MCP SERVER | NO CODE | CLOUD HOSTED

Landbot MCP

Manage every step of the customer conversation flow.

Landbot MCP lets your AI client manage entire conversational funnels from start to finish. You can pull detailed customer records, view full chat histories, send automated follow-up messages, or instantly hand off a conversation from a bot to a live agent—all without leaving your current workspace. It gives your agents the ability to act directly within complex customer support pipelines.

A+ Quality Score 100/100

chatbots | conversational-ai | lead-capture | automated-messaging | customer-engagement | no-code

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

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.

Landbot MCP

8 tools available
Cloud-hosted on Vinkius

Managing customer conversations used to mean jumping between five different dashboards just to get context: one for chat history, one for lead status, another for active bots, and maybe a fourth just to send a quick follow-up text. This MCP changes that. It connects your AI agent directly into the Landbot system, letting it see and manipulate live conversations as if it were a human operator sitting at the desk.

Your agent can search for specific customers by email or pull up an entire chat sequence log to understand exactly what happened before you got involved. If the conversation is complicated, your agent can programmatically reassign that thread directly to a specialized human agent using simple commands. Plus, it lets you send automated text messages right into the active dialogue channel. Because this MCP lives in the Vinkius catalog, your AI client connects once and gets access to all these critical communication tools for your whole operation.

Core Capabilities

01 — List available bots

View a list of every bot operating within the Landbot system.

02 — Get specific bot details

Fetch detailed information about one particular bot by its unique ID.

03 — Find customer records

Search for a specific customer profile using an email address.

04 — Retrieve customer data

Pull core details and metadata for a single identified customer.

05 — Fetch chat message history

Download the complete chronological transcript of messages exchanged with a customer.

06 — Hand off conversation to staff

Programmatically reroute an active bot dialogue and assign it to a live human agent.

07 — Send automated text replies

Send a structured message directly into the customer's ongoing chat thread.

One Click on Vinkius — From Prompt to Execution

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

01 First, you authorize your AI client connection through Vinkius and provide the Landbot API token in the credentials.

02 Next, you instruct your agent to perform a specific action, like fetching chat logs or rerouting an interaction. The MCP executes that command against the Landbot system.

03 Finally, your agent receives structured data—whether it's the customer's full message history, their profile details, or confirmation that the conversation has been assigned to Sarah.

The bottom line is: you use natural language prompts in your AI client instead of logging into multiple service dashboards.

Built For

This MCP is for Operations Engineers, Sales Marketers, and Product Managers who are tired of manually hopping between customer support tools to get a complete view of a single interaction. You need immediate context, reliable data retrieval, and the ability to programmatically change the flow when needed.

Customer Success Manager

On a Tuesday afternoon, you use this MCP to retrieve a customer's full chat history (`get_messages`) before calling them. This ensures your conversation starts with the correct context and shows the customer that you actually read their previous messages.

Sales Operations Specialist

When an AI bot collects a lead, you use this MCP to search for that contact's record (`search_customers`) and pull their complete profile details before triggering the next sales sequence. This prevents wasted effort on unqualified leads.

Product Manager

You examine conversation flows by listing all active bots (`list_bots`) to understand which customer journeys are being run. You can then use `get_customer` data to map out common friction points in the user path.

What Changes When You Connect

01 Instantly see full context. Instead of asking a customer, 'What were we talking about?', your agent runs `get_messages` and has the entire chat transcript ready to analyze and respond to, ensuring zero friction in service.

02 Control the handoff process. If the AI can't solve it, you don't want the conversation to stall. Use `assign_agent` to immediately route a complex ticket to a human agent, guaranteeing continuity of care.

03 Deep customer profiling. Need more than just a name? Running `get_customer` retrieves key metadata about the account, letting your agent tailor responses based on subscription level or purchase history.

04 Targeted outreach. You can't wait for a lead to come back online. Use `send_text_message` to send a specific follow-up message directly into their chat channel when they need it most.

05 Operational oversight. By using `list_bots`, your team can quickly audit which conversational pathways are running and check the status of every bot without logging into separate monitoring dashboards.

Real-World Applications

A sales rep needs to qualify a new lead.

The rep doesn't want to rely on memory. They prompt their agent to search for the prospect by email using `search_customers`. The agent instantly pulls up the contact's full profile, allowing the rep to immediately reference specific details—like which bot they interacted with last week—and ask highly targeted questions.

Support needs to escalate a complex technical issue.

The initial chatbot interaction failed because it hit a knowledge gap. The support agent tells their AI client to run `get_messages` for the customer ID, pulling up every single message exchanged. This history allows the agent to understand the full scope of failure and then use `assign_agent` to hand off the case with a perfect summary.

Product team needs to analyze chat bottlenecks.

The product manager wants to know why customers drop off. They run `list_bots` first, identifying all active funnels. Then they use `get_bot` on a specific bot and cross-reference that data with customer profiles (`list_customers`) to pinpoint exactly where the conversation fails or gets stalled.

Marketing needs to re-engage an inactive lead.

The marketing team identifies high-value leads who went silent. They use `get_customer` to pull up their last known details, then instruct the agent to send a personalized message using `send_text_message`, restarting the conversation loop without human intervention.

Patterns to Avoid

Searching for customer data manually

✗ AVOID

The user opens the chat interface, scrolls up through dozens of messages to find a specific detail, and then has to copy that information into an internal CRM ticket. This takes five minutes per client.

✓ INSTEAD

Don't scroll or copy anything. Ask your agent to run `get_customer` first. The tool pulls the metadata instantly and provides it in clean, structured format for immediate use.

Assuming a bot is still running

✗ AVOID

A developer assumes that because they set up a new bot matrix, it's live and functional when testing. They waste time manually checking multiple internal dashboards to verify its status.

✓ INSTEAD

Before you test anything, always run `list_bots`. This confirms if the system sees the bot as active and available for use before you start running complex simulations.

Trying to fix a conversation without history

✗ AVOID

A support agent jumps in chat with 'Hi, how can I help?' but realizes they have no idea why the customer is upset because they didn't pull the log first. This leads to frustration and repeat questioning.

✓ INSTEAD

Always start by calling `get_messages` for that customer ID. The full transcript provides immediate context so your agent can address the root problem right away.

The Right Fit

Use this MCP if your workflow involves managing, analyzing, or acting upon conversational data—specifically chat transcripts and lead funnel status. If you need to know who a customer is, what they said, where they are in the pipeline, or if an automated message needs to be sent, this MCP provides the necessary tools

(`get_messages` , `search_customers` , `send_text_message`). Don't use it if your goal is simple data retrieval that doesn't involve a conversation context. For instance, if you just need a list of all employee names from an HR database, you need a general CRM tool, not one designed for chat flow control. You must be focused on the lifecycle of customer interaction.

The Struggle of Context Switching

Today, understanding a single customer's journey requires painful context switching. You start in the primary chat tool to see their last message, then you jump to your CRM dashboard to verify if they are paying customers, and finally, you open an internal knowledge base just to find out which bot was supposed to handle them. This manual process means critical details—the payment status or the specific funnel name—are often missed in the rush.

With this MCP, all that data lives within your AI agent's scope. You simply ask for a customer's profile and their chat history simultaneously. Your agent retrieves the metadata (`get_customer`) *and* the full transcript (`get_messages`), delivering one single answer that gives you 100% context right when you need it.

Landbot MCP: Take Action, Not Just Notes

The biggest waste of time is gathering data only to have to switch back to a UI button to act on it. You pull up the full history and see the bot failed to resolve the issue; you then have to manually change the ticket status, find the right human team, and write an internal note about the handoff.

Now, your agent handles the action for you. After reviewing the chat log, you simply tell it to route the conversation using `assign_agent` . The MCP takes care of updating the system, changing the ownership, and notifying the correct person—all from a single prompt.

Landbot: 8 Conversation Tools

Use these tools to interact with customer data, list bots, retrieve message transcripts, or assign agents directly from your AI client.

#	TOOL	DESCRIPTION
01	`assign_agent`	Programmatically hands off a conversation thread from the bot to a live human agent.
02	`get_bot`	Retrieves specific details for one Landbot instance using its unique identifier.
03	`get_customer`	Fetches the core profile and metadata for a single customer account.
04	`get_messages`	Pulls the complete, chronological transcript of all messages exchanged with a customer.
05	`list_bots`	Lists every Landbot active in your organization's account.
06	`list_customers`	Retrieves a list of the most recently interacting customers with the bots.
07	`search_customers`	Searches for and locates a specific customer profile using their email address.
08	`send_text_message`	Sends an automated message directly into the ongoing conversation with a customer.

See It in Action

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

U List standard bots running active pipelines right now.

Loading Landbot configurations. There are currently 3 standard matrix bots running active logic mapping sequences.

U Fetch the entire transcription log for customer ID 98453.

Extracted session log for 98453. The prospect answered 4 initial screening questions mentioning 'Price points'.

U Force assign the highest severity angry customer ticket to Agent Sarah.

Operation complete. I've intercepted the live interaction and injected the routing update pointing to Agent Sarah gracefully.

Frequently Asked Questions

01 **How do I use Landbot MCP to see all my active bots?**

You run `list_bots` to get an inventory of every bot running in your account. This is the first step if you need to audit or understand what conversational pipelines exist.

02 **What information does Landbot MCP provide about a customer?**

You can use `get_customer` to pull core profile details and metadata on a single individual. You also run `search_customers` if you only know their email address.

03 **Can I send messages through the Landbot MCP?**

Yes, you can use `send_text_message` to programmatically inject an automated reply into a customer's live chat conversation. This is useful for follow-ups or confirming information.

04 **How do I make sure a difficult chat gets to a human?**

Use the `assign_agent` tool. Your agent intercepts the dialogue and routes it away from bot control, assigning ownership to a live team member immediately.

05 **Is Landbot MCP only for viewing data?**

No, it's much more than that. You can perform actions like sending messages and reassigning agents, making it an active control layer over your customer service system.

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 `"landbot": { "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

Landbot 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 Landbot. 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	Landbot MCP
Server ID	`019d75c4-4c78-711d-a87b-9a3179f9989d`
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/landbot.