MCP SERVER | NO CODE | CLOUD HOSTED

Device Update Calendar MCP

Automate firmware deployment schedules based on real hardware constraints.

Device Update Calendar MCP acts as a scheduling engine for your device fleet. It looks at release dates, maintenance windows, and hardware limits like battery health or storage to find the best time for updates. Your AI client uses this to plan, verify, and validate every step of the firmware rollout process.

A+ Quality Score 100/100

firmware | maintenance | scheduling | hardware | updates

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · OCTOBER 2026
vinkius.com/en/ai-agent-connect/device-update-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.

Device Update Calendar MCP

4 tools available
Cloud-hosted on Vinkius

Managing firmware rollouts across hundreds or thousands of devices is a logistical nightmare. You can't just push an update and hope for the best. This MCP handles the heavy lifting by evaluating the actual state of your hardware before any code is deployed. It checks if a device has enough battery to survive the flash process or enough storage to hold the new image.

Instead of manually checking every device, you let your agent do the math. It looks at your predefined maintenance windows and finds the first available slot that doesn't conflict with critical operations. It also ensures that the version you are pushing won't break anything by validating the dependency chain. You get a coordinated, data-driven schedule that respects the physical realities of your hardware fleet.

Core Capabilities

01 — Hardware Constraint Checking

Your agent checks battery and storage levels to prevent failed updates.

02 — Maintenance Window Planning

The AI finds the first available time slots that fit your deployment timeline.

03 — Version Compatibility Validation

Your agent verifies that the dependency chain is intact before scheduling.

04 — Fleet-wide Scheduling

The AI generates a complete update plan for all eligible devices at once.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/en/ai-agent-connect/device-update-calendar** — connect your AI agent in three steps.

01 Connect your preferred MCP-compatible client to Vinkius.

02 Ask your agent to check the readiness of specific devices.

03 Use the agent to find the best maintenance windows for your rollout.

04 Validate that all dependencies are met for the target firmware version.

05 Generate and review the final update schedule for the entire fleet.

Connect your AI client to the Vinkius-hosted MCP to start managing your device updates.

Built For

This MCP is built for technical teams managing large fleets of connected hardware. You hand off the complex math of scheduling and hardware verification to your agent.

Embedded Systems Engineer

You use the agent to verify that firmware versions match hardware capabilities.

IoT Fleet Manager

You delegate the coordination of maintenance windows across thousands of nodes.

DevOps Engineer

You automate the validation of deployment readiness for edge devices.

What Changes When You Connect

01 Prevents failed updates by checking battery and storage before deployment.

02 Reduces manual planning by finding the earliest available maintenance windows.

03 Eliminates version conflicts by validating dependency chains automatically.

04 Provides a centralized view of the update schedule for all eligible devices.

Real-World Applications

Fleet Rollouts

Schedule firmware updates for a large group of sensors while respecting their specific battery levels.

Pre-update Verification

Run a check on a specific device to see if it has enough storage for a new software image.

Conflict Avoidance

Ensure new firmware versions don't break existing software dependencies on your hardware.

Maintenance Planning

Identify the best time to perform updates without interrupting critical device uptime.

Patterns to Avoid

Blindly pushing firmware updates

AVOID

You trigger a massive firmware rollout across your entire sensor fleet without checking if the devices can actually handle the new image.

✓ INSTEAD

Use check_device_readiness to confirm battery health and storage capacity before you start the deployment.

Ignoring version dependencies

AVOID

You schedule an update that installs a new version which is incompatible with the existing software configuration on the device.

✓ INSTEAD

Run validate_dependency_chain to ensure the current configuration satisfies all version requirements first.

Manual window hunting

AVOID

You spend hours looking through logs to find a gap in your maintenance schedule that won't disrupt operations.

✓ INSTEAD

Let your agent find the first available slot using find_earliest_window.

The Right Fit

Technical teams managing large fleets of connected hardware should connect this MCP. If you only manage a handful of devices that you can check manually, you don't need this. The deciding factor is whether you need to automate the validation of deployment readiness across thousands of nodes.

Device Update Calendar 4 Tools

These tools handle the technical math of hardware readiness, maintenance scheduling, and version dependency checks.

#	TOOL	DESCRIPTION
01	`check_device_readiness`	This tool evaluates a single device against all technical requirements. It checks things like battery health and storage capacity to ensure an update won't fail.
02	`find_earliest_window`	This tool identifies the first available slot in your maintenance windows. It finds the best time for an update based on your specific timeline.
03	`get_update_schedule`	This tool retrieves the planned update windows for all eligible devices. It gives you a clear view of when every device is set to update.
04	`validate_dependency_chain`	This tool checks if the current device configuration satisfies complex version requirements. It prevents version mismatches before you start a rollout.

See It in Action

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

U What is the update schedule for my devices given these maintenance windows?

Device ID 101 is scheduled for the window 2024-10-01T08:00:00Z to 2024-10-01T12:00:00Z. Device ID 102 is blocked due to insufficient battery.

U Is my device ready for the new firmware update?

No, the device is not ready. It is missing the required storage capacity and the backup status is not confirmed.

U Check if the current dependency versions are compatible with the update.

The dependency chain is valid. All installed versions meet or exceed the minimum requirements.

Frequently Asked Questions

01 **How does this MCP prevent failed firmware updates?**

It uses the check_device_readiness tool to verify that hardware constraints like battery health and storage capacity are met before an update is scheduled.

02 **Can I use this with Claude or Cursor?**

Yes, you can connect this MCP to any MCP-compatible client including Claude, Cursor, and Windsurf.

03 **How does it handle version compatibility?**

The validate_dependency_chain tool checks the current device configuration against required versions to ensure the update is safe to perform.

04 **Does Vinkius host the MCP?**

Yes, Vinkius hosts and manages the MCP, so you don't have to worry about the underlying infrastructure or credentials.

05 **Can I schedule updates for multiple devices at once?**

Yes, the get_update_schedule tool allows your agent to retrieve planned windows for all eligible devices in your fleet.

06 **How does the system handle blocked devices?**

A device is marked as blocked if it fails any mandatory constraint, such as insufficient battery or missing backups. You can use `get_update_schedule` to see a list of all blocked devices and the specific reason for each block.

07 **Can I check if a specific device is ready for an update?**

Yes, you can use the `check_device_readiness` tool to evaluate a device's battery level, storage, and dependency versions against the required update parameters.

08 **How are maintenance windows selected?**

The system uses `find_earliest_window` to identify the first available time slot that occurs on or after the official release date of the update.

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 `"device-update-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

Device Update 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 Device Update 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	October 2026
MCP Server	Device Update Calendar MCP
Server ID	`01a0ee75-d6d0-70dc-98d2-1099b407de1b`
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/en/ai-agent-connect/device-update-calendar.

© 2026 Vinkius · vinkius.com · support@vinkius.com