

Client-Side ASR - Whisper In The Browser - Engineering Cheat Sheet

Fora Soft Learn · AI for Video Engineering

1 - Untangle the name (the #1 mistake)

'faster-whisper' = CTranslate2 server/desktop library - NOT a browser engine.
Real browser engines: whisper.cpp (WebAssembly) + Transformers.js (WebGPU).
Built-in Web Speech API sends audio to Google by default - not on-device.

2 - The in-browser pipeline (all in the tab)

Mic -> getUserMedia -> resample to 16 kHz mono -> VAD gate -> Whisper -> caption.
No audio leaves the device. The VAD gate skips silence and saves the battery.

3 - Two engines: WebAssembly vs WebGPU

WebAssembly (whisper.cpp): runs on the CPU; SIMD + threads; the fallback.
-> threads need cross-origin isolation (COOP + COEP headers) or they crawl.
WebGPU (Transformers.js): runs on the GPU; up to ~100x faster; default late 2025.

4 - The cost ledger (no cloud bill - cost moves to the device)

Download (once, cached): tiny 75 / base 142 / small 466 MB; quantized 31 / 57 / 182.
Memory: a few hundred MB for tiny, more for larger. Battery: only while running.
Sweet spot for live: tiny/base, quantized, VAD-gated, WebGPU when available.

5 - Does it keep up live? The real-time factor (RTF)

RTF = processing time / audio length. Below 1.0 = keeps up; above 1.0 = falls behind.
Example: base on CPU 1.2 s / 1 s = 1.2 (too slow); on WebGPU 0.25 s / 1 s = 0.25 (fine).
Always measure on the device your users actually have. Moonshine suits tight live loops.

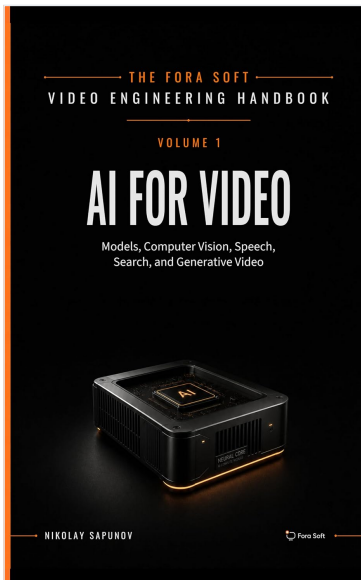
6 - Client or server? Checklist

- ☐ Audio must stay private / must work offline / 1:1 or single user -> run on the CLIENT.
- ☐ Group call you record, search, or translate / need top accuracy -> run on the SERVER.
- ☐ Prefer WebGPU; fall back to WebAssembly; ship COOP + COEP headers for Wasm threads.
- ☐ Size the model to your WORST supported device - not the developer's laptop.
- ☐ Recover accuracy with VAD + noise suppression, not with a bigger model.

Companion to: 'Client-side ASR - Whisper in the browser'

www.forasoft.com/learn/ai-for-video-engineering/articles-ai/client-side-asr-faster-whisper-wasm-browser

Fora Soft · www.forasoft.com



GO DEEPER · VOLUME 6 OF 7

AI for Video: A Complete Guide to Intelligence in Video Products

From Models and Inference to Computer Vision, Speech, Search, and Generative Video

By Nikolay Sapunov, CEO at Fora Soft · 1,444 pages · Kindle & paperback on Amazon

[Read it on Amazon →](#)
amazon.com/dp/B0HH6TKPZ1

This sheet untangles the browser ASR engines and gives the pipeline, cost ledger and real-time-factor check for running Whisper in a tab. Volume 6 covers speech in depth: ASR, TTS and diarization, model artifacts and deployment topologies, and the ML minimum you need to choose between client and server.

WHAT YOU WILL FIND INSIDE

- Speech recognition, TTS and diarization
- Model artifacts and how they ship
- Deployment topologies: device, edge, cloud
- ML basics for product decisions

THE FULL HANDBOOK · 7 VOLUMES · 9,000+ PAGES · ALL ON AMAZON

Vol. 1 · Video Encoding — pixels and color to H.264, HEVC, AV1, quality metrics and AI

Vol. 2 · Video Streaming I — network transport, contribution and ingest, HLS, MPEG-DASH, CMAF

Vol. 3 · Video Streaming II — ABR, CDNs, players, WebRTC, DRM and the economics of delivery

Vol. 4 · Audio for Video I — PCM, codecs, loudness, accessibility, real-time voice processing

Vol. 5 · Audio for Video II — jitter buffers, lip sync, Atmos, quality metrics, neural codecs

Vol. 6 · AI for Video I — models and inference, computer vision, speech, search, generative video

Vol. 7 · AI for Video II — real-time pipelines, agents, serving, cost, compliance, verticals

Want live captions that run in the browser without sending audio to a server?

We will benchmark Whisper builds on your users' devices and ship a WebGPU pipeline with a WebAssembly fallback. 250+ video products shipped since 2005.

[Book a 30-min call →](#)
[WhatsApp →](#)
[Email us →](#)