

WHITE PAPER

Real-Time Voice AI on CPU

Turn-taking, echo and capacity for assistants that answer a small business's visitors and callers.

Ron Alterio, Founder

November 2026

EXECUTIVE SUMMARY

The conversation has a clock

A voice assistant has one constraint a chat window does not: the conversation has a clock. A pause that would be unremarkable in text is a dropped call in speech, and an assistant that talks over someone, or hears itself and answers its own voice, is not a slower chat window but a broken one.

This paper presents the design behind Xorventa's voice assistants on websites and phone lines: how a turn is ended by silence and by whether the thought is complete, how an interruption stops the assistant without its own voice being mistaken for the caller, and how capacity is measured and enforced so that one caller too many is refused rather than degrading everyone.

0.17 core

Cost of each additional live session, fitted from 1 to 6 concurrent sessions, with no audible degradation on a workstation.

2.8–4.4 s

From pressing voice to the first word, down from over 10 s, mostly by moving browser costs ahead of the press.

43 ms

Median round trip across measured browser sessions (95th percentile 66 ms), with 0.2% median packet loss.

0 model calls

To decide whether a turn is finished, whether audio is echo and whether a caller can be admitted.

Written for: engineers and technical leaders building or evaluating real-time voice assistants.

1. The problem

Text retrieval tolerates delay and forgives interruption, because the reader is in control of both. Speech gives neither back. Three properties make voice a different engineering problem.

1.1 Turn-taking has no correct answer in silence alone

Nothing in the audio says a person has finished. People pause to think, and they pause when they are done. End a turn too early and the assistant cuts them off, then answers half a question. End it too late and every reply feels slow. A single silence threshold can only trade one failure for the other.

1.2 The assistant can hear itself

On a laptop or phone speaker, the assistant's own voice reaches the visitor's microphone. If that audio is transcribed as the visitor, the assistant starts answering itself. The obvious defence is to mute the microphone while it speaks, but then no one can interrupt it, and a person who cannot interrupt has lost the ordinary control every conversation has.

1.3 Capacity fails for everyone at once

A real-time media connection keeps accepting callers past what the machine can serve. There is no queue to grow and no request to retry. When the CPU runs out, every live call breaks up together, at exactly the moment interest peaks. For a small business's assistant on shared infrastructure, the operating constraints are also tight: CPU rather than GPU, and real-time media that may be tied to a single machine, which limits how far horizontal scaling can help.

2. Why the conventional approaches fall short

Common approach	Why it falls short for this problem
End the turn after a fixed silence	Any value cuts off a person who is thinking or adds lag to one who has finished. The two needs conflict, so no tuning resolves them.
A learned end-of-turn predictor	In our use it mis-guessed in both directions, stalling on long questions and cutting people off, and each wrong guess is hard to explain.
Mute the microphone while the assistant speaks	It stops echo by removing interruption, and the microphone tends to reopen before playback has actually finished.
A speech engine that invents words from noise	Whisper-family models are known to produce plausible text from silence and degraded audio. In a live call those words become a turn the assistant then answers.
A spoken filler after a timer	"One moment" at 1.5 s fired on ordinary turns, because ordinary replies begin at about that point. It added words without removing any real wait.
Admit every caller, then cope	The next caller's experience is fine for a moment, and everyone else's gets worse. Per-customer limits do not help, because the exhausted resource is the machine.
Size capacity from a synthetic test or one run	A media-only test measured well under half the real cost of a session (Section 4), and a single run of the real pipeline did not reproduce.

3. The solution

One assistant serves text, browser voice and phone calls, answering from the same knowledge. Voice adds a real-time path around it, where each stage has a specific job and a specific way to fail safely.

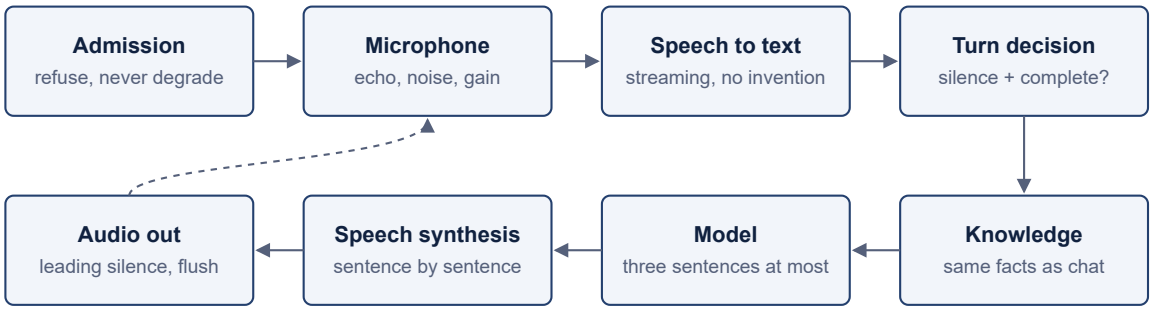


Figure 1. The path of one turn, clockwise from the top left. The dashed line is interruption: speech while the assistant is talking stops synthesis and clears audio already buffered in the browser.

3.1 Clean audio before it reaches the server

The microphone is requested with echo cancellation, noise suppression and automatic gain control all on, because the browser is the only place that knows exactly what is coming out of the speaker. Recognition then uses a non-Whisper streaming engine with biasing toward the business's own terms. The engine is selected by configuration, since it may need to change. Both engines we ran emit the same interim and final transcripts, so nothing downstream depends on which is active.

3.2 Ending a turn on silence and on completeness

The recognizer's silence window is deliberately short, at 800 ms, so that a finished thought commits quickly. When it fires, a deterministic check reads the accumulated words. If they trail off on a conjunction, preposition, article, hesitation or dangling auxiliary, with no closing question mark or exclamation, the assistant holds the floor for a further 1.5 s to let the person finish.

- **No model call.** The check adds no latency and cannot be slow or unavailable.
- **It fails towards answering.** Anything unclear counts as complete, so a real end of turn can never hang. A backstop bounds every hold: 3 s in a browser, 1.2 s on a phone line.
- **A greeting is not a turn.** "Hi there" is held, with interim speech extending it up to 6 s on screen surfaces, so the person's actual question is answered rather than the hello.

This resolves the conflict in Section 1.1 instead of tuning around it: the silence threshold is free to be short because a separate signal protects the person who is still thinking.

3.3 Interruption without self-hearing

Interruption depends on four independent gates: the pipeline must allow it, the speaking signals must reach it, a transcript arriving during playback must not be dropped, and the browser must be told to clear audio it has already buffered. Without that last gate the server stops generating and the visitor still hears the rest of the sentence. Any one gate left closed silently disables the others, so one setting drives all four.

Two checks keep the assistant's own voice from starting a turn. They run before the interruption gate, so opening it does not open a path for echo: a transcript that appears verbatim in the last reply, or that shares at least 70% of its words with it, is treated as echo, within a recency window and with a guard for real questions. An interruption must also reach two words, so a cough or a stray "uh" does not stop the sentence.

3.4 Clean speech out

A WebRTC jitter buffer is empty at the start of each reply, and if the first audio outruns it, the listener hears a hitch in the first word. A 200 ms leading silence fills it first. The injection is anchored to the single start-of-speech event for the turn and never inferred from a gap in the audio, because a late chunk mid-sentence would then be mistaken for a new reply, and silence spliced into a word is an audible click. Spoken replies are limited to three sentences, because a voice reply that runs long cannot be skimmed.

3.5 Starting fast

Much of the wait from pressing the button to hearing a voice is not in the server. A browser's first peer connection in a tab took 3.5 to 4.3 s, and every later one took 0.003 s. So the widget pays those once-per-tab costs when the visitor opens the window, not after the press. It also opens the microphone once, holding the permission check's stream until the call connects, and waits at most 0.6 s for connection candidates, letting late ones arrive afterwards.

3.6 Knowledge and prompt injection

Voice retrieves from the same knowledge as chat, using the same fact records and retrieval described in our paper on grounding AI chat, at the same cost of milliseconds per turn. Retrieved text reaches the model between delimiters with a per-turn random nonce, as data. Spoken answers sound more certain than written ones, which is a reason to keep this path the same as chat rather than build a second one. A prompt-cache warm-up during the greeting keeps the first real turn from paying the full prompt cost.

3.7 Ending a call safely

A call should be able to end because it is over, not only because a timer ran out. The two call-ending actions are deliberately opposite:

Action	If the caller speaks again	Reasoning
Hang up	Cancelled. Re-checked just before the line is cut, with a short grace window.	"Oh wait, one more thing" must work. If the check itself fails, the call is not ended, because cutting someone off costs more than a little dead air.
Transfer to a person	Not cancelled.	The caller has been told they are being put through, and the transfer cannot be taken back once it is scheduled.

Hard limits still exist for abandoned sessions: 15 minutes in total, and two minutes without audio in either direction, or one minute on a phone line.

3.8 Admission: refuse the next caller

A platform-wide limit applies across all customers, because the exhausted resource is the machine. Browser and phone sessions draw from the same count. Past the limit, a new caller is told at once and offered text chat, and calls already in progress are not disturbed. The limit is set from the measured cost per session and kept below saturation, allowing for the fact that one interpreter process may not be able to use every core. The operator is alerted before the limit is reached, ahead of anyone being refused.

4. The evidence

4.1 How we test

Question	Setup	What was recorded
What does media alone cost?	Real WebRTC with a speech-band tone: encryption, Opus decode and encode, resampling. 1 to 8 sessions.	Cores per session, late frames, 95th percentile frame gap.
What does the full pipeline cost?	Real backend and live speech vendors, 1 to 6 concurrent sessions, on a workstation.	The same, per process.
Is that repeatable?	Six 30-second runs, three on each of two speech engines.	Multiple of the media-only cost.
How does it behave in use?	Browser sessions on our service, October 2026.	Network quality from the browser's own statistics.

Frames arrive every 20 ms, and a gap above 60 ms, three frame periods, is counted as a glitch because it is audible as break-up. The media and pipeline runs were August 2026.

4.2 Capacity

Sessions	Media only: cores	Per session	Glitch %	p95 gap, ms
1	0.09	0.090	0.19	32.5
2	0.18	0.090	0.29	33.1
4	0.37	0.093	0.34	33.3
8	0.81	0.101	0.36	33.2

Live sessions	Full pipeline: cores	Per session	Glitch %	p95 gap, ms
1	0.281	0.281	0.08	33.9
2	0.510	0.255	0.12	33.1
3	0.684–0.711	0.211–0.237	0.13–0.29	32.6–33.1
4	0.783	0.183	0.38	33.2
5	1.014	0.193	0.14	33.0
6	1.132	0.180	0.15	33.7

A straight-line fit of the full pipeline gives about 0.12 cores fixed plus 0.17 per session. That is about three sessions per core at 70% load and about five at saturation.

> **Dividing a core by the cost of one session understates capacity, because a single session carries all the fixed overhead. Dividing it by the media-only cost overstates capacity. Only the full pipeline at several levels gives the number that matters.**

4.3 Why one run is not a measurement

Speech engine	Run 1	Run 2	Run 3	Mean multiple of media-only cost
Non-Whisper streaming engine	2.80	2.22	2.52	2.51
Whisper-family streaming engine	3.07	2.58	2.62	2.76

A single earlier run had given a multiple of 1.69, which would have made a planning assumption of 2 look cautious. It did not reproduce: repeated runs came out 25 to 40% above that assumption. The two engines overlap, so the engine is not what decides capacity.

4.4 Start-up

From pressing voice to the first word, on a microphone already permitted, the design in Section 3.5 reduced the time from 10.3–11.0 s to 2.8–4.4 s, measured with a real microphone against the live service on one laptop.

4.5 In use

Network quality per session	Median	95th percentile
Round-trip time	43 ms	66 ms
Packet loss	0.2%	2.3%
Jitter	2 ms	5 ms
Audio concealed by the browser	0.6%	3.3%

Browser sessions on our service with at least five quality samples, from the browser's own statistics. None of them needed a reconnect. These figures describe the network path to our service, not the model's response time.

4.6 Scope of these measurements

- **Cost per session.** The capacity runs used a many-core workstation, so they establish the cost per session and the shape of the curve. Absolute limits depend on the hardware.
- **Connection and media.** The test audio was a tone, which isolates the cost of holding a live session open. The cost of generating speech is additional and should be allowed for when sizing.
- **Reply timing.** The 1.1 to 1.5 s figure is what we see on ordinary turns, rather than a controlled benchmark across questions and load.

5. How this behaves in a call

Moment	Time	What sets it
Press voice to the first word	2.8–4.4 s	Browser costs paid in advance
Silence before a finished thought commits	800 ms	Recognizer window
Extra hold when the words trail off	1.5 s	Completeness check, no model call
Longest possible hold while speech continues	3 s browser, 1.2 s phone	Anti-hang backstop
End of turn to the start of the reply	About 1.1–1.5 s, observed	Retrieval in milliseconds; most of the rest is the model and synthesis
Leading silence on each reply	200 ms	Fills the jitter buffer
Caller past the limit	At once	Refused, offered text chat

The result is a clear division of labour. Decisions in the call's path are deterministic and take no time: completeness, echo and admission. The time that remains is spent where it improves the answer, in the model and the voice.

6. Guidance for teams building similar systems

If you are deciding	Our recommendation	What would change it
How to end a turn	Keep the silence window short and add a deterministic completeness check that fails towards answering.	A language where sentence endings are harder to read from words alone.
How to allow interruption	Turn on browser echo cancellation, check for echo before the interruption gate, and clear the browser's buffer.	Phone lines, where echo behaves differently.
Which speech engine	Test on noise and silence, not only clean speech, because inventing words costs more than missing them.	Batch transcription, where a person reads the result.
How to size capacity	Measure the full pipeline at several levels, fit the line and repeat it. Never divide by a single session.	Nothing. Then measure on the production machine as well.
What to do at the limit	Refuse the next caller, across all customers, and offer another channel.	A real queue may suit a contact centre with staff behind it.
Whether to fill silence	Do not speak on a timer. Acknowledge real work when it starts, and show state on screen.	Telephone calls, where a silent line can sound dead.

7. Conclusion

Most of what makes a voice assistant feel right is not the model. It is a set of small, deterministic decisions about whose turn it is, whose voice that was and whether another caller can be served well, with each one designed to fail in the direction that costs the person least. These can be built, tested and reasoned about without a GPU.

These results come from our own engineering tests and from sessions on our own service, within the scope in Section 4.6. The methods transfer directly, and we encourage teams building real-time voice to run them on their own stack.

References

The standards, models and findings this work builds on.

- W3C, WebRTC: Real-Time Communication in Browsers
- W3C, Media Capture and Streams
- IETF RFC 6716, Definition of the Opus Audio Codec
- IETF RFC 3550, RTP: A Transport Protocol for Real-Time Applications
- Radford et al., Robust Speech Recognition via Large-Scale Weak Supervision
- Koenecke et al., Careless Whisper: Speech-to-Text Hallucination Harms
- Python documentation, global interpreter lock

THE PRACTICAL CONCLUSION

Most of what makes voice feel right is not the model. It is deciding, quickly and safely, whose turn it is.

Silence and completeness, not silence alone

Interruption, with echo checked first

Capacity measured on the full pipeline, repeated

Refuse the next caller rather than degrade everyone

Explore Xorventa at xorventa.ai