MCP SERVER | NO CODE | CLOUD HOSTED

Google Civic Information MCP

Audit public records and political data instantly.

Google Civic Information delivers authoritative access to localized government data for political auditing and civic research. Your agent instantly identifies representatives, audits election timelines, and retrieves detailed polling metadata without needing manual searches or jumping through complex government portals.

A+ Quality Score 100/100

civic-engagement | political-data | elections | representatives | public-records | polling-locations

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/google-civic-information

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.

Google Civic Information MCP

5 tools available
Cloud-hosted on Vinkius

This MCP connects your AI agent directly to official civic intelligence sources. You can run deep political audits on local governments, turning complicated policy questions into simple conversations for your agent. For instance, instead of navigating multiple state and county websites, your agent finds the specific representatives tied to a street address or checks out upcoming election dates across different regions. It handles everything from finding electoral divisions by name to locating polling sites near an address. When you connect this MCP through Vinkius, you get access to all these data streams in one place, making sure your civic research is always verified and precise.

Core Capabilities

01 — Find local political representatives

The agent searches for current or past political officials using a specific street address.

02 — Audit election timelines

You retrieve comprehensive lists of both upcoming and historical elections supported by Google Civic.

03 — Locate voter resources

The agent pulls polling site locations and specific ballot information linked to an address and a given election.

04 — Identify administrative regions

You search for electoral divisions (OCD-IDs) using either the division's name or its general location.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/google-civic-information** — connect your AI agent in three steps.

01 Subscribe to this MCP and input your Google Civic Information API Key.

02 Tell your AI client what civic data you need, like 'Who represents 1600 Pennsylvania Avenue?'

03 Your agent executes the appropriate tool calls, returning verified representative lists or election timelines directly to your chat.

The bottom line is that your AI agent handles all the complex querying and data aggregation from disparate civic sources in one conversation.

Built For

Political analysts, investigative journalists, and civic engagement teams use this MCP when they need rapid access to official public records. They struggle with fragmented government websites that require jumping between multiple portals just to track a single piece of data.

Political Analyst

They audit representative metadata by address and cross-reference different election dates to build historical models.

Investigative Journalist

They perform rapid audits of political officials or locate polling sites for articles, without manual web searches.

Civic Engagement Lead

They verify election timelines and find necessary voter information across multiple regions to plan outreach efforts efficiently.

What Changes When You Connect

01 Stop cross-referencing government websites. Instead, use the agent to run a representative audit by address using get_representatives_by_address. You get official metadata—party and office titles—in one query.

02 Don't waste time confirming if your data source is down. Use check_api_status first; it gives you immediate confirmation that your political research workflow is operational before running complex queries.

03 Need to track election changes? Instead of manually searching, get_google_civic_elections provides a full catalog of upcoming and past elections, giving you clear civic timelines.

04 Planning voter outreach? Use get_voter_information. You feed it an address and an election name, and it gives back the necessary polling sites and ballot details—no clicking required.

05 Understanding regional boundaries is key. search_civic_divisions lets you find electoral divisions (OCD-IDs) just by knowing a region's name or location.

Real-World Applications

A journalist needs to write about election changes in Chicago.

Instead of visiting the Cook County website, they ask their agent for upcoming elections and local representatives. The agent first uses get_google_civic_elections, then runs search_civic_divisions using 'Chicago' to get the relevant IDs, finally pulling representative lists with get_representatives_by_address.

A policy researcher needs to verify polling locations for a new guide.

They input a specific address and the name of an election. The agent calls get_voter_information, which immediately returns confirmed polling sites and ballot details, saving hours of manual verification.

A local activist needs to know who represents their neighborhood.

They simply ask about a street address. The agent uses get_representatives_by_address to quickly pull the official names, parties, and office titles for that exact location.

A team is building a data pipeline for state comparisons.

They use get_google_civic_elections to build an initial dataset of election IDs and dates. They then use search_civic_divisions to map out the administrative reach of those elections across different states.

Patterns to Avoid

Searching for data by city name.

✗ AVOID

The user types 'Give me all political reps in Washington D.C.' This is too vague and provides no actionable results because the tool needs a specific address or division ID.

✓ INSTEAD

To get accurate data, use get_representatives_by_address with an exact street address like '1600 Pennsylvania Avenue NW' OR use search_civic_divisions to find the official OCD-ID for the area first.

Forgetting to specify an election period.

✗ AVOID

The user asks, 'Where is the polling place?' The system fails because it doesn't know if they mean a past or upcoming vote, leading to vague results.

✓ INSTEAD

Always pair get_voter_information with the correct election name and address. You can use get_google_civic_elections first to confirm the exact names of supported elections.

Assuming all data is available on one portal.

✗ AVOID

The user gets frustrated when a single government site doesn't have both representative and polling data. They spend time switching between multiple tabs.

✓ INSTEAD

Use this MCP, which aggregates information from various sources. You can chain tools like search_civic_divisions to contextualize the results of get_representatives_by_address.

The Right Fit

You use this MCP if your job involves auditing political boundaries or public records where location and timeline matter. If you need specific data tied to a physical address, like who represents it (get_representatives_by_address) or where people vote (get_voter_information), this is the right tool. Don't use it if you just need general Wikipedia information on politics; you need official records. Also, don't use it if your goal is to draft a press release—you'll need a different writing MCP for that. However, if you only want to know what *types* of data are available without checking specific locations, start by running get_google_civic_elections.

Tracking civic information used to be a nightmare of tabs and forms.

Right now, if you're doing political research, you have to copy-paste addresses into multiple government websites. You check one site for representatives, then jump to another for polling dates, and maybe use a third just to find the right electoral division ID. It’s slow, it’s exhausting, and you always worry about missing an update.

With this MCP, your agent handles all that jumping around. You simply ask it what you need—say, 'What are the representatives for this address?'—and the tool pulls verified data from authoritative sources without you ever needing to touch a single government portal.

Google Civic Information gives you verifiable representative details.

Manually finding who represents an area requires knowing the exact OCD-ID, then searching multiple directories for names and titles. This process is prone to human error because data sources don't talk to each other.

Now, you just give your agent a location via get_representatives_by_address. It returns verified official metadata immediately. The manual cross-referencing stops here.

Google Civic Information: 5 Tools

These tools allow your AI agent to perform specific civic tasks like finding representatives, checking election dates, or locating voter resources.

#	TOOL	DESCRIPTION
01	`check_api_status`	Confirms whether the Google Civic Information API is currently operational for use.
02	`get_google_civic_elections`	Retrieves a list of both upcoming and past elections documented by Google Civic.
03	`get_representatives_by_address`	Identifies the political representatives associated with a specific street address.
04	`get_voter_information`	Pulls polling sites and ballot information for an address, provided an election is also specified.
05	`search_civic_divisions`	Finds electoral divisions (OCD-IDs) based on either the division's name or its geographical location.

See It in Action

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

U Who are my political representatives for '1600 Pennsylvania Avenue NW, Washington, DC'?

I've retrieved the representatives for that address! Notable officials include the President of the United States and Congressional representatives from the District of Columbia. I can provide the party affiliation and office titles for each if you'd like.

U Search for civic divisions related to 'Chicago'.

I've identified several electoral divisions for Chicago! These include the City of Chicago and Cook County. I can provide the OCD-IDs for these divisions to assist in further representative audits.

U What elections are upcoming in the United States?

I've retrieved the election catalog from Google Civic! Notable upcoming events include state primaries and the next General Election. I can provide the unique election IDs and dates for each of these if you'd like.

Frequently Asked Questions

01 **How do I check the status of Google Civic Information with this MCP?**

Run the check_api_status tool. This simply confirms if the API is currently operational, ensuring your workflow won't fail because the service is down.

02 **What addresses can get_representatives_by_address find reps for?**

It finds representatives tied to a specific street address. You must provide a full, accurate address string in your query.

03 **Do I need an election name to use get_voter_information?**

Yes, you do. The tool requires both the voter's physical address and the specific name of an election to retrieve relevant polling sites or ballot data.

04 **How can I find out what elections are available using get_google_civic_elections?**

You simply ask your agent to call get_google_civic_elections. It returns a catalog of both past and upcoming election events, giving you the full scope.

05 **Does search_civic_divisions help with representative data?**

Yes. If you don't have an address, but you know the name or general location of a region, use search_civic_divisions to get the necessary OCD-ID before auditing representatives.

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 `"google-civic-information": { "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

Google Civic Information 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 Google Civic Information. 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	Google Civic Information MCP
Server ID	`019d8443-1c22-70fd-b8af-5c180572cc8f`
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/google-civic-information.