

1 — Распутаем название (ошибка №1)

'faster-whisper' = серверная библиотека CTranslate2 — НЕ браузерный движок.
Реальные движки: whisper.cpp (WebAssembly) + Transformers.js (WebGPU).
Встроенный Web Speech API по умолчанию шлёт звук в Google — не на устройстве.

2 — Конвейер в браузере (всё во вкладке)

Микрофон -> getUserMedia -> 16 кГц моно -> VAD-гейт -> Whisper -> субтитр.
Звук не уходит. VAD-гейт пропускает тишину и бережёт батарею.

3 — Два движка: WebAssembly против WebGPU

WebAssembly (whisper.cpp): на CPU; SIMD + threads; запасной путь.
-> потокам нужна cross-origin isolation (COOP + COEP), иначе ползёт.
WebGPU (Transformers.js): на GPU; до ~100х быстрее; по умолч. с конца 2025.

4 — Реестр затрат (нет облачного счёта — затраты на устройстве)

Загрузка (раз, кеш): tiny 75 / base 142 / small 466 МБ; квант. 31 / 57 / 182.
Память: сотни МБ для tiny, больше для крупных. Батарея: только пока работает.
Золотая середина для живого: tiny/base, квант., VAD-гейт, WebGPU где есть.

5 — Успевает ли вживую? Коэффициент реального времени (RTF)

RTF = время обработки / длина звука. Ниже 1,0 = успевает; выше 1,0 = отстаёт.
Пример: base на CPU 1,2 с / 1 с = 1,2 (медленно); на WebGPU 0,25 / 1 = 0,25 (ок).
Меряйте на реальном устройстве пользователя. Moonshine — для тесных живых циклов.

6 — Клиент или сервер? Чек-лист

- ☐ Звук приватный / должно работать офлайн / 1:1 или один юзер -> на КЛИЕНТЕ.
- ☐ Групповой звонок с записью, поиском, переводом / макс. точность -> на СЕРВЕРЕ.
- ☐ Предпочитайте WebGPU; запасной WebAssembly; шлите COOP + COEP для потоков Wasm.
- ☐ Подбирайте модель под ХУДШЕЕ поддерживаемое устройство, не под ноут разработчика.
- ☐ Добирайте точность VAD + шумоподавлением, а не более крупной моделью.