

Cahier des Charges du Projet de Loterie Décentralisée sur Blockchain Solana

1. Objectif du projet

Le présent projet a pour but de créer une plateforme de loterie entièrement décentralisée et transparente, basée sur la blockchain Solana. Les tirages seront automatiques et se baseront sur la possession du token \$BALL, un token SPL natif à l'écosystème. L'objectif est de permettre aux utilisateurs de participer passivement à des tirages horaires et journaliers, tout en garantissant la sécurité, la traçabilité et la décentralisation des opérations.

2. Fonctionnalités principales

2.1 Frontend (Interface utilisateur)

- **Tableau de bord interactif :**
 - Compte à rebours vers le prochain tirage horaire
 - Compte à rebours vers le prochain tirage journalier
 - Estimations des jackpots (USD & SOL)
 - Affichage des 3 derniers gagnants (adresse, gains, slot Solana, ticket)
 - Nombre de participants et de tickets
 - Bouton pour vérifier la transaction sur la blockchain
- **Design responsive :**
 - **Compatible mobile et desktop**
 - Intégration portefeuilles Solana (Phantom, Solflare)

2.2 Backend / Smart Contract

2.2.1 Token SPL \$BALL

- Création du token SPL \$BALL avec un supply limité
- Utilisation exclusive pour la participation à la loterie

2.2.2 Mécanisme de participation

- 1 ticket = 10 000 \$BALL détenus
- Participation automatique sans inscription si les tokens sont en portefeuille

2.2.3 Génération des tirages

- **Hourly PowerBall** : tirage toutes les heures (10% des transactions)
- **Mega Daily PowerBall** : tirage toutes les 24h (5% des transactions)

2.2.4 Tirages et gains

- Tirage via VRF (Chainlink ou Switchboard)
- Sélection aléatoire d'un wallet éligible
- Gain converti en SOL et transféré automatiquement

2.2.5 Conversion et stockage

- 10% des achats \$BALL => Jackpot Heure
- 5% des achats \$BALL => Jackpot Jour
- Conversion automatique en SOL et stockage dans le contrat



3. Architecture technique

3.1 Smart Contracts

- **Langage** : Rust (Anchor Framework)
- **VRF** : Intégration Chainlink ou Switchboard pour tirages aléatoires

3.2 Backend

- **Langage** : Python (Django REST Framework)
- **Fonctions** :

- Lecture holdings des wallets
- Déclenchement des tirages off-chain (si nécessaire)
- Calcul des jackpots
- Enregistrement des gagnants et tirages

3.3 Base de données

- **SGBD** : PostgreSQL
- **Données stockées** :
 - Utilisateurs et adresses wallet
 - Historique des tirages
 - Montants des jackpots et transactions

3.4 Frontend

- **Framework** : Next.js (React)
- **Temps réel** : WebSocket ou Polling
- **Wallets supportés** : Phantom, Solflare via Solana Wallet Adapter

3.5 Interface Admin

- Accès via Django Admin (Superuser)
- Fonctions :
 - Visualisation des utilisateurs et wallets
 - Historique complet des tirages
 - Contrôle manuel exceptionnel des tirages (mode debug)
 - Export de rapports

Module	Responsable	Techno	Durée estimée
Smart contract	Dev Solana	Rust	5-7 jours
Intégration VRF	Dev Solana	Chainlink/Switchboard	2 jours
Backend API REST	Dev Backend	Django	5-7 jours
Gestion des jackpots	Dev Backend	Python	2-3 jours
Interface utilisateur	Dev Frontend	React/Next	5 jours
Intégration wallet	Dev Frontend	Solana Adapter	1-2 jours
Interface Admin	Dev Backend	Django Admin	1 jour
Test global & sécurité	Fullstack	-	3 jours

4.Sécurité & Transparence

- **Traçabilité blockchain** : tous les tirages et transferts seront consultables via Solscan
- **VRF** : assurera une randomisation transparente
- **Audit** : le contrat smart sera auditable par un organisme tiers (ex : Certik)
- **Protection Backend** : API REST sécurisée avec authentification JWT + rate limiting

5. Répartition des tâches et planning

5.1 Tâches

5.2 Planning prévisionnel

Semaine	Objectifs
Semaine 1	Smart contract + base backend
Semaine 2	Intégration VRF + frontend de base
Semaine 3	Connexions complètes et tests
Semaine 4	Tests finaux + sécurité + déploiement

5.3 Budget estimatif

Poste	Coût (€)
Smart contract & VRF	150 - 200
Backend & Admin	100 - 150
Frontend	100 - 150
Intégration blockchain	50 - 100
Test & Sécurité	50 - 100
Total	350 - 750 €

6. Livrables

- Smart contract déployé sur Solana
- API REST sécurisée
- Dashboard frontend complet
- Interface administrateur Django
- Documentation technique (PDF/Notion)
- Rapport d'audit (si budget validé)