MCP SERVER | NO CODE | CLOUD HOSTED

Hygraph (Headless CMS) MCP

Control your entire content backend via conversation.

Hygraph (Headless CMS) MCP connects your AI agent directly to your entire content backend. You can inspect schemas, run complex queries, publish drafts, or wipe out old documents—all through natural conversation. It gives you full control over federated content and structured data without ever leaving your chat client.

A+ Quality Score 100/100

graphql | content-modeling | api-first | schema-management | structured-content | federated-content

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/hygraph-headless-cms

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.

Hygraph (Headless CMS) MCP

10 tools available
Cloud-hosted on Vinkius

This MCP lets you treat your headless CMS like a database you talk to. Instead of writing specific GraphQL endpoints or hitting REST URLs, you simply ask your AI agent what you need done with your content models. You can discover all the available fields just by asking about a model's structure, then run complex queries that pull out exactly the data points you want—like titles and slugs from published posts. Need to update a draft? Tell it which document ID needs changing, or instruct it to promote existing drafts to live content. If you're managing multiple international versions of content, you can audit all your locales in one go. Because this MCP lives on Vinkius, you connect once across any compatible client and instantly get access to Hygraph's entire suite of tools.

Core Capabilities

01 — Discovering Content Structure

The MCP lets your agent automatically list all content models and inspect the specific fields they contain.

02 — Querying Structured Data

You can run complex queries to fetch related data, like getting details for the last three published blog posts.

03 — Creating and Updating Content

The agent handles creating new content drafts or safely mutating existing documents with specific field values.

04 — Managing Media Assets

You can get a list of attached media files, retrieving the necessary cloud URLs for your frontend code.

05 — Controlling Document Status

The MCP allows you to move content from a draft state directly to published status.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/hygraph-headless-cms** — connect your AI agent in three steps.

01 Subscribe to this MCP, providing your Hygraph API Endpoint and Auth Token.

02 Your AI client uses the provided credentials to connect to the CMS backend via a secure tunnel.

03 You talk naturally to your agent, telling it exactly what content action you need—like 'get me all product slugs' or 'publish draft post 123'.

The bottom line is that you interact with your entire headless CMS using natural language commands instead of manual API calls.

Built For

This is for the developer who hates switching context between their IDE, the GraphQL playground, and the actual content management dashboard. It's for anyone whose job involves reading data models or updating structured content regularly.

Frontend Developer

Testing complex queries and inspecting content models directly within your chat environment without opening a separate Hygraph playground.

Content Architect

Auditing the entire schema definition across multiple content types to verify field configurations and ensure data consistency.

Digital Editor

Quickly publishing draft content or updating specific fields on published documents by simply commanding their AI agent.

What Changes When You Connect

01 Stop context switching. You can check model schemas, run complex queries, and update documents—all in one chat window with your agent.

02 Maintain data cleanliness by using the 'wipe_cms_document' tool to irreversibly delete nodes when you need a completely clean content repository.

03 Manage localization complexity easily. Use list_project_locales to audit all translation spaces, ensuring your internationalized content is mapped correctly before publishing.

04 Speed up development cycles by letting your agent run 'execute_graphql_query' to pull data points like titles and slugs without needing a custom API call every time.

05 Control the entire content lifecycle. You can publish drafts using 'publish_cms_document', moving content from draft status directly to live visibility.

Real-World Applications

Getting metadata for a new feature launch

The developer needs all product slugs and titles for the next marketing campaign. They prompt their agent: 'Give me the title and slug of every published blog post.' The agent automatically runs an execute_graphql_query, returning a clean list that can be copied directly into code.

Verifying content model changes

The architect needs to know if a new field was added to the 'Product' model. They ask their agent to run a schema introspection tool, which immediately lists all available fields and confirms the structure without logging into the Hygraph console.

Fixing an old draft post

The editor realizes that Draft ID 45 needs its cover image updated and published. Instead of navigating three different screens, they command their agent to use update_cms_document, providing the new asset handle and triggering a live status change.

Auditing multilingual content

The team lead needs to know which locales have been mapped for French translation. They simply ask the MCP to list all project locales, getting an immediate report via list_project_locales, saving hours of manual checking.

Patterns to Avoid

Using the CMS API directly

✗ AVOID

The user has to write boilerplate GraphQL mutations and manage authentication tokens in their IDE's terminal. They have to manually track which fields are available before writing any query.

✓ INSTEAD

Just talk to your agent. Ask it to run 'list_schema_introspection' first. Then, ask for the data using 'execute_graphql_query'. The MCP handles the complex connection details and payload construction.

Using a generic database connector

✗ AVOID

The agent pulls back raw JSON objects that require manual filtering to find usable content slugs or specific document IDs. The output is messy, requiring heavy post-processing.

✓ INSTEAD

Use the specialized Hygraph tools. For instance, asking for 'title and slug' ensures the query structure only returns those two needed fields via execute_graphql_query.

Manual content publishing

✗ AVOID

The editor finds a draft post, clicks through multiple status tabs, updates the dates, and finally hits 'Publish.' This process is slow and prone to human error.

✓ INSTEAD

Simply tell your agent to perform a structural extraction of properties for an explicit live shift. The publish_cms_document tool handles the entire state change in one step.

The Right Fit

You should use this MCP if your job requires you to interact with structured, federated content—meaning you need to read data from models that are related to each other, or modify those models programmatically. This is ideal for developers writing integration code, architects verifying schemas, or editors managing publication workflows.

However, don't use this if your goal is simply to store unstructured text (like a blog post body) without worrying about its associated metadata (slugs, authors, dates). If you just need simple storage, look at a plain document repository. You should also avoid using it if your content source isn't built on Hygraph or doesn't expose GraphQL endpoints; this MCP is strictly for that platform.

If you are stuck writing repetitive queries, use 'execute_graphql_query'. If you need to change the data itself, use 'update_cms_document'.

The mess of content management APIs today.

Right now, managing a headless CMS means jumping between dozens of tabs. You write a query in one place, check the schema definitions in another, and then if you want to publish it, you have to manually hit a separate endpoint with specific headers and body payloads. It's a frustrating cycle of copy-pasting identifiers and debugging boilerplate code.

With this MCP, that entire process vanishes. You tell your agent what data structure or content state you need—whether it’s listing all available media assets using list_media_assets, or figuring out the full scope of model fields with get_model_fields. The agent handles the complex API calls and gives you a clean answer.

Using Hygraph (Headless CMS) MCP for Content Schema Operations

You no longer need to manually run schema introspection queries just to remember what fields exist on the 'Product' model. You simply ask your agent, and it uses list_schema_introspection to give you a clear map of every content type available in your project.

It’s direct. The complexity is hidden behind natural conversation. It lets you manage structured data as if you were talking to an intelligent teammate who already knows the CMS structure inside out.

Hygraph (Headless CMS) with 10 Tools

Use these tools to manage everything from querying structured data and inspecting model fields to publishing live content directly through your AI agent.

#	TOOL	DESCRIPTION
01	`create_cms_document`	Automates the creation of a new dynamic model instance.
02	`wipe_cms_document`	Irreversibly deletes specified content arrays, leaving clean states.
03	`list_project_locales`	Identifies active translation spaces across your project's different languages.
04	`get_model_fields`	Inspects the specific internal fields available within a given content model.
05	`list_schema_introspection`	Determines and lists all content models that your entire project hosts.
06	`list_media_assets`	Lists all attached media files, providing structured rules about media limits.
07	`publish_cms_document`	Performs the process of promoting content properties to a live, published state.
08	`execute_graphql_mutation`	Sends a structured payload that executes direct data writes to your schema.
09	`execute_graphql_query`	Runs pure GraphQL queries to fetch specific, bounded routing data from Hygraph.
10	`update_cms_document`	Changes global web content boundary values for a document, safely updating drafts.

See It in Action

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

U List all content models in my project

I've performed a schema introspection. Your project contains 8 content models including 'BlogPost', 'Author', 'Category', 'Product', and 'CustomerReview'. Which one would you like to inspect?

U Show me the fields for the 'BlogPost' model

The 'BlogPost' model has the following fields: 'title' (String), 'slug' (String), 'content' (RichText), 'author' (Relation to Author), 'publishedDate' (DateTime), and 'coverImage' (Asset). Would you like to create a new draft post?

U Execute a query to get the title and slug of the last 3 published posts

Running GraphQL query... I've retrieved the last 3 posts: 1) 'AI in 2024' (slug: ai-2024), 2) 'Mastering Hygraph' (slug: mastering-hygraph), and 3) 'Headless CMS Trends' (slug: cms-trends). All are currently in the PUBLISHED stage.

Frequently Asked Questions

01 **Can Hygraph (Headless CMS) MCP query content models?**

Yes, absolutely. You can use execute_graphql_query to fetch specific data points from any model, such as titles or slugs, and you can even run complex nested queries.

02 **How do I list all my language locales with Hygraph (Headless CMS) MCP?**

You use the 'list_project_locales' tool. This command immediately identifies every active translation space configured in your project, which is useful for localization audits.

03 **Can I update a document using Hygraph (Headless CMS) MCP?**

Yes, you can use the 'update_cms_document' tool to safely change data within existing documents. This lets you mutate draft values without needing manual API calls.

04 **What if I want to delete content permanently with Hygraph (Headless CMS) MCP?**

You can use the 'wipe_cms_document' tool for irreversible deletion of content arrays, ensuring that specific nodes are completely removed from your repository.

05 **Does this MCP help me discover my project schema?**

Yes. You can run list_schema_introspection to automatically determine every content model the project hosts, giving you a full overview of your data architecture.

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 `"hygraph-headless-cms": { "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

Hygraph (Headless CMS) 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 Hygraph (Headless CMS). 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	Hygraph (Headless CMS) MCP
Server ID	`019d75b5-b78b-7247-a18e-509aae5ca0df`
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/hygraph-headless-cms.