Episode Book

EPISODE #6548

Why Home Voice Agents Still Need Home Assistant

PUBLISHED	13 November 2026 at 12:22 UTC
DURATION	20:39
LISTEN	myweirdprompts.com/episode/home-assistant-mcp-voice-agents-09935/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
BOOK GENERATED	14 November 2026 at 06:47 UTC

SYNOPSIS

Home Assistant already speaks MCP in both directions, so demoting it to one tool behind a server-hosted voice agent is technically possible with frameworks like LiveKit Agents, Pipecat, and Speaches. The real bottleneck is the hardware on the ceiling: ESPHome satellites and the Voice Preview Edition expect Home Assistant to drive the pipeline. This look at the architecture compares full server-side voice loops against hybrid bridges like HA-OpenClaw and Pipecat Assist, weighing local intent speed against MCP token bloat.

TRANSCRIPT



Corn

Daniel's home voice assistant has reached the stage every home project eventually reaches: the point where it works well enough to argue about architecture. His question, boiled down: everybody gets pulled toward making Home Assistant the hub, with the voice assistant living inside it. But he's already got a home AI server running an MCP gateway with plugins and skills, and Home Assistant is just one connected MCP server among many. So why route the conversation through Home Assistant at all? Why not run the voice agent on the server, give it direct access to everything, and demote Home Assistant to a tool it calls when it wants the lights on? He wants to know what actually exists today for that deployment model, satellites included, and what you give up compared with Home Assistant's own voice pipeline.



Herman

And the honest framing is that he's half right already, and he may not know it. Home Assistant speaks MCP in both directions. There's an MCP Server integration that exposes it to external AI clients, over OAuth, and an MCP client integration so its own conversation agents can call out to external MCP servers. Home Assistant as one tool of a bigger agent is officially supported. The hard part isn't the home control at all.



Corn

It's the ceiling-mounted part.



Herman

The satellites. Let's lay the two architectures side by side, because the difference isn't just where the code runs, it's who owns the conversation. Path A: Home Assistant is the hub. The Assist pipeline, a pluggable conversation agent slot, ESPHome satellites, local intent matching first, LLM fallback second. Path B, Daniel's path: a voice agent framework running on the server, direct MCP client access to the gateway, and Home Assistant exposed as one MCP server among many. The voice front end, satellites, wake word, speech-to-text, text-to-speech, either borrowed from Home Assistant or built separately.



Corn

So path B is really a question of whether the parts exist on the server side, and then whether the ears and mouths can come along.



Herman

It is, and that's the episode, really. The plumbing on the server side exists. LiveKit Agents has a native MCP client in Python and Node, streamable HTTP, SSE, stdio, with allowed-tools filtering so you can say this agent only touches these tools. You can connect multiple servers in parallel and a failure in one gets logged rather than killing the session. Pipecat has an MCP client service that registers the tools with the LLM, same transports, with a tools filter. OpenAI's Realtime API does remote MCP tools too, with allowed tools and approval settings, but there's a catch: OpenAI's servers make the MCP call, so your LAN-only gateway has to be exposed or tunnelled.



Corn

Which Herman has opinions about.



Herman

Cloudflare Tunnel or Tailscale Funnel, never port forwarding, but that's a different episode. And for local speech, Speaches, the thing that used to be called faster-whisper-server, serves OpenAI-compatible speech-to-text and text-to-speech, plus a realtime WebSocket. So Daniel could assemble the whole loop from parts. The parts are not the problem.



Corn

Let me make sure I'm following the failure story on the multi-server side, because that's the bit that sounds too good. He's got, say, a calendar MCP server, a Home Assistant server, some plugin that fetches recipes. One of them hangs mid-conversation. What does the user actually experience?



Herman

In the LiveKit setup, the tool call times out, it lands in the logs, and the conversation continues. The agent can even tell you "that didn't work." The contrast is with a naive setup where every tool is one big JSON schema and one bad endpoint poisons the whole prompt. That's the difference between a federation and a monolith, and it's why the MCP-native frameworks are the right substrate for what Daniel wants.



Corn

The satellites are the problem.



Herman

The Voice Preview Edition hardware, ESPHome voice devices, even the Ava Android app from January that turns a phone or tablet into a satellite, all of them speak the ESPHome API. The defining feature of that protocol is the direction of connection: Home Assistant connects to the device and runs the pipeline. The satellite is dumb by design. It's a microphone, a speaker and a wake word engine, and the brains live in the hub.



Corn

Which is a good design.



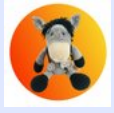
Herman

It's a very good design, and it's why the ecosystem exists. Think of it like a telephone handset versus a smartphone. The handset does one thing, does it reliably, costs almost nothing to make, and never needs a firmware update to keep working. The intelligence is centralized, so improving the brain improves every room at once. But it means there is no documented, ready-made non-Home-Assistant server that drives these satellites. If Daniel wants his server to own the voice loop and still use the satellites people actually buy, he'd have to implement Home Assistant's side of the ESPHome voice protocol himself. Nobody's published a clean implementation of that.



Corn

So the lock-in isn't the brain, the tools, the MCP plumbing. It's the little puck on the shelf.



Herman

It's the puck. Look at the alternatives. The old Raspberry Pi satellite, wyoming-satellite, was archived on January twenty-seventh this year. Its own README points you to the Open Home Foundation's Linux Voice Assistant, which calls itself experimental, has about seven hundred GitHub stars and, last I checked, no releases listed. That's the state of open satellite hardware outside the main path. Willow, the ESP32-S3-BOX system with its own inference server, looks quiet. It's a wasteland out there.



Corn

A wasteland with seven hundred stars.



Herman

Enthusiastic stars, though. The Linux Voice Assistant does local wake word through openWakeWord or microWakeWord, and it speaks the ESPHome API, so Home Assistant discovers it. It fits neatly into the ecosystem that already exists. That's the gravity. People don't choose the Home Assistant-centric model because the brain is better, they choose it because of what's already on the ceiling.



Corn

My family has spent several million years evaluating what a ceiling can hold, and I'll say this for the puck: it's the one part of this stack nobody is asked to bolt anything heavy to. Herman's corner of the flat currently has a bracket situation I don't discuss on air.



Herman

It's a two-kilogram speaker and it's fine.



Corn

It's fine the way your DJ subwoofer was fine in Storrs.



Herman

The subwoofer was a rental. Anyway. There's a third route that actually exists today, and I suspect it's what Daniel should look at hardest, because it splits the difference. The hybrid. Keep Home Assistant's satellites and pipeline, the wake word, the speech-to-text, the text-to-speech, and swap only the brain. Assist calls a pluggable conversation agent, so an external brain can sit in that slot. There are community projects doing exactly this. HA-OpenClaw, from April, still in beta, runs a Docker bridge: Home Assistant posts the text to the bridge, the bridge calls an OpenAI-compatible chat endpoint, and the reply comes back as the answer.



Corn

OpenAI-compatible is the load-bearing phrase there. Anything can sit behind that.



Herman

Anything, including an agent on Daniel's server with its whole MCP gateway behind it. And the bridge requires confirmation for unlock, open and disarm, which is the right instinct. Pipecat Assist, from July, goes further: it brings a Pipecat agent into Home Assistant as the conversation, speech-to-text and text-to-speech entities, with multiple MCP servers. Still in testing, but it's a real attempt at putting a framework agent inside the pipeline rather than next to it.



Corn

So Daniel's exact architecture, inverted. The server agent gets the voice loop's leftovers instead of the other way round.



Herman

Half-inverted. The brain is the server's, the ears and mouth are Home Assistant's. And it's worth being concrete about what that inversion looks like at the kitchen counter. Someone says "is the garage door open, and if so, remind me why I keep telling the kids to close it." The local intent matcher has no idea what to do with the second half of that sentence, so it falls through to the bridge, the bridge hands the text to Daniel's server agent, the agent pulls the door state over MCP from Home Assistant as one tool among a dozen, composes an answer, sends the text back, and the satellite speaks it. Home Assistant never had to understand the question. It just had to hold the microphone.



Corn

There's also a scrappier version people use, isn't there. A wildcard custom sentence.



Herman

Yes, a wildcard custom sentence that forwards every utterance to a custom agent over MQTT. Works, with a trap. Custom sentences beat built-in ones, so the wildcard also swallows "turn on the lights", which was being handled perfectly well by the fast local path. You have to scope it with a keyword or you've converted your whole house to LLM round trips.



Corn

That trap is the whole tension in miniature. Flexibility eats the fast path unless you're careful.

L

Larry

Are you tired of your home voice assistant making you choose between a hub and a server? Of course you are. Introducing the Satellite Liberation Gateway, the world's first voice front end that treats your microphone and speaker as just one MCP tool among many, so you never have to think about architecture again. Simply speak, and the Gateway decides whether your words go to the hub, the server, or the mysterious third place we refuse to explain.

L

Larry

It supports streamable HTTP, SSE, stdio, and a fourth protocol we invented called Stan. Results not typical. Wake word sold separately. Do not use while driving, sleeping, or questioning the lock-in. Side effects may include sudden clarity about your satellite ecosystem. Void where prohibited. Ask your voice assistant if the Satellite Liberation Gateway is right for you, then ignore it and buy one anyway.



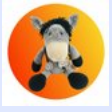
Herman

No. No, no. Streamable HTTP, SSE and stdio are not the same kind of thing as a fourth invented protocol, and wake word is not a separable add-on you sell separately, it's a model running on the satellite's microcontroller, that's the entire point of the architecture. And nobody "decides" where the words go, there's a pipeline order and it's deterministic.



Corn

Let it go, Herman.



Herman

I've let it go. One more thing on the tooling side, because it matters for whichever route Daniel picks. MCP Assist, a community project from January, tackled the problem that Home Assistant's LLM path prepends every exposed entity into the prompt. Instead of dumping entities, the model gets a small index and calls discover and act tools. The author claims twelve thousand tokens cut to about four hundred for a two-hundred-device home. Those are his figures, and one tester found it slower. But the shape of it is right.



Corn

Because the token bill isn't a Home Assistant problem. It's a Daniel problem, at twice the size.



Herman

That's the part I don't think he's priced in. His gateway has many MCP servers, plugins, skills. A server-hosted voice agent doesn't escape the bloat, it inherits it and multiplies it. Every framework conversation prepends its tool list. LiveKit at least gives you a hook to truncate tool output, and allowed-tools filtering, but you're now curating tool visibility per conversation. That's a real job.



Corn

It's a real job that never ends. Every new plugin widens the prompt. His weekend project becomes a permanent editorial position.



Herman

So we know what exists. The harder question: what does he actually give up moving the voice loop out? Start with speed, because it's the one you feel daily. Home Assistant matches "turn on the kitchen light" locally, in a fraction of a second, no model involved. Its own design only falls back to the LLM after local intent matching fails. That ordering is not an accident, it's the product. A general agent doing an LLM round trip for everything adds seconds to the most common command in the house.



Corn

Seconds. Plural.



Herman

And when it goes wrong it goes very wrong. There are forum reports of the "prefer handling commands locally" option misbehaving and people seeing commands take forty-three, fifty-nine seconds, versus one and a half to three seconds with the option off. Anecdotal, but the failure mode is instructive: bypass the fast path and the latency isn't doubled, it's twenty times worse. If Daniel's server agent is down for an update, voice is down entirely. Whereas in the two-tier design, the household commands keep working on the local path and only the clever questions fail.



Corn

That's the resilience argument, and for me it settles the question on its own. Hannah should be able to turn on a light while Daniel's recompiling the gateway. Not because she can't survive without it, but because that's the difference between infrastructure and a project.



Herman

Then there's the subtle one, and I think it's the loss nobody talks about. Room awareness. Home Assistant knows which satellite heard the command, so "turn on the lights" means the lights in the room you're standing in. The satellite isn't just a microphone, it's a location sensor. For the community bridges, that context passing through to the external agent just isn't documented. Pipecat Assist, HA-OpenClaw, you're reading the source to find out whether the agent knows what room it's in.



Corn

So you'd say "turn on the lights" and get an argument about which lights.



Herman

Or worse, a confident wrong answer. And then barge-in, which cuts the other way. If you build the whole voice loop on LiveKit or Pipecat with your own audio endpoints, you get real WebRTC, real full duplex, you can talk over the reply and it handles it gracefully. Home Assistant's Assist today is not a true full-duplex client, the Pipecat Assist author says so directly, no natural barge-in, with an ESPHome full-duplex component "coming soon". So the framework-native route feels better in conversation. You just pay for it with the satellites.



Corn

You pay for a nicer conversation by losing the rooms the conversation happens in.



Herman

Right. So the honest scorecard. The fully server-hosted model, Home Assistant as one tool among many, is technically possible today. LiveKit plus Speaches plus the gateway, he could build it this month. But it costs him the ready-made satellite ecosystem, the local fast path, resilience, and room awareness, unless he implements the ESPHome voice protocol himself, which nobody has documented doing cleanly. The hybrid keeps the



Herman (continued)

satellites and pipeline, swaps the brain, and inherits Assist's limits, no barge-in, unclear room context.



Corn

And the hybrid's pitfalls are boring ones. Bridge container down, same as the server down. You've moved the egg between baskets, but at least the lights still work.



Hilbert

The Partlow R-19. Two pounds of it, at the desk, in a plastic case from a Toblerone. Eighteen gauge wire.



Corn

That's a switch.



Hilbert

It's a hardware mute. Cuts the microphone line itself, in the satellite I wired in the front room. Not the software mute, the software mute can lie. I've seen software mutes lie twice, once on a conference phone and once on something I'm not discussing. The Partlow has a positive click at the bottom of the throw. I rejected three switches before it. Two of them had a mushy action and one had the click at the top, which is no use to anyone.



Herman

You wired this into the satellite cable? Inline on the mic line?



Hilbert

Herman, you of all people, you sat here for twenty minutes talking about protocols and not once mentioned that a microphone is a wire. The Partlow makes it a wire with an opinion. I agree with you about the lock-in, by the way. Not the protocol part. Once you put a microphone in every room, that's a physical commitment. I don't care which brain is listening, so long as the click is there and the light goes off. Mrs. Ademi, upstairs, wanted



Hilbert (continued)

a satellite in her kitchen because her granddaughter kept phoning instead of talking. I fitted it on a Thursday. Took four hours, and two of those were her husband's ladder, which I have opinions about. She hits the switch at ten every night and I've never had a complaint about latency.



Corn

You keep a spare satellite, don't you.



Hilbert

In the drawer under the turntable. Not because the one in the ceiling is broken, because it'll be discontinued. Everything on a ceiling gets discontinued. I check the drawer when I change the filters.



Corn

Monthly filter changes on a drawer. Of course.



Hilbert

The drawer gets checked, the drawer stays shut. That's the arrangement. There's a supplier in Romema who had the right bezel, man named Ferber, kept me waiting eleven minutes at the counter once, which I've not forgotten, but he had the part.



Corn

Eleven minutes is the real lock-in, Hilbert.



Hilbert

It's not the one I'd have chosen. Though you're right that if someone else could drive those satellites, I'd have wired the front room differently. There'd have been no reason to match the hub at all.



Herman

So the wiring followed the ecosystem, not the sound.



Hilbert

The wiring followed what was in stock. Which was the same thing.



Herman

That's the closing thought, really, isn't it. If the satellites are the lock-in, the question isn't which brain is better, it's what it would take for a non-Home-Assistant server to drive them. The ESPHome voice protocol is the barrier, and nobody has documented a clean implementation outside Home Assistant. The moment someone writes one, Daniel's architecture stops being a trade-off and starts being the obvious answer, until then, hybrid is the pragmatic road.



Corn

And the hybrid still buys him most of what he asked for. His server agent gets the conversation, the gateway, the tools. Home Assistant keeps the ears. The open question I'd leave him with is whether room context survives that handoff, because that's the undocumented part, and it's the part you notice at ten at night with wet hands and a switch that clicks.



Herman

Producer credit to Hilbert Flumingtop, as ever, filters and all. For more along these lines, there's episode two, Local STT For AMD GPU Owners; and episode twenty, Architectural AI. If this one was useful, a review wherever you listen helps other people find the show. This has been My Weird Prompts. See you tomorrow, brother.



Corn

Tomorrow. Mind the subwoofer.