

Voice Intent Models: Why You See Ads for What You Talked About

Machine Learning · Voice · Privacy

The evidence says nobody is transcribing your dinner: the ad was predicted from proxies, not overheard. But the capability the public suspects is real, because extracting intent from audio is not speech-to-text. The arithmetic of small decision spaces, on-device ears, and why the cheap model is also the private one.

doi: 10.5281/zenodo.23141636

Assets & Materials

The live voice demo. A ~400 kB intent model running inside the browser tab: talk about products, the dashboard and the exploration shelf react. Audio never leaves the tab.	https://booster-demo.vski.ai/app/voice
The voice-intent model page on vski.ai.	https://vski.ai/models/voice-intent
Model-Driven UI: Why Drilldowns Die (related). The interface half of the same story: what happens downstream of the intent.	https://blog.vski.ai/posts/model-driven-ui/

You mention hiking boots at dinner. By morning there is a boot ad on your phone. Most people have a story like this, and most people settle on the same theory: something heard them.

The evidence says something less cinematic. No study has caught a mainstream app streaming microphone audio into an ad auction. The one company that loudly sold exactly that service turned out, in the FTC's words, to not have been listening at all. The boot ad was predicted: from your location, your searches, the people near you, and the fact that you were already the kind of person who walks past that store. Prediction is cheaper than eavesdropping and, being invisible, considerably creepier.

But the public intuition contains a truth worth taking seriously: *listening is no longer expensive*. Pulling intent out of ambient audio no longer requires transcribing speech, shipping it to a GPU cluster, or anything in the speech-to-text family at all. This post is about that class of model, the voice intent model: what it is, what it costs (arithmetic, not vibes), where the label-space sizes put it on hardware from a microcontroller to one GPU serving a fleet, and why the honest version of this technology keeps the audio on the device.

What the evidence actually says

Start with the debunking, because it is solid.

The academic overview of phone eavesdropping (Kröger and colleagues, 2019) concludes that covert listening is technically feasible but a poor instrument: continuous audio streaming eats battery and data in ways users and researchers can measure, and modern iOS and Android both restrict background microphone access. Tests looking for it in the wild keep not finding it. When researchers demonstrate the trick, they do it with an app they wrote themselves, which proves feasibility, not deployment.

The loudest counterexample collapsed in court. Cox Media Group marketed “Active Listening” to small businesses: an AI that supposedly eavesdropped on consumers through phones and smart speakers and targeted ads at what was said aloud (404 Media broke the pitch deck in 2024). In 2026 the FTC settled with CMG and two partners for \$930,000, and the finding is the interesting part: the service **never used voice data**. Customers were getting marked-up email lists resold from data brokers. Consumers had not opted into anything. And the FTC added that had the service actually worked as advertised, collecting voice without consent would itself have been a violation. Selling fake surveillance is fraud; running real surveillance without consent is illegal; and the punished offense was the fraud.

Where ambient-audio-for-advertising is documented, it lives on the TV side of the living room. Vizio paid the FTC \$2.2 million in 2017 for tracking what people watched through automated content recognition without meaningful consent, and 2018 reporting showed smartphone apps carrying Alphonso's SDK using the microphone to identify TV audio for ad measurement. The pattern: ambient audio as a measurement signal is a real product category, disclosures notwithstanding, and it lives where the device is plugged into the wall.

And where ambient audio *classification* already runs at scale, it runs on your phone, benignly, on-device. Pixel's Now Playing feature (Google Research published the system) recognizes music continuously at low power, entirely locally, against an on-device database. Android's Sound Notifications and Apple's Sound Recognition classify smoke alarms, doorbells, and crying babies on the device, no upload involved. Your pocket already contains an always-on acoustic classifier or three. That is the hardware precedent for everything below. So the corrected picture: nobody is transcribing your dinner, because nobody needs to, and because transcribing is the expensive way to listen anyway. The rest of this post is about the cheap way.

The costly assumption: that listening means transcribing

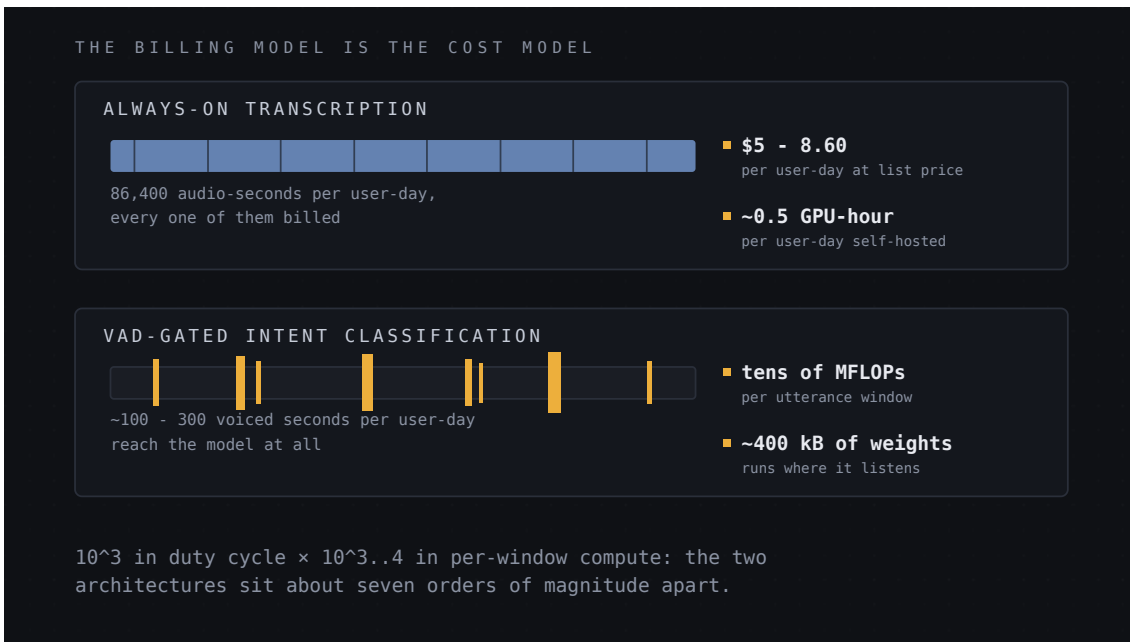
Speech-to-text is the wrong mental model, and it is what makes "listening is expensive" feel true.

STT is open vocabulary and sequential. Whisper, the reference open model, spans 39M parameters (tiny) to 1.55B (large-v3), and it bills by the audio-minute: the hosted API lists \$0.006 per minute, and competitors cluster around half that. The cost is structural. Every second of audio must be encoded, then decoded token by token, in a model that carries a language inside it. You pay for all of that whether the second contains a purchase intent or a cough.

A voice intent model makes a different trade. It does not produce words at all. It classifies an utterance into a **closed set of meanings**: the taxonomy your software can act on. Not "transcript: the kids need boots for the trip", but `outdoor__hiking-boots`, with a confidence, or `unknown`. This is the keyword-spotting and audio-tagging family, not the transcription family, and that family has always been tiny:

- MLPerf Tiny's reference keyword spotter, a depthwise-separable CNN, runs in single-digit milliseconds on a Cortex-M4 microcontroller at microjoule-scale energy, with weights around a megabyte quantized.
- Google's YAMNet classifies audio into 521 AudioSet classes from a few megabytes of MobileNet weights, comfortably real-time on phones.
- Our own voice intent model, 129 meanings, ships as roughly 400 kB and runs in a browser tab through WebAssembly.

None of this is exotic. It is the same lineage as "Hey Siri" detection on a DSP: when the action space is small, the model over audio can be small. The catalog decides the size of the model, not the language.



The arithmetic

An always-on microphone produces 86,400 seconds of audio per user per day. Route that through hosted STT at list price and one user costs \$5 to \$8.60 per day, somewhere between \$1,800 and \$3,100 a year. Transcribing one always-on microphone for a year costs more than the phone around it. Self-hosting does not save you: even at a generous 50 $\times$ real-time on one GPU, 24 hours of audio costs about half a GPU-hour per user-day. A million always-on users is 480,000 GPU-hours a day.

The intent model's bill looks different because it is gated twice.

Duty cycle. A voice-activity detector runs on raw energy and costs nothing. Only voiced utterances reach the model: on the order of 100 to 300 seconds per user-day, about 0.1 to 0.3 percent of the day. Three orders of magnitude never enter the building.

Compute per window. A small convnet over one second of mel-spectrogram, plus a classifier head, is tens of MFLOPs. For scale: a 10,000-label head on a 512-dimensional embedding is $512 \times 10,000 \approx 5\text{M}$ multiply-accumulates, about 5 MB of weights. The label space barely registers; the audio encoder dominates, and it is shared by every label count. A 5,000-label retailer taxonomy and a 10,000-label one are the same model with a different final matrix.

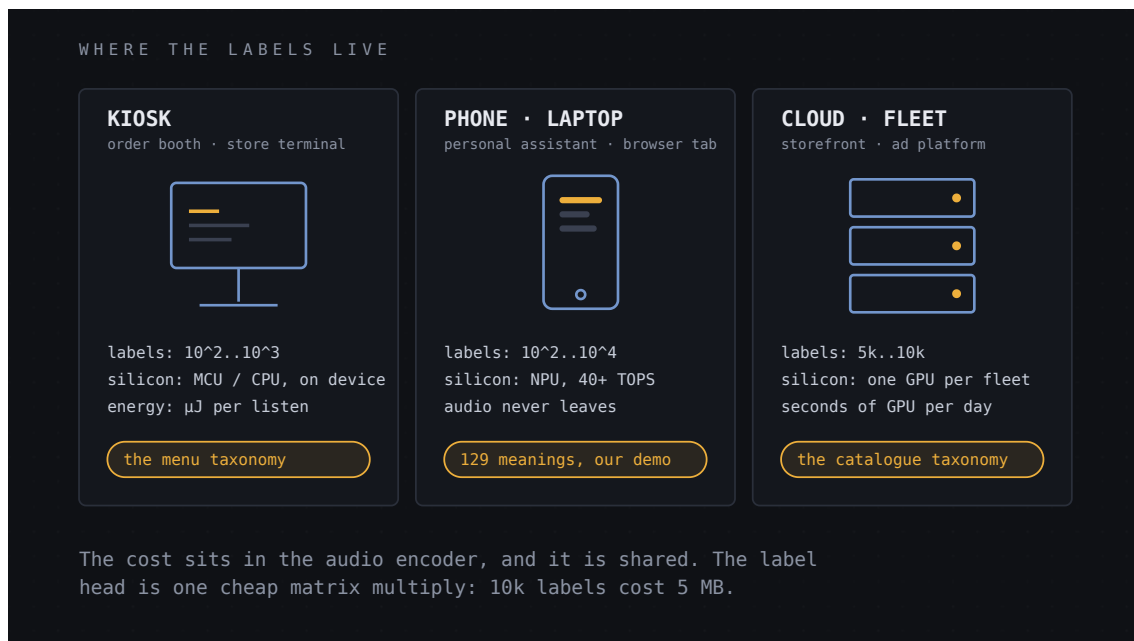
Put the fleet together. A million users at 200 utterances a day, 50 MFLOPs each:

$$10^6 \text{ users} \times 200 \text{ utterances} \times 5 \times 10^7 \text{ FLOPs} = 10^{16} \text{ FLOPs per day}$$

One modern GPU sustaining an effective 100 TFLOPS chews through that in about 100 seconds. Against 480,000 GPU-hours for the always-on STT route, that is roughly seven orders of magnitude: three from billing per utterance instead of per second, three to four more from classification instead of autoregressive decoding. The point of the arithmetic is the shape, not the constants. (Whisper is also solving a much harder problem. It has to handle

every word in every context. The intent model has to handle your catalog. Most of what makes STT expensive is exactly what the intent model refuses to do.)

Model sizes tell the same story from the memory side: 400 kB for the intent model, ~39 MB for quantized Whisper tiny, ~1.6 GB for quantized large-v3.



Three tiers of decision space

Where the label count puts you on hardware:

The kiosk (10^2 to 10^3 labels). A fast-food menu, a store terminal, a vending wall. The taxonomy fits in hundreds to a thousand labels, which is microcontroller territory: MLPerf-Tiny-class silicon classifies an utterance for microjoules, on the device, with no network round trip. Notice that this is the *cheap* half of what the drive-thru industry is currently building the expensive way. McDonald's shut down its IBM automated order taker test across 100+ restaurants in 2024 after the wrong-order virality; Wendy's FreshAI (Google Cloud) keeps going because an order must capture every word and modifier, and that is the STT-plus-LLM stack. Exact transcription of arbitrary orders is genuinely hard. But the order booth where two people discuss what they want while a shelf quietly surfaces what they mentioned needs only taxonomy-level intent, and that problem was solved by a microcontroller some years ago.

The phone and laptop (10^2 to 10^4 labels). A Copilot+ PC carries at least 40 NPU TOPS; Snapdragon's Hexagon does 45. Fifty MFLOPs per utterance is microseconds of that budget even at one percent utilization. This is where our demo lives: 129 meanings, ~400 kB, WebAssembly in a browser tab, the microphone armed on arrival, zero audio uploaded. Now Playing already proved the duty cycle: continuous, on-device, battery-invisible.

The cloud fleet (5 to 10k labels). A full storefront taxonomy. The fleet arithmetic above says one GPU spends under two minutes a day serving a million users. The label head

grows linearly and negligibly: 10k labels cost 5 MB. When someone tells you voice understanding needs a GPU farm, they are describing transcription, not intent.

What the small model cannot do

The honesty section, because the failure modes are structural.

It is closed-world. An intent model knows its taxonomy and nothing else. A product name, a SKU, a street address, anything off the vocabulary lands in `unknown`, and that is the correct behavior: the model should refuse rather than guess its nearest training example. The escape hatch is whatever exact interface sits next to it: a menu, a keypad, exact search. This is the same tiering as model-driven UI: suggest-and-confirm covers the head, exact input absorbs the tail.

Coverage is trained, not assumed. Voices, accents, tempos, and microphone conditions are all distribution. Our own first ear overfit the one voice it was trained on and had to be re-trained on more. Any deployment claim should be read with that in mind.

False positives exist. Every voice model confuses something. Small models earn trust not by never being wrong but by gating: commit only on a rise above the microphone's own measured baseline, veto when the input looks like babble, and never act on weak evidence. Confusion rates near a third of committed utterances are survivable when the failure is "wrong chip shown", and unacceptable when the failure is "wrong item ordered". Match the commit threshold to the blast radius.

Privacy is not automatic. On-device inference keeps the audio local, but the extracted label is still data. A session that transmits "user mentioned moving apartments" upstream has done surveillance with extra steps. The FTC's finding applies here with full force: consent to ambient listening has to be visible and session-scoped, an armed indicator the user can see and end, not a clause in a clickwrap. The architecture should make silent listening impossible, not merely discouraged.

Principles for voice intent

Collected from the arithmetic and the failure modes, the rules we develop against:

1. **Classify, don't transcribe.** Produce the smallest sufficient artifact: a meaning from the action space, with a confidence. Words are for the exact-input tier.
2. **Bill per utterance, not per second.** Gate on voice activity and measured norms. What never reaches the model costs nothing.
3. **Match the label space to the action space.** The catalog sizes the model, not the language. 10k labels is a 5 MB head, not a bigger encoder.
4. **Audio stays where it is heard.** The label may leave, while the ear is armed and visible. The waveform never does.
5. **Unknown is an answer.** Refusal to a typed fallback beats a confident guess outside the taxonomy.

6. **The cost story is the privacy story.** What is cheap enough to run locally never needs to be uploaded. Cheap is not the threat; cheap with a network route is.

The demo

The voice page of our demo store arms the microphone on arrival and listens while the dashboard fills. The model is ~400 kB, runs in the tab, and commits exactly one intent per utterance: the argmax label, only when it rises above its own measured baseline on your microphone, so the gates survive retraining and room noise without retuning. Every committed intent lands in the dashboard, and the exploration shelf reshuffles so the last thing spoken about sits on the first row. Nothing is recorded and nothing is sent. The downstream half, why a shelf should behave that way at all, is the previous post's business.

Sources

- Kröger, J. et al., “Is My Phone Listening in? On the Feasibility and Detectability of Mobile Eavesdropping”, FDN/TMA 2019 (Springer):
https://link.springer.com/chapter/10.1007/978-3-030-22479-0_6
- FTC, “FTC to Require Cox Media Group, Two Other Firms to Pay Nearly \$1 Million to Settle Charges They Deceived Customers About Their ‘Active Listening’ Ad Service”, May 2026: <https://www.ftc.gov/news-events/news/press-releases/2026/05/ftc-require-cox-media-group-two-other-firms-pay-nearly-1-million-settle-charges-they-deceived>
- FTC, final orders with the same firms, August 2026: <https://www.ftc.gov/news-events/news/press-releases/2026/08/ftc-finalizes-orders-cox-media-group-two-other-firms-settling-charges-they-deceived-customers-about>
- 404 Media, “Marketing Company Claims That It Actually Is Listening to Phones, Smart Speakers for Ads Marketing”, December 2023: <https://www.404media.co/cmg-cox-media-actually-listening-to-phones-smartspeakers-for-ads-marketing>
- 404 Media, “Here’s the Pitch Deck for ‘Active Listening’ Ad Targeting”, August 2024: <https://www.404media.co/heres-the-pitch-deck-for-active-listening-ad-targeting>
- FTC press release, Vizio ACR settlement, February 2017 (\$2.2M): <https://www.ftc.gov>
- The New York Times, “How Smart TVs in Millions of U.S. Homes Track More Than What’s On Tonight”, June 2018 (Alphonso smartphone-mic audio recognition): <https://www.nytimes.com>
- Google Research, “Now Playing: Continuous low-power music recognition” (on-device, always-on): <https://research.google/pubs/archive/46522.pdf>
- OpenAI, Whisper model card and sizes (tiny 39M to large-v3 1.55B): <https://github.com/openai/whisper>
- OpenAI API pricing, Whisper at \$0.006/min: <https://platform.openai.com/pricing>

- Banbury et al., “MLPerf Tiny Benchmark”, NeurIPS Datasets and Benchmarks 2021 (DS-CNN keyword spotting, Cortex-M latency and energy):
<https://arxiv.org/abs/2106.07550>
- YAMNet, AudioSet-trained audio event classifier (521 classes, MobileNet backbone):
<https://www.tensorflow.org/hub/tutorials/yamnet>
- Hershey et al., “CNN architectures for large-scale audio classification”, ICASSP 2017 (AudioSet): <https://arxiv.org/abs/1609.09430>
- McDonald’s ends IBM drive-thru AI test, June 2024, as widely reported (CNBC, Restaurant Business): <https://www.cnn.com>
- Wendy’s FreshAI drive-thru ordering with Google Cloud, as reported from 2023:
<https://www.aboutwendys.com>
- Microsoft, Copilot+ PC requirements (40+ NPU TOPS):
<https://www.microsoft.com/windows/copilot-plus-pcs/>