Episode Book

EPISODE #6705

Why Voice Agents Use Frameworks Like Pipecat

PUBLISHED	20 November 2026 at 13:33 UTC
DURATION	30:38
LISTEN	myweirdprompts.com/episode/voice-agent-orchestration-frameworks-92325/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	omnivoice
PIPELINE	V5.36
SOURCES	25 cited by the research (listed at the end)
BOOK GENERATED	20 November 2026 at 21:20 UTC

SYNOPSIS

A 200-millisecond pause can make a caller think the line is dead, yet voice agent builders still add orchestration layers like Pipecat on top of realtime APIs like Gemini Live. Breaking down the four-player market—Pipecat, LiveKit Agents, Vapi, and Retell—reveals what frameworks actually buy you: transport independence over WebRTC, modifying transcripts before an LLM sees them, non-conversation pipelines, and multi-vendor hedging. Plus, the latency trade-off against OpenAI Realtime and what Deepgram's acquisition of Daily means for open-source neutrality.

TRANSCRIPT



Corn

Two hundred milliseconds. That's roughly the pause a caller tolerates before they decide the line is dead. And an entire layer of infrastructure now exists to make sure that pause never happens.



Herman

And it happened quietly. Voice agents stopped being a model demo somewhere in the last year and a half and became a market with valuations and acquisition drama and vendor politics. Which is presumably why Daniel went where he went.



Corn

His question is about the glue. He wants the frameworks that string speech-to-text, an LLM and text-to-speech, or a single speech-to-speech model, into a live voice agent, rather than just calling one vendor's SDK. Pipecat is the one he cares about most, and the sharp end of his question is: why would anyone pick Pipecat over something like Google's Gemini Live API, now that the model vendors ship their own realtime voice stacks? What does the framework actually buy you?



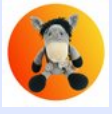
Herman

And the second half of it, which I think is the more interesting half honestly, is who really runs these things. Hobbyists, startups, call-centre vendors, actual big enterprises. And what an enterprise reaches for today.



Corn

So let's start by naming the players, because there are fewer than you'd think.



Herman

Four, really. Two open-source frameworks, Pipecat and LiveKit Agents, and two managed platforms, Vapi and Retell. There's a long tail, TEN Framework, Dograh, Vision Agents, Vocode, but LiveKit's own comparison post says none of those ends up on a production shortlist. Which is a competitor's framing, so take it with the appropriate salt, but the shape of the market backs it up.



Corn

Four names. For a layer everyone calls infrastructure.



Herman

It's worth sitting on that number for a second, because it tells you something about where we are in the hype cycle. Two years ago the answer to "what frameworks exist for voice agents" was a shrug and a GitHub link to someone's weekend project. Now there's a comparison-post economy. Every one of these four has written or been featured in a "which voice framework should you pick" post, and the posts mostly agree with each other, which either means the market has consolidated or means everyone's reading the same posts.



Corn

Or both. Consolidated markets and echo chambers look identical from the outside.



Herman

The layer is real, though. The way I'd cut the stack is four layers deep. Model weights at the bottom, then the realtime APIs, Gemini Live, OpenAI Realtime, the things that speak audio directly. Then orchestration. Then your application code on top. Kwindla Hultman Kramer, who created Pipecat, has a line I'd tattoo on the episode: the Gemini Live API occupies the API layer, Pipecat occupies the orchestration layer. Not rivals, different floors of the building.



Corn

And the cleanest one-sentence map of the four, I think, is the one AssemblyAI put together. Retell hands you a finished phone agent builder, Vapi hands you a managed pipeline you configure by API, LiveKit hands you a room your agent joins, and Pipecat hands you the parts.



Herman

That's the whole market in one sentence. The finished thing, the configured thing, the room, and the parts.



Corn

There's a twist worth flagging before we go further, because it hangs over everything. Pipecat's whole pitch is vendor neutrality, you pick your speech-to-text, your text-to-speech, your transport. And on October twenty-ninth, Deepgram, a speech-to-text company, acquired Daily, Pipecat's maker.



Herman

Terms undisclosed. Pipecat stays open source under the BSD two-clause licence, an independent Pipecat Foundation is, quote, being established, and Kramer joins Deepgram as Head of Open Source. I looked for a foundation site or a governance page and there isn't one yet.



Corn

The vendor-neutral framework now has an owner with a vendor.



Herman

Which doesn't kill anything, but it changes how you read every claim the framework makes about neutrality. Hold that thought.



Corn

I'll hold it slowly.



Herman

The cleanest way to see why anyone picks one of these is to look at what the two open-source frameworks actually are, because they're not the same bet. Pipecat is a pipeline you assemble. Voice activity detection, streaming speech-to-text, the LLM call, text-to-speech, each one is an explicit stage in code you can inspect, reorder, fork, insert logic between. LiveKit Agents is a higher abstraction over the agent itself. Your agent joins a WebRTC room as a participant and reacts to events. You can swap stages in and out, but you can't add new ones, reorder them, or branch them.



Corn

So one sells you parts and the other sells you a room with staff.



Herman

And what you get for giving up that flexibility is a lot. Dispatch, scaling, multi-party sessions, the transport itself, all handled underneath. If you're building a phone agent that needs to handle a thousand simultaneous calls, that's not nothing.



Corn

Can you give me a concrete picture of where that difference bites? Because "pipeline versus room" is clean but abstract.



Herman

Sure. Say you want the agent to check a CRM record before it answers a question about an order status. In Pipecat, that's a stage you insert between the LLM call and the text-to-speech call, sitting right there in your code with your logic in it. In the room model, you don't insert a stage, you attach to the events the room gives you, and you work within what those events expose. Most of the time that's fine. The times it isn't fine are exactly the times someone picked the wrong abstraction for the job, and by then it's expensive to find out.



Corn

The managed pair are different products again, right? Vapi versus Retell isn't a distinction without a difference.



Herman

Not at all. Vapi is API-first. You write a system prompt, pick your providers, register tools through the API, and Vapi runs the whole loop. Five cents a minute for hosting, plus the providers at cost, and the base package lists four concurrent calls.



Corn

Four. The concurrency of a small pizzeria.



Herman

For testing, yes, you scale up from there. Retell is the managed low-code builder. Handles the phone numbers, SIP, transfers, the whole telephony side, and lists between seven and thirty-one cents a minute. Retell is the one where a non-engineer can get something answering a phone. Vapi is the one where an engineer gets the same thing faster.



Corn

And that telephony side is the invisible half of the whole business. Everyone obsesses over the model, and then discovers that buying phone numbers, dealing with SIP trunks, and not getting your calls flagged as spam is its own profession.



Herman

It's the classic pattern. The interesting part of the stack gets the blog posts, and the boring part decides whether the demo survives contact with a real phone network.



Corn

Let's take Daniel's central question head on, because it's the good one. Why would anyone choose Pipecat over just using Gemini Live? Google ships a realtime voice stack. It handles the audio, the turn taking, all of it. What's the framework for?



Herman

Four concrete things, and I want to go through them properly because this is the heart of the question. First, transport independence. Pipecat runs over Daily's WebRTC, Twilio Media Streams, a raw WebSocket, even LiveKit's transport, and your pipeline code doesn't change. With Gemini Live you're on a bidirectional WebSocket, period, which means you own audio chunking, playback buffering and interruption handling yourself.



Corn

That's the forty-line barge-in problem.



Herman

Roughly forty lines of code just to handle the case where the caller starts talking over the agent. Pipecat handles turn detection and barge-in out of the box. And Pipecat's own docs argue WebRTC is the better choice for client-to-server audio anyway, because TCP over WebSocket brings head-of-line blocking, codec coupling, and no jitter buffering. One dropped packet and everything behind it waits.



Corn

And in practice, what does that look like when it goes wrong? Because "head-of-line blocking" is the kind of phrase that slides past people.



Herman

The caller is mid-sentence, one packet drops somewhere between their phone and your server, and now every chunk of audio behind it is queued up waiting for the retransmit. The agent's picture of the conversation is suddenly half a second stale, it's deciding turn boundaries on audio from the past, and the caller hears the agent reply to something they said a moment ago. It doesn't crash. It just feels wrong, in the way that makes people say "these things are creepy" and hang up.



Corn

Second thing.



Herman

Modifying the transcript before the LLM sees it. This is the one I think is undersold. Per AssemblyAI's comparison, Pipecat is the only one of the four where you can modify the transcript before the LLM call, get full access to advanced speech-to-text features like keyterms and endpointing, and insert arbitrary code between components. Vapi, no. Retell, no. LiveKit, limited, through event handlers.



Corn

And why does that matter to someone who isn't building this?



Herman

Because of what a misheard account number does. The transcription gets it wrong, the LLM answers the wrong question confidently, the text-to-speech reads that confident wrong answer aloud. The errors compound down the pipeline. If you can't touch the transcript between hearing and reasoning, you can't put a guard on the one point where a small mistake becomes a spoken one.



Corn

A wrong number spoken politely is still a wrong number.



Herman

And the guard can be dumb and still work. A regex that catches sixteen digits where an account number should have twelve. A lookup against the customer database before the LLM ever sees the text. None of that requires intelligence, it just requires being allowed to stand in the middle of the pipe, which is precisely what three of the four options don't permit.



Corn

Third, and this is the one I find elegant, non-conversation pipelines. Because a Pipecat pipeline is a list you write rather than a session you configure, nothing in it assumes a user speaks and then an agent replies. Continuous translation is an ordinary pipeline. A listen-only agent is an ordinary pipeline. A stage that reacts to audio without generating any speech at all, ordinary pipeline. In an agent-shaped framework you'd be fighting the session model to build any of those.



Corn

Continuous translation. I'd use that.



Herman

You would use it once, at two in the morning, to argue with a French website.



Corn

The listen-only case is the one I want more people to take seriously, though. An agent that joins a call, says nothing, transcribes everything, and flags moments for a human. That's a compliance product, a training product, a quality-assurance product, and it's a pipeline with the speech stages deleted.



Herman

And the moment your framework assumes "agent speaks, then user speaks," that whole category becomes an argument with your tools instead of a Tuesday afternoon.



Corn

Fourth thing.



Herman

Multi-vendor hedging. You can mix a fast voice model with a slower backend model running your tools while the voice keeps talking, that's a pattern Pipecat ships called LLMWithBackend. Or you swap your speech-to-text provider per stage. A single vendor's realtime stack locks the whole loop to that vendor's model, that voice, and that price list.



Corn

Now give me the honest counter-case, because there is one.



Herman

There is, and it's measured. A third-party latency comparison, a developer named Vadim Alakhverdov, prices as of June, measured median latencies around three hundred and eighty milliseconds for Gemini Live over WebSocket, about four hundred and ten for OpenAI Realtime over WebRTC, and about four hundred and forty for Pipecat. So the framework costs you tens of milliseconds and a layer of dependency, and Gemini Live was



Herman (continued)

the cheapest of the three.



Corn

Sixty milliseconds. Does a human notice?



Herman

A human notices dead air, and the question is where those milliseconds actually come from. The counter-argument to the counter-case is that hand-rolled barge-in on a raw WebSocket is where the real latency lives, and the bugs. Your forty lines handle the happy path, and then a caller coughs.



Corn

The one thing the demo never does is cough.



Herman

And there's one more piece that dissolves the premise of the either/or question entirely. Pipecat ships a first-class Gemini Live service and dedicated docs for building with Gemini Live. The framework treats the vendor's realtime stack as a component, not a rival. Some teams literally run Pipecat pipelines over LiveKit transport. Pipecat and LiveKit aren't competitors in practice, they're occasionally the same stack.



Corn

So the misconception that it's Pipecat or Gemini Live, pick a religion, is wrong at the architecture level.



Herman

Wrong at the level below the argument. The real split in this market isn't open versus closed. It's who owns the audio transport.



Herman

Which is the technical case. The obvious next question is who's actually running this stuff, and who's paying for it.



Corn

Before we get into the money, can we agree on one caveat up front? Every adoption number we're about to say comes from a vendor's own blog, a customers page, or a competitor's comparison post. Nobody has independently measured production minutes by framework.



Herman

Nowhere. And I looked for it wanting to find it. So LiveKit's million downloads a month and Deepgram's world's most popular are marketing claims, not measurements. We use them anyway, with that standing.



Corn

Named and shamed, forward we go.



Herman

LiveKit is the enterprise infrastructure story. Hundred million dollar Series C in January, a billion dollar valuation, led by Index Ventures with Salesforce Ventures and others in. And the customer list is the part that matters. OpenAI built ChatGPT's Advanced Voice Mode on LiveKit Cloud. Salesforce's Agentforce voice agents run customer support for big brands. Tesla uses voice AI for sales, support, insurance and roadside assistance.



Corn

Tesla has a voice agent for roadside assistance. Somewhere right now a man is being talked out of a ditch by a language model.



Herman

Their homepage claims billions of calls in production for OpenAI, Nvidia, Oracle, Spotify, others. But here's the bit the Series C post itself admits. LiveKit's own read is that large enterprises are evaluating and building, and many of these use cases are still in the proof-of-concept stage. The most confident company in the market put the quiet part in its own announcement.



Corn

That's a rare act of corporate honesty.



Herman

And it's worth being precise about what that honesty implies. When a company with OpenAI and Tesla on its homepage says many use cases are still proof-of-concept, the read is that the logos represent experiments with serious budgets, not necessarily serious call volumes. Which is still a real signal, enterprises don't run cheap experiments forever, but it's a different signal than "billions of calls" suggests on its own.



Corn

The logo wall is a photo of the future, not the present.



Herman

Pipecat skews differently, toward developer teams and platform builders. Pipecat Cloud ran a nine-month beta with over a thousand teams before general availability in January. And the README's example set tells you who it's for. Phone agents across Twilio, Telnyx, Plivo, Exotel, Vonage, Genesys. IVR navigation with DTMF tones. Restaurant reservations configured in YAML. That reads like contact-centre and business-agent work.



Corn

Not hobbyists.



Herman

The hobbyists are real but they're the on-ramp, not the story. Pipecat ships an ESP32 client SDK, there's a whole microcontroller angle, AI companions, interactive storytelling. You can have a talking gadget on your desk by the weekend. Nobody's routing a call centre through it by the weekend.



Corn

Which is, to be fair to the hobbyists, how a lot of open-source infrastructure stories start. The weekend gadget people learn the tool, some of them get jobs, and five years later the tool is quietly inside a bank.



Herman

The on-ramp is a talent pipeline as much as anything. Every hobbyist who ships a talking desk toy is a future engineer who already knows your API.



Corn

The managed pair, then. Vapi.



Herman

Vapi is the mid-market and enterprise managed pick. Fifty million Series B in May at roughly half a billion valuation, seventy-two million total raised, over a billion calls handled, enterprise revenue up ten times year over year. And the case study that made the rounds: Amazon Ring evaluated more than forty vendors and now routes one hundred percent of inbound calls through Vapi.



Corn

Forty vendors. Someone at Ring lived through a vendor evaluation the length of a pregnancy.



Herman

And picked the one that charges five cents a minute for hosting plus providers at cost. Retell is the call-centre workhorse. Hundred million dollar annualized run rate announced just this week, on four point six million raised.



Corn

A hundred million on four point six. That's not a company, that's a printing press.



Herman

It's the capital-efficiency outlier in the whole set. Single seed round, four thousand plus customers, over a hundred million calls a month across healthcare, financial services, insurance, logistics, retail, and ranked the fourth fastest-growing software vendor of last year in a Brex benchmark built on spend data from thirty-five thousand companies. Somebody's receptionist is a Retell agent, statistically several somebodies.



Corn

And the sectors on that list are the telling part. Healthcare, insurance, financial services. These are the industries with the most compliance rules about who says what on a recorded line, and they're the ones handing the phone to a managed platform. Which cuts against the naive read that regulated industries must build everything themselves.



Herman

It cuts both ways, honestly. Some of them are buying Retell precisely because the compliance posture is the vendor's problem, and some are building on LiveKit for the same reason, they want the audit trail in their own logs. The regulation doesn't pick the layer for you. It just makes the choice expensive to get wrong.



Corn

So, what's the enterprise go-to? Because Daniel asked that directly and there's a temptation to just name a winner.



Herman

I'd name the fault line instead. LiveKit argues the enterprise answer is LiveKit, that businesses scaling voice agents in production start with or migrate to open source, and it generally comes down to two frameworks. AssemblyAI argues it depends on the layer: managed, Vapi or Retell, when you want a working phone number fast, open source when you need custom tools, latency control, millions of minutes and compliance. There's a rough threshold floating around too, from a smaller consultancy, that below about ten thousand minutes a month you pick Vapi to validate, and above it self-hosting usually pays for itself within a quarter.



Corn

The pattern I hear under all of that is simpler than any of them state it. Enterprises with engineering depth go open source. Enterprises that want a phone agent next quarter go Vapi or Retell.



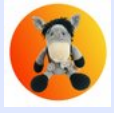
Herman

And the tell is who they hire, not what they buy. A company that just posted a platform engineering role for a voice team has already made its framework decision, whatever the procurement document says. Anyway, one more piece before we leave the adoption question, because it decides the future of the whole layer. Retell's own engineers say roughly ninety-nine percent of their agents still run a cascading pipeline, speech-to-text, LLM, text-to-speech, with fallbacks at every layer, despite speech-to-speech models being fully available.



Corn

Ninety-nine. That's the number that should scare the speech-to-speech evangelists.



Herman

It's the strongest signal there is that the pipeline abstraction, which is Pipecat's core bet, isn't going anywhere. The cascade won because when the speech-to-text layer fails you fall back to another provider and keep the call. One model doing everything is one model failing at everything.



Corn

Redundancy is the product nobody demos.



Herman

Redundancy is exactly what the framework sells, even though no framework's landing page says so.



Hilbert

Medieval Pottage Facts. Residue analysis of fats soaked into the cooking pots of a medieval village in Northamptonshire showed that the peasants' central dish was stews and pottages of beef, mutton, cabbage and leeks. That was the good news.



Corn

I'm not asking what the bad news was.



Herman

Don't, he'll tell you. So, the acquisition, and the strategic question underneath it, because I think this is where Daniel's curiosity is really pointed.



Corn

The Deepgram thing.



Herman

Deepgram's framing is that the deal creates the first complete, production-ready, open stack for conversational voice AI. Read that sentence slowly. A speech-to-text vendor now owns the orchestration layer that lets you pick any speech-to-text vendor. The BSD licence is real, the foundation is promised, Kramer running open source there is reassuring. But the neutrality pitch is now being made by an interested party.



Corn

"Pick any vendor you like, we recommend the one signing the paycheques."



Herman

And here's the deeper strategic problem, which Kramer himself articulated better than any critic could. His line is that as a problem becomes common, its solution tends to move down the stack. Turn detection started as application code he wrote himself. A few months later it had moved into Pipecat. By the time he gave that talk, it was available inside the live API surface itself. The vendor's API absorbed the framework's feature.



Corn

If that keeps happening, what's left for Pipecat to own in two years?



Herman

Kramer's answer, and he's careful to call it an arbitrary personal estimate, is that none of the listed realtime voice challenges is more than about fifty percent solved. Turn taking in noisy rooms, handling overlapping speech, real understanding of hesitations. The framework's job, in his telling, is to keep absorbing the next unsolved problem, because new use cases keep creating new application-layer responsibilities.



Corn

That's a bet, not a guarantee.



Herman

It's a bet with good historical support, but the pace of the model vendors absorbing features has never been faster. The two directions, features moving down the stack and use cases multiplying at the top, are in a race, and the orchestration layer lives or dies on which one wins.



Corn

The race, though, hasn't stopped anyone from being interrupted mid-sentence by a machine that then apologizes.



Hilbert

Forty lines is generous, Herman.



Herman

Sorry, which forty?



Hilbert

The barge-in code on Gemini Live. You said roughly forty lines. I counted, on the Python quickstart, because I was going to say something about it. It's nearer sixty before it behaves when somebody breathes in. You can leave the forty in, it changes nothing, but it's sixty.



Herman

He's right, actually. The forty is the happy-path count.



Hilbert

Of course I'm right. The interruption thing you two keep circling, that's the product, not the orchestration. You spent ten minutes on transcript modification and transport abstraction, and the whole business lives or dies in the half-second after a caller stops talking. I had a hold system put in at a place I worked, late seventies, a man called Groll installed the PABX, he came from a firm on the industrial estate and he was there every week for a month. The music cut in over people. Not at the end of the silence, over the words. Groll said it was a timing relay and it was set at the factory. He adjusted it with a screwdriver while the receptionist watched. Took him one visit, he'd been telling the firm for a decade.



Corn

The screwdriver was the whole product.



Hilbert

The point is the pause. Every one of these agents is tuned for interruption, none of them is tuned for hesitation. A caller who stops mid-sentence to think gets talked over. I keep a note of it, in the ledger book, the A5 one, the date and what I was about to say. Since the first one talked over me in spring of twenty twenty-three. Two hundred and thirty-one entries.



Corn

Two hundred and thirty-one things you were about to say.



Hilbert

Most of them were "yes" or "stop", I'm not claiming they were literature. Corn, you leaned into your microphone on "Redundancy is the product", I've pulled the level down on it in post, don't lean again, it's a condenser not a handheld. Two hundred and thirty-one. Groll would have set that relay to two seconds flat and gone home.



Herman

Two seconds is longer than the whole barge-in window most of these agents get.



Hilbert

Then it's set wrong. Groll charged eleven pounds for the visit and the firm thought that was steep.



Herman

I want to pull on what he said about hesitation, because there's a real technical point buried in it. Turn detection models are trained to detect the end of an utterance, and a hesitation is an utterance that hasn't ended. The framework's endpointing settings, how many milliseconds of silence before the agent decides you're finished, that's exactly



Herman (continued)

Groll's timing relay with a config file instead of a screwdriver.



Corn

And the factory setting is the wrong one, same as it was in nineteen seventy-eight.



Herman

Pipecat is actually one of the few places you can tune that per stage, because the turn detector is a swappable component rather than baked into the vendor's session. But nobody tunes it until customers complain, and customers don't complain, they just hang up.



Corn

Which loops back to the transcript point, in a way. The two problems rhyme. The framework's value isn't that it's smarter, it's that it has doors in the wall where you can put a person with a screwdriver. The managed platforms are faster until the day the timing is wrong, and then the timing is wrong for every customer at once.



Herman

That's a fair distillation. The parts are a liability right up until the day they're the only thing that lets you fix the one setting that matters.



Corn

Which is the misconception, then, before we land this. The one most people hold.



Herman

I'd say it's that Pipecat and Gemini Live are rivals. They're different layers of one stack, and Pipecat literally ships first-class support for Gemini Live as a component. The framework's own docs walk you through it.



Corn

And the runner-up misconception, the one I'd want buried, is that speech-to-speech killed the cascade. Ninety-nine percent of Retell's agents still run the three-stage pipeline with fallbacks, and that's the vendor whose whole business is voice agents saying so about itself.



Herman

The cascade survives because it fails in pieces, and a phone call is the one place where failing in pieces is the entire requirement.



Corn

So the open question Daniel leaves us with, and it's the right one to end on. If the model vendors keep absorbing orchestration features, turn detection, barge-in, tool calls, what is the durable business of a framework? Kramer's bet is that new use cases always outrun the absorption. It's a bet, and the Deepgram acquisition makes it harder to read the neutrality argument at face value, and the foundation that's supposed to answer that still doesn't have a governance page. That's the space to watch. Meanwhile the split stands, engineering depth goes to LiveKit or Pipecat, a phone agent by next quarter goes to Vapi or Retell, and Groll's screwdriver tells us the last mile was always timing.



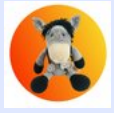
Herman

Thanks as always to our producer, Hilbert Flumingtop, at the desk and at the ledger book.



Corn

If you enjoyed this one, try episode seven, Building Custom ASR Tools; episode four, If Your Voice Ages, Does Your Fine-Tune Become Useless; and episode nine, Benchmarking Custom ASR Tools - Beyond The WER. This has been My Weird Prompts. If the show's been worth your commute, a review wherever you listen helps more than you'd think.



Herman

We'll be back soon. See you tomorrow.



Corn

Tomorrow. Slowly.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Pipecat README , retrieved 2026-11-20. [primary]**
   https://github.com/pipecat-ai/pipecat

2. **Deepgram, acquisition announcement, 2026-10-29 , retrieved 2026-11-20. [primary]**
   https://deepgram.com/blog/daily-pipecat-deepgram

3. **LiveKit, Series C announcement, 2026-01-22 , retrieved 2026-11-20. [primary]**
   https://livekit.com/blog/livekit-series-c

4. **LiveKit customers page , retrieved 2026-11-20. [primary]**
   https://livekit.com/customers

5. **LiveKit homepage , retrieved 2026-11-20. [primary]**
   https://livekit.com/

6. **Retell, 2026-11-12 , retrieved 2026-11-20. [primary]**
   https://www.retellai.com/blog/retell-100m-revenue-run-rate

7. **Retell about page , retrieved 2026-11-20. [primary]**
   https://www.retellai.com/about-us

8. **Vapi, Series B post, 2026-05-12 , retrieved 2026-11-20. [primary]**
   https://vapi.ai/blog/series-b

9. **Pipecat docs, Building with Gemini Live , retrieved 2026-11-20. [primary]**
   https://docs.pipecat.ai/pipecat/features/gemini-live

10. **Pipecat docs, Choosing a Transport , retrieved 2026-11-20. [primary]**
    https://docs.pipecat.ai/client/concepts/choosing-a-transport

11. **Daily, Pipecat Cloud GA, 2026-01-08 , retrieved 2026-11-20. [primary]**
    https://www.daily.co/blog/pipecat-cloud-is-now-generally-available/

12. **LiveKit, The best open source frameworks for building realtime voice and video AI agents [primary, competitor-authored], retrieved 2026-11-20.**
    https://livekit.com/blog/best-open-source-voice-and-video-ai-agent-frameworks

13. **AssemblyAI, Vapi vs LiveKit vs Retell (and Pipecat), pricing as of Nov 2026, retrieved 2026-11-20.**
    https://www.assemblyai.com/blog/vapi-vs-pipecat-vs-livekit

14. **Slator, 2026-11-03, retrieved 2026-11-20.**
    https://slator.com/deepgram-to-acquire-daily-pipecat/

15. **Unite.AI, retrieved 2026-11-20.**
https://www.unite.ai/deepgram-acquires-daily-commits-to-keeping-pipecat-open-source/

16. **New Runtime, retrieved 2026-11-20.**
https://newruntime.com/posts/deepgram-is-acquiring-daily-maker-of-the-pipecat-4e2126/

17. **TechCrunch, 2026-01-22, retrieved 2026-11-20.**
https://techcrunch.com/2026/01/22/voice-ai-engine-and-openai-partner-livekit-hits-1b-valuation/

18. **Unite.AI, 2026-01-23, retrieved 2026-11-20.**
https://www.unite.ai/livekit-reaches-1-billion-valuation-as-voice-ai-infrastructure-heats-up/

19. **Sacra, Retell revenue/funding estimates, retrieved 2026-11-20.**
https://sacra.com/c/retell-ai/

20. **TechCrunch, 2026-05-12, retrieved 2026-11-20.**
https://techcrunch.com/2026/05/12/vapi-hits-500m-valuation-as-amazon-ring-chose-its-ai-platform-over-40-rivals/

21. **AI Engineer World's Fair 2025 talk, Kramer (Daily) & Basu Mallick (Google DeepMind), retrieved 2026-11-20.**
https://ai.engineer/talks/fWY1FQwpWkY-milliseconds-magic-real-time-workflows-using

22. **Vadim Alakhverdov, latency comparison, prices as of June 2026, retrieved 2026-11-20.**
https://vadimall.com/posts/openai-realtime-vs-gemini-live-vs-pipecat-voice-ai-typescript

23. **Dealroom, Daily company profile, retrieved 2026-11-20.**
https://dealroom.co/companies/daily/

24. **Hakim, LiveKit vs Pipecat vs Vapi, retrieved 2026-11-20.**
https://tryhakim.ai/en/blog/livekit-vs-pipecat-vs-vapi

25. **Dreaming Press, retrieved 2026-11-20.**
https://dreaming.press/posts/livekit-vs-pipecat-vs-vapi-voice-agents.html