LiveKit Agents vs PolyAI

G2 voice-platform leaderboard, proof of concept on MIVAS single-agent tasks. Runs of 21–22 October 2026. LiveKit: Muse Voice Transcribe 1.0 / GPT-5.4 (reasoning off) / ElevenLabs Eleven v4 Turbo, 630 scored calls (126 callers × 5). PolyAI: GPT-4o Transcribe / Raven 3.5 / Eleven v4 Turbo, 126 scored calls (126 callers × 1). Same tasks, tool servers and deterministic task check. PolyAI's call-measured numbers are provisional while its STT and TTS configuration is still being explored.

Board points (out of 100)

Criterion	Points	LiveKit	PolyAI	Note
Task success	40	**27.3**	**15.1**	40 × clean pass rate (68.2% vs 37.8%)
WER	6	**4.7**	**4.2**	WER term only; semantic WER and form-filling not yet computed
Latency	8	**0.1**	**0.8**	Both far above the 1 s / 2 s targets
UI	5	**3.4***	**3.5***	Time to first agent decided by judgment on both
DevX	6	**4.5**	**4.2**	
Platform customizability	6	**4.4***	**3.6***	Interruption handling decided by judgment on both
Go live	8	**5.6**	**4.6***	Load: LiveKit from the run; PolyAI decided by judgment
MCP	6	**2.6**	**5.0**	
Docs	5	**3.0**	**3.9**	
Pricing clarity	5	**3.1***	**0.5***	Cost per hour decided by judgment on both
Language support	5	**5.0***	**5.0***	Both by configuration; live probes not run
Total	**100**	**63.5**	**50.4**	

* Includes an item decided by judgment on 22 October from documentation and run evidence, not by the live test the methodology specifies. Open for further discovery: barge-in test, metered cost hour, UI stopwatch and the five language probes on both platforms; PolyAI load ramp.

Measured on calls

Measure	LiveKit	PolyAI	Basis
Task success, clean, all industries	**68.2%**	**37.8%**	LiveKit 307/450 at k=5; PolyAI 34/90 at k=1
Task success, clean, customer support	**70.0%**	**16.7%**	
Task success, clean, healthcare	**58.0%**	**60.0%**	
Task success, clean, legal	**76.7%**	**36.7%**	
Task success, background-noise variants	**93.3%**	**66.7%**	LiveKit 90 calls; PolyAI 18
Task success, degraded-signal variants	**52.2%**	**33.3%**	LiveKit 90 calls; PolyAI 18

WER, clean, all industries	**8.3%**	**8.0%**	Platform's own caller transcript vs the caller's exact text, Whisper-normalized
WER, clean, customer support	**8.3%**	**13.7%**	
WER, clean, healthcare	**5.4%**	**6.3%**	
WER, clean, legal	**13.4%**	**7.5%**	
WER, background noise	**12.8%**	**17.4%**	
WER, degraded signal	**58.5%**	**83.3%**	
Latency, median per turn	**3.95 s**	**3.58 s**	End of caller speech to first agent audio. LiveKit 3,619 turns, PolyAI 672
Latency, p95 per turn	**6.60 s**	**5.86 s**	On Bluejay's per-turn metric (used for PolyAI) LiveKit is 3.80 s / 6.44 s
Concurrency sustained	**26–28 calls**	**2 calls**	LiveKit 0 breaks over two hours; PolyAI returned 429 above 2

Task success. LiveKit's first-call-only rate is 71.1% and its arguments-only rate (the PolyAI grading rule) is 70.0%, so neither k nor the grading rule moves the picture. LiveKit's legal fails are dominated by Muse dropping spoken phone numbers after a name; PolyAI's customer-support rate reflects Raven 3.5 skipping required lookups (get_customer_summary 21 times).

WER. Near-tied clean overall. LiveKit is better on customer support and healthcare, worse on legal because of the digit drop; PolyAI degrades much more under noise and bad signal. Scored on the WER term only; semantic WER and form-filling accuracy are not computed yet.

Latency. Neither platform approaches the 1 s / 2 s targets. LiveKit's turns are Muse end-of-utterance ~1.2 s plus GPT-5.4 ~0.9 s, with tool-call turns adding a second LLM pass (4–6 s). PolyAI on Raven is about 3 s faster per reply than it was on Opus 5.5.

Rubric scores by criterion

Each item is graded full (1), half (0.5) or none (0) and multiplied by its weight; a criterion adds to 10. * = decided by judgment today.

UI

#	Category	Weight	LiveKit	PolyAI	Note
1	Time to first agent	2.5	half* 1.25	half* 1.25	Quickstarts claim minutes; no stopwatch run on either
2	Build without code	2.0	full 2.0	half 1.0	LiveKit's browser Builder needs no code; PolyAI tools are Python functions
3	Test in the browser	1.5	full 1.5	full 1.5	
4	Call review	1.5	half 0.75	full 1.5	LiveKit insights are admin-only and kept 30 days
5	Errors you can see	1.5	half 0.75	half 0.75	
6	Team workspace	1.0	half 0.5	full 1.0	
	Total	**10**	**6.75**	**7.0**	

> **Findings.** LiveKit's browser Builder makes an agent with HTTP tools and no code; PolyAI's console needs Python for every tool but has the stronger call review (transcript, recording, per-call tool diagnosis) and team roles. Neither console has been stopwatched by a new builder.

DevX

#	Category	Weight	LiveKit	PolyAI	Note
1	Agent-as-code	2.5	full 2.5	full 2.5	
2	Call control	1.5	full 1.5	half 0.75	PolyAI web calls need a provisioned connector token
3	Data out	2.5	half 1.25	full 2.5	LiveKit transcripts are dashboard-only; PolyAI's Conversations API exports turns, latency, tool events, audio
4	API correctness	1.5	half 0.75	half 0.75	No idempotency on either
5	SDK quality	1.0	full 1.0	none 0	PolyAI has no client SDK
6	Scoped keys and auth	1.0	half 0.5	half 0.5	
	Total	**10**	**7.5**	**7.0**	

> **Findings.** LiveKit is code-first with typed Python and TypeScript SDKs and programmatic call control; PolyAI has no client SDK but a far better data path: its Conversations API exports turns, per-turn latency, tool events and audio, which is what made its WER measurable, while LiveKit's had to be captured by a hook in our own worker.

Platform customizability

#	Category	Weight	LiveKit	PolyAI	Note
1	Model choice	1.5	full 1.5	full 1.5	
2	Conversation behaviour	1.0	full 1.0	half 0.5	PolyAI silence and idle timeouts undocumented
3	Interruption handling	3.0	half* 1.5	half* 1.5	Barge-in on by default on both; stop time and backchannel rate not measured
4	Prompt and context	1.0	full 1.0	full 1.0	Full MIVAS prompts fit on both
5	Custom tools	2.5	half 1.25	half 1.25	LiveKit: no built-in retry. PolyAI: Python-only, no HTTP tool
6	Speech controls	0.5	full 0.5	half 0.25	PolyAI has no speaking-rate setting
7	Agent handoff	0.5	full 0.5	none 0	PolyAI handoffs are SIP transfers only
	Total	**10**	**7.25**	**6.0**	

> **Findings.** LiveKit lets you pick every model and hand off between agents with context; PolyAI's tools are Python-only, with no HTTP tool and no agent handoff. Barge-in is on by default on both and was not measured: the LiveKit run used the SDK default interruption mode, PolyAI allows up to five interruptions per call after the first turn.

MCP

#	Category	Weight	LiveKit	PolyAI	Note
1	MCP tools on a live call	3.5	full 3.5	full 3.5	Both docs-graded
2	MCP connection controls	1.5	half 0.75	half 0.75	
3	Platform MCP server	3.0	none 0	full 3.0	PolyAI ships Builder and Data MCP servers; LiveKit's is docs-only
4	What that server can do	2.0	none 0	half 1.0	PolyAI about 74% of documented REST endpoints
	Total	**10**	**4.25**	**8.25**	

> **Findings.** PolyAI's largest lead. It ships two management-plane MCP servers (Builder and Data) covering about three quarters of its REST API; LiveKit's only official MCP server is docs search, so a coding agent cannot build or manage LiveKit agents over MCP. Both can call MCP tools on a live call.

Go live

#	Category	Weight	LiveKit	PolyAI	Note
1	Telephony and numbers	2.0	half 1.0	half 1.0	LiveKit numbers inbound-only; neither documents porting or STIR/SHAKEN
2	Release management	1.5	full 1.5	full 1.5	
3	Runtime under load	2.5	full 2.5	none* 0	LiveKit held 26–28 concurrent with 0 breaks; PolyAI 429 above 2 concurrent, 26% errors at 3 in September
4	Integrations at the edge	1.0	half 0.5	full 1.0	PolyAI has guided CCaaS setups
5	Ops and incident response	1.5	half 0.75	half 0.75	
6	Compliance and trust	1.0	half 0.5	full 1.0	
7	Support and escalation	0.5	half 0.25	full 0.5	LiveKit publishes no numeric SLA; very responsive to us in practice
	Total	**10**	**7.0**	**5.75**	

Findings. The overnight runs are the load evidence: LiveKit's self-hosted workers held 26–28 concurrent calls with zero breaks, PolyAI returned 429 above two concurrent and errored on 26% of calls at three in September. PolyAI is ahead on contact-centre integrations, compliance attestations and a published 99.9% SLA; LiveKit numbers are inbound-only and its SLA is Enterprise-only, though the team was very responsive to us.

Pricing clarity

#	Category	Weight	LiveKit	PolyAI	Note
1	Public price	2.0	full 2.0	none 0	PolyAI publishes no price
2	Every line item named	1.5	full 1.5	none 0	
3	Cost per hour matches the estimate	3.0	half* 1.5	none* 0	LiveKit estimate possible, bill not reconciled; PolyAI has no estimate to check against
4	Usage and spend in the product	1.5	half 0.75	half 0.75	
5	Spend controls	1.5	none 0	none 0	
6	Free start	0.5	full 0.5	half 0.25	
	Total	**10**	**6.25**	**1.0**	

Findings. LiveKit prices every unit publicly, so a cost estimate is possible and only the bill reconciliation is open. PolyAI publishes no price at all, which the rubric scores as none on three items. Neither platform documents spend caps or alerts.

Docs

#	Category	Weight	LiveKit	PolyAI	Note
1	Reference accuracy	2.0	half 1.0	full 2.0	PolyAI publishes OpenAPI specs; LiveKit is protobuf-only
2	Payload and event schemas	1.5	half 0.75	full 1.5	
3	Concepts and lifecycle	1.0	half 0.5	full 1.0	
4	Telephony and go-live guides	1.5	half 0.75	half 0.75	
5	Errors, limits and gotchas	1.0	half 0.5	half 0.5	
6	Examples and quickstarts	1.0	full 1.0	half 0.5	PolyAI has no TypeScript examples
7	Change management	1.0	half 0.5	half 0.5	
8	Machine-readability	1.0	full 1.0	full 1.0	
	Total	**10**	**6.0**	**7.75**	

> **Findings.** PolyAI publishes OpenAPI 3.1 specs and full payload schemas for every REST family; LiveKit's reference is protobuf-only with several SIP fields present in the proto but absent from the docs. LiveKit's quickstart and llms.txt are the stronger entry points.

Language support

#	Category	Weight	LiveKit	PolyAI	Note
1	Spanish	2.0	full* 2.0	full* 2.0	Both graded by configuration: all five languages listed for STT, LLM and TTS on each side. Live probes not run.
2	French	2.0	full* 2.0	full* 2.0	
3	Mandarin Chinese	2.0	full* 2.0	full* 2.0	
4	Hindi	2.0	full* 2.0	full* 2.0	
5	Arabic	2.0	full* 2.0	full* 2.0	
	Total	**10**	**10**	**10**	

> **Findings.** Both graded by configuration rather than by probe: every layer on each side lists the five languages (ElevenLabs Eleven v4 Turbo speaks all five; Raven and GPT-5.4 answer in them; GPT-4o Transcribe covers them; Muse's realtime language list was not checked against all five). The five live probes are owed on both before this is final.

Sources: mivas/g2/rerun/V4_RUN_NOTES.md, V4_AUDIT.md, livekit_v4_scores.csv; Ronnie's PolyAI CSV and 49-item rubric (22 Oct 2026). Run audit and per-task attribution are kept in a separate findings document.