Dictation App Architecture Checklist

The five stages, the on-device-vs-cloud decision, the latency budget, and the tests to run before shipping

1 · The five stages of a dictation app

Stage	What it does	Off-the-shelf block (2026)	Where it runs
1 Capture	Mic → buffered slices	OS microphone APIs / Electron	On-device
2 Detect	Speech vs silence (VAD)	Silero VAD	On-device
3 Transcribe	Speech → raw text	faster-whisper (NVIDIA) / whisper.cpp (Mac)	Device or server
4 Clean up	Raw → polished text	Language model + short prompt	Device or server
5 Deliver	Insert at cursor	OS paste / accessibility APIs	On-device

Whisper is the engine (a model); Wispr Flow is the car (an app built on one). Stage 4 — the language-model clean-up that strips fillers and fixes punctuation — is what makes a modern app "write, not transcribe."

2 · On-device or cloud? The audio-leaves-the-machine question

Criterion	On-device (e.g. Superwhisper)	Cloud (e.g. Wispr Flow)
Audio leaves machine	No — never	Yes — travels to a server
Privacy ceiling	Highest (legal/medical/finance)	Depends on the vendor's retention policy
Model quality	Limited by user's hardware	Same strong models for everyone
Works offline	Yes	No
Cost shape	One-time / free local tier	Per-minute → subscription
Choose when	Audio must stay private	Top quality on any device

3 · The one-second latency budget (single dictated sentence)

Stage	Budget	Note
End-of-turn detection (VAD)	~120 ms	Notice you stopped talking
Network round trip	~150 ms	Cloud build only
Transcription (ASR model)	~400 ms	Under half the budget
Clean-up (language model)	~250 ms	Strip fillers, punctuate
Deliver to cursor	~80 ms	Paste the text

VAD + clean-up = ~370 ms — more than a third. The transcription model is never the whole story; the work around it decides whether the app feels instant.

4 · Pre-flight before you ship

- [ ] Named the job: dictation (talk → text in any app) vs file transcription vs live captions.
- [ ] Chose where each stage runs: on-device (private) or cloud (powerful on any hardware), or a hybrid.
- [ ] Confirmed the privacy requirement first — it decides the architecture before any code is written.
- [ ] Picked the engine: faster-whisper on NVIDIA, whisper.cpp on Mac — both run the same Whisper models.
- [ ] Added the clean-up layer (a language model) — this is what makes it "write, not transcribe."
- [ ] Set a latency target (< 1 s feels quick) and measured it end-to-end from your users' real locations.
- [ ] Tuned end-of-turn detection (VAD) on real, noisy user audio — accents, background noise, mid-sentence pauses.
- [ ] Handled the 30-second chunk limit: chunk and transcribe incrementally; never make the user wait for a full window.
- [ ] Guarded against hallucination: never auto-commit a medical/legal transcript to a record without human review.
- [ ] Checked language coverage for every language you must support, including mid-sentence switching if needed.
- [ ] Confirmed data residency & retention: where does the audio travel, and does that meet your regulatory posture?
- [ ] Costed compute: per-minute server bill (cloud) vs one-time on-device — this drives your pricing model.