MCP SERVER | NO CODE | CLOUD HOSTED

CoreMedia Content Cloud MCP for AI Agents

Programmatic access to digital experience platform data and site assets

CoreMedia Content Cloud gives your AI agent direct access to a headless digital experience platform. It lets you query all site data—from articles and navigation menus to global images and brand configurations—using structured GraphQL requests. You can search across the entire content catalog or fetch specific assets by path, making it ideal for developers needing programmatic control over CMS architecture.

A+ Quality Score 100/100

graphql | digital-experience | composable-dxp | content-delivery | enterprise-cms | api-first

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/coremedia-content-cloud

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.

CoreMedia Content Cloud MCP

10 tools available
Cloud-hosted on Vinkius

This MCP connects your AI agent directly to CoreMedia's full content cloud. Instead of jumping through a web UI and copying data piece by piece, your agent talks straight to the source. You can ask it things like, 'Give me all articles related to sustainability' or 'What are the navigation links for our European site?' It executes complex GraphQL queries under the hood, handling everything from nested menus to global asset URIs. When you connect this through Vinkius, your agent gains total control over your digital experience platform, letting you manage content and audit schemas right inside your chat window or IDE. You'll get reliable data instantly, whether you need article bodies, site context, or brand hierarchy details.

Core Capabilities

01 — Execute complex GraphQL queries

Send arbitrary payloads to define precise content structures within the CoreMedia Delivery Schema.

02 — Retrieve full articles and channels by path

Fetch complete HTML grids, metadata, and structural details for specific content nodes like articles or entire site channels.

03 — Search all global assets

Run deep, full-text searches across every piece of content in the system to find matching records instantly.

04 — Map navigation and site context

Determine the exact structure of your website's menus, hierarchies, and brand settings for any given locale.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/coremedia-content-cloud** — connect your AI agent in three steps.

01 Subscribe to this MCP on Vinkius and supply your CoreMedia GraphQL Host URL along with an Access Token.

02 Your AI client uses the credentials to establish a secure connection, allowing it to read the entire content schema.

03 You prompt your agent naturally (e.g., 'What are our product categories?') and the MCP executes the necessary structured query to return precise data.

The bottom line is you get programmatic access to deep CMS data without writing boilerplate GraphQL queries yourself.

Built For

This MCP is critical for developers and architects who can't rely on the UI. It's for content marketers who need rapid, verifiable answers about site structure or asset status. If your job involves auditing a complex digital experience platform, this is what you need.

Front-end Developer

Tests GraphQL queries and pulls specific metadata for assets directly from the IDE or chat environment.

Content Marketer

Searches for articles or verifies site navigation paths without having to log into the CoreMedia Studio platform.

Product Architect

Audits complex brand hierarchies and global site configurations across multiple environments in real-time.

What Changes When You Connect

01 Instantly audit content structure and asset locations. Instead of navigating the CoreMedia Studio UI, your agent finds article details or navigation paths directly via `get_cmarticle_path`.

02 Reduce development time by accessing complex schema information. Use `get_introspection_query` to validate active model extensions and verify data types without manual setup.

03 Improve content governance by visualizing site context. The `get_site_context` tool confirms brand variations and multi-brand environment rules in a single query.

04 Accelerate debugging with asset details. Use `get_cmpicture_asset` to pull explicit image metadata, solving placement issues without checking file systems.

05 Boost search capability across the entire site. The `search_global_content` function lets you find content matching keywords anywhere in the global catalog.

06 Optimize performance by using compiled queries. Running a `execute_persisted_query` ensures your front-end always loads optimized data, skipping unnecessary steps.

Real-World Applications

Finding all legal disclaimers for the EU site

A marketer needs to verify if a new product page has the required GDPR disclaimers. Instead of manually checking 10 different pages, they prompt their agent: 'Show me the navigation tree and associated content context for European locales.' The MCP uses `get_navigation_tree` and `get_site_context` to return all relevant paths instantly.

Building a data validation endpoint

A front-end developer needs to ensure their CMS connection handles future content types. They ask the agent to validate the schema using `get_introspection_query`. The MCP returns the full list of dynamic headless types, allowing the dev to code against the correct structure before testing.

Global content audit for a new campaign

A product team needs to find every mention of 'AI Ethics' across all global articles and channels. They prompt the agent with: 'Search globally for anything about AI ethics.' The MCP uses `search_global_content` to return every matching article path, saving hours of manual auditing.

Checking an old page's content structure

An architect is reviewing a legacy site section and needs to know its exact layout components. They ask the agent for the channel details for the specific URL. The MCP executes `get_cmchannel_page`, returning the full structured ruleset so the architect can report on the component breakdown.

Patterns to Avoid

Treating CMS data like a simple database

✗ AVOID

A developer tries to query for articles using only simple filters, assuming a flat structure. They might run a basic search that fails because the content is deeply nested and structured by brand context.

✓ INSTEAD

Don't rely on simple searches. To correctly pull article data, use `get_cmarticle_path` or `execute_graphql_payload`. These tools force you to define the exact structural path needed for a complex DXP.

Assuming all content is available everywhere

✗ AVOID

A user assumes that because they found an image asset, its usage rights and metadata are visible through basic API calls. They might miss locale-specific or brand-gated context.

✓ INSTEAD

Always check the site boundaries first. Use `get_site_context` to confirm if the content is multi-brand aware before attempting to retrieve specific assets or articles. This prevents showing irrelevant data.

Overlooking performance needs for high traffic

✗ AVOID

A team builds a front-end that retrieves and processes large amounts of static content on every load, leading to slow page speed because the CMS isn't optimized.

✓ INSTEAD

For critical paths or highly trafficked components, use `execute_persisted_query`. This runs pre-compiled hashes, ensuring edge caching and peak performance delivery right from the start.

The Right Fit

Use this MCP when your goal is to programmatically audit, validate, or extract structured data from a complex digital experience platform (DXP). If you need to know *how* content is nested—for example, verifying navigation hierarchies with `get_navigation_tree` or checking brand context with `get_site_context`—this is the right tool. Don't use it if you simply need basic CRUD operations on a single record; those might be handled by specialized services. If your requirement is only to fetch raw text and don't care about the surrounding site structure, a simpler content retrieval API may suffice. But for true DXP control, this MCP provides the necessary depth.

CoreMedia Content Cloud: Auditing Digital Experience Structure with CoreMedia Content Cloud

Today, auditing a large digital experience platform means clicking through dozens of admin tabs and running multiple reports. You copy the menu structure here, check asset metadata there, and then manually cross-reference brand guidelines in another tool. It's slow, error-prone, and takes hours just to verify site consistency.

With this MCP, you prompt your agent with a single request like, 'What is the full navigation hierarchy for our main corporate site?' The system uses `get_navigation_tree` to instantly map out every link, parent/child relationship, and required locale metadata. You get a clean, structured JSON output ready for immediate use.

CoreMedia Content Cloud: Global Content Search and Asset Discovery with CoreMedia Content Cloud

Finding specific assets or articles used across multiple brands is currently a scavenger hunt. You might find an image, but you'll spend time verifying if that asset can actually be placed on the correct page type or in the right locale.

Now, ask your agent to 'Find all images tagged for Q1 2024 campaigns.' The MCP uses `get_cmpicture_asset` and `search_global_content`. It not only finds the image details but verifies its URI template, telling you exactly where and how it can be placed in your digital experience. You get certainty.

CoreMedia Content Cloud: 10 GraphQL Tools for CMS Schema Operations

These tools let your agent perform specific operations like fetching article paths, checking asset metadata, or executing complex schema-locked queries.

#	TOOL	DESCRIPTION
01	`execute_graphql_payload`	Runs custom GraphQL queries to map out specific data routing spaces within the CMS schema.
02	`get_cmarticle_path`	Extracts structured properties and full details for an active article node using its path.
03	`get_cmpicture_asset`	Retrieves specific asset information, tracing detailed logs for image assets.
04	`get_navigation_tree`	Identifies the exact array structure of native navigation menus across the site.
05	`get_cmchannel_page`	Lists structured rules and layouts for an entire content channel or section page.
06	`get_site_context`	Retrieves structural details confirming multi-brand site environment variations.
07	`get_cmviewtypes`	Determines available application nodes and the underlying database structure for content views.
08	`get_introspection_query`	Checks the system schema automatically to validate active model extensions and data types.
09	`execute_persisted_query`	Runs pre-compiled, optimized queries against global content boundaries for high performance.
10	`search_global_content`	Performs deep full-text searches across all available nodes to pinpoint specific content records.

See It in Action

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

U What are all my current navigation options for a new product line?

Navigation Hierarchy Audit

• **Root:** Products
* Product A: (Path: /products/a) - Active
* Product B: (Path: /products/b) - Needs Review
* Service Line X: (Path: /services/x) - Active
• **Site Context:** This structure applies to the 'English-US' locale and is confirmed against the primary brand configuration. You can also check sub-channels for specific regional requirements.

U I need a full report on all articles mentioning 'sustainability' last year.

Global Content Search Results

Found 4 matching articles:

1. `/global/csr/2023-report`: Full article content retrieved (HTML body available).
2. `/news/eco-initiatives`: Title: Eco-Friendly Practices. Last updated: 2023-11-01.
3. `/products/solar-panel`: Section details show sustainability features section is active.
4. `/corporate/investor-relations`: Mentions ESG goals in the metadata.

U What's the structure of our main 'About Us' page?

CMChannel Layout Details

The 'About Us' channel is composed of three primary sections:

- **Section 1: Our Mission:** Uses a rich text grid (Metadata Type: `TextBlock`).
- **Section 2: Leadership Team:** Requires an image asset and name metadata. Asset ID required.
- **Section 3: History Timeline:** Structured using timeline components.

Need me to find the best image asset for the leadership team section?

Frequently Asked Questions

01 **How can CoreMedia Content Cloud help me audit my site's content structure?**

This MCP allows your AI agent to programmatically map out every menu, hierarchy level, and section component of your digital experience. You don't have to click around; you simply ask the system to verify the full navigation tree or brand context for any given page.

02 **Can I use CoreMedia Content Cloud MCP to find all articles about a specific topic?**

Yes. The agent uses global search capabilities across all content nodes, finding every article and channel that matches your keywords. It returns the full path and summary for quick auditing.

03 **Is CoreMedia Content Cloud MCP good for front-end developers?**

Absolutely. Developers can use this MCP to test complex GraphQL queries, retrieve asset metadata (like image details), and validate schemas directly from their chat or IDE. It's perfect for pre-build testing.

04 **Does CoreMedia Content Cloud help with multi-brand sites?**

Yes. You can use the MCP to retrieve site context, verifying which brand configurations and locale metadata apply to a given content piece. It ensures your code respects all global environment rules.

05 **What if I need to know what types of assets are available?**

The agent can perform schema introspection queries to show you every active model extension and data type the CMS supports, giving you a complete map of your content capabilities.

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 `"coremedia-content-cloud": { "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

CoreMedia Content Cloud 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 CoreMedia Content Cloud. 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	CoreMedia Content Cloud MCP
Server ID	`019d757c-9575-71ba-8848-5e0632c7d71e`
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/coremedia-content-cloud.