Fiche Technique – Pipeline STT Temps Réel Android (VOSK)

Objectif

- Reconnaissance vocale (STT) **offline** sur tablette **Galaxy Tab S10+**.
- Latence totale **≤ 2 s**.
- Contexte : **environnement médical bruité**.
- Pipeline : `MICRO → Prétraitement → VAD → Buffer → STT (Vosk) → NLP → Logique → Alerte`.

Architecture Générale

[Microphone]
│
▼
[AudioRecord 16 kHz PCM mono]
│
▼
[Prétraitement : AEC / AGC / RNNoise]
│
▼
[WebRTC VAD]
├─ Silence → chunk ignoré
└─ Parole → chunk → Buffer circulaire
│
▼
[VOSK STT - Streaming partiel]
│
▼ Texte partiel
[NLP (DistilBERT TFLite Quantifié)]
│
▼
[Logique métier]
│
▼
[UI Compose → Alerte médecin]

Paramètres Audio

Paramètre	Valeur	Justification
Fréquence d'échantillonnage	16 kHz	Suffisant pour la voix humaine
Format	PCM 16 bits mono	Standard faible poids

Paramètre	Valeur	Justification
Taille de chunk	0.5 s (8 000 échantillons)	Bon équilibre latence / performance
Overlap	50 % (0.25 s)	Évite pertes de syllabes
Buffer circulaire	2–3 chunks	Flux stable et continu
Timestamp	Oui (millisecondes)	Journalisation médicale fiable

Modules et Librairies

Module	Outil / Lib	Format	Rôle
Capture audio	`AudioRecord` Android	PCM 16 kHz	Capture en flux continu
Réduction de bruit	RNNoise + WebRTC ANS	JNI / C	Améliore SNR et clarté
Écho / Gain	WebRTC AEC + AGC	JNI / C	Stabilité vocale
VAD	WebRTC VAD	JNI	Détection rapide de parole
STT	**VOSK Android SDK (v0.3.40)**	`.zip`	Reconnaissance vocale offline
NLP	DistilBERT TFLite (int8)	`.tflite`	Détection d'intention / entités
Logique	Kotlin + `Flow`	-	Traitement métier / cohérence
Interface	Jetpack Compose	-	Alerte / feedback utilisateur

Latences Estimées

Étape	Latence approximative
Capture + Prétraitement	200–300 ms
VAD	5–10 ms
STT (Vosk streaming)	400–700 ms
NLP + Logique métier	400–600 ms
Total estimé	**≈ 1.2 – 1.6 s** ☑

Optimisations Techniques

- Thread 1 : capture + prétraitement + VAD
- Thread 2 : STT (streaming partiel avec `recognizer.partialResult()`)
- Thread 3 : NLP + logique métier
- Coroutines Kotlin (`Flow`, `Channel`) → non-bloquant.
- Quantisation TFLite (int8) : latence -40 %.
- **Lazy loading du modèle STT** après détection de parole.
- **Contrôle CRC / SHA-256** des modèles `.zip`.

- Utilisation de `ByteBuffer.allocateDirect()` pour le ring buffer.

🔒 Sécurité & Confidentialité

- Opération 100 % **offline** → confidentialité médicale garantie.
- **Audio sandboxé** : permissions `RECORD_AUDIO` isolées.
- Aucun stockage audio brut.
- **Logs textuels horodatés**, stockés localement, chiffrés si nécessaire.
- Audit interne : traces VAD/STT/NLP stockées et vérifiées.

📊 Évaluation & Qualité

Indicateur	Méthode de calcul	Objectif
SNR post-traitement	RMS(voix)/RMS(bruit)	> 15 dB
Faux positifs VAD	% faux positifs	< 5 %
Latence `VAD_end → STT_out`	Δ timestamps	< 1 s
Charge CPU (Android Profiler)	—	< 40 %

📓 Références Techniques

- VOSK Android SDK
- VOSK Models (FR)
- WebRTC VAD
- RNNoise
- DistilBERT TFLite Models

☑ Résumé

Cette architecture **VOSK + WebRTC + TFLite** garantit :

- Latence < 2 s
- 100 % offline (confidentialité médicale)
- Résilience au bruit
- Modularité (STT/NLP interchangeables)
- Évolutivité vers pipeline multimodal (vision + audio)