MCP SERVER | NO CODE | CLOUD HOSTED

LiveKit MCP

Control real-time audio, video, and phone calls.

LiveKit MCP lets your AI agent fully manage real-time media sessions using natural language commands. You can programmatically create rooms, adjust participant permissions, mute tracks on demand, and start high-quality audio/video recordings directly from any compatible client.

A+ Quality Score 95.83/100

webrtc | real-time-audio | real-time-video | media-streaming | participant-management | session-recording

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

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.

LiveKit MCP

41 tools available
Cloud-hosted on Vinkius

Managing live video and audio streams used to require diving into complex dashboards or writing dedicated SDK calls for every small change. Now, your AI agent handles that complexity through this MCP. You can use natural language prompts to orchestrate real-time communications: spinning up a new meeting room, managing who joins, and even recording specific segments of the call. For instance, if you need to mute a noisy participant or record an entire web page in use, you just tell your agent what's happening. It handles the underlying WebRTC protocols, giving you deep control over session setup, media routing, and data signaling without touching any code. Connecting this LiveKit MCP through Vinkius means your AI client gets instant access to industry-standard tools for communications, making it easier than ever to build complex agentic workflows that handle real-time interactions.

Core Capabilities

01 — Control Room Lifecycle

Your agent can create new meeting rooms with specific rules or forcibly delete existing ones and all connected participants.

02 — Manage Participants

You can list who's currently in a room, view participant details, kick users out, or explicitly invite other agents to join the session.

03 — Control Media Streams

Your agent can mute or unmute specific audio and video tracks for any user in the room, or record an entire composite web layout.

04 — Handle SIP Calls

The MCP lets your agent define inbound and outbound telephone trunks, allowing it to manage virtual phone numbers and transfer live calls into a meeting room.

05 — Send Real-time Data Signals

You can send custom data packets (like Base64 encoded information) directly to specific participants or broadcast them across the whole room for signaling purposes.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your LiveKit Server URL and API Token/Secret credentials.

02 Your AI client connects, giving it access to all communication tools, allowing you to issue commands like 'Create a new meeting room for the marketing team.'

03 The agent executes the necessary calls, performing actions in real-time—like creating the room and notifying participants—all through natural language conversation.

The bottom line is, your AI client treats this entire media infrastructure like an extension of its own conversational logic.

Built For

This MCP is essential for telecom engineers and backend developers who build systems relying on live, multi-party communications. You're the person tired of manually jumping between a dashboard to check room status, then running separate code just to mute someone.

Telecom Engineer

They use this MCP to programmatically set up complex call flows, like defining SIP inbound trunks or transferring an active call into a virtual meeting space.

Backend Developer

They integrate it to give their application's AI agent control over live media—such as monitoring participant presence or initiating room recordings based on user conversation.

DevOps Engineer

They leverage it for session management, automating cleanup by listing active rooms and then deleting them to prevent resource leaks across multiple services.

What Changes When You Connect

01 Automate participant management: Instead of manually listing users to check who's present, use `list_participants` to get a roster instantly, or use `remove_participant` to kick out disruptive attendees.

02 Programmatic recording control: You don't have to stop and start recordings via a dashboard. Use `start_room_composite_egress` or `start_web_egress` commands to capture specific events automatically.

03 Deep communication routing: Handle phone calls seamlessly by using tools like `create_sip_inbound_trunk` and `transfer_sip_participant`, making your AI agent a full-fledged telephony system.

04 Fine-grained media control: Need to silence someone without interrupting the flow? Use `mute_published_track` to target just an audio or video feed for one user while everyone else stays connected.

05 Flexible data signaling: Send custom, structured information across the wire using the `send_data` tool. This lets your agent communicate non-media context—like a form ID—to all participants in the room.

Real-World Applications

Automated Interview Recording

A hiring manager needs to record every step of an interview. The agent uses `create_room` to start the session, then waits for the conversation to finish before triggering `start_room_composite_egress` to capture a full video archive.

Customer Support Call Routing

A customer calls into a generic number. The agent uses `create_sip_dispatch_rule` and `list_phone_numbers` to identify the caller, route them to the correct specialized room, and then manage the call using `transfer_sip_participant`.

Live Conference Broadcast

The event organizer needs a live feed of a presentation screen. The agent uses `start_web_egress` to capture the web page, and then uses `update_layout` if they need to switch the recording focus mid-stream.

Session Audit Logging

A compliance officer needs a record of every person who entered a sensitive meeting. The agent first calls `list_rooms`, then uses `get_participant` multiple times on each room to build a complete, auditable attendance log.

Patterns to Avoid

Treating it like simple messaging.

✗ AVOID

Trying to use this MCP just to send plain text messages or transfer files. It's overkill and doesn't support those basic data types.

✓ INSTEAD

If you only need text chat, use a standard message service MCP. This LiveKit MCP is strictly for real-time media (audio/video) sessions and calling infrastructure.

Assuming persistence.

✗ AVOID

Running `create_room` and assuming the room will stay active forever without cleanup. You'll quickly run out of resources or have stale data.

✓ INSTEAD

Always follow up session creation with a plan to manage its lifecycle, using tools like `delete_room` once the meeting is over to ensure clean resource management.

Ignoring media flow state.

✗ AVOID

Trying to send data (`send_data`) before you've confirmed that participants are actually connected. The packets will fail without an active session or room context.

✓ INSTEAD

First, call `list_participants` to validate the roster and ensure users are in a room. Then, execute your media actions like `mute_published_track`.

The Right Fit

Use this MCP if your workflow absolutely requires deep control over live audio and video streams or integrated telephony features. Specifically, you need to manage participant presence (`list_participants`), programmatically record a web page or room (`start_web_egress`, `start_room_composite_egress`), or route phone calls through the system using SIP tools. Don't use it if your primary

goal is simple asynchronous chat; for that, you need a messaging service MCP. Also, don't use it if you only need to store static meeting notes—use a database connector instead. This tool is about controlling what happens *right now*, in real-time, across multiple media streams.

Managing live conferences used to be a dashboard nightmare.

Today, if you want an AI agent to manage a conference call, you'd have to jump through hoops. You'd log into the platform dashboard to check who was in the room. To record something, you might manually toggle recording on and off. If someone started making noise, you'd have to find their user ID and hit another button just to mute them. It’s slow, it requires multiple clicks, and it breaks your natural conversation flow.

With this MCP connected via Vinkius, the process disappears. You tell your agent: 'Start a room for Acme Corp.' The agent handles creating the space and inviting everyone. If the meeting needs recording, you just say, 'Record everything from now on.' The agent uses `start_room_composite_egress` to handle the entire complex media workflow automatically.

LiveKit MCP: Full Control Over Real-Time Media

You no longer need separate scripts or manual API calls just to check who's present in a call. You can ask your agent, 'Who are the speakers?' and it uses `list_participants` to give you an immediate, accurate count, making resource checks effortless.

The difference is that this MCP treats real-time media control as just another conversational topic. Your AI client doesn't need a manual guide; it simply knows how to talk to the infrastructure.

LiveKit MCP: 30 Tools for Real-Time Media

Use these tools to manage every aspect of your communication infrastructure, from setting up SIP trunks and creating rooms to muting participants and recording sessions.

#	TOOL	DESCRIPTION
01	create_dispatch	It tells your agent to explicitly trigger a named user account to join a specific meeting room.
02	create_ingress	This sets up an entry point for media feeds, whether they come from RTMP, WHIP protocols, or a simple URL pull.
03	create_room	You can use this tool to generate a new meeting room with custom rules and settings.
04	create_sip_dispatch_rule	This maps incoming phone calls based on specific phone numbers or PINs directly into a designated LiveKit room.
05	create_sip_inbound_trunk	It defines the initial setup for how all incoming SIP voice calls are received and handled by your system.
06	create_sip_outbound_trunk	This sets up a necessary connection point that allows your agent to dial out using external phone lines.
07	create_sip_participant	The tool dials an actual SIP number and brings the resulting call directly into a LiveKit room for participation.
08	delete_dispatch	It removes a dispatch rule that was previously set up to route calls automatically.
09	delete_ingress	This tool cleans up and removes an existing media ingress point.
10	delete_room	It forcefully disconnects everyone in a room and permanently deletes the meeting space.
11	delete_sip_dispatch_rule	This removes a specific SIP rule that directs calls based on phone numbers or PINs.
12	delete_sip_trunk	It cleans up and deletes an entire configured SIP trunk setup.
13	get_participant	You can use this to retrieve detailed information about a single user who is in the room.
14	list_dispatch	It shows you all the current rules set up for automatically dispatching users into rooms.

#	TOOL	DESCRIPTION
15	list_egress	This lists any active recording or media export jobs that are currently running.
16	list_ingress	It shows you a list of all the media ingress points provisioned for your system.
17	list_participants	This lists every participant currently connected and active within a specified room.
18	list_phone_numbers	It retrieves a list of all phone numbers that are owned by your current project account.
19	list_rooms	This shows you an overview of every active or open meeting room in your LiveKit instance.
20	list_sip_inbound_trunk	It lists all the SIP trunks configured to handle incoming telephone calls.
21	list_sip_outbound_trunk	This shows you a list of all the SIP trunks set up for making outgoing phone calls.
22	mute_published_track	Your agent can mute or unmute a specific audio or video track belonging to any participant.
23	purchase_phone_number	It allows you to buy a new phone number and optionally link it right away to a call dispatch rule.
24	release_phone_numbers	This returns a purchased phone number back into the general inventory pool.
25	remove_participant	It kicks a specific participant out of a room, regardless of their status or permissions.
26	search_phone_numbers	You can search for available phone numbers using criteria like country and area code.
27	send_data	This sends custom data packets to one or more participants within the room.
28	start_participant_egress	The tool initiates a recording that captures only the audio and video feeds of a single participant.
29	start_room_composite_egress	This starts an advanced recording job capturing the entire room using a defined web layout.
30	start_track_composite_egress	It begins a composite recording that captures one specific audio track and one specific video track together.

#	TOOL	DESCRIPTION
31	start_track_egress	This exports a single media track without needing to re-encode the raw feed.
32	start_web_egress	It captures and records any web page content that is visible in the browser window.
33	stop_egress	This stops any recording or media export job that was previously started by your agent.
34	transfer_sip_participant	It redirects an active SIP call from one phone number to another number or URI.
35	update_ingress	This updates the settings for a media ingress point used by multiple sessions or participants.
36	update_layout	You can change the visual arrangement of a room while an active composite recording is running.
37	update_participant	This allows you to modify metadata or permissions for any participant in the system.
38	update_phone_number	The tool changes the SIP dispatch rule attached to a phone number, redirecting where calls go next.
39	update_room_metadata	It updates key-value pairs that store context data across an entire meeting room.
40	update_stream	You can add or remove output URLs (RTMP/SRT) from a stream while it is active.
41	update_subscriptions	This changes which specific media tracks a participant is allowed to receive or send.

See It in Action

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

U List all currently active rooms in my LiveKit instance.

I've retrieved the active rooms. You have 3 rooms running: 'Main-Lobby' (2 participants), 'Gaming-Zone' (5 participants), and 'Support-Desk' (0 participants).

U Create a new room called 'Strategy-Meeting' with a max of 10 participants.

The room 'Strategy-Meeting' has been successfully created with a limit of 10 participants and a default empty timeout.

U Mute the audio track for participant 'user_99' in the 'Main-Lobby' room.

I've sent the command to mute the track 'TR_XXXXXXXX' for 'user_99' in 'Main-Lobby'. The participant is now muted.

Frequently Asked Questions

01 **How do I use the LiveKit MCP to record an entire meeting room?**

You trigger recording using `start_room_composite_egress`. This tool records all participants and the web layout simultaneously, giving you a full composite video file.

02 **Can I make my AI agent handle phone calls with LiveKit MCP?**

Yes. You use tools like `create_sip_inbound_trunk` to set up receiving lines and then use `transfer_sip_participant` to move the live call into a managed room.

03 **How do I mute a user’s microphone using LiveKit MCP?**

You must use the `mute_published_track` tool, specifying which participant and which track (audio or video) needs to be muted. This is much more precise than simply kicking them out.

04 **What if I need to move a call from one room to another?**

You use the `transfer_sip_participant` tool. It handles the complexity of maintaining the active connection while rerouting the user's SIP stream.

05 **Is LiveKit MCP just for video, or can it handle data too?**

It manages both media and data. You use `send_data` to send custom Base64 encoded packets alongside the audio/video streams, making it useful for signaling.

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 `"livekit": { "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

LiveKit 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 LiveKit. 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	LiveKit MCP
Server ID	`019e38b9-6998-7159-ae8e-197b4a43b2e1`
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/livekit.