

Памятка: архитектура приложения диктовки

Пять стадий, выбор «на устройстве против облака», бюджет задержки и тесты перед запуском

1 · Пять стадий приложения для диктовки

Стадия	Что делает	Готовый кирпич (2026)	Где работает
1 Захват	Микрофон → ломтики	API микрофона ОС / Electron	На устройстве
2 Детекция	Речь или тишина (VAD)	Silero VAD	На устройстве
3 Расшифровка	Речь → сырой текст	faster-whisper (NVIDIA) / whisper.cpp (Mac)	Устр. или сервер
4 Очистка	Сырой → чистый текст	Языковая модель + промпт	Устр. или сервер
5 Доставка	Вставка у курсора	Вставка ОС / accessibility API	На устройстве

Whisper — это движок (модель); Wispr Flow — машина (приложение поверх него). Стадия 4 — очистка языковой моделью, убирающая паразитов и чинящая пунктуацию, — и делает так, что современное приложение «пишет, а не расшифровывает».

2 · На устройстве или в облаке? Вопрос «аудио покидает машину»

Критерий	На устройстве (Superwhisper)	В облаке (Wispr Flow)
Аудио покидает машину	Нет — никогда	Да — едет на сервер
Потолок приватности	Высший (юр./мед./фин.)	Зависит от политики хранения вендора
Качество модели	Ограничено железом	Одни сильные модели для всех
Работает офлайн	Да	Нет
Структура цены	Единоразово / free локально	За минуту → подписка
Когда выбирать	Аудио должно быть приватным	Топ-качество на любом устройстве

3 · Секундный бюджет задержки (одна продиктованная фраза)

Стадия	Бюджет	Заметка
End-of-turn детекция (VAD)	~120 мс	Заметить, что вы замолчали
Сетевой round trip	~150 мс	Только облачная сборка
Расшифровка (модель ASR)	~400 мс	Меньше половины бюджета
Очистка (языковая модель)	~250 мс	Убрать паразитов, пунктуация
Доставка к курсору	~80 мс	Вставить текст

VAD + очистка = ~370 мс — больше трети. Модель расшифровки — никогда не вся история; работа вокруг неё решает, ощущается ли приложение мгновенным.

4 · Чек-лист перед запуском

- ☐ Назвали задачу: диктовка (речь → текст в любом приложении), расшифровка файла или живые субтитры.
- ☐ Выбрали, где работает каждая стадия: на устройстве (приватно) или в облаке (мощно на любом железе), либо гибрид.
- ☐ Подтвердили требование приватности первым — оно решает архитектуру до написания кода.
- ☐ Выбрали движок: faster-whisper на NVIDIA, whisper.cpp на Mac — обе запускают одни модели Whisper.
- ☐ Добавили слой очистки (языковую модель) — это и делает «пишет, а не расшифровывает».
- ☐ Задали цель по задержке (< 1 с ощущается быстрым) и измерили её end-to-end из реальных локаций пользователей.
- ☐ Настроили end-of-turn детекцию (VAD) на реальном шумном аудио — акценты, фон, паузы посреди фразы.
- ☐ Обработали лимит куска в 30 секунд: расшифровывайте инкрементно; не заставляйте ждать заполнения окна.
- ☐ Защитились от галлюцинаций: никогда не записывайте мед./юр. транскрипт в карту без проверки человеком.
- ☐ Проверили покрытие языков для всех нужных языков, включая смену языка посреди фразы, если требуется.
- ☐ Подтвердили data residency и хранение: куда едет аудио и устраивает ли это вашу регуляторику?
- ☐ Посчитали вычисления: посекундный счёт за сервер (облако) против единоразового на устройстве — это задаёт цену.