should connect this MCP. If your documentation process is already fully automated or involves very low-volume, simple claims, you don't need this. The decision comes down to whether you spend more time chasing people for files than actually evaluating the claims themselves.

Documentation Request Plan: 4 Essential Tools

These tools manage the entire lifecycle of document gathering, from mapping owners to scheduling follow-ups and finding backup evidence.

#	TOOL	DESCRIPTION
01	assign_document_owners	This tool maps specific claim requirements to the most appropriate document holders.
02	compose_request_messages	This tool generates the actual text content for requests sent to document holders.
03	identify_unavailability_contingencies	This tool finds requirements at risk due to unavailable holders and suggests alternative evidence paths.
04	generate_request_plan	This tool creates a structured schedule of communications and follow-up dates.

See It in Action

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

U Assign the required medical records to the available holders.



The medical records have been assigned to Hospital Provider A and Clinic B.

U Generate a follow-up plan for the missing police report.



The follow-up plan includes an initial request today, a first reminder in 3 days, and an escalation in 7 days.

U Draft a message to the claimant for the missing invoice.



Dear Claimant, please provide the missing invoice for your claim by Friday to avoid processing delays.

Frequently Asked Questions

01 How does this MCP help with missing documents?

It identifies which requirements are at risk and suggests alternative ways to find the necessary evidence.

02 Can I use this with Claude or Cursor?

Yes, you can connect this MCP to any compatible client including Claude, Cursor, and Windsurf.

03 Does it write the emails for me?

Yes, the `compose_request_messages` tool drafts the actual text content for the requests.

04 How does it handle follow-ups?

The `generate_request_plan` tool creates a specific schedule of communication dates and reminders.

05 Do I need to host the MCP myself?

No, Vinkius hosts and manages the MCP so it is ready to use immediately after connection.

06 How does the tool assign document owners?

The `assign_document_owners` tool maps each claim requirement to the most appropriate holder based on the provided list of known documents and available holders.

07 What happens if a document holder is unavailable?

You can use `identify_unavailability_contingencies` to find alternative paths, such as requesting secondary evidence or digital forensic copies, when a primary holder is unreachable.

08 Can I automate the follow-up process?

Yes, `generate_request_plan` creates a complete schedule of initial requests and subsequent chase dates to ensure deadlines are met.

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>"documentation-request-plan": { "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

Documentation Request Plan 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 Documentation Request Plan. 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	September 2026
MCP Server	Documentation Request Plan MCP
Server ID	01a0e105-0ce6-7172-8330-0f905ea30534
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/en/ai-agent-connect/documentation-request-plan.