Implémentation — OCR ML Kit + voix whisper.rn (offline-first)

Le scan de reçu et la dictée sont désormais **offline-first** : après capture photo (caméra ou galerie), l'utilisateur recadre la zone utile, puis ML Kit lit le ticket localement ; pour la voix, un compte à rebours précède un enregistrement WAV PCM 16 kHz analysé par whisper.rn (`base-q5` / `tiny-q5`). Les modèles ne sont **pas** téléchargés au démarrage de l'app : le download (CDN ou Hugging Face) se déclenche à la **1re dictée** (pendant le compte à rebours / init Whisper), puis reste en cache local. Un parseur local extrait montant, commerçant, date et catégorie (lexicon FR + matching sur les catégories de l'app), calcule un score (montant 40 %, reste 20 % chacun) et, si le score est ≥ au seuil (`EXPO_PUBLIC_RECEIPT_OFFLINE_SCORE_THRESHOLD` , défaut 0,7), garde le résultat sans appeler le cloud. L'API Grok n'intervient qu'en secours lorsque la licence `receipt_online_access` est présente, le réseau est dispo et le score est trop bas ; hors ligne, un simple warning s'affiche sans bloquer. Les images jointes sont compressées en JPEG 0,72 à résolution inchangée, et la saisie manuelle reste toujours disponible.

État : livré côté app (offline d'abord, Grok en secours sous seuil).
i18n : clés `receipt.*` en **FR** (`lang/fr.js`) et **EN** (`lang/en.js`) — crop, countdown, étapes loader, offline, Whisper.

Ops (CDN, env, licence) : `docs/RECEIPT-OFFLINE-OPS.md`
Réf. outils : `docs/OFFLINE-OCR-VOICE.md`
Réf. licence : `docs/LICENSE.md`

---

1. Décisions

Besoin	Solution livrée
Scan offline	`@react-native-ml-kit/text-recognition`
Dictée offline	`whisper.rn` + modèles ggml `base-q5_1` / `tiny-q5_1`
Audio	WAV PCM 16 kHz mono (`receiptPcmWavRecorder` + `receiptWavEncode`)
Scan / dictée online	API Grok (`receipts/scan`, `receipts/dictate`) — **escalade** seulement
Recadrage photo	`ReceiptCropModal` avant OCR (`expo-image-manipulator@~14.0.8`)
Catégorie locale	`receiptCategoryMatch` (mots-clés FR + similarité noms catégories)
Saisie manuelle	Toujours libre

Principe : offline d'abord pour tout le monde. Online = coût → uniquement si `receipt_online_access` + réseau + score offline < seuil.

---

2. Matrice licence × réseau

Licence `receipt_online_access`	Internet	Comportement
Non	—	**Offline seul** (ML Kit / whisper.rn)

Oui	Non	**Offline seul** + **warning** non bloquant
Oui	Oui	Offline d'abord → score ≥ seuil : **garder offline** → score < seuil : **Grok**

Manuel : toujours disponible.

Slug licence

Clé `FEATURES`	Slug API	Rôle
`RECEIPT_ONLINE`	`receipt_online_access`	Autorise le secours Grok uniquement

```
// constants/Features.js
RECEIPT_ONLINE: "receipt_online_access",
```

Permissions OS (≠ licence)

Caméra / photos / micro avant capture. Ordre :

OS → capture → (crop photo) → offline → (éventuel online) → formulaire

---

3. Scoring & seuil (détails)

Implémentation : `helpers/receiptScore.js` + `helpers/receiptOfflineConstants.js`.

Pondération champs (`scoreExpenseFields`)

Champ	Points
`amount` > 0	0.40
`merchant` non vide	0.20
`date` `YYYY-MM-DD`	0.20
`category_id`	0.20

Si confiance OCR dispo (`0..1`) : `score *= 0.7 + 0.3 * ocrConfidence`.
Dictée multi-dépenses : **moyenne** des scores (`scoreExpenses`).

Seuil

Source	Valeur
Env	`EXPO_PUBLIC_RECEIPT_OFFLINE_SCORE_THRESHOLD`
Défaut	`0.7`
Constante runtime	`RECEIPT_OFFLINE_SCORE_THRESHOLD`
Test	`isOfflineScoreEnough(score)`

Score	Si licence + online	Sinon
`>= seuil`	Garder offline — **0** appel Grok	Offline
`< seuil`	Escalade Grok	Garder offline / « À compléter »

Sans réseau : **jamais** d'appel online.

---

4. Warning hors-ligne

Si `hasOnlineReceipt && !isOnline` au démarrage scan / dictée (`warnOfflineIfNeeded` dans `useReceiptScanFlow`) :

- Alert i18n `receipt.offlineWarning*`
- Continuer → analyse offline seule
- Pas de blocage abo

Réseau : `@react-native-community/netinfo` via `helpers/network.js` .

---

5. Flux réel (as-built)

```
caméra | galerie
    │
    ├─ crop (ReceiptCropModal) — cadre libre / image entière / annuler
    │
voix (VoiceRecorderModal)
    │  compte à rebours 2 s → AudioRecord PCM → WAV 16 kHz
    │
    ├─ !isOnline && hasOnlineReceipt → warning
    │
    ▼
offline
  photo  → ML Kit → parseReceiptText → matchCategorySmart → score
  voix   → whisper.rn → repairVoiceTranscript → parseVoiceTranscript → score
    │
    ├─ score >= SEUIL                          → formulaire (path: offline)
    ├─ score < SEUIL && licence && online      → POST receipts/* (Grok)
    └─ sinon                                   → formulaire / « À compléter »
```

Entrées UI : `TransactionTypeModal.js` , `NewExpenseBudgetBtn.js`
Orchestration : `hooks/useReceiptScanFlow.js`
Loader étapes : `ReceiptScanLoader.js`
Debug `__DEV__` : `ReceiptScanDebugPanel.js`

---

6. Modules livrés

6.1 Photo / OCR

Fichier	Rôle

`components/ReceiptCropModal.js`	Recadrage avant OCR (crop natif, pas de resize)
`services/receiptOcr.js`	ML Kit → texte + parse + score
`helpers/receiptOfflineParse.js`	Montant / commerçant / date / heure
`helpers/receiptCategoryMatch.js`	Catégorie par lexicon + fuzzy sur noms WatermelonDB
`helpers/receiptScan.js`	Fichiers scan, persist JPEG **qualité 0.72**, même résolution (`actions: []`)

`expo-image-manipulator` : **`~14.0.8`** (pas `^58` — incompatible Expo 54 / Gradle Publication).

6.2 Voix / STT

Fichier	Rôle
`components/VoiceRecorderModal.js`	UI + countdown 2 s
`helpers/receiptPcmWavRecorder.js`	Capture AudioRecord → chunks PCM
`helpers/receiptWavEncode.js`	En-tête WAV + assemblage
`helpers/receiptVoiceRecording.js`	Flags upload / PCM
`helpers/receiptWhisperModel.js`	Cache local, download CDN→HF→miroir (à la 1re dictée, pas au boot)
`helpers/receiptOfflineConstants.js`	Seuil, prompt Whisper, catalogues modèles
`helpers/receiptVoiceLexicon.js`	Correction homophones FR (`repairVoiceTranscript`)
`helpers/receiptOfflineParse.js`	`parseVoiceTranscript` + split multi-dépenses
`services/receiptStt.js`	Contexte whisper **lazy** + analyse

Patch : `patches/whisper.rn+0.7.4.patch` (exports Metro).

6.3 Online Grok (existant)

`redux/features/receipt/receipt.services.js` — `scanReceipt` / `dictateExpense` uniquement en escalade.
422 / 429 / 503 inchangés.

6.4 UX / i18n

- Étapes animées loader (`receipt.scanStep*` , `receipt.voiceStep*`)
- Countdown dictée (`receipt.voiceCountdown*`)
- Crop (`receipt.crop*`)
- Offline / modèle manquant (`receipt.offline*` , `receipt.whisperModel*`)

---

7. Détails receipt offline

7.1 Parse OCR (`parseReceiptText`)

- Montant : regex FR/FCFA, préférence **max** (total ticket).

- Commerçant : premières lignes « propres » (hors montants / dates).
- Date / heure : formats numériques + relatifs.
- Catégorie : `matchCategorySmart` — si null après OCR brut, le lexicon / similarité noms catégories comble le trou.

7.2 Parse voix (`parseVoiceTranscript`)

1. `repairVoiceTranscript` (lexicon homophones).
2. Découpe multi-dépenses (`splitVoiceItemAmounts`) — ex. « Taxi 2000 et pain 500 ».
3. Par item : montant, objet (`extractVoiceMerchant`), date relative, catégorie.
4. Prompt Whisper initial : `WHISPER_DICTATION_PROMPT` (biaise vocabulaire dépenses FR).

7.3 Modèles Whisper

Quand le téléchargement a lieu

Moment	Comportement
Démarrage app	**Rien** — pas de prefetch au boot (`prefetchWhisperModelOnWifi` existe mais n'est pas branché)
1re ouverture du modal dictée	`warmWhisperContext()` → `ensureWhisperModel()` pendant le compte à rebours
Analyse si pas encore chaud	`analyzeDictationOffline` rappelle `ensureWhisperModel()`
Dictées suivantes	Lecture du cache `documentDirectory/whisper/` uniquement

Sur **réseau** sans cache : télécharge d'abord `tiny-q5` (~~31 Mo) ; sur **Wi-Fi**, enchaîne ou upgrade en fond vers~~ ~~`base-q5`~~ (57 Mo).
Les ~140 Mo évoqués ailleurs = ancien `ggml-base.bin` **non quantifié** — **non** téléchargé par défaut (ce n'est pas 31+57).

Priorité	Fichier	~Taille	Rôle
Cache local	meilleur déjà téléchargé	—	Utilisé en premier
Qualité	`ggml-base-q5_1.bin`	~57 Mo	Cible (Wi-Fi / fond)
Secours	`ggml-tiny-q5_1.bin`	~31 Mo	1re dictée / data limitée
Legacy	`ggml-base.bin` / `ggml-tiny.bin`	—	Réutilisation cache ancien seulement

Ordre URL : **CDN** (`EXPO_PUBLIC_WHISPER_MODEL_BASE_URL`) → Hugging Face → `hf-mirror.com` .
Hors ligne + aucun cache → `whisper_model_missing` → saisie manuelle.
Pas de modèle full ~140 Mo embarqué dans l'APK.

7.4 Audio

- Format fixe : **PCM 16-bit LE, mono, 16 kHz**, enveloppé WAV.
- Évite m4a / MediaRecorder (incompatible whisper.rn natif).
- URI normalisée `file://` : `helpers/receiptNativeUri.js` .

7.5 Crop photo

Après caméra/galerie, **avant** ML Kit :

- Cadre ~92 % initial, drag + poignée coin.
- Confirm → `manipulateAsync([{ crop }])` (résolution de la zone gardée).
- « Image entière » → URI d'origine.
- Annuler → stoppe le lot (scans déjà faits conservés).

7.6 Persistance pièce jointe

À l'enregistrement : JPEG compressé **0.72**, **sans resize** (`RECEIPT_SAVE_JPEG_QUALITY`).

7.7 Variables env (rappel)

Variable	Défaut	Rôle
`EXPO_PUBLIC_RECEIPT_OFFLINE_SCORE_THRESHOLD`	`0.7`	Escalade Grok sous ce score
`EXPO_PUBLIC_WHISPER_MODEL_BASE_URL`	vide	CDN modèles ; sinon HF

Détail ops / checklist CDN : `docs/RECEIPT-OFFLINE-OPS.md` .

---

8. Travaux hors app (toujours à faire)

Équipe	Travail
API / billing	Attacher `receipt_online_access` aux offres Grok
Infra	CDN Whisper (`ggml-tiny-q5_1.bin`, `ggml-base-q5_1.bin`)
QA	Matrice §2 + checklist §11

---

9. Critères de done (app)

- [x] Offline d'abord pour tout user
- [x] `score >= SEUIL` → pas d'appel Grok
- [x] `score < SEUIL` + licence + online → escalade Grok
- [x] Sans internet → offline seul
- [x] Licence cloud + offline → warning non bloquant
- [x] Sans `receipt_online_access` → jamais Grok
- [x] Crop après photo
- [x] PCM WAV + whisper base-q5 / tiny-q5
- [x] Catégorie locale (keywords)
- [x] Compression JPEG à la sauvegarde (même résolution)
- [ ] CDN prod + feature licence branchée en billing (ops)

---

10. Notes iOS / Android

- Enregistrement voix : WAV PCM 16 kHz mono — pas MediaRecorder/m4a.
- URIs : `file://` (`receiptNativeUri.js`).
- `useFrameworks: "static"` dans `app.json` (pods ML Kit / whisper).

- Patch whisper.rn via `patch-package`.
- Android : `expo-image-manipulator@~14.0.8` obligatoire (versions 58.x cassent `settings.gradle` Publication).
- Build device :
  - Android : `npx expo run:android --device`
  - iOS : `npx expo prebuild --platform ios` puis `npx expo run:ios`

---

11. QA manuelle (device)

Rebuild obligatoire après deps natives.

#	Setup	Attendu
1	Sans `receipt_online_access`	Offline seul, 0 appel API receipts
2	Avec permission, airplane ON	Warning puis offline
3	Licence + online, ticket lisible	Offline, pas de Grok
4	Licence + online, photo floue	Escalade `receipts/scan`
5	Après photo	Écran crop → zone / entière / annuler
6	Dictée FR, modèle en cache	Transcript + 1..n cartes dépenses
7	Dictée offline sans modèle	Message modèle manquant
8	Multi-photos	Crop puis analyse une à une
9	Manuel	Inchangé