

● WHITEPAPER · VAHA SPEECH-TO-TEXT

On-device speech-to-text for VAHA: the model, where transcription happens, and a first accuracy test

AUTHORS

Ritesh Bonthalakoti

REVIEWER

Bharat Meda

VERSION

v1.0

DATE

2026-10-22

STATUS

Published

DISTRIBUTION

Public

CONTENTS

Abstract	6 Analysis and discussion
1 Introduction and objective	7 Limitations and threats to validity
2 Requirements and acceptance criteria	8 TRL assessment
3 System under test	9 Conclusions and next steps
4 Method	10 References
5 Results	Appendices

ABSTRACT

Abstract

VAHA turns each voice note into text on the device, an Arduino UNO Q, using `faster-whisper` on the CPU in int8. This paper describes that pipeline, checks the current build, as of 2026-08-17, against the product README, and reports a first accuracy and speed test of the two models the README and the code name. The device runs `base.en` by default, while the README's model table names `small`, and its own pipeline description names `base.en`. Captures from the device are always transcribed on the device. Notes recorded on the phone are not: with no device connection and a Groq API key set, their audio goes to a cloud service, and with no key their recording is deleted and the note keeps a placeholder text. On 12 synthetic sentences (120 words) on a development computer, `base.en` made 3 word errors and `small` 2, a difference of one word; `base.en` took 2.46 s per sentence on average and `small` 7.18 s (measured). The two are not separable on accuracy by this test; `small` is about 2.9 times slower (calculated). The transcription stage is at technology readiness level (TRL) 3.

KEY POINT

Key findings

- **Model:** the device runs `base.en` by default; the README's model table says `small` (277M), and its pipeline description says `base.en`.
- **Where it runs:** device captures never leave the device for transcription. Phone-recorded notes can be sent to Groq, and with a Groq key set, note text goes to Groq for titles.
- **Lost recordings:** a phone-recorded note's audio is deleted after every transcription attempt, including when no transcript was made.
- **Accuracy and speed:** one word separates the two models on a 120-word synthetic test; `small` took about 2.9 times as long. Not yet measured on the UNO Q or with real speech.

SECTION 1

Introduction and objective

VAHA's privacy position rests on transcription happening at home: the README states that speech-to-text runs on the UNO Q and that "no audio or text ever touches the cloud". Transcription sits between wake-word and stop-phrase detection, covered in a separate paper (reference 5), and local storage and sync, covered in the sync paper (reference 4). Which model does the work also matters, because the Whisper models trade accuracy against speed and memory, and a small single-board computer has little of either to spare.

This paper answers four questions:

1. Which speech-recognition model does the device run, and does the documentation say the same?
2. Where is each recording transcribed, and when does audio or text leave the home network?
3. Can a recording be lost on its way to a transcript?
4. How accurate and how fast are the two candidate models, `base.en` and `small`?

In scope: the device's transcription pipeline; the device's transcription endpoint used by the app; the app's handling of notes recorded on the phone, including its Groq fallback and Groq title suggestions; and a first accuracy and speed test of the two models.

Out of scope: wake-word and stop-phrase detection; Notion export; the sync of captures from device to app; and accuracy or speed on the UNO Q itself, which has not been measured yet (section 7).

SECTION 2

Requirements and acceptance criteria

The requirements come from the product README (reference 1) and the product's purpose. They were written down for this paper, after the code, so they state what the build is checked against, not acceptance criteria set before testing (section 8).

ID	Requirement	Criterion	Basis
R1	Transcription stays local	No audio or transcript leaves the home network	README: "No audio or text ever touches the cloud"
R2	Documented model	The README names the model the device runs	README model table and pipeline description
R3	No silent loss	No recording is deleted unless its transcript has been made	Implied by the product's purpose: capturing what was said
R4	Accuracy and speed	Word error rate and transcription time on the UNO Q within stated limits	No limits set yet; choosing them is a product decision that has not been made

SECTION 3

System under test

3.1 Transcription on the device

Each capture is transcribed once, after recording stops (Figure 1). The device records at 48 kHz, resamples to the model's 16 kHz, removes any DC offset and scales the clip to its own peak, then calls `faster-whisper` with the language fixed to English and its voice-activity filter off. The result is tidied: filler words (um, uh, hmm) are removed, the first letter is capitalised and a full stop added if missing. Transcript and audio go into the capture package that the companion app later syncs.

ONE CAPTURE, FROM MICROPHONE TO TRANSCRIPT, ON THE DEVICE

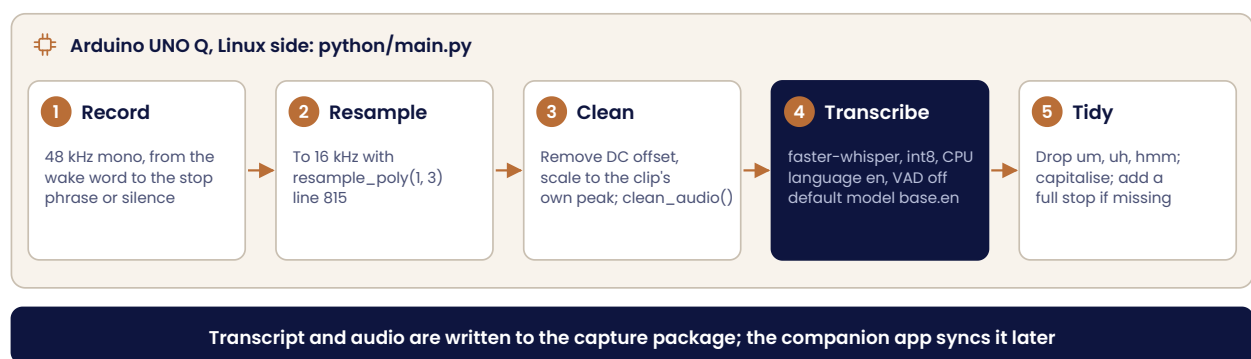


Figure 1 One capture from microphone to transcript on the device (`python/main.py` ; line numbers in Appendix A).

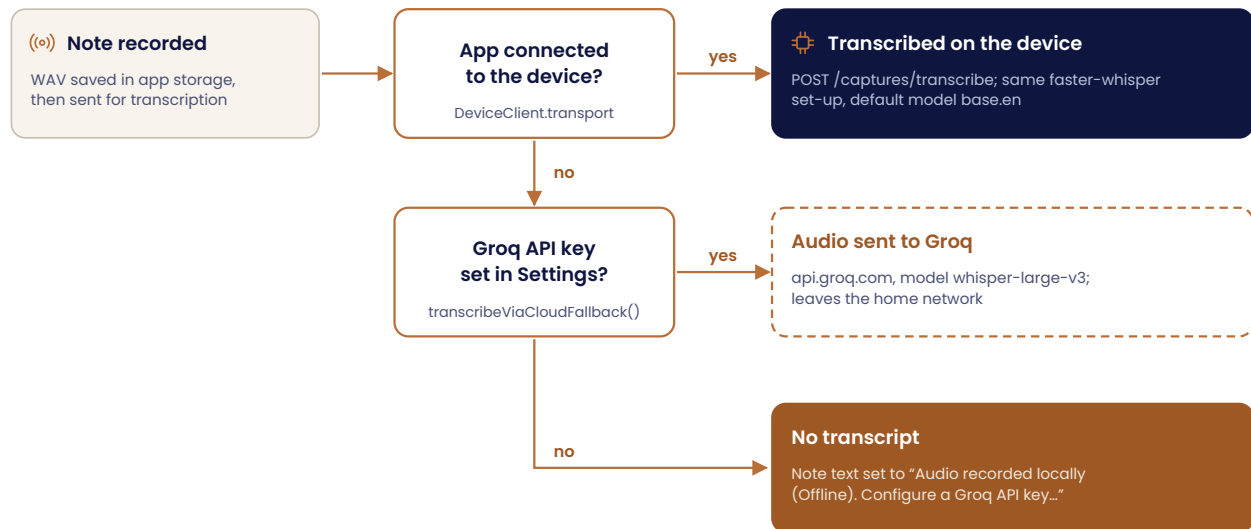
The model is chosen by name through the `WHISPER_MODEL_NAME` environment variable, with `base.en` as the default. No model weights are kept in the repository: on first use, `faster-whisper` downloads the named model to the device's model folder, and loads it from there afterwards.

The device server also offers a transcription endpoint, `POST /captures/transcribe`, which the app uses for notes recorded on the phone. It loads the model the same way, with the same default, and applies the same decode options and tidying. It does not apply the DC-offset and peak scaling step.

3.2 Notes recorded on the phone

The app can also record a note with the phone's microphone. Figure 2 shows what happens to that recording. If the app is connected to the device, it uploads the WAV to the device's transcription endpoint. If it is not, it falls back to Groq, a cloud service, provided the user has entered a Groq API key in Settings; without a key it makes no network call. "Connected" here means the app holds a device connection, which it creates when it connects to the device and clears when it disconnects. If the connection exists but the device does not answer, the upload fails and the app reports that transcription failed; it does not fall back to Groq.

● A NOTE RECORDED ON THE PHONE: WHERE ITS AUDIO GOES



In every case the WAV is then deleted; only the note text is kept.

With a Groq key set, the transcript is also sent to Groq (llama-3.1-8b-instant) to suggest a title.

Figure 2 Where a phone-recorded note's audio goes, as the app's code runs it (`new-capture.tsx` , `LocalWhisperTranscriptionService.ts` , `DeviceClient.ts` ; line numbers in Appendix A).

Groq is also used for titles. When a Groq key is set, the app sends each new phone note's transcript to Groq's chat service (model `llama-3.1-8b-instant`) to suggest a title, and does the same with the combined text when the user merges notes. Merged notes can include captures transcribed on the device.

3.3 Technology stack

Layer	Technology
Device	Arduino UNO Q (ABX00087), Linux side
Speech recognition, device	faster-whisper on CTranslate2, CPU, int8; default model <code>base.en</code>
Audio	Recorded at 48 kHz mono; transcribed at 16 kHz
Decode options	<code>language="en"</code> , <code>vad_filter=False</code>
Speech recognition, cloud fallback	Groq API, <code>whisper-large-v3</code> (phone notes only, when a key is set)
Title suggestions	Groq API, <code>llama-3.1-8b-instant</code> (when a key is set)

● SECTION 4

Method

4.1 What was examined

Item	Details
Source	VAHA source repository (reference 1), as of the last change to its code on 2026-08-17
Device	python/main.py : recording, resampling, cleaning, model loading and transcription; python/server/routes/captures.py : transcription endpoint
Companion app	Phone recording screen, transcription service, device connection, and the captures list (merging)
Models tested	base.en and small , faster-whisper int8 on CPU, downloaded by name as on the device
Test audio	12 sentences of the kind people dictate as notes, 120 words in all, spoken by Windows text-to-speech (two voices, Microsoft David and Microsoft Zira, at rate 165) and saved as 22.05 kHz mono WAV (analysis/tts_references/ , reference 6)
Test machine	Development computer: Intel Core i5-10210U at 1.60 GHz, Windows build 10.0.26200, Python 3.12.10
Software versions	faster-whisper 1.2.1, ctranslate2 4.8.3, pyttsex3 2.99, NumPy 2.5.2, SciPy 1.17.1, pinned in analysis/requirements.txt

4.2 Procedure

The scripts and data for the accuracy and speed test are published with this paper (reference 6). They sit in the analysis/ and data/ folders of the VAHA repository, and paths such as analysis/wer_latency_test.py below are relative to its root. They were added on 2026-10-22 without changing the code this paper examines.

1. Read the device's transcription code from recording to transcript, and the transcription endpoint.
2. Find how the model is chosen, its default, and when the default changed (Appendix B).
3. Compare with the README's model table and pipeline description, and their history.
4. Follow a phone-recorded note through the app: transcription on the device, the Groq fallback, title suggestions, and what happens to the recording.
5. Generate the test audio with analysis/gen_tts_references.py .
6. Transcribe each sentence once with each model on the test machine, with the device's decode options, cleaning and tidying, and record each transcript and its transcription time (analysis/wer_latency_test.py). Model loading is timed separately and not included.
7. Score each transcript against its sentence: lower-case, strip punctuation and count word substitutions, deletions and insertions. Word error rate is the total of these over the 120 reference words (analysis/summarise_wer.py).

Not yet done: the same test on the UNO Q; real recorded speech, in quiet and in a bathroom; the phone-note paths run on a handset.

SECTION 5

Results

5.1 Known gaps in the current build

CAUTION

Gap 1: the README names two different models

The device has run `base.en` by default since 2026-08-07, when the hard-coded `small` was replaced with the `WHISPER_MODEL_NAME` setting to speed up transcription (`main.py` line 56). The README's pipeline description gives `base.en` (lines 90 and 109), but its model table gives "`faster-whisper small (277M)`", described as a "fine-tuned English speech model" (line 234). All three were written on 2026-08-17. The repository holds no fine-tuned weights and no training code; the model is downloaded by name. The published sizes are 74 million parameters for `base` and 244 million for `small` (reference 2, datasheet), so the README's 277M matches neither. A comment on the device's transcription endpoint also still says it loads `small` (`captures.py` line 34).

Listing 1: The model the device runs, and what the documentation says

```
python/main.py
56 WHISPER_MODEL_NAME = os.environ.get("WHISPER_MODEL_NAME", "base.en")

python/server/routes/captures.py
34     Lazy-load faster-whisper small. Thread-safe singleton.

README.md
90     StopDetect -->|faster-whisper base.en int8| STT["On-Device Transcription"]
109 4. **On-Device STT**: Audio is downsampled to 16 kHz and transcribed locally
    using `faster-whisper` (`base.en`, int8 quantized, running on CPU).
234 | **C: Speech-to-Text (STT)** | `faster-whisper small` (277M) | CTranslate2 (int8, CPU) |
    Fine-tuned English speech model running 100% offline on-device. |
```

CAUTION

Gap 2: phone notes and note titles can go to a cloud service

When the app has no device connection and a Groq key is set, it sends a phone-recorded note's audio to `api.groq.com` for transcription with `whisper-large-v3` (`LocalWhisperTranscriptionService.ts` lines 27–29 and 55–66). With a key set, it also sends each new phone note's transcript, and the combined text of merged notes, to Groq to suggest a title (`new-capture.tsx` lines 158–160; `captures/index.tsx` lines 275–277). The Groq key is optional and empty until the user adds one. The README's setup steps mention a "Groq token for fallback" (line 285) but not what is sent or when, and its privacy statement says no audio or text ever touches the cloud (line 70). Captures recorded on the device are always transcribed on the device.

Listing 2: The cloud fallback in LocalWhisperTranscriptionService.ts

```
27     if (!transport) {
28         this.logger.warn("[TRANSCRIPTION] UNO Q device offline. Attempting cloud fallback...");
29         return await this.transcribeViaCloudFallback(audioUri);
30     }
31     ...
43     private async transcribeViaCloudFallback(audioUri: string): Promise<Result<string, Error>> {
44         const groqKey = useSettingsStore.getState().groqApiKey;
45         if (!groqKey) {
46             this.logger.warn("[TRANSCRIPTION] No Groq API Key configured for fallback.");
47             return Result.ok("Audio recorded locally (Offline). Configure a Groq API key in Settings to enable free internet transcription fallback.");
48         }
49     }
50     ...
54     const formData = new FormData();
55     formData.append("model", "whisper-large-v3");
```

CAUTION

Gap 3: a phone note's recording is deleted even when no transcript was made

After every transcription attempt the app deletes the phone recording, whatever the outcome (`new-capture.tsx` line 148); a phone note keeps only its text. If the device upload fails, the user is told transcription failed and can type the note; the recording is already gone. If the app has no device connection and no Groq key, the transcription service returns "Audio recorded locally (Offline). Configure a Groq API key in Settings to enable free internet transcription fallback." as a successful result (`LocalWhisperTranscriptionService.ts` line 48). That text becomes the note, and the recording it refers to has been deleted.

Listing 3: After transcription, in new-capture.tsx

```
145     const uploadResult = await transcriptionService.transcribe(localWavPath);
146     try {
147         await deleteAsync(localWavPath, { idempotent: true });
148     } catch (e) {
149         // ignore cleanup failures
150     }
151     if (uploadResult.isSuccess) {
152         const text = uploadResult.getValueOrThrow();
153         setTranscript(prev => prev ? prev + "\n" + text : text);
154     }
155     ...
158     const groqKey = useSettingsStore.getState().groqApiKey;
159     if (groqKey && text.trim()) {
160         const suggested = await suggestTitleUsingGroq(text, groqKey);
```

The recording is deleted at line 148, before the result is checked at line 153.

Table 5.1: Gaps summary

Gap	Where	Since	Consequence
README names two models	README lines 90, 109, 234; <code>main.py</code> line 56	2026-08-17 (README)	A reader cannot tell which model produced any accuracy or speed claim; "fine-tuned" and "277M" are not supported by the repository

Gap	Where	Since	Consequence
Phone notes and titles to Groq	LocalWhisperTranscriptionService.ts lines 27–66; new-capture.tsx lines 158–160; captures/index.tsx lines 275–277	2026-08-06 (fallback); 2026-08-08 (titles)	With a key set, phone-note audio, and note text including device transcripts, leave the home network, contrary to the README
Recording deleted without a transcript	new-capture.tsx line 148; LocalWhisperTranscriptionService.ts line 48	2026-07-22 (deletion); 2026-08-06 (placeholder text)	A phone note made with no device connection and no key, or whose upload fails, loses what was said; in the first case the note says the audio was kept

5.2 Accuracy and speed of the two models

Each model transcribed the same 12 sentences once (section 4.2). The sentences were 3.0 to 4.6 s long. Table 5.2 summarises the results; Table 5.3 lists every word error.

Table 5.2: Word error rate and transcription time, development computer, synthetic speech (measured, n = 12 sentences, one run)

Model	Word errors / words	Word error rate (95 % interval)	Time per sentence, mean ± SD	Min–max	Real-time factor, mean
base.en (device default)	3 / 120	2.5 % (0.9–7.1 %)	2.46 ± 0.48 s	1.91–3.20 s	0.63
small (README table)	2 / 120	1.7 % (0.5–5.9 %)	7.18 ± 0.60 s	6.40–7.97 s	1.85

Source: `data/raw/wer_latency_test_2026-10-22.csv`, summarised by `analysis/summarise_wer.py` into `data/processed/wer_latency_summary.csv` (reference 6). The 95 % interval is the Wilson score interval, treating each word as an independent trial (calculated); words in one sentence are not independent, so the true intervals are wider. Real-time factor is transcription time divided by the length of the audio; below 1 is faster than the speech. Loading the model took 13.9 s for `base.en` and 19.4 s for `small`; the device loads it once, on first use.

Table 5.3: Every word error in the test

Model	Sentence spoken	Transcript	Word errors
base.en	the meeting with the design team got pushed to three thirty	The meeting with the design team got pushed to 330.	2
base.en	idea for the app, add a dark mode toggle in settings	Idea for the app, add a dark mode toggle and settings.	1
small	the meeting with the design team got pushed to three thirty	The meeting with the design team got pushed to 3.30.	2

Both models wrote "three thirty" as a number, which the scoring counts as two errors; neither misheard it. The only misheard word is `base.en`'s "and" for "in". On this test the two models are equally accurate within one word, and `small` takes about 2.9 times as long (calculated from the

means in Table 5.2). The two intervals overlap almost entirely: a set this small, with speech this clean, cannot show an accuracy difference between them.

SECTION 6

Analysis and discussion

6.1 Results against the requirements

Table 6.1: Requirement conformance

ID	Criterion	Result	Status
R1	No audio or transcript leaves the home network	Device captures: transcribed on the device. Phone notes and note titles: sent to Groq when a key is set (Gap 2)	Partly met
R2	The README names the model the device runs	The model table names <code>small</code> ; the device runs <code>base.en</code> (Gap 1)	Not met
R3	No recording deleted before its transcript is made	Phone recordings are deleted after any attempt, including failed ones (Gap 3)	Not met
R4	Accuracy and speed on the UNO Q within limits	No limits set; measured only on a development computer with synthetic speech (Table 5.2)	Not measured

6.2 Which model the device should run

The switch to `base.en` was made for speed, and the test supports that reason: on the test machine `small` ran slower than the speech it transcribed (real-time factor 1.85), while `base.en` ran faster (0.63). The UNO Q's processor is slower than a development computer's, so both will take longer on the device; how much longer has not been measured. The test does not show `small` to be meaningfully more accurate on clean speech. Real speech in a bathroom, with fan noise, running water and echo, is where a larger model may earn its time, and that is the test to run before choosing (section 9). Whichever model is chosen, the README should name it, and both accuracy and speed should be measured on the device with that model.

Word error rate here also depends on how numbers are scored. "Three thirty", "330" and "3.30" mean the same to a reader, and a scorer that normalises numbers would count neither model's transcript as wrong. The scoring in this paper does not, so its error counts are, if anything, high.

6.3 Why the phone-note gaps matter

Gaps 2 and 3 affect only notes recorded on the phone, and only when a Groq key is set (Gap 2) or the device is not reachable (Gap 3). They are still the ones a user would not expect. A user who adds a Groq key to get transcription away from home also sends every new note's text to Groq for a title, and merged notes, including device captures, with it. A user without a key who records a note away from home is told the audio was recorded locally, and it was, until the app deleted it.

Both are small to fix: keep the phone recording until a transcript has been saved, show a "not transcribed" state instead of placeholder text, and ask before sending anything to Groq, with the README stating what is sent and when.

SECTION 7

Limitations and threats to validity

- **Not on the UNO Q.** The accuracy and speed test ran on a development computer. The device's processor is slower and was not measured; Table 5.2 shows the models' relative speed, not their speed on the device.
- **Synthetic, clean speech.** Two text-to-speech voices in silence have none of the variation of real speakers or the noise of a bathroom. Error rates on real speech will be higher.
- **Small sample, one run.** 12 sentences and 120 words per model, each transcribed once. Five word errors in total cannot separate the two models on accuracy.
- **Scoring.** Numbers are not normalised, so "three thirty" written as digits counts as two errors (section 6.2).
- **Code read, not run, for the gaps.** Gaps 2 and 3 are what the app's code does; they have not been observed on a handset.
- **Model in shipped units.** The default can be changed with `WHISPER_MODEL_NAME`, and nothing in the repository sets it. One developer checkout has an uncommitted local `.env` that sets `small`; it is excluded from the repository and is not part of any build or release. Units are therefore taken to run `base.en`; this was not checked on a unit.
- **Requirements set after the code.** R1 to R3 come from the README and the product's purpose, and R4 has no limits yet.

SECTION 8

TRL assessment

TECHNOLOGY READINESS: WHERE THIS WORK SITS

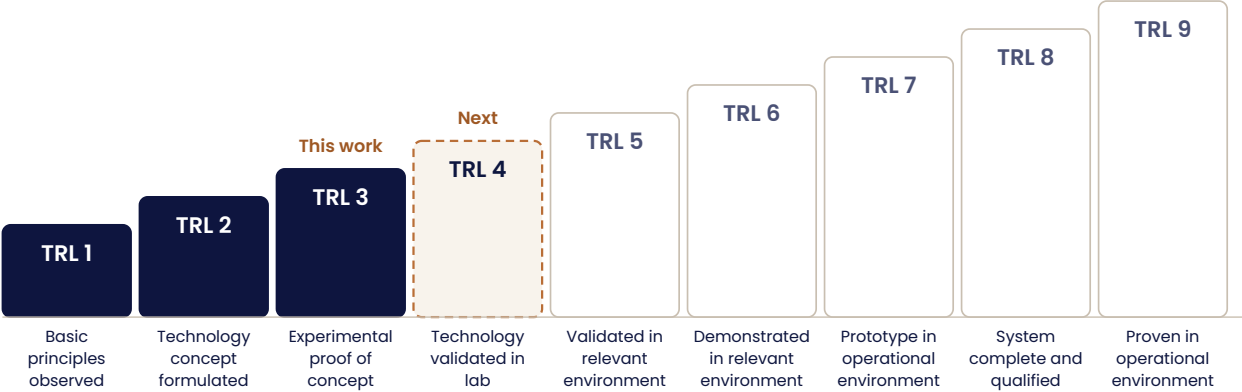


Figure 3 Where this work sits on the EU Horizon TRL scale. Definitions from the Dharanova publishing guidelines (EU Horizon TRL scale, ISO 16290 criteria, reference 3).

TRL demonstrated	Below TRL 4. The evidence supports an experimental proof of concept (EU Horizon TRL 3) for the transcription stage
Integration level	Component: speech-to-text, on the device and in the app's phone-note path
Fidelity	Prototype: real device code, real app code

Environment	Development computer, synthetic speech
Evidence	Sections 3 and 5: the code, and one accuracy and speed test of both models off the device
Why not TRL 4	TRL 4 requires validation in the lab against defined criteria. No accuracy or speed limits are set (R4), the test has not run on the UNO Q or with real speech, and R2 and R3 are not met
Gaps to TRL 4	Set limits for R4; correct the README (R2); keep phone recordings until transcribed (R3); measure both models on the UNO Q with real recorded speech
Gaps beyond TRL 4	TRL 5 needs the relevant environment defined (a candidate: a home bathroom with fan, running water and the device at its mounting distance) and the model tested there

SECTION 9

Conclusions and next steps

The device transcribes its own captures on the device, as designed, with `base.en` by default. The documentation does not say so consistently, phone-recorded notes can leave the home network or be lost, and the first accuracy test cannot tell the two candidate models apart on clean speech while showing `small` to be about three times slower. The transcription stage is a proof of concept, below TRL 4.

Next steps:

- 1. Correct the README:** one model name throughout, without "fine-tuned" or a parameter count it cannot support, and a plain statement of what goes to Groq and when.
- 2. Keep phone recordings until a transcript is saved,** and show a "not transcribed" state instead of placeholder text.
- 3. Ask before sending note text to Groq for titles,** and keep title suggestions off unless the user turns them on.
- 4. Set limits for R4:** a word error rate and a transcription time per second of speech on the UNO Q. This is a product decision, not yet made.
- 5. Measure on the device:** run `analysis/wer_latency_test.py` on the UNO Q with both models, with real recorded speech in quiet and in a bathroom, with more sentences and speakers and several runs, and choose the model against the limits.
- 6. Publish the results as a new version** of this paper.

SECTION 10

References

1. Dharanova, VAHA source repository, <https://github.com/dharanova-code/vaha>. README sections "100% On-Device & Offline Privacy", "AI / ML Model Specifications" and "Performance Metrics"; source files as listed in Appendix A.
2. A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision", *Proceedings of the 40th International Conference on Machine Learning*, 2023, Table 1 (model sizes).

- 3. ISO 16290:2013, *Space systems: Definition of the Technology Readiness Levels (TRLs) and their criteria of assessment*.
- 4. Dharanova, *Device-to-app sync for VAHA: architecture, and the integrity and security of the current build*, v1.0, 2026-10-22.
- 5. Dharanova, *Wake-word and stop-phrase detection for VAHA: dataset, validation accuracy and first field check*, v1.0, 2026-10-21.
- 6. Scripts and data for this paper, in the VAHA source repository (reference 1): `analysis/` and `data/` , added 2026-10-22.

● APPENDICES

Appendices

A. Source locations

All paths are in the VAHA source repository (reference 1). Line numbers refer to the repository as of 2026-08-17.

Table A.1: Files and lines cited

File	Lines	What is there
<code>python/main.py</code>	40	Model input rate, 16 kHz
	56	Model chosen by <code>WHISPER_MODEL_NAME</code> , default <code>base.en</code>
	251–262	<code>clean_audio()</code> : DC offset and peak scaling
	266–276	Model loaded by name into the device's model folder
	277–295	<code>transcribe()</code> : decode options and tidying
	815–818	Resampling to 16 kHz and the call to <code>transcribe()</code>
<code>python/server/routes/captures.py</code>	32–55	Endpoint's model loading; comment at line 34 says <code>small</code>
	58–75	Endpoint's transcription and tidying
	101	<code>POST /captures/transcribe</code>
<code>mobile/src/features/captures/services/LocalWhisperTranscriptionService.ts</code>	16–30	Transcription: device upload, or cloud fallback with no device connection
	43–49	No Groq key: placeholder text returned as a successful result
	55, 66	<code>whisper-large-v3</code> ; Groq transcription request
<code>mobile/src/features/devices/client/DeviceClient.ts</code>	63–78, 106–113	Device connection created on connect, cleared on disconnect
<code>mobile/app/(modals)/new-capture.tsx</code>	20–40	Title request to Groq, <code>llama-3.1-8b-instant</code>

File	Lines	What is there
	145–166	Transcription, deletion of the recording (line 148), title suggestion (158–160), failure message (166)
mobile/app/(tabs)/captures/index.tsx	25, 275–277	Title request for merged notes
README.md	70	"No audio or text ever touches the cloud"
	90, 109	Pipeline description: base.en
	234	Model table: faster-whisper small (277M), "fine-tuned"
	248	Reported CPU use during transcription
	285	Setup: "Groq token for fallback"

B. When each behaviour was introduced

Table B.1: Transcription history, from the repository

Date	Change
2026-07-08	Device transcription pipeline created, with faster-whisper small hard-coded
2026-07-22	Model folder resolved at start-up, with download if the model is missing; the app's transcription service and phone-recorded notes introduced, the recording deleted after transcription
2026-08-06	Groq transcription fallback and the Groq key setting added to the app
2026-08-07	Device default changed from small to base.en , selectable with WHISPER_MODEL_NAME , to speed up transcription
2026-08-08	Groq title suggestions added for new and merged notes
2026-08-17	README model table (small) and pipeline description (base.en) written; last change to the code

C. Performance figure quoted in the product README

Table C.1: Reported transcription performance (README, method unknown)

Metric	Value	Source
CPU use on the UNO Q during transcription	about 30 % at peak	README line 248 (reference 1)

No script, log or note in the repository states how this was measured, with which model, or over how many runs.

D. Revision history

Version	Date	Author	Changes
v0.1	2026-10-19	Ritesh Bonthalakoti	First draft
v0.2	2026-10-22	Ritesh Bonthalakoti	Restructured to the publication template; Groq fallback condition traced; pipeline and fallback diagrams added
v0.3	2026-10-22	Ritesh Bonthalakoti	First accuracy and speed test of both models on a development computer with synthetic speech
v0.4	2026-10-22	Ritesh Bonthalakoti	Test machine and package versions recorded; R4 limits stated as an open product decision; local <code>.env</code> noted in Limitations. Corrected: the README names both models, and its model table was written on 2026-08-17, not edited after the switch; the Groq fallback dates from 2026-08-06; the fallback applies to phone-recorded notes only. Accuracy result restated: one word separates the models; word error rate pooled over all words. Added: Groq title suggestions and the deletion of phone recordings without a transcript (Gaps 2 and 3), requirements R1 to R4, TRL assessment, source locations. Diagrams redrawn by script; code screenshots replaced by code listings; 95 % intervals on word error rate; internal R&D log removed as a source
v1.0	2026-10-22	Ritesh Bonthalakoti	First published version. Test audio moved to <code>analysis/tts_references/</code> ; scripts and data published in the VAHA repository (reference 6). Reviewed by Bharat Meda



We build intelligence into everyday objects.

● HOW TO CITE

Ritesh Bonthalakoti. *On-device speech-to-text for VAHA: the model, where transcription happens, and a first accuracy test.* Dharanova whitepaper, v1.0, 2026.

dharanova.com · info@dharanova.com