

MCP SERVER

NO CODE

CLOUD HOSTED

# HappyFox MCP

Manage support cases from conversation.

HappyFox connects your entire customer service operation directly into any AI agent. Manage tickets, track contacts, and update staff notes without ever leaving your chat window. You get full control over complex help desk workflows—all from natural conversation.

**A+** Quality Score 100/100

ticket-management

automated-workflows

customer-service

support-automation

case-tracking

helpdesk



# 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.

### 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.

### 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.

### 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**

Encryption at rest

**Ed25519**

PKI vault signatures

**24h TTL**

Ephemeral session keys

**V8 Isolate**

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.

### 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.

### 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.

### 06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

**FinOps**

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

**Firewall & DLP**

PII redaction activity, sensitive data protection counters, and security event timeline.

**Agent Activity**

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

**Tool Health**

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

**Incident Log**

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

Get started at [cloud.vinkius.com](https://cloud.vinkius.com) — connect your AI agent in under 60 seconds.

# HappyFox MCP

10 tools available

Cloud-hosted on Vinkius

This MCP lets you take the manual work out of managing a support help desk. Instead of clicking through five different tabs to find ticket status, contact history, or category rules, you just talk to your agent. It acts like having a dedicated Support Coordinator on your team.

Your agent can check all open tickets, pull up client details by name, and even add staff updates directly to the case file—all without needing an API key or complex commands. You'll find this entire suite of functionality managed through Vinkius, giving you one connection point for thousands of tools across different services. This makes your AI agent a true operational extension of your team.

Need to create a new ticket because a client just called? Your agent handles the required subject, text, category ID, and contact details in sequence. Need to know who needs help? You can list contacts or search through entire ticket histories using simple text queries.

---

## Core Capabilities

### 01 — Search support tickets by keywords

Your agent searches the entire history of cases for specific words or topics.

### 02 — Check all open service requests

It pulls a list of all support tickets, allowing you to filter and see what needs attention.

You can get comprehensive information about any specific support ticket ID.

### 03 — Pull details on one ticket case

It lets you log a staff update or response directly into the case file for the customer to see.

### 04 — Add responses to an active ticket

The agent searches and lists all existing contacts in your help desk database.

### 05 — Find and list customer records

You can create an entirely new ticket by providing the subject, body text, category, and contact information.

### 06 — Start a new support ticket

# One Click on Vinkius — From Prompt to Execution

Available at [vinkius.com/mcp/happyfox](https://vinkius.com/mcp/happyfox) — connect your AI agent in three steps.

- 01 Subscribe to this MCP on Vinkius and enter your HappyFox API Key, Auth Code, and Account Name.
- 02 Connect your agent client (like Claude or Cursor) to the MCP via your profile settings.
- 03 Start by asking your agent a question like, 'List all pending tickets in the billing category.' and watch it pull the data for you.

The bottom line is, you get to talk to your support desk system using natural language instead of navigating complex dashboards.

---

## Built For

This MCP is built for Support Managers and Customer Success Teams who are tired of switching between the ticketing platform and their AI agent just to pull status updates or log notes. If your job involves reviewing ticket history and coordinating with staff, this saves you hours a week.

### Support Manager

You use it to list all available categories, check team workloads, or retrieve general reports on service statuses without pulling up the full dashboard.

### Customer Success Agent

On a follow-up call, you ask your agent to pull details for a specific contact and then add a staff update right into the ticket history.

### IT Support Staff

When a new issue comes up, you use it to create an internal technical ticket instantly by providing all required data points like subject and category ID.

## What Changes When You Connect

- 01 Stop manually compiling reports. Instead of digging through dashboards to find ticket statuses, your agent lists all tickets instantly. Need to check the status or list details for a specific case?
- 02 Keep clients in the loop without copy-pasting. Use the `add_staff_update` tool to log responses directly into the ticket file, ensuring customers see real-time updates.
- 03 Never start from scratch. If you need to create a new support request, your agent handles the required subject, text, category ID, and contact details with just a few prompts using `create_ticket`.
- 04 Get full visibility on who needs help. You can use `list_contacts` to find customer records or run `search_tickets` to trace every historical interaction across all clients.
- 05 Streamline internal coordination. Need to know who's available? Use `list_staff` to get a roster of your team members and their assigned roles in seconds.

---

## Real-World Applications

### Handling an Escalation

A high-priority client calls about a recurring bug. Instead of opening the manual search panel, you ask your agent to `'search_tickets'` for their name and 'bug'. The agent pulls up all related history so you can address them immediately.

### Onboarding New Staff

A new team member needs access to the system. You use your agent to `'list_staff'` to see who is currently set up and then check ticket categories using `'list_categories'` to understand workflow rules.

### The Daily Huddle

At morning standup, you ask: 'Show me all pending tickets in the General category.' Your agent uses `list\_tickets` to pull a filtered list of items that need attention right now.

---

## Patterns to Avoid

---

### Trying to find information across systems

#### ✗ AVOID

Manually opening your ticketing system, then switching to your CRM, and copying/pasting data to draft a follow-up email.

#### ✓ INSTEAD

Use this MCP. Your agent can `list\_contacts` to get the customer's details and then use those same details when you run `get\_ticket` to pull up their history in one conversation.

---

### Forgetting required fields for a new ticket

#### ✗ AVOID

Drafting an email saying, 'There's a problem,' but forgetting which category it falls under or who the client is.

#### ✓ INSTEAD

Just prompt your agent to `create\_ticket`. It guides you through collecting all mandatory information—subject, text, category ID, and contact details—before submitting.

---

### Needing a simple status check

#### ✗ AVOID

Scrolling down pages of tickets just to see if the ticket was reassigned or closed.

#### ✓ INSTEAD

Ask your agent to `list\_statuses` first, then use `list\_tickets` with filters to get exactly the view you need without scrolling.

---

## The Right Fit

Use this MCP if your workflow involves constant context switching between communication (chat) and data management (ticketing system). This is perfect for agents who need to check status, update notes, and create records frequently. Don't use it if you only need to read a static report or generate external documentation; in those cases, a simple document generation MCP will work better. If your goal is to build complex logic that requires multiple sequential actions (e.g., 'If ticket priority is high, then notify staff X and assign

category Y'), look for an automation workflow tool, not just a data retrieval one.

---

---

## The pain of jumping between tabs all day long

Right now, if you need to know what's going on with a client, your day looks like this: open the ticketing platform, navigate to 'Contacts,' then click into the specific customer record. Next, you have to switch tabs to view their entire history and scroll through pages of notes just to see if someone already updated it. If you need to create something, you're forced to remember three different IDs—the category ID, the status ID, and the contact ID—before you can even start typing.

With this MCP connected through Vinkius, that entire process vanishes. You don't manage tabs or click through complex menus anymore. You simply speak to your agent: 'What did John Doe say about his API issue?' The agent handles all the back-end lookups and data retrieval for you in a single conversation. It just works.

---

## HappyFox MCP gives you control over every case interaction

Before, logging an update was manual work: remember to open the correct ticket ID, find the right section for internal notes, and then type out your response. If you missed a step or used the wrong category, the whole record was messy.

Now, you simply tell your agent, 'Add a staff update to ticket 102: We found the root cause.' That's it. The agent logs the note correctly, tracks the action, and keeps the entire system clean. It's instant process control.

---

# HappyFox: 10 Tools for Support Operations

These tools let you perform every core function of a ticketing system, from listing contacts to creating new cases, all through conversation.

#	TOOL	DESCRIPTION
01	<code>add_staff_update</code>	Logs a response or update directly into an existing support ticket case.
02	<code>create_ticket</code>	Generates a brand new support ticket when given the subject, text body, category ID, and contact details.
03	<code>get_ticket</code>	Pulls complete, detailed information for one specific support case using its ID.
04	<code>list_categories</code>	Provides a list of all available ticket categories that your team uses.
05	<code>list_priorities</code>	Shows every possible priority level assigned to a support ticket, from low to high.
06	<code>list_staff</code>	Retrieves names and roles for all staff members currently using the help desk system.
07	<code>list_statuses</code>	Lists every current status a ticket can be in, like 'Open' or 'Closed'.
08	<code>list_tickets</code>	Fetches a comprehensive list of all support tickets using filtering and pagination parameters.
09	<code>list_contacts</code>	Pulls a list of every user or contact recorded in the help desk system.
10	<code>search_tickets</code>	Finds relevant tickets by running a general text search across the entire ticket history.

---

## See It in Action

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

**U** List all pending tickets in the 'General' category.



I've retrieved your tickets. You have 4 pending items in 'General', including 'Login issue' (ID: 102), 'Billing question' (ID: 105), and 'API feedback' (ID: 110). Which one would you like to update?

**U** Add a staff update to ticket ID 102: 'Looking into this now'.



Update added! I've successfully logged your response to ticket 102. The customer will be able to see this in their support portal. Would you like to change the ticket status to 'In Progress'?

**U** Create a new ticket for 'App crash on startup' in the Technical category.



Process started! I found the 'Technical' category (ID: 5). To create the ticket, I'll need the customer's name and email address. Could you provide those?

---

## Frequently Asked Questions

### 01 How does HappyFox MCP help me list tickets?

You use the `list\_tickets` tool to pull a comprehensive list of all open support requests. You can also run `search\_tickets` if you're looking for something specific in the history.

### 02 Can HappyFox MCP create new tickets?

Yes, using the `create\_ticket` tool. Just give your agent the subject, the text body, and the category details, and it builds the entire record for you.

---

**03 Is listing categories part of HappyFox MCP?**

Yes, the `list\_categories` tool lets you see every available ticket category. This is critical when you need to create a new case correctly.

---

**04 What if I just want to check client information with HappyFox MCP?**

Use the `list\_contacts` tool. It pulls all existing users in your help desk database, so you can quickly verify who is associated with a ticket.

---

**05 How do I log notes using HappyFox MCP?**

You use the `add\_staff\_update` tool to record notes. You just tell your agent what update you want, and it logs it directly into the appropriate ticket.

---

# Go Live in 60 Seconds

Get your connection token from [cloud.vinkius.com](https://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 `"happyfox": { "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

# HappyFox 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](https://cloud.vinkius.com) →

[vinkius.com](https://vinkius.com) · [support@vinkius.com](mailto: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 HappyFox. 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	HappyFox MCP
Server ID	019d75ae-1701-72cd-837a-2542b74c2619
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

### 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/happyfox](https://vinkius.com/mcp/happyfox).