MCP SERVER | NO CODE | CLOUD HOSTED

ChangeDetection.io MCP for AI Agents

Automate Website Monitoring and Track Price Changes

ChangeDetection.io monitors websites automatically to track visual and text updates across any URL. It lets you manage multiple 'watches' for competitors or documentation sites, retrieving historical snapshots so you can compare differences exactly. Use it through your AI client to get alerts on price drops, content changes, or broken links without checking pages manually.

A+ Quality Score 98.33/100

web-monitoring | change-detection | automation | data-tracking | notifications

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

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.

ChangeDetection.io MCP

14 tools available
Cloud-hosted on Vinkius

Use this MCP to automate web monitoring and data extraction simply by talking to your AI agent. You connect your ChangeDetection.io instance once via Vinkius's catalog, giving your agent access to powerful tools for tracking external websites.

Instead of clicking into multiple competitor sites or documentation pages just to see if something changed, you tell your agent what to watch. It monitors the URLs and takes historical snapshots. When you need answers, the agent can retrieve those snapshots and run a comparison (a diff) showing exactly what text or visual element was added or modified.

This process lets data analysts track price movements across competitors automatically, while developers monitor API documentation for breaking changes before they happen. You manage everything from setting up new watches to organizing them with tags, all through natural conversation. It's about getting timely, actionable intelligence without the manual overhead.

Core Capabilities

01 — Monitor and Manage Websites

Create, update, or delete specific web pages (watches) you need to monitor for changes.

02 — Analyze Content Differences

Retrieve historical snapshots of a page and compare them against each other to see exactly what text or element changed.

03 — Review Web History

Access the full history of any watch, allowing you to analyze trends in content over time.

04 — Organize Monitoring Projects

Apply tags and categories to your watches, helping manage large-scale monitoring efforts across many URLs.

05 — Retrieve Specific Page Data

Get the most recent snapshot of a watch or retrieve historical data for deep analysis.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP on Vinkius and provide your ChangeDetection.io Instance URL and API Key.

02 Next, you instruct your AI client via natural conversation, telling it which URLs need monitoring or what data history you want to compare.

03 Finally, the agent uses the connected tools to pull the necessary historical snapshots and deliver a clean summary of changes.

The bottom line is that you talk to your agent about web monitoring tasks—like 'compare yesterday's pricing page to today's'—and it handles all the data retrieval.

Built For

This MCP is essential for anyone who spends time manually checking websites for updates. Data Analysts tracking competitor prices, SEO Specialists monitoring SERP changes, or Developers needing to know if an API documentation page broke—you need this.

SEO Specialist

Uses the MCP to keep a continuous eye on competitor landing pages or search engine results pages (SERPs) for modifications.

Data Analyst

Monitors multiple product listing pages, using the tools to track price changes or availability across different e-commerce sites automatically.

Web Developer

Uses this MCP to monitor key API documentation websites, ensuring that external dependencies haven't broken or changed without notice.

What Changes When You Connect

01 Stop manually checking websites. Use the `get_snapshot_diff` tool to instantly compare two pages and see precisely what changed, whether it's a price or a paragraph.

02 Manage your entire monitoring portfolio with ease. You can use `list_watches` and then apply organization using `create_tag`, keeping large projects structured.

03 Get historical context that simple checks miss. The `get_watch_history` tool lets you analyze trends, seeing how content has evolved over months or years.

04 Maintain control over your data sources. Use `list_watches` to audit all active monitors and use `update_watch` if a URL needs adjusting.

05 Quickly check specific pages without digging through history. The `get_snapshot` tool gives you the most up-to-date information immediately.

Real-World Applications

Competitor Price Tracking

A data analyst wants to know if three key competitors changed their flagship product price overnight. They ask their agent to run a comparison on the relevant URLs, using `get_snapshot_diff` to receive an immediate report showing only the dollar amount changes.

Broken Documentation Monitoring

A developer needs to know if an external API documentation page has been updated or broken. They set up a watch and ask their agent to run `get_watch_history` on that URL, allowing them to confirm the integrity of the latest version.

Marketing Campaign Audit

An SEO specialist runs into an unexpected change on a competitor's main landing page. They ask their agent to check the differences using `get_snapshot_diff` and find that a new promotional badge was added, giving them actionable intelligence.

Project Scale Management

A team has hundreds of URLs they need to track across different business units. They use the agent to first `list_watches`, then apply specific tags using `create_tag` and `search_watches` to isolate only 'Q3 Marketing' pages for review.

Patterns to Avoid

Monitoring a Single Page Manually

✗ AVOID

A user opens Chrome, types in five competitor URLs, and manually checks the price on each one every day. This is slow, prone to human error, and impossible at scale.

✓ INSTEAD

Instead, use this MCP's `create_watch` tool through your AI client to monitor all five pages simultaneously. Then, ask the agent to run a comparison using `get_snapshot_diff` for all of them in one go.

Forgetting Historical Context

✗ AVOID

A user only checks the current page and sees a change but has no idea if that change is normal or part of an ongoing trend.

✓ INSTEAD

Always use `get_watch_history` to pull previous versions. This lets you see patterns, determining if the content shift was a one-time event or a sustained modification.

Overloading Tags/Watches

✗ AVOID

A user creates 500 watches without any system for grouping them, leading to an unmanageable list and wasted time searching for the right page.

✓ INSTEAD

Use `list_tags` first. Then, use `create_tag` to build a clear taxonomy (e.g., 'Competitor A,' 'API Docs'). You can then filter everything using `search_watches`.

The Right Fit

You should use ChangeDetection.io if your job requires you to track changes on external websites—especially pricing, documentation, or competitor content—without manually visiting those pages repeatedly. This MCP is perfect when you need the AI agent to perform side-by-side comparisons of historical data using `get_snapshot_diff`. Don't use this MCP if you only need simple scraping (e.g., 'give me today's title'). For that, a simple web scraper tool will suffice. You also don't need it if your changes are confined to internal systems; in that case, an internal database connection is better. If you need structured data output for the changes, focus on using `get_snapshot` and then asking your agent to format the results into JSON or a table.

ChangeDetection.io MCP: Tracking Competitor Price Changes

Right now, tracking competitor pricing means logging into multiple sites, navigating through product pages, and manually comparing the listed dollar amounts in a spreadsheet. This process is incredibly time-consuming, unreliable, and often misses subtle changes.

With this MCP, you simply tell your agent which URLs to monitor for price movements. The system takes over all the clicking and checking; you just ask it to use `get_snapshot_diff` and get an immediate report showing exactly where prices changed between snapshots.

ChangeDetection.io MCP: Auditing API Documentation Changes

Developers typically have to manually bookmark key API documentation pages, relying on inconsistent emails or forum posts to know if the endpoint has changed, leading to painful downtime when code breaks.

Connect this MCP and let your agent set up watches for critical docs. By running `get_watch_history`, you get a verifiable audit trail of every change that occurred, allowing you to stay ahead of breaking updates.

14 ChangeDetection.io Tools for Web Monitoring Data Tracking

Use these specific tools within your agent conversation to create, manage, search, and analyze historical data from monitored websites.

#	TOOL	DESCRIPTION
01	`create_tag`	Creates a new category tag for your monitoring watches.
02	`create_watch`	Sets up a new monitor to track changes on a specific website URL.
03	`delete_watch`	Removes an existing web monitor from your list of watches.
04	`get_snapshot_diff`	Compares two snapshots (using 'previous' and 'latest') to show exactly what changed on a page.
05	`get_snapshot`	Retrieves the most recent snapshot of a watch for immediate review.
06	`get_tag`	Fetches details about a single tag you've created or used.
07	`get_watch_favicon`	Retrieves the small icon (favicon) associated with a specific monitored website.
08	`get_watch_history`	Pulls all recorded data and historical versions for a given watch monitor.
09	`get_watch`	Retrieves the general details of a single monitored watch, including its status and URL.
10	`import_urls`	Imports multiple URLs into your monitoring system via an API process.
11	`list_tags`	Generates a list of every tag currently in use across all your watches.
12	`list_watches`	Provides a complete list of all web pages you are currently monitoring.
13	`search_watches`	Allows you to search for specific watches by criteria like tags or keywords.
14	`update_watch`	Modifies the settings, URL, or schedule of an existing web monitor.

See It in Action

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

U Show me the difference between the last two snapshots for the competitor's main pricing page.

Pricing Comparison Report: Competitor X

- **Original Price (Snapshot 1):** $49.99 (Displayed in red text)
- **New Price (Snapshot 2):** $39.99 (Displayed in green text)
- **Status:** Price reduced by $10.00.

Additionally, the 'Buy Now' button added a new warning badge: *Limited Stock*.

U List all my active watches and tell me which ones are related to our Q3 marketing campaign.

Active Web Monitors

WATCH NAME	URL	LAST CHECK	TAGS
Competitor Pricing	example.com/price	2 hours ago	#pricing, #competitors
API Docs	docs.example.com/api	Yesterday	#dev-docs
Campaign Landing	campaign.com	30 minutes ago	#Q3_marketing

You have 3 active monitors.

U What was the favicon for the API documentation site last week?

The favicon for `docs.example.com/api` (UUID: 123e4567) is available. It's a simple black and white shield icon. If you need to use this image, I can provide the direct link.

Frequently Asked Questions

01 **How does ChangeDetection.io help me track price changes across multiple websites?**

It sets up automatic monitors on specific product pages. When you ask for a comparison, it retrieves historical snapshots and highlights only the changed prices or promotional text, saving you hours of manual checking.

02 **Can I use ChangeDetection.io to monitor API documentation changes?**

Yes. You can set up watches on your documentation URLs. By viewing the history, you'll see exactly when a parameter was added or if a required field was deprecated.

03 **Is ChangeDetection.io better than just using Google Alerts?**

Yes. Google Alerts track keywords mentioned in articles; this MCP tracks the actual content and visual elements on specific, live web pages, giving you much deeper structural data.

04 **What happens if a website changes its structure completely?**

The system will still capture the change. When you run `get_snapshot_diff`, it shows what was there before and what is there now, even if the page layout looks totally different.

05 **How do I keep my monitoring projects organized with ChangeDetection.io?**

You use tags to group watches by project or client. You can list all available tags and then apply new ones using `create_tag` to categorize your entire portfolio.

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 `"changedetectionio": { "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

ChangeDetection.io 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 ChangeDetection.io. 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	ChangeDetection.io MCP
Server ID	`019e3876-9d29-73cb-9748-6d4f36017eed`
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/changedetectionio.