MCP SERVER | NO CODE | CLOUD HOSTED

Chaindesk MCP for AI Agents

Programmatic Management of Enterprise Knowledge Bases and Document Ingestion

Chaindesk gives you the ability to build and control custom AI knowledge agents trained exclusively on your company's private data. It lets developers programmatically manage multiple specialized bots, ingest external documents like URLs and PDFs, and query deep context-aware answers using any compatible AI client.

A+ Quality Score 95.83/100

llm-training | custom-chatbots | knowledge-retrieval | no-code-ai | rag-pipeline

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

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.

Chaindesk MCP

11 tools available
Cloud-hosted on Vinkius

Building smart internal tools used to mean complicated APIs or manual data dumps. Now, you can build dedicated AI agents that act as subject matter experts for your business—all without writing complex code. This MCP lets your preferred AI client manage the whole process conversationally. You control everything from creating specialized bots configured with specific goals, to continuously feeding them fresh knowledge by adding entire websites or documents into a central datastore. Need to know what information is available? Your agent can check and monitor all your connected data sources for instant status reports. If you're connecting this through the Vinkius catalog, you get access to this full orchestration suite from one place. The result is an AI that doesn't guess; it answers using only your approved corporate knowledge.

Core Capabilities

01 — Build and Manage Specialized Knowledge Bots

Create multiple distinct AI agents, assigning each one a specific role and providing core instructions to guide its behavior.

02 — Ingest Data from External Sources

Add or update data sources—like entire website URLs or uploaded documents—to build a real-time, comprehensive knowledge base for your agents.

03 — Run Contextual Queries Against Proprietary Data

Send questions to a specific agent and receive detailed answers grounded in your company's private data, not general internet knowledge.

04 — Monitor AI Knowledge Bases

View the status and directory of all connected knowledge collections (datastores) directly through your AI client for quick reporting.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP, then grab your API Key from your Chaindesk dashboard.

02 Use your AI client to issue commands, such as asking the agent to create a new bot or add a data source URL.

03 The system processes the request, updates the knowledge base, and provides you with confirmation of the action.

The bottom line is that you use natural conversation to manage complex AI infrastructure tasks like building bots and feeding them information.

Built For

This MCP is essential for technical teams, product managers, and support leads who are tired of manually updating knowledge bases or coordinating multiple specialized chatbots. It gives you centralized control over your entire AI architecture.

Technical Product Manager

Coordinates the deployment of several specialized AI assistants for different business units, making sure each bot stays informed and accurate.

Developer/Ops Engineer

Automates document ingestion workflows by using natural language commands to upsert data sources into internal tools. Manages agent creation and deletion programmatically.

Support Team Lead

Monitors how agents are responding and updates the core knowledge base in real-time without having to switch between multiple dashboards.

What Changes When You Connect

01 Manage your entire bot fleet from one place. Use the `list_agents` tool to see every specialized assistant you've deployed, giving you a clear view of your AI infrastructure.

02 Keep knowledge current instantly. The `upsert_datasource` tool lets you add or update data sources like website URLs in real-time, ensuring your agents never use outdated facts.

03 Maintain context across interactions. By accessing complete session histories via the `get_messages` tool, your agent always remembers what was discussed earlier in the conversation.

04 Build specialized bots easily. Use `create_agent` to build a highly focused AI assistant for one department or topic without needing dedicated code for each one.

05 Know your data sources at a glance. The `list_datastores` and `get_datastore` tools give you immediate visibility into what information is available for querying.

Real-World Applications

Handling complex customer support queries

A support agent needs to answer a question that spans three different manuals. Instead of searching three separate systems, the agent uses `query_agent` to pull context from all relevant datastores and gives one single, accurate answer.

Onboarding new departmental knowledge

The legal team publishes a new compliance guide. The operations lead doesn't have to manually upload it; they use `upsert_datasource` on the main datastore, and all relevant agents immediately gain access.

Debugging bot performance

A developer suspects an agent is behaving oddly. They check the conversation history using `list_conversations` and review the agent's prompt via `get_agent` to pinpoint exactly where the logic failed.

Scaling AI services across teams

A company grows departments rapidly. Instead of building one monolithic bot, they use `create_agent` several times to build separate, dedicated assistants for HR, IT, and Sales, keeping their knowledge bases isolated.

Patterns to Avoid

Treating the AI like a search bar

✗ AVOID

Asking your agent general questions without pointing it to specific data sources. The bot might give a generic answer that doesn't match company policy or procedure.

✓ INSTEAD

Always ensure you use `upsert_datasource` first, feeding the agent new documents or URLs into the knowledge base. This forces the AI to ground its response in your actual proprietary information.

Building one mega-bot for everything

✗ AVOID

Creating a single 'Ultimate Bot' that tries to handle legal compliance, payroll questions, and sales leads simultaneously. It ends up being vague and unreliable.

✓ INSTEAD

Use `create_agent` multiple times. Build separate bots for distinct functions (e.g., one Legal Agent and one HR Agent). This keeps the scope narrow and the accuracy high.

The Right Fit

Use this MCP if your core problem is that your AI agents need to answer questions based on a constantly changing, complex body of proprietary documentation. You're not just looking for general chat; you need orchestration—the ability to manage *where* the knowledge comes from and *who* gets access to it.

Don't use this if all you need is a simple wrapper around an existing API call or a basic Q&A system that only reads static files. For those simpler needs, other tools might suffice. However, if your process requires dynamically creating new agents, ingesting data from multiple live URLs, and monitoring which knowledge sources are active, then this MCP is necessary.

Chaindesk MCP for AI Agents: Managing Corporate Knowledge Retrieval

Right now, getting a comprehensive answer often means jumping through hoops. You might open the internal wiki, search a SharePoint site, and then manually cross-reference a compliance PDF. Then you copy chunks of text into your chat window for an AI to read. It’s slow, it's error-prone, and critical context is always lost in the transfer.

With this MCP, that entire process goes away. You simply ask your agent a natural language question. The system automatically finds all relevant internal sources—be they URLs or documents—and feeds them to the bot for synthesis. You get one clean answer based only on verified company data.

Chaindesk MCP for AI Agents: Orchestrating Specialized Bot Workflows

When multiple teams use generic bots, they often step on each other's toes. The Sales bot might accidentally pull data meant for the Legal team, leading to compliance risks or mixed messaging.

This MCP lets you define specialized roles using `create_agent`. You build isolated knowledge agents—one for HR policy, one for product specs—ensuring that every department gets its own accurate, dedicated AI expert. It’s precision control over your bot fleet.

Manage 11 Tools for Custom Agent Development and Knowledge Management

Use these tools to build, manage, query, and update the complex systems that power your custom AI agents.

#	TOOL	DESCRIPTION
01	`list_agents`	Retrieves a list of all AI bots currently configured within your system.
02	`list_conversations`	Gets a record of past chat sessions, which can be filtered by the specific bot ID.
03	`list_datastores`	Shows you a list of all connected knowledge collections (datastores) available to your agents.
04	`get_datastore`	Retrieves detailed information about one specific knowledge collection by its ID.
05	`query_agent`	Sends a message to an agent so it can answer questions using your custom knowledge base.
06	`update_agent`	Modifies the settings or instructions of an existing AI bot.
07	`create_agent`	Builds a brand new AI bot by providing its name, linking it to a knowledge base, and setting its core operational prompt.
08	`delete_agent`	Removes an existing AI agent from your system entirely.
09	`get_agent`	Retrieves all the detailed configuration and settings for a specific AI bot.
10	`get_messages`	Fetches the complete history of messages from a particular conversation thread.
11	`upsert_datasource`	Adds new content, like a URL or document, to an existing knowledge collection, or updates it if it already exists.

See It in Action

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

U List all my current bots and check the status of our main knowledge base.

System Overview

Here are your active agents:

- `Support Bot` : Handles general FAQs. (Status: Online)
- `Legal Analyst` : Only uses compliance data. (Status: Online)
- `Product Research` : Focuses on market trends. (Status: Maintenance Mode)

Datastore Status

The primary knowledge base (`ds_main`) is healthy and fully updated. We successfully added the Q3 reports yesterday. You're good to go.

U I need a new bot for onboarding new hires, using the HR manual data.

Agent Created Successfully!

- **Name:** New Hire Onboarding Assistant
- **ID:** `agent_hr123`
- **Knowledge Source:** Linked to `ds_hr_manual` .
- **Instructions Set:** The bot is now programmed to answer questions strictly based on HR policy and cannot hallucinate. You can start testing it right away.

U What's the latest info about our return policy? (Assume this data was added via a URL link)

🔍 **Retrieval Complete**

According to the updated documentation linked from `https://company.com/returns`:

- Returns must be requested within 30 days of purchase.
- Items must include original packaging and proof of purchase.
- We accept returns for a full refund, provided the item is unused.

Frequently Asked Questions

01 **How does Chaindesk help me keep my AI bots up to date with new policies?**

You update your bot's knowledge by feeding it fresh data sources, like a URL or PDF. Instead of rebuilding the whole thing, you just use the data ingestion tools to 'upsert' the information, and the agents instantly incorporate the changes.

02 **Can I run multiple specialized AI bots for different teams?**

Yes. You can create separate bots, each with its own specific knowledge base and purpose. This prevents them from mixing up data or giving confusing answers across departments.

03 **What if my agent needs to answer questions about data that isn't in the main database?**

You must first connect the required information by using the knowledge ingestion tools. The agent can only access and report on what you programmatically feed it, ensuring accuracy.

04 **How do I know if my AI bots are working correctly?**

The MCP lets you monitor everything. You can list your available agents to check their status or retrieve conversation histories to review exactly how the bot responded and what data it used.

05 **Is Chaindesk only for developers, or can a non-technical person use it?**

While it has powerful developer features, the goal is making it accessible. You manage complex configurations through natural conversation with your AI client, meaning you don't need to write code.

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 `"chaindesk": { "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

Chaindesk 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 Chaindesk. 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	Chaindesk MCP
Server ID	`019dd0cb-0cc0-72c0-b7f9-fc5469da33dc`
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/chaindesk.