

# CAHIER DES CHARGES — AIGE-APRON-SMART v2.0

Système intelligent de gestion des 18 postes de stationnement + Flight Radar Live

## 1. Contexte & justification

L'Aéroport International Gnassingbé Eyadéma fonctionne actuellement avec :

- attribution **manuelle** des postes de stationnement,
- absence d'historisation et de traçabilité,
- coordination limitée entre ATC, Ops, Handling,
- absence de visualisation du **trafic aérien en temps réel** (vols en approche, départs, retards...).

### Objectif :

Moderniser, digitaliser et automatiser le processus via une plateforme unique qui :

- attribue intelligemment les 18 postes de stationnement,
- affiche l'état du tarmac en temps réel,
- montre le trafic aérien live autour de Lomé grâce à un module **Flight Radar / ADS-B** intégré.

## 2. Objectif global du système

Créer une plateforme appelée **AIGE-APRON-SMART**, comprenant :

1. **Gestion intelligente des 18 postes** de stationnement.
2. **Attribution automatique et manuelle** des postes.
3. **Visualisation en temps réel** (statut des postes).
4. **Historique complet + export PDF/CSV**.
5. **Module Flight Radar** intégré pour afficher :
  - les vols dans un rayon de 200 km autour de Lomé,
  - les vols en approche (ETA),
  - les vols au départ (ETD),
  - les informations : immatriculation, altitude, vitesse, compagnie, provenance/destination.

Ce module permettra à l'aéroport :

- d'anticiper les arrivées,
- d'améliorer l'attribution des postes,
- d'augmenter la sécurité et la rapidité opérationnelle.

## 3. Périmètre fonctionnel (Scope)

### Inclus

- Système complet d'attribution des 18 postes.
- Interface Web (dashboard + radar).
- Règles d'allocation automatiques.
- Module Flight Radar intégré (API ADS-B ou OpenSky).
- Carte radar live (Google Maps / Leaflet).
- Historique, recherche, export.
- API REST (vols, postes, assignations, radar).
- Authentification légère (Admin / Ops / ATC).
- Dockerisation optionnelle.

### Exclus (version prototype)

- Intégration officielle avec systèmes ATC (AMHS/AFTN).
- Interfaçage direct avec systèmes de compagnies aériennes.
- Capteurs IoT physiques sur le tarmac.
- Application mobile.

## 4. Parties prenantes

- **AIGE – Direction des opérations** (client principal)
- **ATC sol (Ground Control)** – utilisateurs directs
- **Handling (ASKY, Ethiopian, Air France, etc.)**
- **Sécurité piste**
- **ANAC** – audit et conformité
- **Équipe projet AIGE-APRON-SMART**

# 5. Exigences fonctionnelles (RF)

## RF-01 — Gestion des 18 postes

- RF-01.1 : Créer 18 postes : ST-01 à ST-18.
- RF-01.2 : Attributs : type, max\_code, statut, contraintes.
- RF-01.3 : Modification manuelle possible.

## RF-02 — Enregistrement des vols

- Enregistrer : callsign, aircraft\_type, nature, horaires, etc.
- Valider données et stocker en base.

## RF-03 — Allocation automatique

1. Vérifier compatibilité avion / poste
2. Vérifier disponibilité horaire
3. Priorité selon nature (PAX, CARGO, VIP, MIL)
4. Scoring basé sur :
  - proximité porte,
  - type de poste,
  - historique.

Résultat : proposer automatiquement ST-xx ou null.

## RF-04 — Dashboard tarmac

- Visualisation des **18 postes** (zones colorées).
- Mise à jour toutes les 3 secondes.
- Détails d'un poste au clic.

## RF-05 — Module Flight Radar Live (NOUVEAU)

### RF-05.1 : Connexion API Flight Radar

Le système se connecte à l'une des sources :

- **OpenSky API (gratuit)**
- **ADS-B Exchange**

- **AirSense local** (si récepteur ADS-B installé)

#### **RF-05.2 : Affichage des vols live**

- Carte dynamique (Leaflet / Mapbox / Google Maps).
- Avions représentés par icônes + direction + vitesse.
- Mise à jour toutes les 5-10 secondes.

#### **RF-05.3 : Informations affichées pour chaque avion**

- callsign
- immatriculation
- altitude
- vitesse
- compagnie
- provenance
- destination
- ETA / ETD

#### **RF-05.4 : Focus sur AIGE**

- Afficher uniquement les vols dans un rayon configurable (ex : 200 km).
- Liste des **vols en approche**.
- Liste des **départs**.

#### **RF-05.5 : Lien entre Flight Radar et allocation**

Fonction essentielle :

Lorsqu'un vol entre dans un rayon défini (ex : 60 km), il apparaît automatiquement dans la liste des vols entrants de l'aéroport.

### **RF-06 — Historique / audit**

- Historique complet des assignations.
- Recherches filtrées.
- Export CSV & PDF.
- Journal des actions opérateurs.

## **RF-07 — API interne**

Endpoints :

/spots  
/flights  
/assignments  
/radar/live  
/radar/approach  
/radar/departure

## **RF-08 — Authentification**

- Login minimal : Admin / Ops / ATC.
- Journaux des actions.

# **6. Exigences Non Fonctionnelles (RNF)**

## **RNF-01 — Performance**

- Radar live mise à jour  $\leq$  5 s.
- Attribution automatique  $\leq$  1 s.

## **RNF-02 — Sécurité**

- Accès local réseau AIGE.
- HTTPS si production.
- API-key pour accès radar externe.

## **RNF-03 — Portabilité**

- Fonctionne sous Linux/Windows.
- Docker disponible.

## **RNF-04 — Extensibilité**

- Prévu pour ajouter :
  - notifications,
  - IA prédictive,
  - intégration AMHS.

# 7. Architecture générale

## Backend :

- FastAPI
- Database SQLite / PostgreSQL (prod)
- Module Flight Radar (Python)

## Frontend :

- Dashboard HTML / React
- Carte radar (LeafletJS + OpenSky API)

## Flux radar :

1. API OpenSky → backend → JSON réduit
2. Backend filtre vols autour de l'AIGE
3. Frontend affiche sur carte + liste

# 8. Modèle de données

## Nouvelles tables Flight Radar :

### Table flights\_live

| id | icao24 | callsign | altitude | speed | lat | lon | heading | origin | destination | last\_seen |

### Table radar\_log (optionnel)

| id | timestamp | raw\_data |

# 9. Cas d'usage du module radar

## Cas 1 : Vol entrant détecté automatiquement

- OpenSky détecte un avion Asky en approche.
- L'application l'affiche automatiquement dans “Arrivées imminentes”.
- L'opérateur peut préparer l'attribution.

## Cas 2 : Alerte de saturation

- Si >2 vols en approche simultanée → alerte pour ATC.

## Cas 3 : Aide à l'allocation

L'algorithme peut anticiper en fonction de :

- ETA,
  - type avion,
  - poste adéquat.
- 

# 10. Interface utilisateur (UI/UX)

## Écran 1 — Dashboard Tarmac

- Vue 18 postes
- Liste vols enregistrés
- État (occupé / réservé / libre)
- Graphiques légers (temps d'occupation, etc.)

## Écran 2 — Flight Radar Live

- Carte Live
- Icônes avions (rotation = heading)
- Sidebar : Arrivées / Départs / Zone 60 km
- Fiche avion au clic

## Écran 3 — Historique

- Table filtrable
  - Export CSV / PDF
- 

# 11. Tests & Validation

## Tests fonctionnels

- Test attribution automatique.
- Test de surcharge (plusieurs vols).
- Test affichage radar.
- Test arrêts / reprises du flux radar.

## Tests d'intégration

- Radar API → backend → UI
  - Flights → allocation → visualisation
- 

# 12. Déploiement

## Environnement

- Serveur local / PC
- Python 3.10+
- Docker optionnel

## Procédure

- Installer dépendances backend
  - Configurer API OpenSky
  - Lancer frontend
  - Vérifier radar live
  - Former opérateurs
- 

# 13. Risques & solutions

Risque	Solution
API Radar instable	Mise en cache 10 sec
Perte connexion internet	Mode offline (sans radar)
Erreur d'allocation	Modification manuelle
Charge élevé radar	Filtrage rayon 200 km

---

# 14. Livrables

- Cahier de charges (celui-ci)
- Diagrammes d'architecture
- Backend (FastAPI) complet
- Frontend (dashboard + radar)

- Scripts d'installation
  - Export CSV/PDF
  - Manuel opérateur
  - Pitch deck
- 

## 15. Conclusion

AIGE-APRON-SMART v2.0 devient un **système de nouvelle génération**, combinant :

- **gestion intelligente des postes,**
- **visualisation temps réel des avions,**
- **support décisionnel pour ATC/Ops,**
- **anticipation des arrivées,**
- **réduction des risques et retards,**
- **modernisation du processus aéroportuaire.**