جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

أكاديمية كاوست
KAUST ACADEMY

OnKith: What Happens at Home, Stays at Home

Sami Al Irhain
sami.alirhain@gmail.com

Abdullah Alhulays
abdullah.alhulays@gmail.com

Turki Alshuaibi
turki_alshuaibi_999@hotmail.com

Faris Alshehri
farisabdulah007@gmail.com

Yahya Alsharif
yahya.alsharif567@gmail.com

Khalid Alharbi
kh.alharbi32@gmail.com

Mentor: Mojtaba AlShams
mojtaba.alshams@kaust.edu.sa

Repository: `https://github.com/YahyaAlsharif/OnKith`

August 2026

Contents

Abstract **2**

1 Introduction **2**
1.1 Scope and threat boundary . . . 2
1.2 Contributions . . . 3

2 Related Work **3**
2.1 Private voice interfaces and edge speech . . . 3
2.2 PII detection and redaction . . . 3
2.3 Speech errors as a privacy problem . . . 4

3 System Design **4**
3.1 Implemented pipeline . . . 4
3.2 Storage and failure semantics . . . 4
3.3 Implemented assistant boundary . . . 5
3.4 Physical edge target . . . 5

4 Data and Model Development **5**
4.1 Released English corpus . . . 5
4.2 Ontology and hybrid recognition . . . 6
4.3 Model progression and selection . . . 7

5 Experimental Methodology **7**
5.1 Evaluation firewall . . . 7
5.2 Privacy metrics . . . 7
5.3 Benchmark views . . . 8

6 Results and Discussion **8**
6.1 Final text-model performance . . . 8
6.2 Why DeBERTa replaced TinyBERT . . . 8
6.3 FP32 to INT8 quality cost . . . 9
6.4 Cross-device privacy parity . . . 10
6.5 End-to-end audio privacy . . . 10
6.6 Recognizer accuracy–cost tradeoff . . . 11
6.7 Raspberry Pi throughput and memory . . . 12
6.8 Assistant integration and outbound privacy . . . 13

7 Engineering and Error Analysis **14**
7.1 Production decoder defect . . . 14
7.2 ASR and privacy-model interaction . . . 14
7.3 Detector-only spoken-form normalization . . . 15
7.4 Operational boundary findings . . . 15

8 Limitations and Future Work **15**

9 Conclusion **16**

Project Resources and Reproducibility **16**

References **16**

A Final PII Ontology **18**

B Reproducibility Snapshot **19**

Abstract

Always-on assistants create a boundary problem: useful speech may need remote reasoning, but private values should remain on the trusted device. We present OnKith `FullPipeline_V6`, a Raspberry Pi voice assistant whose V4 speech-and-privacy core is followed by a post-mask assistant layer. Audio segmentation, faster-whisper `base.en` recognition, DeBERTa INT8 masking, placeholder allocation, reply validation, rehydration, text-to-speech, and the user interface run locally; only a masked request is sent to Groq. The privacy model retains its established evidence: 0.9527 typed F1 and 0.0027 private-character leakage on held-out text, but 0.6244 F1 and 0.1610 leakage on frozen OOD-final. Canonical V6 device runs give 98.35% predicted-span agreement across 2,000 workstation and Raspberry Pi rows. On 44 audio clips, the Pi obtains 0.0903 number-normalized aggregate WER, 0.8300 private-character recall, 0.1700 leakage, and a mean full-pipeline real-time factor of 0.634; its P90 is 1.109, so not every clip is faster than real time. A live six-case workstation assistant run captured six outbound payloads: none contained a raw transcript or any of the nine values the masker had replaced. Two of 11 gold entities were detector misses and therefore remained in the masked request, separating boundary enforcement from detector recall. All six API calls succeeded, and both replies containing placeholders were rehydrated locally. The Pi repeated the six payload and local-safety probes without an API key; it did not execute a live remote round trip. OnKith therefore demonstrates an implemented masked-text assistant boundary while retaining explicit limits from detection misses, local raw-transcript storage, a pre-mask music exception, and the small evaluation fixtures.

1 Introduction

Smart speakers and voice assistants are convenient precisely because they accept natural, unstructured speech. That same interface can capture unintended activations, names, addresses, account identifiers, credentials, and other context-dependent private material. Prior user studies report both persistent privacy concern and active privacy-seeking behavior around smart speakers [1], [2]; measurements of misactivations show that the boundary between intended and unintended capture is technically imperfect [3]. Sending every transcript to a remote assistant therefore makes privacy depend on the remote service's collection, retention, and interpretation policies.

OnKith explores a testable systems boundary: speech recognition and PII masking run locally, the assistant receives typed placeholders rather than their values, and any placeholder-bearing reply is validated and rehydrated locally before speech or display. The design does not claim to solve all speech privacy. It targets textual entity disclosure, measures failure at multiple layers, and makes exceptions visible. Speaker identity, paralinguistic attributes, contextual secrets outside the ontology, and undetected PII remain outside the guarantee.

1.1 Scope and threat boundary

The intended adversarial event is raw or insufficiently masked textual content leaving the trusted edge device. Local audio and local SQLite storage are considered inside the device boundary, while any network service is outside it. This boundary has two important qualifications. First, the implementation stores the raw transcript locally before the masking update, so compromise or inspection of the device database can expose unmasked content. Second, the optional music-command path can send up to five pre-mask command words to a YouTube lookup worker. It is an explicit egress exception, not part of the generic privacy-preserving route.

V6 implements the downstream route. Questions are sent through the Groq API only after masking; the conversation-scoped placeholder map, history needed for placeholder continuity, reply validation, rehydration, local TTS, and localhost UI remain on the device. Ordinary speech is held locally unless wake/intent routing selects the assistant path. Local commands remain local. The optional music route is still a separate pre-mask exception. The measured scope is split deliberately: workstation evidence includes a live remote round trip, whereas the Pi evidence is a payload audit with no request sent.

1.2 Contributions

Among the open systems and literature reviewed for this work, OnKith is best positioned as an edge systems engineering and evaluation contribution rather than a new neural architecture. Its principal contributions are:

- an implemented V6 voice-assistant pipeline combining local VAD, faster-whisper ASR, detector-only spoken-form normalization, compact DeBERTa masking, conversation-scoped placeholders, masked-only remote requests, local reply rehydration, TTS, and UI;
- a released English PII token-classification corpus with 284,619 examples, 2,088,335 labelled spans, 31 entity types, Saudi-context English identifier formats, spoken password disclosures, and hard negatives;
- a leakage-aware evaluation firewall separating development views from a once-evaluated frozen OOD-final set, plus deterministic text, audio, and assistant fixtures with explicit live-versus-offline scopes;
- a deployment-path case study showing that a tokenizer/decoder defect can resemble quantization damage, and that corrected INT8 quality loss is small; and
- end-to-end failure attribution demonstrating a large gap between clean-text privacy metrics and production audio privacy, together with direct workstation and Raspberry Pi latency and memory evidence; and
- byte-level outbound-payload auditing that distinguishes successful post-mask boundary enforcement from the detector misses that occur before that boundary.

2 Related Work

2.1 Private voice interfaces and edge speech

Local voice stacks reduce routine disclosure by moving capture and inference closer to the user, but the privacy value depends on what actually leaves the device. Benazir and Lin study an edge speech-privacy design that filters sensitive entities before remote inference and report Raspberry Pi deployment, providing a close systems point of comparison [4]. OnKith differs in its evidence emphasis: it audits the actual post-mask request boundary, keeps rehydration local, makes the optional raw command exception explicit, and reports clean-text, OOD, device, audio-path, and assistant evidence separately.

Whisper is an encoder–decoder ASR family trained at scale from weakly supervised audio–text data [5]. OnKith V6 runs the English `base.en` checkpoint through faster-whisper and CTranslate2 on CPU with INT8 compute [6], [7]. Compact streaming ASR continues to advance through systems such as Moonshine v2 and other on-device models [8], [9]. They motivate future substitution studies, but the current evaluated system intentionally retains the V4 Whisper core; the intervening Parakeet configuration remains an archived workstation-only experiment.

2.2 PII detection and redaction

PII detection combines contextual sequence labelling with rules for structured identifiers. Microsoft's Presidio is a representative hybrid system with recognizers and de-identification operators [10]. Recent data and model work emphasizes the need for privacy-specific ontologies rather than assuming that general NER is sufficient. SPY studies synthetic PII data across legal and medical settings [11]; CAPID adds contextual relevance for question-answering systems [12]; and PRvL compares accuracy, leakage, utility, and computational cost for larger redaction models [13]. OnKith

similarly uses hybrid recognition, but focuses on a compact local encoder and deployment-path evaluation.

TinyBERT supplies the distilled encoder used in the project's Model V1 [14]. The final Model V2 uses DeBERTaV3's disentangled attention and replaced-token-detection pre-training [15]; adopting that architecture is an engineering choice, not an architectural novelty claim. The deployed graph runs through ONNX Runtime [16].

2.3 Speech errors as a privacy problem

ASR error and downstream entity error are not interchangeable. Szymański et al. show that spontaneous-speech NER can fail even without conventional word errors and argue that aggregate F1 alone is inadequate [17]. Flechl et al. study ASR trained from de-identified data [18], while Blau et al. improve recognition of personal identifiers through text injection [19]. These findings motivate our separate accounting of entity survival through ASR, privacy-model misses, and postprocessing.

Speech privacy also extends beyond transcript content. VoicePrivacy evaluates speaker anonymization and privacy–utility tradeoffs [20]. OnKith does not anonymize a speaker's voice, and a text-only mask cannot protect vocal identity, accent, health cues, or emotion. This distinction bounds the claims made in the remainder of the report.

3 System Design

3.1 Implemented pipeline

Figure 1 is the V6 architecture. Audio is captured at the edge, segmented with Silero VAD 6.0 at 16 kHz, and decoded locally by faster-whisper `base.en` (`Systran/faster-whisper-base.en`) through CTranslate2 1.2.1 on CPU with INT8 compute, English decoding, punctuation and capitalization, and beam size 5. The speech-and-privacy core is machine-identical in output to V4 for the canonical fixtures; V6 adds the assistant only after immediate final masking.

The final privacy model is `microsoft/deberta-v3-xsmall`, exported to an INT8 ONNX graph. Its 70,706,367 parameters include a 70,682,112-parameter backbone and a 24,255-parameter token-classification head. The accepted graph is 82,278,782 bytes (78.47 MiB), with SHA-256 `bc8d1d7a...048106a8`. Long transcripts use 512-token windows with a 128-token stride. Before detection, a detector-only normalizer rewrites observed spoken address forms such as "local at domain" and "domain slash path" into detector-friendly email or URL syntax. It projects detected spans back to the original transcript; storage and released/displayed transcript text are not replaced by the normalized copy. Character offsets are decoded into candidate spans and combined with cue-gated structured recognizers. Deterministic consolidation resolves overlaps and assigns typed placeholders before any assistant request is built.

3.2 Storage and failure semantics

The runtime first inserts each ASR result into a local SQLite database with a `raw_transcript` field and status `waiting_for_mask`; it later updates the record with the masked transcript. SQLite uses write-ahead logging and normal synchronous mode. This sequence supports local audit and recovery but means that raw transcript persistence is an explicit privacy tradeoff. Disk encryption, retention limits, or storing only masked text would reduce this exposure but are not implemented in the evaluated system.

The masker wrapper is "fail closed" only at the exception boundary: if the masker throws an exception or returns an explicitly unsafe result, the wrapper emits a blocked outcome instead of raw text. An ordinary detector miss is not detectable by that mechanism and can pass through unmasked. For that reason, character leakage and end-to-end misses are more meaningful than an unqualified fail-closed claim.

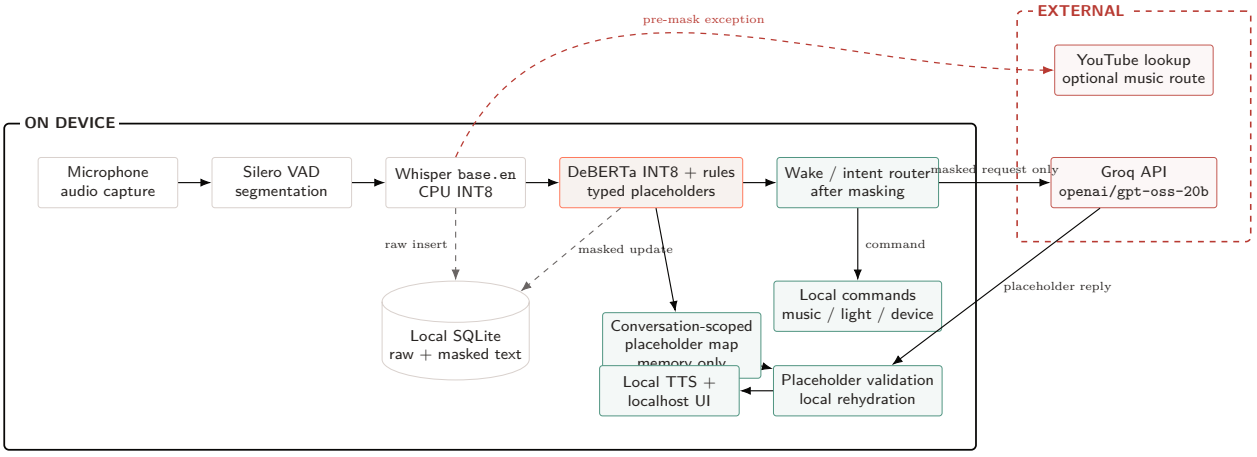


Figure 1: FullPipeline_V6 system and trust boundary. The black enclosure contains local processing, storage, the conversation-scoped placeholder map, reply rehydration, TTS, and UI. Groq receives a post-mask request and returns a placeholder-bearing reply; placeholder values never form part of that request. The red dashed music lookup remains an explicit pre-mask exception.

3.3 Implemented assistant boundary

After masking, V6 routes wake-qualified questions to Groq's API using `openai/gpt-oss-20b`, with a configured 120B fallback [21]. The remote model receives typed placeholders such as `[GIVENNAME_1]`, never the corresponding value. A conversation-scoped in-memory map renumbers and retains placeholder associations across turns; it is not written to SQLite. The reply is checked locally for intact and known placeholders, unresolved forms are kept visible rather than guessed, and known placeholders are rehydrated only on the device. The restored answer is then rendered by the localhost UI and spoken by a local TTS backend. Assistant history and rehydrated replies remain in process memory and are not persisted.

The router separates ordinary speech, local commands, and questions. Ordinary speech is retained in the existing local pipeline. Music, light, and device commands are handled locally, except that a music search can invoke the separately marked YouTube network path. A question alone can reach the assistant worker, and the worker receives only the post-mask text plus local entity records that never become arguments to the Groq client.

3.4 Physical edge target

The deployment target is a Raspberry Pi 5 with 4 GB RAM. Figure 2 shows the physical test arrangement with local cooling and a tabletop microphone. Canonical V6 benchmarks ran the same privacy artifact and Whisper configuration on the workstation and ARM device, then exercised assistant payload construction on both. The Pi had no Groq API key, so its assistant run stopped at the audited request boundary. No controlled thermal-soak study was performed; the results establish feasibility for the recorded fixtures rather than sustained operation.

4 Data and Model Development

4.1 Released English corpus

We release an English PII token-classification dataset containing 284,619 examples and 2,088,335 labelled spans across 31 entity types. It harmonizes four public sources and extends them with Saudi-context English PII coverage and identifier formats, including transliterated names, Saudi IBANs, ZATCA VAT numbers, and Saudi bank names, together with spoken password-disclosure examples and hard negatives. The public release is available on Hugging Face.

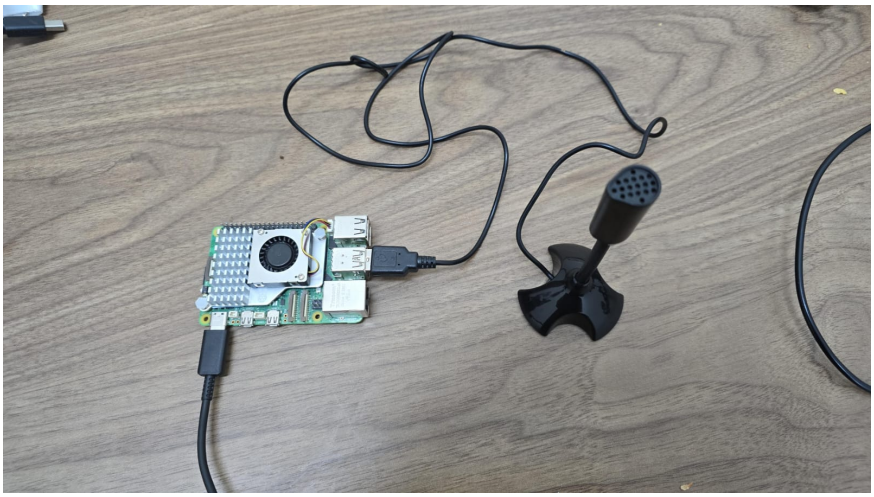


Figure 2: Physical Raspberry Pi 5 deployment used for final validation. The reported measurements cover the deterministic benchmark runs; they do not constitute a controlled thermal-soak experiment.

Table 1: Released dataset composition. Counts are taken from the final public release.

Split	Examples	Labelled spans	Role
Training	217,667	1,621,186	Model fitting
Validation–original	40,909	311,469	Primary IID development view
Validation–supplementary	6,907	22,442	Targeted challenge view
Held-out test	19,136	133,238	Final IID reporting
Total	**284,619**	**2,088,335**	31 entity types

The corpus is English throughout and does not establish Arabic-script or Arabic-language model support. "Saudi-context" describes English examples and local identifier formats, not multilingual capability. Table 1 gives the released split counts. The supplementary validation slice concentrates difficult Saudi-context, spoken-credential, and negative examples, which explains why it is not expected to match the easier original validation distribution.

The additional construction material includes 450 Saudi/MENA-context English rows (81 train, 270 supplementary validation, and 99 test), 2,145 spoken password-disclosure rows, 146 hard negatives, and 43,533 augmented training rows. These targeted additions improve coverage of deployment-relevant forms, but synthetic and template-driven generation can underrepresent linguistic ambiguity and real acoustic behavior.

4.2 Ontology and hybrid recognition

The final ontology contains 31 entity types and 63 BIO labels: one outside label plus begin/inside labels for every entity. It removes legacy categories that were not retained in Model V2 and adds `PASSWORD` with disclosure-specific rules. Appendix A lists the ontology.

Neural context is combined with structured recognizers rather than forcing one mechanism to solve every pattern. Cue-gated recognizers handle forms whose validity depends on both surface structure and nearby language. Spoken digit words are interpreted for contextual recognition. The inherited detector-only spoken-form normalizer remains active. The final rendering stage remains after masking; no postprocessing-amplified identifier occurred in the V6 fixture, but that zero count does not prove that amplification is impossible under other inputs.

4.3 Model progression and selection

Development moved from a BiLSTM baseline to TinyBERT Model V1 and DeBERTa Model V2. The BiLSTM established feasibility but its 0.6051 binary token F1 came from its own validation protocol and is not directly comparable with the later shared split. TinyBERT reached strong IID performance with a compact graph, while DeBERTa provided a modest IID improvement and a much larger robustness improvement. The decisive evidence was OOD behavior, not a small clean-validation gain.

Model V2 was exported to FP32 and then dynamically quantized to INT8. Architecture and checkpoint selection used the training split, validation views, and OOD-dev, but not the held-out test. The later deployment-acceptance gate did include the held-out test alongside the validation and OOD-dev views, requiring every measured typed-F1 drop to remain within 0.02. The corrected decoder met that gate on all recorded views, including a maximum observed drop of 0.0182 on OOD-dev. This separates the true quantization cost from the obsolete decoder artifact in Section 7.1.

5 Experimental Methodology

5.1 Evaluation firewall

The evaluation design separates model selection, deployment acceptance, and final reporting (Figure 3). Training, original validation, supplementary validation, combined validation, and OOD-dev were available during model and engineering decisions; the held-out test did not select the architecture or checkpoint. The test later participated in the FP32-versus-INT8 deployment-acceptance gate. Frozen OOD-final was evaluated once after acceptance. Final V6 evidence then uses three independent fixture families: a deterministic 2,000-row cross-device text subset, a 44-clip audio route, and six assistant cases. Their outcomes are never merged into a single success rate.

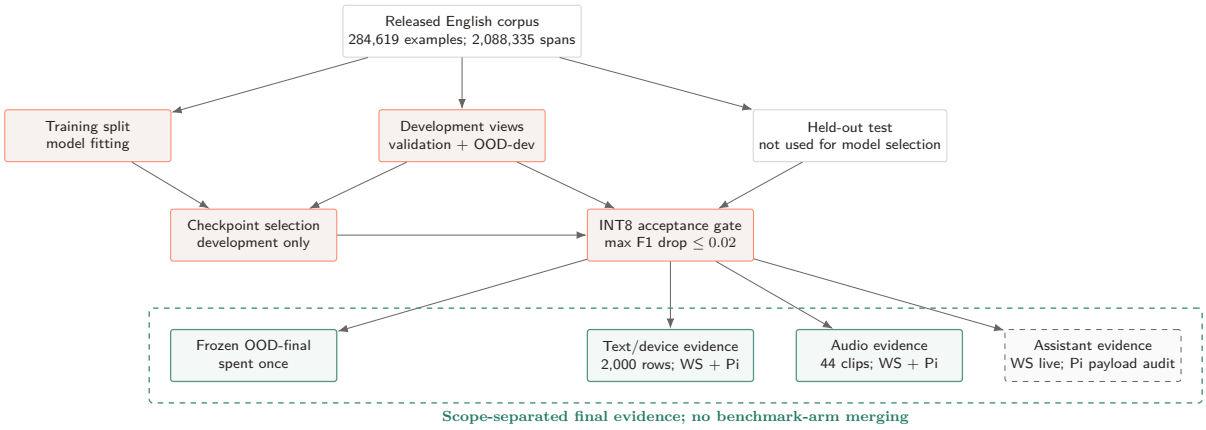


Figure 3: Evaluation firewall. OOD-dev influenced development; frozen OOD-final was spent once. Final text, audio, and assistant evidence uses separate pinned runs. The workstation assistant arm is live; the Raspberry Pi arm is a payload audit only.

5.2 Privacy metrics

Typed precision, recall, and F1 require both span and type correctness. Binary F1 collapses entity types and asks only whether private spans are detected. For a privacy system, character coverage is also important because a partially exposed phone number or account identifier may still be harmful. If C_{private} is the set of reference-private characters and C_{covered} the subset covered by predicted masks, then

$$\text{character recall} = \frac{|C_{\text{covered}}|}{|C_{\text{private}}|}, \qquad \text{leakage} = 1 - \text{character recall}. \tag{1}$$

Overmasking measures non-private characters hidden by the system. Hard-negative false-positive rate reports the fraction of negative challenge rows or clips that receive an incorrect private prediction.

False negatives and false positives have asymmetric costs. A false negative discloses content that the privacy gate exists to protect; a false positive reduces utility but usually does not reveal the protected content. We therefore report F1 together with character leakage, overmasking, and hard-negative behavior rather than optimizing a single aggregate score.

5.3 Benchmark views

Clean and OOD text. The final INT8 graph is measured on the held-out test, original validation, supplementary validation, their combined view, OOD-dev, and frozen OOD-final. TinyBERT and DeBERTa are compared on matched OOD views where the artifacts align.

Quantization. FP32 and INT8 use the same corrected offset decoder. Split-wise typed-F1 deltas determine acceptance; historical values produced by the faulty `word_ids()` path are excluded from quantization conclusions.

Device parity. A fixed 2,000-row subset sampled from the held-out test is run through the workstation and Raspberry Pi 5. We report output agreement, model quality, latency, throughput, and peak resident memory. These are text-row privacy measurements, not live audio latency or an independent generalization benchmark.

Production audio. The 44-clip fixture contains 32 synthetic privacy clips and 12 public LibriSpeech clips with 279.699 seconds of audio. It exercises actual loading, VAD, Whisper decoding, transcript handling, privacy masking, and display rendering. The fixture includes 25 clips containing private entities and seven hard-negative clips. Canonical V6 run `20260823T204339Z_linux-aarch64_audio-v6-whisper_full` contains the Pi WER, privacy, attribution, latency, RTF, stage, and memory evidence in one machine-readable bundle; the paired workstation run is `20260823T142306Z_windows-amd64_audio-v6-whisper_full_workstation`.

Assistant boundary. Six fixed cases exercise wake routing, local masking, request construction, placeholder behavior, and local commands. The workstation run `20260823T142736Z_windows-amd64_e2e-v6-whisper_full_workstation` uses a configured Groq key and measures the live remote leg. Pi run `20260823T204627Z_linux-aarch64_e2e-v6-whisper_full` has no API key: it captures and audits six payloads, but sends none. A masker-replaced value crossing the outbound boundary, a raw transcript crossing it, and a detector miss before masking are reported as three different events.

6 Results and Discussion

6.1 Final text-model performance

Table 2 shows that IID clean-text results are strong but distribution-sensitive. Held-out test leakage is 0.0027, while supplementary-validation leakage rises to 0.0157. OOD-dev leakage is 0.1275 and frozen OOD-final leakage is 0.1610. The widening gap is more consequential than the small difference among IID F1 scores.

Privacy interpretation. A 0.9527 held-out typed F1 does not imply near-perfect deployment privacy. The frozen OOD-final leakage of 0.1610 shows that a materially larger share of private characters remains visible under distribution shift.

6.2 Why DeBERTa replaced TinyBERT

On the shared clean validation view, TinyBERT INT8 reached 0.9502 typed F1 and DeBERTa reached 0.9588. That 0.0086 difference alone would be a weak reason to replace a compact model.

Table 2: Final DeBERTa INT8 text evaluation. HN FP is the hard-negative false-positive rate; the held-out test value is one false positive among four hard-negative rows. OOD-dev influenced development; frozen OOD-final did not.

View	Typed P	Typed R	Typed F1	Binary F1	Char. recall	Leakage	HN FP
Held-out test	0.9456	0.9599	0.9527	0.9638	0.9973	0.0027	0.2500 (1/4)
Validation–original	0.9574	0.9601	0.9588	0.9704	0.9983	0.0017	0.0000
Validation–supplementary	0.8271	0.9514	0.8849	0.8947	0.9843	0.0157	0.0000
Validation–combined	0.9474	0.9596	0.9534	0.9650	0.9969	0.0031	0.0000
OOD-dev	0.6570	0.7225	0.6882	0.7346	0.8725	0.1275	0.2222
Frozen OOD-final	0.5950	0.6568	0.6244	0.6667	0.8390	0.1610	0.2361

The frozen OOD-final comparison is decisive: TinyBERT obtains 0.4649 typed F1, 0.3209 character recall, 0.6791 leakage, and 0.6528 hard-negative FP; DeBERTa obtains 0.6244 typed F1, 0.8390 character recall, 0.1610 leakage, and 0.2361 hard-negative FP. Figure 4 visualizes matched but distinct fixtures: OOD-dev and OOD-final each contain 360 cases with the same category allocation, while using different seeds, templates, domains, numeric values, and manifest hashes.

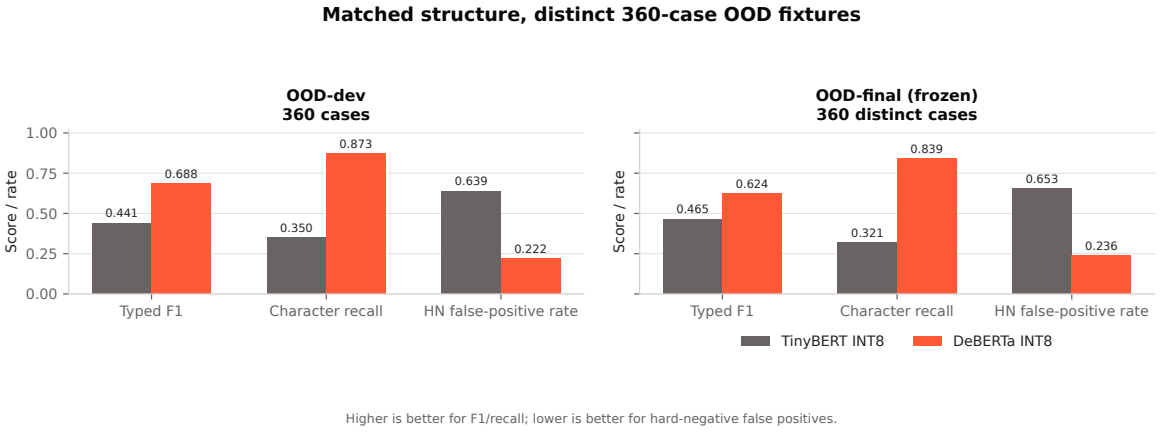


Figure 4: Matched, distinct 360-case OOD fixtures from final artifacts. DeBERTa's selection is justified by private-character coverage and hard-negative behavior under shift, not only by a small IID F1 gain.

The frozen result remains far below IID performance, so DeBERTa is not a solved privacy detector. It is the stronger accepted candidate under the measured conditions. This wording preserves both the model-selection evidence and the remaining generalization risk.

6.3 FP32 to INT8 quality cost

Table 3 reports the corrected shared-decoder comparison. INT8 reduces typed F1 by roughly 0.0022–0.0029 on the four clean views and by 0.0182 on OOD-dev. Every drop remains within the 0.02 acceptance limit. The previously observed result near 0.59 was not a quantization collapse; it came from a production decoding defect that affected span boundaries.

Table 3: FP32 versus INT8 typed F1 after the decoder correction.

View	FP32	INT8	Drop
Held-out test	0.9549	0.9527	0.0022
Validation–original	0.9610	0.9588	0.0023
Validation–supplementary	0.8878	0.8849	0.0029
Validation–combined	0.9558	0.9534	0.0023
OOD-dev	0.7064	0.6882	0.0182

Table 4: Deterministic 2,000-row privacy-model benchmark.

Metric	Workstation	Raspberry Pi 5
Typed F1	0.9452	0.9448
Character recall	0.9950	0.9949
Character leakage	0.0050	0.0051
Mean latency (ms/row)	46.6	91.6
P95 latency (ms/row)	97.2	233.0
Throughput (rows/s)	21.40	10.91
Peak RSS (MiB)	1,671.0	1,928.4

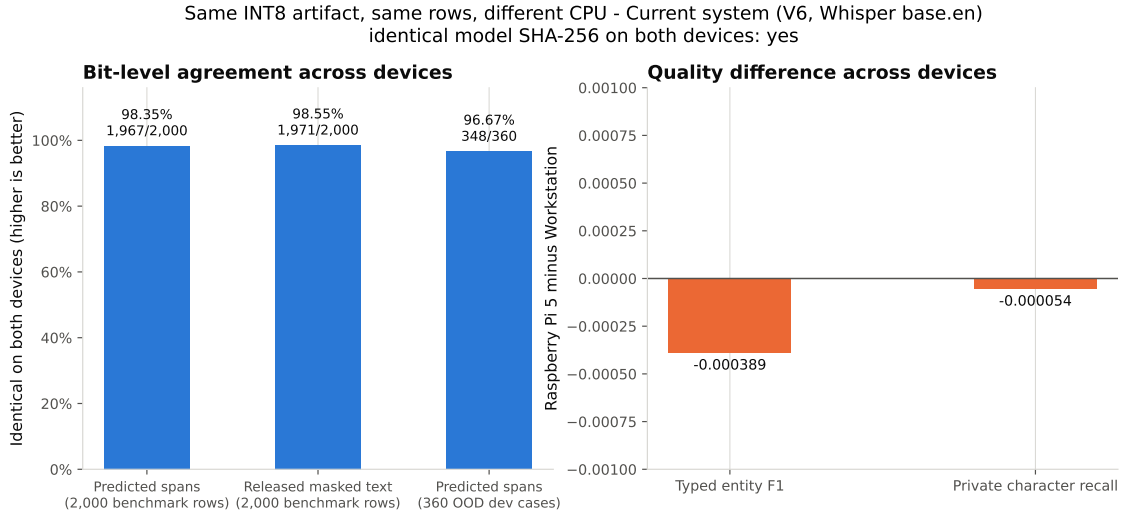


Figure 5: V6 prediction and released-text parity for the fixed text-row fixtures. Full audio latency is measured separately because Whisper dominates that path.

6.4 Cross-device privacy parity

The same deployed artifact behaves consistently on x86 and ARM (Table 4). Predicted spans agree on 1,967 of 2,000 rows (98.35%), released masked text agrees on 1,971 rows (98.55%), and OOD-dev spans agree on 348 of 360 cases (96.67%). Typed F1 differs by only −0.0004. These results establish privacy-model portability for fixed fixtures; they do not supply an independent generalization benchmark. The residual differences are consistent with x86-64 versus aarch64 INT8 kernel arithmetic, not different models: both runs record the same graph hash.

6.5 End-to-end audio privacy

Table 5 contains the production audio result. Across 60 expected private entities, 40 survive STT intact and 33 of those are fully masked. Comparable private-character recall is 0.8300, so 0.1700 of reference-private characters leak through the V6 route. Sixteen of 25 clips containing private information are fully sanitized; one of seven hard-negative clips produces a false positive. All values come from the single canonical Pi audio run.

Raw WER is 0.1841 after lowercasing and punctuation removal. Whisper often emits numerals where the fixture reference uses number words, so raw WER is not comparable with the earlier recognizer's orthography. Applying the same spoken-number folding to reference and hypothesis gives a number-normalized WER of 0.0903. Even that aggregate remains an incomplete privacy proxy because a small error on a name, credential, or number can determine whether the downstream detector recognizes the entity. Figure 6 compares three directly scored privacy routes.

Table 5: Canonical V6 Raspberry Pi production-audio fixture results from run `20260823T204339Z`. The controls in Figure 6 use the same 32 privacy clips.

Metric	Result
Clips / total audio	44 / 279.699 s
Raw aggregate WER	0.1841
Number-normalized aggregate WER	0.0903
Long identifiers exact	3 / 5
Expected private entities	60
Entities preserved by STT	40 / 60
Preserved entities fully masked	33 / 40
Private-character recall	0.8300
Private-character leakage	**0.1700**
Fully sanitized private clips	16 / 25 (64%)
Hard-negative false positives	1 / 7

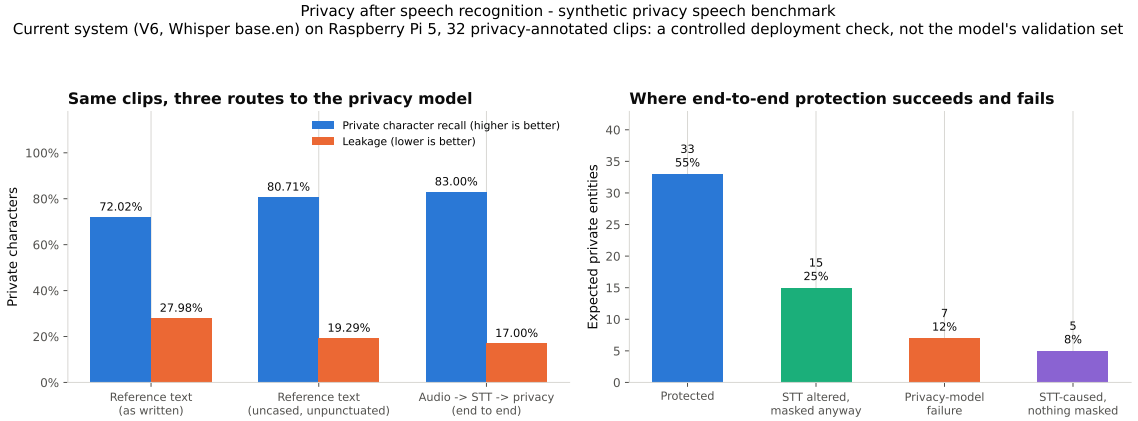


Figure 6: Comparable privacy controls and the V6 audio route. Reference text reaches 0.7202 character recall, uncased/unpunctuated reference text reaches 0.8071, and audio–STT–privacy reaches 0.8300. Detector-only normalization repairs four observed address forms; this does not make STT intrinsically beneficial.

The controls explain why the audio route must not be compared only with held-out IID leakage of 0.0027. On the same fixture, the written-reference control already leaks 0.2798 and the uncased/unpunctuated control leaks 0.1929, showing that fixture composition and preprocessing create substantial difficulty before STT is introduced. The V6 route's 0.1700 leakage is slightly lower because the inherited detector-only normalizer corrected four observed email/URL forms; it is not evidence that recognition errors generally improve privacy.

Failure attribution separates 33 protected entities, 15 entities altered by STT but still masked, seven privacy-stage failures, and five STT-caused failures for which nothing recognizable was masked. Postprocessing amplified zero identifiers in this fixture. This accounting shows that neither STT nor the privacy model alone explains the complete error surface. The production interface between them must be evaluated directly.

6.6 Recognizer accuracy–cost tradeoff

The current system deliberately retains Whisper rather than the more accurate historical Canary recognizer. Table 6 compares only the two canonical Raspberry Pi arms using the transcript-rescored number-normalized WER. Canary is materially more accurate, but its mean STT RTF is above 2; Whisper is more than three times faster by this measure and uses less peak memory. This is an engineering tradeoff, not evidence that Whisper is the better recognizer in every dimension.

Table 6: Canonical Raspberry Pi recognizer tradeoff on the same 44 clips. Canary WER is recomputed from its saved transcripts using the common number-normalized scorer.

Metric	Canary generation	V6 Whisper
Number-normalized aggregate WER	**0.0421**	0.0903
Mean STT RTF	2.022	**0.617**
Mean full-pipeline RTF	2.041	**0.634**
Peak RSS (MiB)	2,030.9	**1,571.2**
Long identifiers exact	**5 / 5**	3 / 5

6.7 Raspberry Pi throughput and memory

The canonical V6 Pi run achieves mean STT RTF 0.617 and mean full-pipeline RTF 0.634. Mean processing is therefore faster than real time for this fixture, although P90 STT and full-pipeline RTF are 1.090 and 1.109, and P95 values are 1.117 and 1.139. Speech recognition accounts for 96.8% of measured processing time and remains the dominant cost. Mean end-to-end processing time is 3.009 s per clip, with P50 2.615 s and P95 5.388 s.

The Pi stage means are 0.4 ms for loading, 70.3 ms for Silero VAD, 2,913.2 ms for Whisper, 0.1 ms for transcript normalization/merge, 24.9 ms for DeBERTa masking, and 0.1 ms for postprocessing. The privacy stage is 0.826% of processing time. Optimizing the detector alone would have little effect on total latency.

Peak resident memory during the complete audio run is 1,571.2 MiB. Loading VAD raises RSS from a 56.6 MiB baseline to 90.4 MiB; loading Whisper reaches 836.6 MiB; adding DeBERTa reaches 1,464.0 MiB, and all loaded models reach 1,465.0 MiB before the sampled-loop peak. This supports feasibility within the Pi's measured 3.95 GiB memory. The 1,928.4 MiB peak in Table 4 belongs to the separate 2,000-row privacy harness. Neither harness establishes thermal, concurrency, or worst-case limits.

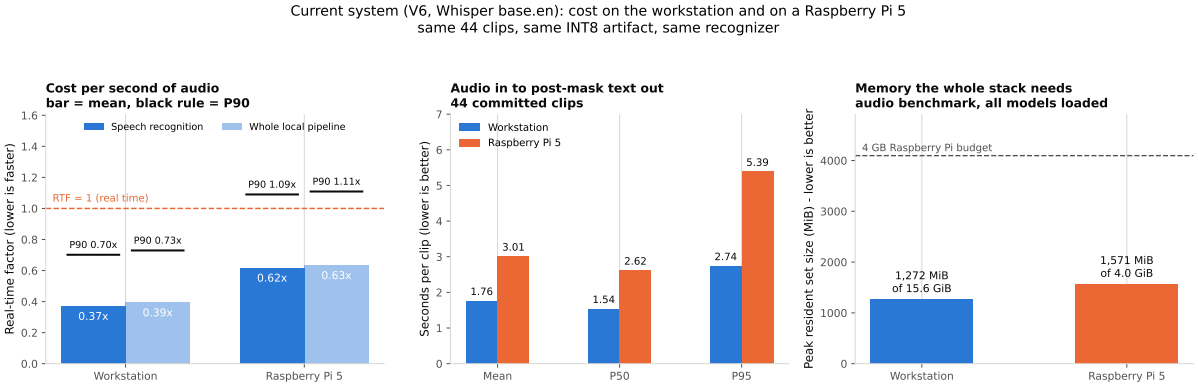


Figure 7: Canonical V6 audio cost on the workstation and Raspberry Pi 5. The black rule is P90; the Pi's shortest clips expose Whisper's fixed per-clip latency floor.

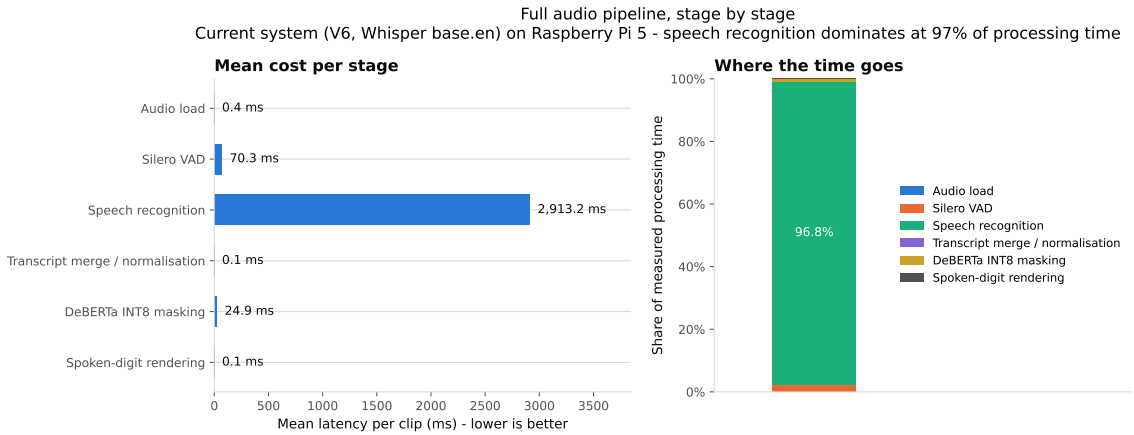


Figure 8: Raspberry Pi 5 V6 stage timing from run `20260823T204339Z`. Whisper decoding consumes 96.8% of measured processing time; privacy masking consumes 0.826%.

6.8 Assistant integration and outbound privacy

Table 7 keeps live and offline evidence distinct. The workstation sent six masked requests and received six replies. Across captured payload bytes, none of nine values that the masker had replaced reappeared, and no raw transcript appeared. This is a zero boundary-failure result, not zero PII leakage: the detector missed two of 11 gold entities (a date and a time), and those unmasked values remained in the request. Both replies that used placeholders contained only intact placeholders and were fully resolved locally. Four deterministic rehydration probes and ten malformed-placeholder probes passed on each device.

The Pi constructed and captured the same six request payloads, with zero masker-replaced values and zero raw transcripts crossing the constructed boundary. Because no Groq key was configured, no request left the Pi and no live reply, network latency, placeholder-preservation, or Pi TTS outcome exists. The local pre-LLM mean is 1,354.8 ms on the workstation and 2,642.1 ms on the Pi; mean live workstation network time is 601.6 ms. Workstation TTS timing was collected for five cases, but speech quality was not evaluated.

Privacy interpretation. The outbound gate prevented values already found by the masker from crossing. It cannot protect a value the detector failed to mark. Boundary enforcement and detector recall must therefore be reported separately.

Table 7: V6 assistant evidence by scope. A dash means the leg was not attempted, not zero latency or failure.

Metric	Workstation live	Pi payload audit
Cases / captured payloads	6 / 6	6 / 6
API requests succeeded	6 / 6	—
Masker-replaced values outbound	0 / 9	0 / 9
Raw transcripts outbound	0 / 6	0 / 6
Detector misses in request	2 / 11	2 / 11
Deterministic rehydration probes	4 / 4	4 / 4
Malformed-placeholder probes	10 / 10	10 / 10
Placeholder replies intact / rehydrated	2 / 2	—
Mean local pre-LLM latency (ms)	1,354.8	2,642.1
Mean network round trip (ms)	601.6	—
Peak RSS (MiB)	1,251.2	1,528.6

7 Engineering and Error Analysis

7.1 Production decoder defect

During deployment validation, FP32 and INT8 production results appeared dramatically worse than notebook evaluation. The behavior initially looked like quantization damage. Investigation instead found that DeBERTa's SentencePiece `word_ids()` grouping could attach punctuation to entity tokens. The production decoder expanded predictions to those groups, harming exact typed spans even when private content remained covered.

The shared decoder was changed to derive grouping from token character offsets rather than `word_ids()`. Regression tests captured punctuation and boundary cases, after which FP32 and INT8 returned to the expected range and the true quantization loss became small (Table 3). This case demonstrates why exporting a graph is not sufficient: evaluation must include the exact production tokenizer, decoder, and postprocessing path.

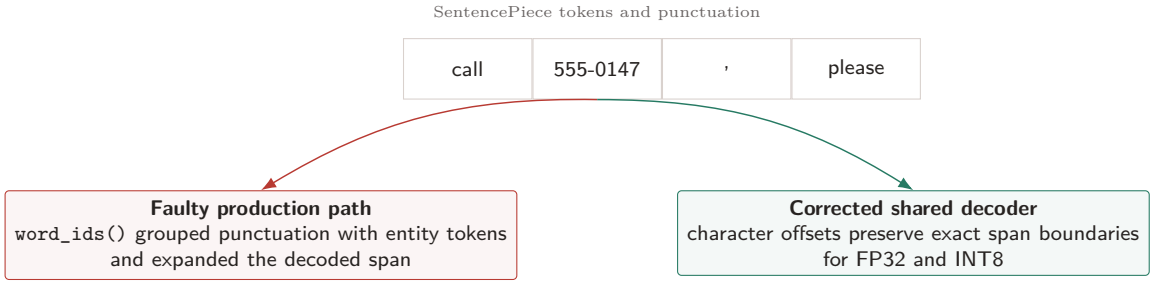


Figure 9: Decoder case study. SentencePiece word grouping expanded entity spans around punctuation. Offset-based decoding restored exact boundaries for the shared FP32/INT8 production path.

7.2 ASR and privacy-model interaction

The 60-entity accounting exposes four qualitatively different outcomes. In 33 cases, the V6 route protects the preserved entity. In 15 additional cases, STT changes the surface form but the detector still finds and masks the altered content. Seven failures occur even though a recognizable private form reaches the privacy stage, indicating model, rule, or partial-boundary limitations. Five failures are attributed to STT changes that leave nothing recognizable for masking. This attribution is supported for the fixed fixture; it should not be generalized to all speech conditions.

WER is an incomplete privacy proxy: context accuracy can hide a corrupted identifier, while structured logic may still mask an altered form. Track private spans across every boundary.

7.3 Detector-only spoken-form normalization

The inherited V4 core contains a narrowly scoped detector input transform for address-like speech forms. It rewrites patterns such as `alex.carter at example.com` to an email form and `example.com slash booking` to a URL form before privacy detection. Four regions were rewritten across the 3,072 texts processed in the fixture's three privacy arms; all four matched the intended form. Mask offsets are projected back to the original transcript, so the normalized text is not substituted into local storage or the released transcript.

The ordering matters scientifically: the transform can improve detector coverage without claiming that the recognizer produced a better privacy representation. The later display renderer still runs after masking. It amplified zero identifiers in the V6 fixture, but future tests should continue to assert privacy invariants across detector input, original masked text, and final rendered output.

7.4 Operational boundary findings

Four implementation details matter to an operator. First, immediate final privacy processing is active; legacy three-second buffer fields are not the runtime behavior. Second, local raw transcripts exist in SQLite before masking. Third, the optional music-command handler is intercepted before privacy and may send up to five command words to a network lookup. Fourth, assistant history and placeholder mappings are process-local and bounded rather than durable; an evicted or unknown placeholder is not guessed. These facts prevent absolute claims that raw text never leaves memory or that every outbound path is masked.

8 Limitations and Future Work

Language and data. The released corpus and final detector are English-only. Saudi-context English identifiers do not establish Arabic-script or Arabic-language support. The corpus combines public, augmented, and synthetic material, which may not reproduce natural disfluency, code-switching, dialect, or sensitive disclosure patterns. Arabic support requires a separately designed ontology, data program, tokenizer/model evaluation, and audio fixture.

Generalization and recall. Frozen OOD-final leakage is 0.1610, and V6 production-audio leakage is 0.1700. The assistant probe itself contains two detector misses among 11 gold entities. None of these results supports a guarantee that all PII is removed. Contextual secrets outside the 31-type ontology can pass through, and exception-level blocking cannot detect an ordinary model miss. More diverse frozen sets, adversarial paraphrases, confidence-aware abstention, and conservative pattern rules should be assessed without contaminating the final view.

Audio propagation. STT errors can destroy the form needed by a downstream detector, while detector-only normalization can repair only the narrow forms it recognizes. Post-mask rendering amplified zero identifiers in the fixed V6 fixture, not a guarantee beyond it. Future work should evaluate identifier-biased decoding, n-best or lattice-aware privacy detection, and direct speech entity methods. Speech recognition remains the first performance priority: Whisper decoding is 96.8% of measured Pi processing time, whereas privacy masking is 0.826%.

Privacy surface. Text redaction does not anonymize speaker identity or protect vocal traits. The system assumes a single speech stream and does not implement diarization. Local SQLite retains raw transcripts, and the music-command exception can create pre-mask raw-text egress. Storage minimization, encryption, retention controls, network consent, and explicit disablement of exceptions should be treated as product requirements before deployment.

Integration and operations. The assistant integration is implemented, but its live evidence is six workstation cases against one provider/model configuration. The Pi run is payload-audit-only and supplies no remote latency, reply, or spoken-output evidence. TTS timing covers five workstation cases and no quality assessment. The evaluation does not include prompt injection, adversarial remote replies, multi-user concurrency, long-duration reliability, controlled thermal soak,

or an independent security audit. Provider policy and network transport remain external trust dependencies even though private values are intended to stay local.

9 Conclusion

OnKith FullPipeline_V6 implements an edge voice assistant in which speech recognition, PII detection, typed placeholder allocation, reply checking, rehydration, TTS, and UI remain local, while Groq receives only the post-mask request. The inherited privacy core retains strong held-out clean-text evidence—0.9527 typed F1 and 0.0027 leakage—but frozen OOD-final leakage is 0.1610. DeBERTa materially outperforms TinyBERT on the matched frozen fixture, and corrected INT8 quality loss remains within the 0.02 gate.

The audio result remains the most important detector limitation. V6 production-audio private-character leakage is 0.1700, with 16 of 25 private-containing clips fully sanitized. Supported failures arise from STT and the privacy stage; detector-only normalization repairs four observed address forms, while postprocessing amplifies none in this fixture. Canary is more accurate on the same clips, but V6 Whisper is substantially faster and lighter on the Pi.

The assistant boundary works for the tested scope: the live workstation run sent six of six requests successfully, and captured outbound bytes contained neither raw transcripts nor any of nine masker-replaced values; two detector misses still remained outbound. Both placeholder-bearing replies were intact and rehydrated locally. The Pi constructed and audited the same six payloads but sent none because no API key was present. Mean Pi full-pipeline RTF is 0.634 and peak audio-harness RSS is 1,571.2 MiB, supporting measured edge feasibility. The result is therefore an implemented and evidenced masked-text boundary, not a universal privacy guarantee.

Project Resources and Reproducibility

Public resources. The public project site is https://onkith.online/, with the team contact page at https://onkith.online/connect. The public source repository is https://github.com/YahyaAlsharif/OnKith. The released English dataset, its card, builder, and Parquet splits are available on the Hugging Face dataset page.

Evidence boundary. The repository provides the implementation and public project materials. Numerical claims in this report are tied to the frozen evaluation JSON, manifests, and benchmark run directories named in the methods and reproducibility snapshot. Some local or gitignored benchmark artifacts are not claimed to be downloadable merely because the repository itself is public.

References

[1] J. Lau, B. Zimmerman, and F. Schaub, "Alexa, are you listening? privacy perceptions, concerns and privacy-seeking behaviors with smart speakers," *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, no. CSCW, 102:1–102:31, 2018. DOI: 10.1145/3274371

[2] N. Malkin, J. Deatrick, A. Tong, P. Wijesekera, S. Egelman, and D. Wagner, "Privacy attitudes of smart speaker users," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 4, pp. 250–271, 2019.

[3] D. J. Dubois, R. Kolcun, A. M. Mandalari, M. T. Paracha, D. Choffnes, and H. Haddadi, "When speakers are all ears: Characterizing misactivations of IoT smart speakers," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 4, pp. 255–276, 2020. DOI: 10.2478/popets-2020-0072

[4] A. Benazir and F. X. Lin, *Privacy-preserving edge speech understanding with tiny foundation models*, arXiv preprint arXiv:2502.01649, 2025. [Online]. Available: https://arxiv.org/abs/2502.01649

[5] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202, PMLR, 2023, pp. 28 492–28 518. [Online]. Available: https://proceedings.mlr.press/v202/radford23a.html

[6] SYSTRAN, *Faster-whisper: Faster whisper transcription with CTranslate2*, Official implementation repository, 2026. [Online]. Available: https://github.com/SYSTRAN/faster-whisper

[7] OpenNMT, *CTranslate2: Efficient inference with transformer models*, Official documentation, 2026. [Online]. Available: https://opennmt.net/CTranslate2/

[8] M. Kudlur, E. King, J. Wang, and P. Warden, *Moonshine v2: Ergodic streaming encoder ASR for latency-critical speech applications*, arXiv preprint arXiv:2602.12241, 2026. [Online]. Available: https://arxiv.org/abs/2602.12241

[9] N. Banfic, D. Fan, K. Vaishnavi, S. Kemp, S. Choi, R. Ren, S. Shaw, and M. Tang, *Pushing the limits of on-device streaming ASR: A compact, high-accuracy english model for low-latency inference*, arXiv preprint arXiv:2604.14493, 2026. [Online]. Available: https://arxiv.org/abs/2604.14493

[10] Microsoft, *Presidio: Data protection and de-identification SDK*, Official documentation, 2026. [Online]. Available: https://microsoft.github.io/presidio/

[11] M. Savkin, T. Ionov, and V. Konovalov, "SPY: Enhancing privacy with synthetic PII detection dataset," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, 2025, pp. 236–246. DOI: 10.18653/v1/2025.naacl-srw.23 [Online]. Available: https://aclanthology.org/2025.naacl-srw.23/

[12] M. Ponomarenko, S. Abedini, M. Shafieinejad, D. B. Emerson, S. Mohapatra, and X. He, "CAPID: Context-aware PII detection for question-answering systems," in *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, 2026, pp. 320–331. DOI: 10.18653/v1/2026.eacl-srw.23 [Online]. Available: https://aclanthology.org/2026.eacl-srw.23/

[13] L. Garza, A. Kotal, A. Piplai, L. Elluri, P. Das, and A. Chadha, *PRvL: Quantifying the capabilities and risks of large language models for PII redaction*, arXiv preprint arXiv:2508.05545, 2025. [Online]. Available: https://arxiv.org/abs/2508.05545

[14] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 4163–4174. DOI: 10.18653/v1/2020.findings-emnlp.372

[15] P. He, J. Gao, and W. Chen, *DeBERTaV3: Improving DeBERTa using ELECTRA-style pretraining with gradient-disentangled embedding sharing*, arXiv preprint arXiv:2111.09543, 2021. [Online]. Available: https://arxiv.org/abs/2111.09543

[16] Microsoft, *ONNX Runtime: Cross-platform machine learning accelerator*, Official documentation, 2026. [Online]. Available: https://onnxruntime.ai/

[17] P. Szymański, L. Augustyniak, M. Morzy, A. Szymczak, K. Surdyk, and P. Żelasko, "Why aren't we NER yet? artifacts of ASR errors in named entity recognition in spontaneous speech transcripts," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 1746–1761. DOI: 10.18653/v1/2023.acl-long.98 [Online]. Available: https://aclanthology.org/2023.acl-long.98/

[18] M. Flechl, S.-C. Yin, J. Park, and P. Skala, "End-to-end speech recognition modeling from de-identified data," in *Proceedings of Interspeech 2022*, 2022, pp. 1382–1386. DOI: 10.21437/Interspeech.2022-10484

[19] Y. Blau, R. Agrawal, L. Madmony, G. Wang, A. Rosenberg, Z. Chen, Z. Gekhman, G. Beryozkin, P. Haghani, and B. Ramabhadran, "Using text injection to improve recognition of

personal identifiers in speech," in *Proceedings of Interspeech 2023*, 2023, pp. 191–195. DOI: 10.21437/Interspeech.2023-2118

[20] N. Tomashenko, X. Miao, P. Champion, S. Meyer, X. Wang, E. Vincent, M. Panariello, N. Evans, J. Yamagishi, and M. Todisco, *The VoicePrivacy 2024 challenge evaluation plan*, arXiv preprint arXiv:2404.02677, 2024. [Online]. Available: https://arxiv.org/abs/2404.02677

[21] Groq, *Groqcloud API reference*, Official documentation, 2026. [Online]. Available: https://console.groq.com/docs/api-reference

A Final PII Ontology

The final model uses the following 31 entity types (63 BIO labels including `O`):

ACCOUNTNUM	AGE	AMOUNT
BANKNAME	BUILDINGNUM	CITY
COUNTRY	CREDITCARDNUMBER	CURRENCY
DATE	DRIVERLICENSENUM	EMAIL
GENDER	GIVENNAME	IDCARDNUM
IPV4	ORGANISATION	PASSPORTNUM
PASSWORD	SALARY	SEX
SOCIALNUM	STREET	SURNAME
TAXNUM	TELEPHONENUM	TIME
TIMEZONE	TITLE	URL
ZIPCODE		

B Reproducibility Snapshot

Table 8: Final implementation snapshot used by this report.

Component	Final setting
Pipeline	`FullPipeline_V6`; V4 local core plus post-mask assistant layer
ASR	faster-whisper `base.en`; `Systran/faster-whisper-base.en`; CTranslate2 1.2.1 CPU INT8; beam 5
VAD	Silero ONNX; 16 kHz audio
Privacy encoder	`microsoft/deberta-v3-xsmall`
Deployment graph	INT8 ONNX; 78.47 MiB; 70,706,367 parameters
Token context	512 tokens with 128-token stride
Privacy output	Detector-only address-form normalization; immediate final masking; 31 entities / 63 BIO labels
Local storage	SQLite WAL; raw transcript inserted before masked update
Assistant route	Groq `openai/gpt-oss-20b`; masked request; local in-memory map, validation, rehydration, TTS, UI
Device	Raspberry Pi 5, 3.95 GiB measured; audio-harness peak RSS 1,571.2 MiB
Final fixtures	Frozen OOD-final; 2,000-row parity subset; 44 audio clips; six assistant cases
V6 text runs	`20260823T142451Z` (WS); `20260823T203840Z` (Pi)
V6 audio runs	`20260823T142306Z` (WS); `20260823T204339Z` (Pi)
V6 assistant runs	`20260823T142736Z` (WS live); `20260823T204627Z` (Pi payload audit)
Public data	284,619 English examples; 2,088,335 labelled spans