



Real-Time Voice Pipeline

Voice and multimodal agents 01 | Cascade (ASR → LLM → TTS) vs half-cascade vs speech-to-speech as an engineering choice, the signal path with echo cancellation, audio formats and token rates, frameworks and realtime APIs as of 2026, prompts written for the ear

#VoiceAgents #Realtime #ASR #TTS #SpeechToSpeech

Three architectures

A real-time voice pipeline

×1

×1

Audio goes in, an answer comes out as audio, quickly enough that it feels like talking to a person.

×10

×10

Cascade: streaming ASR → LLM with tools → streaming TTS. Speech-to-speech: one model with an audio encoder in and codec tokens out. Half-cascade sits between.

- each stage streams to the next
- a transport (WebRTC, WebSocket, SIP/RTP) carries the media

×100

×100

Echo cancellation subtracts the agent's own playback from the microphone; VAD and a turn detector gate the ASR; the LLM writes speakable text; TTS emits 20-40 ms audio chunks; every hop adds buffering you must budget.

ANALOGY

A cascade is an interpreter team (listener, thinker, speaker); speech-to-speech is one bilingual person who never writes anything down.

The pipeline at three depths

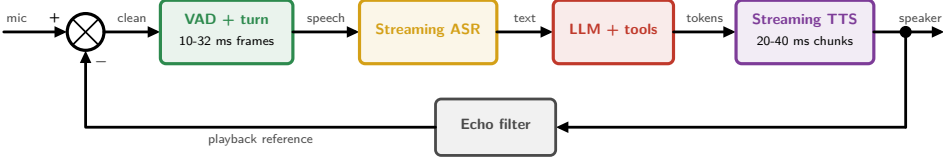
Before Engineer a voice pipeline, can you explain...

	Not yet	Partly	Yes
MUST KNOW ASR, TTS, neural codecs			
Learn: perception-and-generation 05	☐	☐	☐
MUST KNOW Audio-native models, any-to-any			
Learn: llm 16	☐	☐	☐
MUST KNOW Function calling			
Learn: agents 02	☐	☐	☐
HELPFUL Agent frameworks landscape			
Learn: agents 10	☐	☐	☐
HELPFUL Python asyncio			
Learn: python 05	☐	☐	☐

Check these before reading on

	Cascade	Half-cascade	Speech-to-speech
Latency floor	highest: three hops	medium	lowest: one model
Keeps tone, emotion, laughter	✗	input only	✓
Text log of both sides	✓	partial	transcripts on the side
Any LLM, any voice	✓	TTS only	✗
Tool-use control and audit	✓	✓	partial
Per-stage metrics (WER, MOS)	✓	partial	✗
Full duplex, overlap	✗	✗	some models

Engineering trade-offs as of 2026 (the mechanisms; vendors move fast, re-check the details)



Signal path of a cascade. Echo cancellation is a subtraction: the playback signal, filtered by an adaptive model of the room, is removed from the microphone so the agent does not hear itself

⚠ Speech-to-speech is not free of a pipeline

You still need transport, echo cancellation, turn detection (built in or yours), tool execution, logging and guardrails. Most providers return a **text transcript of the model's audio** for logs; it is a side output and may differ from what was spoken.

● Audio formats and token rates

Stream	Format	Rate
OpenAI Realtime API audio	PCM16 mono 24 kHz, or G.711 μ -law / A-law 8 kHz	$24,000 \times 16 \text{ bit} = 384 \text{ kbit/s}$ raw
Gemini Live API audio	PCM16 in at 16 kHz, out at 24 kHz	32 tokens per second of input audio (documented)

WebRTC	Opus, 48 kHz clock, 20 ms frames typical	about 16-40 kbit/s for wideband speech
Phone call (PSTN)	G.711 μ -law or A-law, 8 kHz, 8 bit	64 kbit/s (sheet 04)
Mimi codec (Kyutai, Moshi)	12.5 frames/s $\times$ 8 codebooks of 2,048	100 tokens/s, 1.1 kbit/s
EnCodec 24 kHz	75 frames/s $\times$ 2-32 codebooks of 1,024	1.5-24 kbit/s
Speech as text tokens	$\approx$ 150 words/min $\times$ $\approx$ 1.3 tokens/word (approx.)	$\approx$ 3 tokens/s

What travels on the wire and into the model (provider values as documented, as of 2026)

Wiring it: frameworks and realtime APIs

LiveKit Agents cascade (Python, 1.x API as of 2026; plug-in names change)

```
from livekit import agents
from livekit.agents import Agent, AgentSession
from livekit.plugins import deepgram, openai, cartesia, silero
from livekit.plugins.turn_detector.multilingual import (
    MultilingualModel)
```

```
async def entrypoint(ctx: agents.JobContext):
    session = AgentSession(
        stt=deepgram.STT(model="nova-3"),
        llm=openai.LLM(model="gpt-4o-mini"),

        tts=cartesia.TTS(),
        vad=silero.VAD.load(),
        turn_detection=MultilingualModel(),
    )
    await session.start(room=ctx.room, agent=Agent(
        instructions="Short spoken answers, no lists."))
    await ctx.connect()

if __name__ == "__main__":
    agents.cli.run_app(
        agents.WorkerOptions(entrypoint_fnc=entrypoint))
```

session.update (JSON event)

```
1 {"type": "session.update",
2   "session": {
3     "modalities": ["audio", "text"],
4     "instructions": "You are a booking agent...",
5     "voice": "alloy",
6     "input_audio_format": "pcm16",
7     "output_audio_format": "pcm16",
8     "input_audio_transcription": {"model":
9       "whisper-1"},
10    "turn_detection": {"type": "server_vad",
11      "threshold": 0.5, "prefix_padding_ms": 300,
12      "silence_duration_ms": 500},
13    "tools": [{"type": "function", "name":
14      "find_slot",
15      "parameters": {"type": "object"}}]}
```

- audio and a text transcript of the model's speech
- 24 kHz mono PCM16; g711_ulaw for phone legs avoids resampling
- side transcription of the user for logs; the model hears audio, not this text
- server VAD: 500 ms of silence ends the turn; semantic VAD is the alternative (sheet 03)
- tools as in text function calling; results come back as conversation items (sheet 06)

A speech-to-speech session configured over the OpenAI Realtime WebSocket (beta-era field names; the 2025 GA interface nests audio settings under session.audio, check current docs)

Event	Direction	What to do
input_audio_buffer.append	client $\rightarrow$	send base64 audio chunks (about 20-100 ms each)
input_audio_buffer.speech_started	$\leftarrow$ server	user began speaking: stop local playback (barge-in)
response.audio.delta.events	$\leftarrow$ server	queue the chunk for playback, track ms played
response.function_call_arguments.done	$\leftarrow$ server	run the tool, send the output item, then response.create

Bit rate of a neural audio codec

FORMULA $B = f \times q \times b$

WHERE

- B = bit rate of the token stream (bit/s)
- f = codec frame rate (frames/s)
- q = codebooks (RVQ levels) used per frame (count)
- b = bits per code = $\log_2(\text{codebook size})$; 2,048 entries $\rightarrow$ 11 (bit)

WORKED

f = 12.5
q = 8
b = 11
B = 12.5 $\times$ 8 $\times$ 11
B = 1 100 bit/s

RESULT B = 1 100 bit/s

READ IT Mimi as used by Moshi: 1,100 bit/s and f $\times$ q = 100 audio tokens per second, about 30 times the $\approx$ 3 text tokens per second of the same speech. A model that reads and writes audio tokens spends far more context per second than a cascade.

MISTAKES

- Compare tokens per second, not bits: context length is counted in tokens
- Fewer codebooks lower the rate and the audio quality together
- Provider-billed audio tokens are not codec tokens; use the documented rate

Context budget of a 10-minute support call

PROBLEM

How many input tokens does the user's speech in one call cost, as audio vs as a transcript?

1. Audio tokens

$\Rightarrow S \times a = 240 \times 32 = 7,680$ tokens

why: every second of speech is encoded, whatever its content

2. Transcript tokens

$\Rightarrow S \times w \times \tau = 240 \times 2.5 \times 1.3 \approx 780$ tokens

why: text keeps only the words

3. Ratio

$\Rightarrow 7,680 / 780 \approx 10$

why: audio context grows about ten times faster

4. Policy

$\Rightarrow$ keep the last turns as audio, older turns as transcript or summary

why: tone matters most in recent turns; costs and limits grow with every turn re-sent

GIVEN

- Call length 600 s, user speaks 40 % of it: S = 240 s
- Audio input rate a = 32 tokens/s (Gemini documentation, as of 2026)
- Speaking rate w $\approx$ 2.5 words/s; $\tau \approx$ 1.3 text tokens per word (approx.)

ANSWER

About 7,700 audio tokens vs about 800 transcript tokens: roughly 10 $\times$ more

Check: Agent turns re-sent as audio add their own share; count with the provider's usage fields per response

conversation.item.truncate	client →	cut the agent's message at the ms actually heard (sheet 03)
response.cancel	client →	stop a response in progress

Realtime API events a client must handle (OpenAI names; Gemini Live has equivalents)

Template ≈ tokens

role	You are the phone assistant of <code>{{business}}</code> . You speak with callers in real time.	81 ch 21
instructions	Answer in one or two short sentences. Ask one question at a time. Confirm names, dates and amounts by repeating them.	117 ch 30
constraints	No markdown, lists, emoji or URLs. Write numbers as spoken: 'twenty past three', 'forty-two euros'. Never read IDs longer than six characters; offer to text them.	162 ch 41
context	Caller: <code>{{caller_name}}</code> . Today is <code>{{date}}</code> . Opening hours: <code>{{hours}}</code> .	69 ch 18
output	If a tool will take time, first say a short holding phrase, then call it.	73 ch 19

2130411819

Template ≈ 129 tokens (502 characters / 4, rounded up per block; a rule of thumb for English, real tokenizers differ)

variable	meaning	used in	example value	check
<code>{{business}}</code>	company name spoken in greetings	role	Dupont Dental	ok
<code>{{caller_name}}</code>	from the CRM lookup, or 'unknown'	context	Ms Martin	ok
<code>{{date}}</code>	spoken date, local time zone	context	Monday the fifth of October	ok
<code>{{hours}}</code>	opening hours as a short sentence	context	nine to six, Monday to Friday	ok

Filled example ≈ 137 tokens

You are the phone assistant of `Dupont Dental`. You speak with callers in real time.

Answer in one or two short sentences. Ask one question at a time. Confirm names, dates and amounts by repeating them.

No markdown, lists, emoji or URLs. Write numbers as spoken: 'twenty past three', 'forty-two euros'. Never read IDs longer than six characters; offer to text them.

Caller: `Ms Martin`. Today is `Monday the fifth of October`. Opening hours: `nine to six, Monday to Friday`.

If a tool will take time, first say a short holding phrase, then call it.

{{variable}} slot

example value

no value given

A system prompt written for the ear: the LLM's text becomes speech, so format rules matter as much as content

Myths and recap

YOU MAY HEAR "Speech-to-speech removes the need for ASR" BUSTED ACTUALLY You still want a transcript Logs, analytics, compliance and evaluation run on text; most stacks transcribe the user on the side.	YOU MAY HEAR "Higher sample rate means better recognition" BUSTED ACTUALLY ASR models expect their training rate Most ASR runs at 16 kHz; phone audio is 8 kHz and upsampling adds no information. Resample once, correctly.
YOU MAY HEAR "Any chat prompt works for voice" BUSTED ACTUALLY Text written for the eye sounds wrong Lists, markdown, long numbers and URLs are read out literally; voice prompts constrain format.	

Three pipeline beliefs, busted

Tick a box only when you could explain the point to someone else

☐ Pick the architecture from control needs	☐ Echo cancellation before anything else
cascade for tools and audit, S2S for natural conversation	or the agent interrupts itself
☐ Know your formats	☐ Audio tokens cost ≈ 10 × text
8 / 16 / 24 / 48 kHz; resample once at the edge	summarise or transcribe old turns
☐ Write prompts for the ear	
short, no markup, spoken numbers	

- Q1

Three voice-agent architectures?

A

Cascade (ASR → LLM → TTS), half-cascade (audio in, text out, TTS) and speech-to-speech (audio in and out of one model).
- Q2

Why does a cascade suit regulated, tool-heavy agents?

A

Text between stages gives logs, per-stage metrics, any LLM and tight tool control.
- Q3

What does acoustic echo cancellation subtract?

A

The agent's own playback, filtered by an adaptive model of the echo path, from the microphone signal.
- Q4

Mimi codec tokens per second?

A

12.5 frames/s × 8 codebooks = 100 tokens/s (1.1 kbit/s).
- Q5

Raw bit rate of 24 kHz PCM16 mono?

A

24,000 × 16 = 384 kbit/s; base64 over WebSocket adds a third (≈ 512 kbit/s).
- Q6

Documented Gemini audio input rate?

A

32 tokens per second of audio (as of 2026).
- Q7

Which Realtime event tells you the user started speaking?

A

input_audio_buffer.speech_started: stop playback and prepare to truncate.
- Q8

Name three rules of a voice system prompt.

A

One or two short sentences; no markdown, lists or URLs; numbers written as spoken.
- Q9

Why keep a side transcription in speech-to-speech?

A

The model hears audio; logs, analytics and evaluation still need text.