

# Cadrage Audit – Codex (Standard)

## Objectif

Définir un cadre clair et systématique pour chaque **audit complet réalisé par Codex**. Garantir que l'Owner reçoive une vision consolidée et actionnable, sans dispersion documentaire.

---

## Périmètre de l'audit

Chaque audit Codex doit couvrir :

1. **Arborescence** du dépôt (structure, modularité, solidité).
  2. **Code** : sécurité & bonnes pratiques (lint, types, deps, vulnérabilités).
  3. **Branchements** : routes ↔ API ↔ UI (règle « branché ou rien »).
  4. **Liste des éléments OK** (conformes).
  5. **Liste des éléments KO** (bloquants/majeurs/mineurs).
  6. **Recommandations courtes** (diffs de principe, non commitables).
- 

## Entrées requises

- Vision produit (dernière version).
  - Plan directeur.
  - KS (Invest).
  - Document récapitulatif de référence projet (fourni par Owner).
- 

## Sorties attendues

Chaque audit doit produire **deux niveaux de livrables** :

### 1. Livrables détaillés (techniques)

- `arbo_snapshot.txt` (arbre du repo)
- `arbo_metrics.json` (métriques structurelles)
- `arbo_review.md` (analyse arbo)
- `code_security.md` (constats sécurité & qualité)

- `security_findings.json` (findings détaillés)
- `wiring_report.md` (flux câblés vs manquants)
- `wiring_matrix.csv` (matrice branchements)
- `ok_ko_table.md` (OK/KO par domaine)
- `kill-list_KO.txt` (KO bloquants)
- `patches_de_principe/*.diff` (propositions, non commitables)

## 2. Synthèse centrale (obligatoire)

Un fichier `audit_synthese.md` doit être ajouté à **chaque audit** :

- Résumé clair des constats (OK/KO/manquants/risques).
- Double formulation pour chaque point :
  - **Technique (brut)**
  - **Clair non-tech** (accessible à l'Owner).
- Liens directs vers les fichiers détaillés correspondants.

👉 **Important** : la demande de synthèse doit être rappelée en fin d'audit, pour s'assurer que Codex livre bien le fichier `audit_synthese.md` en complément des rapports détaillés.

---

## ✅ Critères d'acceptation

- Tous les fichiers listés en sorties détaillées doivent exister.
  - La synthèse (`audit_synthese.md`) est rédigée en français.
  - Chaque constat de la synthèse est exprimé en 2 lignes maximum (tech + clair non-tech).
  - La synthèse renvoie vers les fichiers détaillés.
  - La demande de synthèse est explicitement formulée en clôture de l'audit.
- 

## 📌 Exemple de sortie — `audit_synthese.md`

# Synthèse Audit — R2 (exemple)

## ✅ OK

- [Arbo] Structure modules claire (apps/ vs packages/) → voir `arbo_review.md`
- [Code] ESLint sans erreur bloquante → voir `code_security.md`

## ⚠️ KO

- [Câblage] Login → API non branché (`app/login/page.tsx:42`)
- Clair : l'écran de login ne parle pas encore avec l'API.
- Voir `wiring_report.md`

## 🚧 À améliorer

- [Logs] Absence de logs JSON uniformes sur `/projects`

→ Clair : certaines actions ne sont pas encore tracées.

→ Voir code\_security.md

---



## Contraintes

- **Lecture seule** : l'audit ne modifie pas le dépôt.
  - **Neutralité** : si non démontrable → marquer *non vérifiable*.
  - **Sobriété** : constats courts, actionnables.
  - **Rappel** : la synthèse centrale est obligatoire et doit être demandée explicitement en fin d'audit.
- 



## Résumé Mantra

**Détaillé + Synthèse** → L'Owner lit la synthèse, Codex garde les détails.