MCP SERVER | NO CODE | CLOUD HOSTED

Deterministic Datetime Engine MCP for AI Agents

Calculate accurate date spans and business days for complex scheduling

The Deterministic Datetime Engine MCP gives your AI client perfect, local temporal math. It handles complex scheduling needs—like calculating business days that skip weekends or figuring out the precise difference between two dates across years and months. Stop relying on guesswork for deadlines and billing cycles.

A+ Quality Score 100/100

temporal-math | date-calculation | business-logic | v8-engine | local-processing | calendar-automation

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/deterministic-datetime-engine

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.

Deterministic Datetime Engine MCP

3 tools available
Cloud-hosted on Vinkius

Most large language models struggle with basic calendar math. If you ask an agent to calculate a deadline like '45 business days from today,' it'll guess, usually wrong, because it can't programmatically skip weekends or handle varying month lengths. This MCP solves that by offloading all date calculations to a strict V8 JavaScript engine running right inside your client. You get mathematically perfect results every time.

This means you don't have to worry about complex billing cycles or project timelines throwing off your estimates. Whether you need to know the exact gap between two historical dates or just figure out when a deadline falls after accounting for holidays, this engine handles it instantly and locally. Connecting this Deterministic Datetime Engine MCP through the Vinkius catalog means all your temporal logic is centralized, giving your agent one reliable source of truth for time.

Core Capabilities

01 — Calculate business day adjustments

Determines a future date by adding or subtracting specific days while automatically skipping weekends.

02 — Determine exact date spans

Calculates the precise mathematical difference between two dates, providing totals in years, months, and days.

03 — Verify leap year status

Confirms if any given year adheres to the official Gregorian calendar rules for a leap year.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/deterministic-datetime-engine** — connect your AI agent in three steps.

01 You ask your AI client to perform a time-sensitive calculation, such as 'What is the deadline 30 business days from now?'

02 Your client routes this request through the Deterministic Datetime Engine MCP, which runs the logic locally using its V8 engine.

03 The MCP returns a single, mathematically accurate date (e.g., October 15th), eliminating guesswork.

The bottom line is you get reliable, programmatic calendar math that your AI agent can use without failing due to complex scheduling rules.

Built For

Project managers and finance analysts need this. If your job involves SLAs, billing cycles, or anything requiring accurate date arithmetic—especially across international time zones—you're hitting the pain point that LLMs can’t solve on their own.

Project Manager

Calculating project milestones and understanding how many working days remain until a major deliverable.

Finance Analyst

Determining accurate billing cycles or payment due dates that must skip weekends or account for specific fiscal holidays.

Operations Planner

Tracking service level agreements (SLAs) and figuring out the exact time elapsed between two critical operational events.

What Changes When You Connect

01 Stop guessing deadlines. Use add_business_days to accurately move project dates forward or backward while skipping weekends.

02 Eliminate billing errors by using calculate_date_difference, which provides precise total counts of years, months, and days between records.

03 Handle complicated calendar rules instantly. You can use check_leap_year to validate any date logic against the official Gregorian algorithm.

04 Improve SLA compliance. Your agent always knows the exact number of working days that passed, removing ambiguity from contract language.

05 Guarantee accuracy by running all calculations locally. This means zero reliance on external APIs or network latency for critical dates.

Real-World Applications

Calculating project completion estimates

A PM asks their agent, 'If we add 20 business days to the design kickoff date, when is the final build phase due?' The agent uses add_business_days and returns a specific, accurate end date.

Determining client tenure for billing

A Finance Analyst needs to know exactly how long an account has been active. They prompt their agent to calculate the difference between signup and today's date using calculate_date_difference, getting a precise total in months.

Validating historical data sets

A researcher needs to check if 2000 was a leap year for their dataset. They use check_leap_year and get an immediate, definitive True/False answer, ensuring the integrity of their calendar logic.

Patterns to Avoid

Relying on basic AI math

✗ AVOID

Asking a general-purpose agent, 'How many days between 2024 and 2028?' without context. The answer might be mathematically rounded or simply wrong.

✓ INSTEAD

Always use calculate_date_difference to get the precise number of years, months, and days, ensuring your historical data is accurate.

Ignoring business day rules

✗ AVOID

Calculating a deadline by just adding 30 calendar days. This misses weekends and holidays, making the resulting schedule impossible.

✓ INSTEAD

Use add_business_days to adjust your dates; this tool guarantees that every calculated day is a weekday.

Assuming universal leap year rules

✗ AVOID

Building scheduling logic assuming all years are divisible by four. This fails on century years not divisible by 400.

✓ INSTEAD

Check the date validity first using check_leap_year to ensure your schedule handles complex Gregorian calendar exceptions.

The Right Fit

Use this MCP if your process depends on absolute time accuracy, particularly in finance, legal, or operations. If you need to calculate payment due dates that skip weekends, use add_business_days. If the core task is finding the gap between two historical points, rely on calculate_date_difference. Don't use it if you just need a simple 'add 7 days'; those basic tasks are fine for general AI agents. However, never let an LLM handle calendar math alone—it's a recipe for bad data.

Deterministic Datetime Engine MCP: Solving Payroll Billing Cycles

In payroll and billing departments, manually tracking time is hell. You're clicking through spreadsheets, checking if the start date falls on a weekend, then calculating how many *actual* workdays elapsed until the next payment cycle starts. It’s tedious, error-prone copy-pasting that keeps auditors up at night.

With this MCP, you simply tell your agent the two dates and say 'calculate the difference.' The engine handles every rule—weekends, month lengths, leap years—and gives you the exact workdays needed for accurate payroll reports. It's instantaneous, perfect arithmetic.

Deterministic Datetime Engine MCP: Mastering Project Milestone Scheduling

Project teams waste hours trying to estimate deadlines because they have to juggle business days versus calendar days. They might calculate a 30-day extension, only to realize the actual deadline falls on a Saturday and has to be pushed back further.

Now you tell your agent to 'add 30 business days.' The MCP handles the pushback automatically. You get the final, accurate date ready for kickoff—no more rescheduling headaches.

3 Tools in the Deterministic Datetime Engine MCP for Date Calculation

These tools let your agent calculate precise date differences, manage workdays, and verify calendar rules with mathematical certainty.

#	TOOL	DESCRIPTION
01	`add_business_days`	Calculates a future or past date by adding or subtracting specified business days, skipping Saturdays and Sundays automatically.
02	`calculate_date_difference`	Provides the precise mathematical duration between two dates in years, months, and days.
03	`check_leap_year`	Verifies if a specific year is a leap year using the official Gregorian calendar algorithm.

See It in Action

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

U If my contract starts on May 28, 2024, what's the exact end date if it runs for 90 business days?

Contract Period Calculation

- Start Date: 2024-05-28
- Duration: 90 Business Days (Excluding Weekends)
- **Calculated End Date: 2024-08-26**

This date accounts for all intervening Saturdays and Sundays, providing a precise termination milestone.

U How many days passed between 1996-03-01 and today?

Date Difference Report

Based on the provided dates:
- Total Years: 28 years
- Months: 4 months
- Days: 7 days

The total elapsed time is **28 years, 4 months, and 7 days**, or a total of 10,369 calendar days.

U Is the year 2100 a leap year? I need to verify my date logic.

✅ **Leap Year Check**

The Gregorian algorithm dictates that a year is a leap year if it is divisible by 4, unless it's divisible by 100 but not by 400.

- Year: 2100
- Is Divisible by 4? Yes.
- Is Divisible by 100? Yes.
- Is Divisible by 400? No.

Result: False. 2100 is not a leap year.

Frequently Asked Questions

01 **How can the Deterministic Datetime Engine calculate accurate business days for payroll?**

It reliably calculates future dates by adding or subtracting specified workdays. This tool automatically skips Saturdays and Sundays, giving you a precise end date that matches actual working schedules.

02 **What is the best way to find out the exact time difference between two historical records?**

You can use the engine's function to get mathematically perfect totals. It returns the total elapsed duration broken down into years, months, and days for easy record-keeping.

03 **Does this MCP help me with complex calendar rules like leap years?**

Yes. The Deterministic Datetime Engine handles these complex calendar exceptions flawlessly using the official Gregorian algorithm, giving you confidence in your date validation logic.

04 **Can I use the Deterministic Datetime Engine for project scheduling?**

Absolutely. You can set accurate milestones by adding business days to a start date, ensuring your projected deadlines never land on a weekend or holiday.

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 `"deterministic-datetime-engine": { "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

Deterministic Datetime Engine 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

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	Deterministic Datetime Engine MCP
Server ID	`019e3886-ca68-7059-a67c-8c051a87a29f`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE