MCP SERVER | NO CODE | CLOUD HOSTED

Bland AI MCP for AI Agents

Automate Conversational Phone Calls and Call Analytics

Bland AI automates entire phone communication pipelines, letting your AI agents dispatch conversational calls and analyze the results. Connect it to manage inbound numbers, retrieve full call transcripts, or instantly determine if a sales goal was met during a conversation.

A+ Quality Score 98.33/100

voice-ai | conversational-ai | telephony | automated-calling | webrtc | call-analytics

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/bland-ai

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.

Bland AI MCP

10 tools available
Cloud-hosted on Vinkius

This MCP gives your AI client total control over enterprise-grade telephony and voice workflows. You can programmatically manage everything from outbound campaigns to complex customer support monitoring. Need to run a large campaign? Use the bulk dispatch tools to send conversational agents to hundreds of numbers simultaneously. Finished a call, but need to know if the agent hit its goal? You pull the transcript and analyze it instantly for specific outcomes. For operations teams, this means you can retrieve full historical call logs or grab raw recordings as MP3 files for quality assurance. It also lets you view all purchased inbound numbers so your AI client knows where to route incoming customer calls. This capability makes Bland AI a central piece of infrastructure that integrates perfectly with the thousands of tools available in the Vinkius catalog, putting telephony control right into your agent's hands.

Core Capabilities

01 — Execute Bulk Campaigns

Dispatch single or large batches of conversational agents to target phone numbers at scale.

02 — Monitor and Control Live Calls

Instantly end a live call, analyze recordings for goal completion status, or retrieve detailed transcripts from completed calls.

03 — Manage Inbound Routing

Identify available inbound phone numbers and create web-based signaling sockets for browser audio testing.

One Click on Vinkius — From Prompt to Execution

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

01 Connect your Bland AI API key to Vinkius, then link the MCP to your preferred AI client (Claude, Cursor, etc.).

02 Use your agent to list available inbound phone numbers or retrieve historical call logs.

03 Issue a command to dispatch agents, analyze a transcript, or end an active call.

The bottom line is, once connected, you treat the entire telecom system like another toolset within your AI client's natural conversation flow.

Built For

Sales and Operations teams running high-volume campaigns need this. Customer Support managers who rely on call quality monitoring will find it invaluable. Developers building conversational features also depend on its detailed control over live calls.

Sales Development Representative (SDR)

Sending targeted, automated calling campaigns and querying transcripts immediately to determine lead qualification status after the call ends.

Customer Support Manager

Monitoring specific inbound numbers, listening back to raw recordings, and evaluating agent performance against established goal criteria.

Contact Center Developer

Debugging live conversational flows by manually disconnecting rogue calls or testing WebRTC fallback channels without leaving the IDE.

What Changes When You Connect

01 Determine lead qualification status instantly. Use the `analyze_call` tool to query transcripts and confirm if a specific goal was met without manual review.

02 Scale your outreach campaigns immediately. The `send_batch` tool lets you dispatch hundreds of agents concurrently, far faster than any manual dialing system.

03 Keep full control over live interactions. If an agent gets stuck or goes off-script, the `end_call` tool lets you shut down the session instantly.

04 Streamline data collection. The `get_recording` tool pulls native MP3/WAV files directly, so you never have to rely on a limited text transcript for QA logging.

05 Manage your entire phone presence. Use `list_inbound` to see all available numbers and `send_call` to test agent performance against a single target number.

Real-World Applications

Evaluating Sales Campaign Outcomes

An SDR runs a batch campaign using `send_batch`. After the calls, they ask their agent to use `get_call_details` for five specific IDs. The agent reviews the transcripts and reports back, showing which leads achieved the 'demo interest' goal.

Handling Critical Support Incidents

A support team member notices an agent is repeating bad advice during a live call. They use `end_call` to immediately interrupt and shut down the session, preventing further damage while they step in.

Building Internal Testing Tools

A developer needs to test how their new web application handles voice input. They use `create_web_call` first, then simulate a call flow using the resulting signaling socket, all within their IDE.

Auditing Recorded Interactions

The QA manager needs to check audio quality for 50 calls. Instead of relying on text summaries, they use `get_recording` for each call ID, pulling the native MP3 files directly into a bulk analysis folder.

Patterns to Avoid

Treating Calls as Simple Text Logs

✗ AVOID

Assuming that simply fetching data with `get_call_details` provides enough context, and missing nuances about the conversation's success.

✓ INSTEAD

Don't just fetch transcripts. Run a targeted analysis using `analyze_call`. This tool interrogates the raw recording to give you a definitive goal completion status.

Manual Scaling of Outreach

✗ AVOID

Trying to call 500 leads one by one, wasting hours and missing out on necessary data points.

✓ INSTEAD

Use `send_batch` instead. This tool handles the bulk telecom dispatching, allowing you to scale your outreach instantly across entire arrays.

The Right Fit

You should use this MCP if your core business process involves high-volume, automated voice communication or requires granular analysis of call outcomes. Use it when you need to know *why* a call failed—not just that it happened. If your needs are limited to simple messaging (like sending Slack alerts) or basic data retrieval without context, this is overkill. Don't use it if you only need to read the phone numbers; `list_inbound` handles that. But if you need to connect those numbers to an active AI workflow for dialing or monitoring, then this MCP is essential.

Managing Telephony Campaigns with Bland AI

Today, running a campaign means exporting massive CSVs of call logs. You manually cross-reference these against CRM records to figure out if the sales representative successfully pitched the product or if the lead was simply uninterested. The

With Bland AI, your agent handles the entire workflow. Send agents via `send_call` and then use `get_call_details` to retrieve all necessary context in one go. You don't just get a transcript; you get structured data about what happened.

whole process is slow, error-prone data wrangling that takes hours.

Analyzing Call Quality with Bland AI

The current process for quality assurance requires logging into the phone system, finding the recording, downloading it (often in a proprietary format), and then having a human listen to check if the agent hit key compliance points. It's expensive and slow.

Now, you can pull raw audio files using `get_recording` directly via your AI client. You maintain full control over the media extraction process, ensuring perfect logging for later review.

10 Tools for Telephony Campaign Management

These tools let you manage the entire lifecycle of a phone call: sending them out, monitoring them live, analyzing the data afterward, and handling all incoming numbers.

#	TOOL	DESCRIPTION
01	`analyze_call`	Analyzes an active recording and returns whether a specific business goal was achieved during the call.
02	`end_call`	Forcefully disconnects any currently running AI phone call immediately.
03	`get_batch`	Retrieves a summary of how many bulk campaigns are running and their current status.
04	`get_call_details`	Pulls all specific data points, variables, and the full transcript text for one completed call.
05	`list_inbound`	Lists all purchased phone numbers that are available to route incoming customer calls to your AI agent.
06	`list_calls`	Retrieves the full, chronological history of every phone call made or received through the system.
07	`get_recording`	Pulls a direct link to the raw audio file (MP3/WAV) for deep quality assurance review.
08	`send_batch`	Dispatches multiple AI agents at once, scaling up large telecommunications campaigns.
09	`send_call`	Sends an automated conversational agent to a single target phone number for immediate outreach.
10	`create_web_call`	Spawns a temporary, internet-based signaling socket needed for testing browser audio connections.

See It in Action

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

U I need to run a campaign of 100 calls today targeting enterprise CTOs.

Campaign Status:

✅ Ready to Dispatch.

- **Method:** Bulk Batch Send (`send_batch`)
- **Targets:** 100 unique numbers (verified)
- **Agent Profile:** Sales Outreach v3.2 (Approved)
- **Estimated Completion:** Within the next hour.

Note: Keep monitoring the status using `get_batch` for real-time progress.

U What was the outcome of call ID c-1987? Did they want a demo?

Analysis Complete.

Goal Query: Was the customer interested in a demo?
Result: TRUE (Success).

> The prospect explicitly stated, 'I'm free next Tuesday at 2 PM.'

Recommendation: Schedule follow-up immediately.

U End this active call now. It sounds like the agent is stuck in a loop.

Command Executed.

The telephony session for the current call has been forcefully disconnected. A synchronous interrupt flag was sent to Bland AI, and the connection has been successfully shut down.

Frequently Asked Questions

01 **How do I run bulk campaigns using Bland AI with my AI agents?**

You use a batch dispatch tool. This lets your agent send out automated calls to dozens or hundreds of numbers at once, scaling up outreach quickly without manual intervention.

02 **Can I analyze recordings from past calls using Bland AI with my agents?**

Yes, you can. You pull the transcript and then ask your agent to run a targeted analysis on it, determining if specific goals—like scheduling or interest level—were met.

03 **What is the best way to monitor incoming calls with Bland AI?**

First, you use the list inbound tool to check your purchased numbers. Then, you connect these numbers so they can be routed directly to your automated agent for handling.

04 **If an active call goes wrong, how do I stop it using Bland AI?**

You issue a command to end the call. This sends an immediate interrupt signal to the phone system and shuts down the session instantly, letting you take over.

05 **Does Bland AI help me debug my code when testing web audio?**

Absolutely. You can use a WebRTC signaling socket creation tool. This lets developers test browser-based audio flows directly within their coding environment for accurate debugging.

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 `"bland-ai": { "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

Bland AI 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 Bland AI. 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	Bland AI MCP
Server ID	`019d755e-0c87-7330-a360-4ef4bb2e2982`
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/bland-ai.