MCP SERVER | NO CODE | CLOUD HOSTED

Truto Unified Calendar MCP

Manage your schedule across every calendar in one place.

The Truto Unified Calendar MCP lets your agent handle scheduling across any calendar provider—Google, Outlook, and more. Instead of juggling multiple logins or writing complex code for different APIs, you treat all connected accounts as one unified system. Your AI client reads availability, creates meetings, and updates events in real-time across every linked service.

A+ Quality Score 100/100

calendar-integration | scheduling | unified-api | time-management

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/truto-unified-calendar

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.

Truto Unified Calendar MCP

10 tools available
Cloud-hosted on Vinkius

Managing a schedule today means dealing with fragmented tools. You're constantly switching between Google Calendar, Outlook, and whatever else your company uses just to see if two people are free at the same time. This MCP solves that by giving your AI agent access through one unified API layer. It reads all connected calendars—regardless of which provider owns them—and presents a single source of truth for your availability. You can tell your assistant, 'Book a meeting with Jane and Mark next Tuesday,' and it handles checking gaps across three different corporate accounts before writing the event out to all relevant parties. When you connect this via Vinkius, your agent gains immediate access to sophisticated scheduling logic without needing custom code for every single API in existence.

Core Capabilities

01 — Check overall availability

It scans multiple connected calendars to find blocks of time when everyone is free.

02 — Create and modify meetings

Your agent writes new events or changes details, passing the updates directly to the root calendar provider.

03 — List all connected calendars

It retrieves a list of every linked calendar account and its current connection status.

04 — Find specific event details

You can ask for full information on any single meeting, including attendee lists or conferencing links.

05 — Validate connections

It checks if all linked accounts are still active and have the necessary permissions to function.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/truto-unified-calendar** — connect your AI agent in three steps.

01 First, anchor this MCP instance to your local Vinkius agent. This tells your system where to look for scheduling data.

02 Next, provide the Master Token from Truto along with your integrated account ID, pointing it toward the specific calendar provider you need to query.

03 Finally, ask your assistant a natural language question like 'What time are we free next week?' and it handles the rest.

The bottom line is that instead of writing code for Google's API or Microsoft's API separately, you just talk to your agent.

Built For

This MCP is essential for anyone whose job involves coordinating time across multiple people and systems. If you spend more than five minutes switching between calendar apps to book a meeting, this connector saves you hours.

Executive Assistant

You use this to schedule multi-provider meetings by telling your agent to check availability gaps across the CEO's Google account and the department's Outlook group calendar in one go.

Sales Coordinator

You force event creation directly into a client's varied infrastructure, without having to log into their fragmented Google or Microsoft portals yourself.

Platform Architect

You centralize and manipulate the scheduling logic of your agent, enforcing calendar changes using one unified schema rule instead of multiple provider APIs.

What Changes When You Connect

01 Stop writing conditional code for different providers. This MCP gives you a single, normalized schema to read and write events against Google, Outlook, or any other connected service.

02 Instead of guessing if time slots are open, use the `get_free_busy` tool to accurately pinpoint available gaps in your schedule across multiple calendars instantly.

03 When an event changes—like a meeting moving from Google to Outlook—the system handles the write. Use `update_event` to modify details and sync them automatically everywhere.

04 If you need to know who was at last month's Q3 review, use `list_events` or `get_event`. It pulls all attendee lists, regardless of which calendar originally hosted the invite.

05 You can check if your company's connection status is okay by running `validate_connection`, making sure no integrated account has been abruptly decoupled.

Real-World Applications

Booking a cross-departmental meeting

The project manager needs to find 30 minutes next week when the Engineering team (on Google) and Marketing leadership (on Outlook) are both free. They ask their agent, which uses `get_free_busy` against both providers simultaneously, and gets a precise time slot.

Rescheduling client calls

A sales rep needs to change an event for a key client who uses a third-party calendar. The agent uses `update_event`, passing the new date and time, ensuring the change is reflected immediately without manual intervention.

Auditing account health

The IT admin needs to confirm that all 15 department calendars are still linked correctly. They run `list_connections` and receive a clear report on every provider's authentication status.

Capturing meeting details for records

After a major strategy session, the agent uses `get_event` to extract all granular metadata—like who was invited or if there's an attached video link—so the notes are complete and accurate.

Patterns to Avoid

Treating APIs individually

✗ AVOID

Writing separate code blocks for Google Calendar API calls, followed by a completely different block of logic for Microsoft Graph API calls. This creates redundant error handling and complex branching.

✓ INSTEAD

Use this MCP to let your agent call unified functions like `create_event` or `list_events`. You write the command once, and the MCP handles routing it to the correct underlying provider.

Assuming data consistency

✗ AVOID

Running a scheduling script that assumes all calendar providers use the same metadata fields for attendees, which often leads to failed updates or missing information.

✓ INSTEAD

The MCP normalizes this. Use `get_event` to pull details into one schema, ensuring you always get key data points like attendee lists and locations consistently.

Ignoring connection status

✗ AVOID

Running a high-stakes scheduling query when an entire department's calendar connection has quietly dropped authentication. The script fails silently or throws vague errors.

✓ INSTEAD

Always start by calling `list_connections` and then use `validate_connection`. This confirms the account is ready to accept changes before you try to book anything.

The Right Fit

Use this MCP if your primary pain point is managing schedules across *multiple, different* calendar platforms (Google, Outlook, internal systems). You need a single source of truth for availability and event management. Don't use it if you only ever work within one specific platform; then, standard native SDKs might be fine. Also, don’t rely on this just to store contacts or simple notes—it is strictly a scheduling tool. For general data storage or CRM functions, look into dedicated record-keeping MCPs. If your goal is simply listing all calendars and seeing which ones are active, the `list_calendars` tool handles that initial check perfectly.

The Pain of Calendar Fragmentation

Right now, booking a simple cross-department meeting is a multi-step headache. You have to log into Google, manually compare time slots; then switch over to Outlook and do the same thing for the other attendees. If you use an old script, you'll find yourself writing conditional logic just to handle whether the event needs to go through Microsoft or Google.

With this MCP connected via Vinkius, your agent handles the whole sequence in one conversational turn. You ask it what time works, and it checks every linked calendar simultaneously, giving you a single, actionable result instead of three different screens.

Truto Unified Calendar MCP Gives You True Scheduling Power

You no longer have to write custom logic loops or maintain separate API calls for every platform. Instead, you simply tell your agent to perform the action—be it `create_event` or checking availability using `get_free_busy`.

The system now treats all of your calendars as one unified data layer. It's a massive improvement over what was possible just months ago.

Truto Unified Calendar: 10 Available Tools

Use these tools to perform every scheduling action possible: check availability, write new events, modify existing ones, and manage all connected accounts.

#	TOOL	DESCRIPTION
01	`create_event`	Writes a new event to any linked calendar by providing the ID, title, and start/end times in ISO format.
02	`delete_event`	Removes an existing meeting from both Truto and the underlying provider; this action cannot be undone.
03	`get_calendar`	Retrieves specific details about a single calendar, including its name, timezone, and metadata, normalized to one schema.
04	`get_event`	Gets the full details of a specific meeting, returning information like attendees, location, and recurrence patterns.
05	`get_free_busy`	Checks availability within a date range for a calendar by returning time blocks marked as free or busy.
06	`list_calendars`	Retrieves all connected calendars, unifying the data from services like Google and Outlook into one consistent list.
07	`list_connections`	Shows a comprehensive list of all integrated accounts and their current authentication status across providers.
08	`list_events`	Lists every event associated with a calendar ID, showing titles, times, locations, and who is attending.
09	`update_event`	Modifies an existing meeting by passing a specific JSON object containing the fields you need to change.
10	`validate_connection`	Verifies credentials and permissions for a connected account, reporting any errors or accessibility issues.

See It in Action

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

U List all interconnected calendars belonging to this unified account.

Listing calendars for Integrated Account. Retrieved 2 major calendars:
1. 'Primary Operations' (ID: cal_91bx2) [Google Provider]
2. 'Global Holidays' (ID: cal_84zy1) [Google Provider].

U Search my events and get the specific details for event ID evt_7xkP.

Details recovered. Meeting 'Q3 Alignment Sprint' goes from 2026-10-14T09:00Z to 2026-10-14T10:30Z. Video conferencing link appended natively. 4 attendees invited.

Frequently Asked Questions

01 **How does the Truto Unified Calendar MCP handle different time zones?**

It normalizes all times to ISO-compliant formats, handling the complexity of multiple time zones so you don't have to worry about local offsets when scheduling.

02 **Can I use this MCP to read old meeting details using the get_event tool?**

Yes, `get_event` pulls all rich metadata for a specific ID, giving you attendees, location data, and recurrence patterns, even if that information was added years ago.

03 **Does Truto Unified Calendar MCP only work with Google or Outlook?**

No. The key feature is its unification layer; it treats all connected calendars—including those from other major providers—under one consistent schema.

04 **What if I want to change a meeting after it was created? Do I use update_event?**

Yep. Use `update_event` and pass a JSON object with only the fields you need changed, like just updating the location or adding new attendees.

05 **What is the difference between list_calendars and list_connections?**

Use `list_connections` to see if the account *can* connect (checking auth status). Use `list_calendars` once connected to get a list of actual, usable calendars under that account.

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 `"truto-unified-calendar": { "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

Truto Unified Calendar 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 Truto Unified Calendar. 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	Truto Unified Calendar MCP
Server ID	`019d7616-6f17-7081-90b5-fb48a0a8d622`
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/truto-unified-calendar.