MCP SERVER | NO CODE | CLOUD HOSTED

HaloPSA MCP

Manage ticketing, clients, and assets with chat.

HaloPSA connects your service desk and PSA tools directly to your AI client. Manage entire support workflows—from opening a new ticket to tracking assets, reviewing contracts, or updating statuses—all through natural conversation. It lets you run full operations using only text prompts.

A+ Quality Score 100/100

psa | service-desk | it-service-management | ticket-management | client-management

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

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.

HaloPSA MCP

11 tools available
Cloud-hosted on Vinkius

Using this MCP, your agent gains complete control over your service desk and PSA data. Instead of navigating multiple tabs in the HaloPSA portal, you simply ask for what you need. For instance, if a client calls about an issue, you can immediately list all their assets or retrieve details on their current contract status, all without leaving the chat window. You can create new support tickets instantly, look up who used which site, and even perform actions like adding internal notes or changing ticket statuses. It’s like having a dedicated Operations Specialist sitting next to you, ready to execute any command against your system.

Core Capabilities

01 — Create Support Tickets

You generate new support requests in HaloPSA using simple text commands.

02 — Review Ticket Statuses and Details

The agent pulls detailed information on specific tickets or provides a complete list of all open items for review.

03 — Audit Client and User Data

You retrieve comprehensive lists of customers, users, sites, and teams to verify data accuracy across your organization.

04 — Track Assets and Finances

The system pulls records for hardware assets, customer contracts, and invoices, giving you quick financial oversight.

05 — Update Records on the Fly

You perform necessary actions like adding internal notes or updating a ticket's status directly from your chat session.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your HaloPSA credentials (Client ID, Client Secret, etc.).

02 Your AI client authenticates the connection through Vinkius's catalog and makes your agent capable of communicating with HaloPSA.

03 You prompt your agent in natural language—for example, 'List all open tickets for Acme Corp'—and get real-time data or confirmation that an action was performed.

The bottom line is you stop jumping between dashboards and start managing service desk operations entirely through conversation.

Built For

Anyone who spends too much time clicking, copying data, or cross-referencing information across different PSA tabs will need this. It's for the people whose job is to know everything about a client and their service history.

Service Desk Agent

You use it to instantly pull ticket histories, add internal notes during calls, or confirm asset ownership without switching screens.

Operations Manager

You run audits on team assignments, review all active customer contracts, and manage the overall flow of service requests across sites.

IT Director

You get a real-time overview of support volume by listing tickets or checking financial status via invoices to assess departmental load.

What Changes When You Connect

01 Stop manually checking asset lists. You can ask the agent to list all assets linked to a client or site, getting that data instantly for your audit.

02 Never lose context again. Need to add an internal note while talking to a user? Use perform_ticket_action to update status and notes in one go.

03 Get financial clarity without logging into accounting. List invoices or check customer contracts right alongside ticket details to understand the full picture.

04 The agent handles client data retrieval, so you don't have to run separate reports for list_clients or list_users before a meeting.

05 Manage everything from one place. You can create new tickets using create_ticket and then immediately get more details with get_ticket—all in the same chat thread.

Real-World Applications

Investigating an Account Issue

A customer calls, upset about a service gap. You tell your agent to list_clients for their company, then check list_assets to see what equipment they own, and finally retrieve the latest contract status using list_contracts.

Billing Inquiry Resolution

A client questions a charge. You ask the agent to pull the most recent list_invoices, cross-reference it with their contracts using list_contracts, and then use get_ticket if the billing issue started as a support request.

Daily Team Handoffs

It's end of day. Instead of printing out reports, you prompt your agent to list_tickets for all sites and check list_teams to see which teams are currently overloaded or underutilized.

Onboarding New Hardware

A new laptop arrives. You ask your agent to list_assets to verify its serial number is logged, and you can simultaneously create_ticket for setup help while checking if the user exists via list_users.

Patterns to Avoid

Switching between tabs

✗ AVOID

Checking a ticket status on the main dashboard, then copying the client ID to run a separate report on assets, and finally pasting that data into an email draft.

✓ INSTEAD

Just ask your agent: 'For the ticket assigned to Client XYZ, show me their current contract details and list all associated assets.' It handles the cross-referencing for you.

Manual status updates

✗ AVOID

Finding a ticket, manually changing its status from 'Open' to 'Pending', then having to remember to add a note about why.

✓ INSTEAD

Use perform_ticket_action. Tell the agent: 'Set ticket 1025 status to Pending and add an internal note stating I'm waiting for vendor feedback.' Two steps in one prompt.

Overloading prompts

✗ AVOID

Asking the agent to list all users, then list all teams, and then list all sites—all in one massive block of text.

✓ INSTEAD

Break it up. Start with: 'Show me a summary list of active organizational sites.' Then follow up: 'Now show me which support teams are assigned to those sites.'

The Right Fit

Use this MCP if your job requires you to read, write, and act upon data spread across multiple functional areas—support tickets, asset inventories, client records, and financials. If the answer is yes, this connector saves massive amounts of time because it centralizes all that operational knowledge into a single chat interface.

Don't use it if your goal is only to view one type of data (like just viewing a list of users). In those cases, connecting an MCP with dedicated read-only tools might be enough. But if you need the ability to *act*—to create records or change statuses using tools like create_ticket or perform_ticket_action—this is what you need.

The pain of context switching in service desk management

Today, resolving a single issue means bouncing between five different panels. You open the ticketing system to get the ticket details, then switch to the client portal to verify their contact info, move to an asset registry to check warranty status, and finally jump to a financial tab just to see if they're under contract terms. This constant clicking is where hours vanish.

With this MCP, your agent acts like a single brain for all that data. Instead of tabs, you talk to the chat window. You ask about an asset, and it pulls the record; you ask about billing, and it checks contracts. The result? Your AI client handles the complexity, giving you only the answer you need.

HaloPSA: Control your support workflow with HaloPSA MCP

You no longer have to manually copy a ticket ID and paste it into another tool just to update its status or add an internal note. You tell the agent what needs doing, referencing specific tools like get_ticket and perform_ticket_action.

It's not just data retrieval; it's execution. The system becomes your hands on the keyboard, making you a true Operations Specialist right from your chat.

HaloPSA: 11 Tools for PSA Management

These tools let your agent read, write, and act across every part of your service desk system, from user directories to billing records.

#	TOOL	DESCRIPTION
01	`create_ticket`	Generates and submits a brand new support ticket directly into the HaloPSA system.
02	`get_ticket`	Retrieves all detailed information for one specific ticket ID, including its history and notes.
03	`list_assets`	Pulls a comprehensive list of every piece of hardware or software asset recorded in the system.
04	`list_clients`	Provides a full roster of all customer accounts defined within HaloPSA.
05	`list_contracts`	Lists every active and expired customer contract for status review.
06	`list_invoices`	Retrieves a list of all generated invoices, allowing you to check payment statuses.
07	`list_sites`	Shows an inventory of every physical location or operational site tracked by the service desk.
08	`list_teams`	Lists all organizational teams configured for support and resource allocation within HaloPSA.
09	`list_tickets`	Fetches a bulk list of tickets based on criteria like status or assignee.
10	`list_users`	Presents the complete directory of all individual users managed by your service desk instance.
11	`perform_ticket_action`	Allows you to execute specific changes on a ticket, such as updating its status or adding an internal note.

See It in Action

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

U List all open tickets assigned to me.

I've retrieved your tickets. You currently have 3 open items: 'Internet connectivity issue' (ID: 1021), 'Printer setup request' (ID: 1025), and 'Software license renewal' (ID: 1030). Which one would you like to update?

U Add an internal note to ticket ID 1021: 'Waiting for vendor feedback'.

Action performed! I've successfully added the internal note to ticket 1021. Would you like to change its status to 'On Hold' as well?

U Show me the asset list for Client 'Acme Corp'.

Accessing assets... I found 5 assets linked to Acme Corp, including 3 Laptops (MacBook Pro), 1 Server (PowerEdge), and 1 Firewall (FortiGate). Shall I list the serial numbers for these items?

Frequently Asked Questions

01 **How do I set up my HaloPSA API credentials?**

Log in as an Administrator, go to **Configuration > Integrations > Halo API**. In 'View Applications', create a new app with the 'Agent' login type and grant the necessary permissions. You will receive your Client ID and Client Secret there.

02 **What are the Resource and Auth URLs?**

The Resource URL is typically `https://your-tenant.halopsa.com/api` and the Auth URL is `https://your-tenant.halopsa.com/auth`. Replace 'your-tenant' with your actual HaloPSA instance name.

03 Can I perform actions on tickets, like changing status?

Yes! Use the `perform_ticket_action` tool by providing the ticket ID and the specific action ID configured in your Halo instance. You can also add an optional note.

04 Is the integration secure?

Yes, it uses industry-standard OAuth2 Client Credentials flow. Your credentials are encrypted and stored securely within the Vinkius Cloud infrastructure.

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 `"halopsa": { "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

HaloPSA 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 HaloPSA. 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	HaloPSA MCP
Server ID	`019d75ad-caea-70f2-9f9b-01a6492dd145`
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/halopsa.