

Projet Rethagon

Leo Botelho – Alyra Finance décentralisée

Promotion Douglas Adams

Soutenance du 18 décembre 2025

The logo for RETHAGON features the word "RETHAGON" in a bold, white, sans-serif font. The letters are positioned on an orange rectangular background. A white, stylized hexagonal shape is superimposed over the right side of the text, with the letters "GON" partially enclosed within it.

RETHAGON

1 - Qui sommes-nous ?

2 - Contexte du projet

3 - Cadre juridique

4 - État de l'art

5 - Étude d'opportunité

6 - Notre organisation

7 - Démonstration

8 - Notre roadmap

1. Qui sommes-nous ?

Nephelim – *Smart Contract & dApp Developer*

"Nephelim Cohen, développeuse et fan absolue de Battle Rap."

Achaymaa – *Product Owner*

"Achaymaa BEL JEBBAR, 35 ans. Actuellement Lead Data Scientist en banque, je suis passionnée par la DeFi. J'ai la conviction inébranlable qu'elle finira par remplacer la TradFi."

Jean-Pierre – *Business Strategist*

"Je m'appelle Jean-Pierre, j'ai 32 ans. Je travaille dans le secteur de la maintenance des machines et je suis passionné par la crypto et la blockchain. J'ai décidé de me former afin de participer à la croissance de cet écosystème qui me passionne."

Nathalie – *Legal Strategist*

"Nathalie Bez, 60 ans, touche à tout ayant eut mille vies, d'infographiste à manager au Parc Astérix, ambulancière, créatrice d'objets qui dépolluent l'environnement, conductrice de bus, s'intéresse aux nouvelles innovations qui lui semblent ouvrir à d'autres possibilités."

Leo – *Quantitative System Architect*

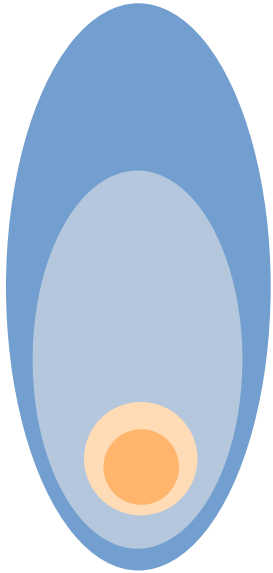
"Leo Botelho, 23 ans. Je travaille dans le secteur de l'informatique et suis passionné par la DeFi et la blockchain. J'ai décidé de me former afin faire de cette passion mon métier et participer activement à la croissance de son écosystème."

2. Contexte du projet - le constat

Bien que la discipline du battle rap soit de plus en plus populaire, les artistes se représentant ne gagnent, de loin, pas suffisamment pour en vivre.

The logo for Rethagon is located in the top right corner. It consists of the word "RETHAGON" in white, uppercase, sans-serif font, positioned to the left of a white hexagonal outline. The entire logo is set against a solid orange rectangular background.

RETHAGON



Solution proposée : **Rethagon**

Une plateforme Web3 qui professionnalise l'économie du Battle Rap en permettant aux artistes de monétiser directement leur performance et leur réputation grâce à une billetterie NFT et un marché de prédiction international.



c'est quoi ?

3. Cadre juridique

Token AURA : Utility Token

- 1 : 1 USDC
- Pas d'investissement
- Pas un Security token

Token GOUV : Soulbound Token

- ve-token (veAURA)
- Sans valeur marchande

NFTs (Billets/Cartes) : Digital Collectible

- Pas de spéculation

Token VOTE : Soulbound Token

- Sans valeur marchande
- Permet de voter l'issue d'un battle

Domaine	Point d'attention	Solution retenue
Structure	Confusion des flux	Ségrégation SAS (Tech) vs Offshore (Pari).
Régulation	Jeu d'argent (FR)	Mode "Compétence" (zéro mise) en France.
Token	AURA = Monnaie ?	"Boucle fermée" sans rendement passif.
Litige	Erreur de jugement	Acceptation du consensus Oracle (CGU).

Calendrier juridique :

- Structuration : Montage Hybride SAS/Offshore & Qualification Token
- Contractualisation : CGU différenciées & Partenariats
- Conformité : Licence iGaming & KYC global
- Sécurité : Audit Smart Contracts

Partenaires & Budget :

- Juriste Web3 : Qualification JONUM & Veille
- Avocat : Structuration Offshore & Contrats
- Expert-Comptable : Fiscalité & Flux Crypto
- Budget estimé : 150'000 – 200'000 €

4. État de l'art

Critère	Modèle Actuel (IRL / Web2)	Solution Rethagon (Blockchain / Web3)
Rémunération	Cachet fixe & plafonné	Revenus dynamiques & illimités
Portée & Marché	Locale & Fragmentée	Mondiale & Unifiée
Technologie	Opacité & Tiers de confiance	Transparence & Smart Contracts
Statut de l'Artiste	Prestataire de service	Actif valorisable (Asset)

Propositions différenciante

Financiarisation de la Réputation :

- Score ELO devient un enjeu financier

Spectateur – Acteur :

- Le fan devient garant de la vérité

Tendance du marché

Rémunération artistes :

- Revenus stables pour vivre de leur art.

Transparence :

- Transparence et équité dans les jugements.

5. Étude d'opportunité

Forces : <ul style="list-style-type: none">• Architecture Juridique Hybride• Engagement Communautaire• Système Anti-Corruption• Monétisation des Artistes	Faiblesses : <ul style="list-style-type: none">• Onboarding• Risque de Dilution de la Gouvernance• Complexité Technique & Opérationnelle
Menaces : <ul style="list-style-type: none">• Requalification Réglementaire (JONUM)• Collusion des Oracles	Opportunités : <ul style="list-style-type: none">• Marché "Blue Ocean"• Scalabilité Mondiale• Gamification de la Réputation

Pourquoi Rethagon est adapté ?

Pourquoi ce projet ?

- Précarité des artistes, Marché océan bleu

Pourquoi la blockchain ?

- Transparence, Liquidité Mondiale

Pourquoi Ethereum ?

- Standards robustes, Sécurité, Interopérabilité

Pourquoi ces tokens ?

- AURA, GOUV, VOTE & NFTs : boucle, soulbound, présence

Pourquoi ces structures ?

- Isoler le risque (SAS / Offshore), Flexibilité commerciale

Nos différences ?

- Le Fan est Juge, Revenus dynamiques

Buisness Model Canvas

Partenaires clés :

- Réseau EVM
- Fournisseurs On-ramp
- Cabinet juridique
- Outil d'audit
- etc.

Ressources clés :

- Base de donnée ELO
- Protocole anti-corruption
- etc.

Activités clés :

- Développement des smart Contracts
- Conformité Legal
- etc.

Propositions de valeur :

Pour les parieurs :

- Paris rapides et transparent

Pour les artistes :

- Revenus directs et automatisés

Pour l'écosystème :

- Standardiser les données

Etc.

Relations clients :

- DAO via le token GOUV
- Relation communautaire
- etc.

Canaux :

- Réseaux sociaux
- Événements physiques
- etc.

Segments de clientèle :

- Fans de Battle Rap
- Parieurs Web3
- Auditeurs de Rap
- Ligues Partenaires

Sources de revenus :

- **Trading Fees** : Commission dynamique sur chaque pari executé.
- **Billeterie** : Commision sur la vente des billets donnant le droit de vote.
- etc.

Structure de coûts :

- **Développement et Audits de sécurité**
- **Frais de Gaz** : Subvention des frais de gaz pour les utilisateurs.
- etc.

6. Notre organisation

- Basé sur des échanges quotidiens sur notre canal discord.
- Utilisation d'outils adapté comme Jira, GitHub Project et tldraw.

Objectif :

Allier agilité et fluidité des échanges, notamment via des sprint régulier.

notes-et-ressources



Ce système peut être étendu à l'alimentation de la base de donnée.

ex. Un détenteur du token de gouvernance le met en jeu pour ajouter une information à la base de données de manière.

@Bobotel @everyone oracle collégial : J'oublie sûrement des points, faites le moi savoir ! (modifié)

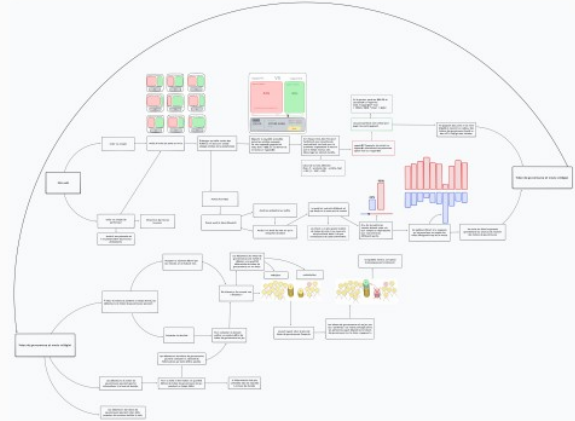
Jean-Pierre BRISEBARRE 17/11/2025 16:19

Super, il faut que je retourne voir le cours pour te dire ça mais ça me semble bon



Bobotel 17/11/2025 17:20

Disponible dans fichier partagé en mode visiteur uniquement, si vous souhaitez apporter des modifications



il s'agit du fonctionnement de:

- système de paris
 - billets et votes
 - oracles via token de gouvernance
- avec les liens les interconnectants

ACHAYMAA 17/11/2025 17:25

ah c'est super. c'est un excellent schéma du modèle économique de l'appli 🌟🌟🌟

Jean-Pierre BRISEBARRE 17/11/2025 17:27

Epics :

- 

DRAFT

9

This item hasn't been started

rethagon #6

Rethagon design System

rethagon #5

Token

rethagon-api #1

Create Supabase POC

rethagon #7

Stabelcoin Natif

rethagon #8

Token de récompense AURA

rethagon #9

Billetterie → droit de vote

rethagon #10

Oracle et token de gouvernance

rethagon #11

Users Stories

rethagon #13

Fonctionnement des pools de pari

leobth opened 3 weeks ago · edited by leobth

Edits · · ·

Détails des calculs

Les calculs sont détaillés en bas de page, suivre les numéros (1), (2), etc.

Vue d'ensemble

Les parieurs arrivent sur un **dashboard** regroupant tous les paris actuellement en cours et sur lesquels ils peuvent miser. Sur chaque fiche, on indique :

- les statistiques fournies par l'oracle avec le pourcentage de chance de gagner ;
- les **cotes actuelles** (1), calculées en fonction des mises déjà collectées ;
- la **liquidité totale pariée** → sert à savoir si sa mise fera beaucoup évoluer les cotes.

Détails d'un pari

Une fois un pari **sélectionné**, le parieur aura **davantage d'éléments** à sa disposition.

Partie statistique

Le parieur aura accès aux **statistiques détaillées** de chaque rappeur, indiquant notamment :

- son **style**,
- ses **victoires** dans cette compétition,
- s'il y a déjà eu un **battle** les opposant, etc.

Partie technique

Se présenteront à lui les **taxes (rappeurs + protocole)** (2), en affichant sa mise sans taxe puis sa mise une fois les taxes prélevées.
De plus, il verra une **jauge** lui permettant de répartir sa mise sur les deux rappeurs.

Les cotes pourront être affichées en :

- mode « **cote actuelle** », ou
- mode « **simulation** », qui montre l'impact de sa mise sur l'évolution des cotes.

Le mode simulation affiche également les **gains potentiels** si les cotes n'évoluent pas.

👉 Miser sur `rappeur1` revient simplement à déposer sa liquidité sur `rappeur1`, qui est un ERC-1155 : sa valeur augmente et donc sa cote augmente avec.

Résultat et gains

Une fois le pari terminé et le **résultat validé par l'oracle**, les **gains** sont crédités automatiquement sur le compte du parieur (3).

Formules

Calculer la cote

(1) Pour calculer la cote du `rappeur#1`, il faut diviser la liquidité déposée sur `rappeur#1` par la liquidité totale, ce qui nous donne la valeur du token `rappeur#1`, puis multiplier le tout par cent :
$$\text{Rappeur_liquidity} / \text{Total_liquidity} * 100 = \text{Cote du rappeur en \%}$$

Calculer la mise après taxation

(2) 15 % de la liquidité revient aux rappeurs et 2 % à 10 % (évolutif dans le temps (4)) revient au protocole. Pour calculer la mise effective du parieur, il faut faire :
$$\text{mise} * (1 - \text{rappeur_fee} - \text{protocol_fee}) = \text{mise_réelle}$$

Calculer les gains

(3) Pour calculer les gains, il faut faire :
$$\text{Total_liquidity} / \text{Winner_Liquidity} * \text{mise_réelle} = \text{gains (mise incluse)}$$
$$\text{gains} - \text{mise_réelle} = \text{gains_nets (mise exclue)}$$

User Story

Écoute active : Feedback de la communauté Discord

Veille : Battle Rap, Paris et Blockchain

User Story :

- "En tant que rappeur, je veux créer un compte artiste pour m'inscrire aux battles."
 - Développement des comptes performeur.
- "En tant que parieur, je souhaite pouvoir répartir ma mise entre les artistes."
 - Développement du slider.
- "En tant que membre de la gouvernance, je souhaite pouvoir contester un résultat que je jugerais erroné."
 - Développement du système de contestation via GOUV (ve-token)

Espaces

Rethagon

Résumé Backlog Tableau Calendrier More 4 +

Recherchez dans... Filter

SCRUM Sprint 1 12 nov. - 19 nov. (7 tickets) 0 28 0 Terminer le sprint

Mettre en place un prototype fonctionnel du système de pari permettant à un utilisateur connecté

SCRUM-6	Story 1 — Créati...	EN COURS	5
SCRUM-10	Story 2 — Créa...	EN COURS	8
SCRUM-14	Story 3 — Wall...	EN COURS	5
SCRUM-26	Story 4 — Swa...	EN COURS	5
SCRUM-29	Story 5 — Dép...	EN COURS	5
SCRUM-90	Task 1 - AURA 1:1 USDC	EN COURS	-
SCRUM-94	Token de gouvernance...	IDÉE	-

+ Créer

SCRUM Sprint 2 19 nov. - 26 nov. (6 tickets) 0 36 0 Terminer le sprint

SCRUM-33	Story 1 — Affichag...	EN COURS	5
SCRUM-36	Story 2 — Choix d...	EN COURS	5
SCRUM-39	Story 3 — Frais pr...	EN COURS	5
SCRUM-42	Story 4 — Confirm...	EN COURS	8
SCRUM-45	Story 5 — Historiq...	EN COURS	5
SCRUM-48	Story 6 — Estimati...	EN COURS	8

+ Créer

SCRUM Sprint 3 26 nov. - 3 déc. (8 tickets) 0 0 0 Démarrer un sprint

Ticket Jira

SCRUM-5 / SCRUM-10

Story 2 — Création compte performeur

En cours

Description

En tant que rappeur, je veux créer un compte artiste pour m'inscrire aux battles

AC :

- GIVEN le rôle performeur sélectionné
WHEN j'envoie le formulaire
THEN mon dashboard artiste apparaît.

DoD :

- Dashboard artiste actif
- Profil artiste créé

Sous-tâches

Progression : 0

Tickets	Prio...	Per...	État
SCRUM-11	Formulaire dédié	M€ N	EN COURS
SCRUM-12	Rôle artiste	M€ N	EN COURS
SCRUM-13	Page dashboard	M€ N	EN COURS

Tickets associés

Ajouter un ticket lié

Contenu Confluence

Exigences produit

DEMANDEZ UN ESS.

Détails

Non assignée

7 – Démonstration

RETHAGON

Animation pari

Smart contract : 0xj487dhw843v284tf6288fr3fddsfcvfe487dhw843e
v284tf487dhw843v284tf487dhw843v284tf

8 - Roadmap

Quarter 1 : Fondations & Sécurité

- Juridique : Création SAS, France + Offshore, Curaçao.
- Tech : Audit de sécurité (Smart Contracts) + Déploiement Geofencing IP.

Quarter 2 : Soft Launch – France

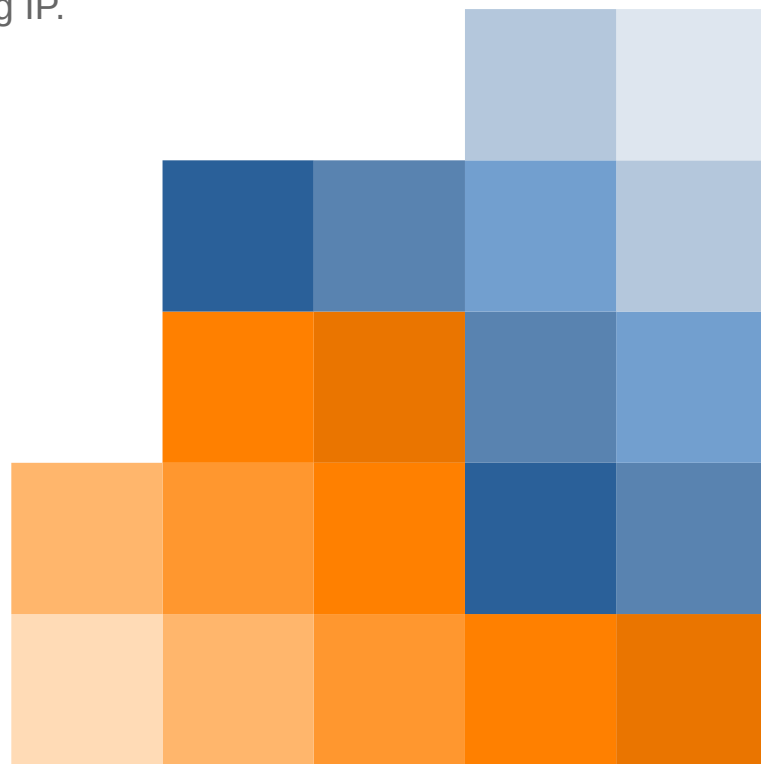
- Produit : Lancement Mode "Skill Game" (Conformité JONUM).
- Offre : Achat de cartes NFT & Compétitions Free-to-play.

Quarter 3 : Hard Launch - Monde

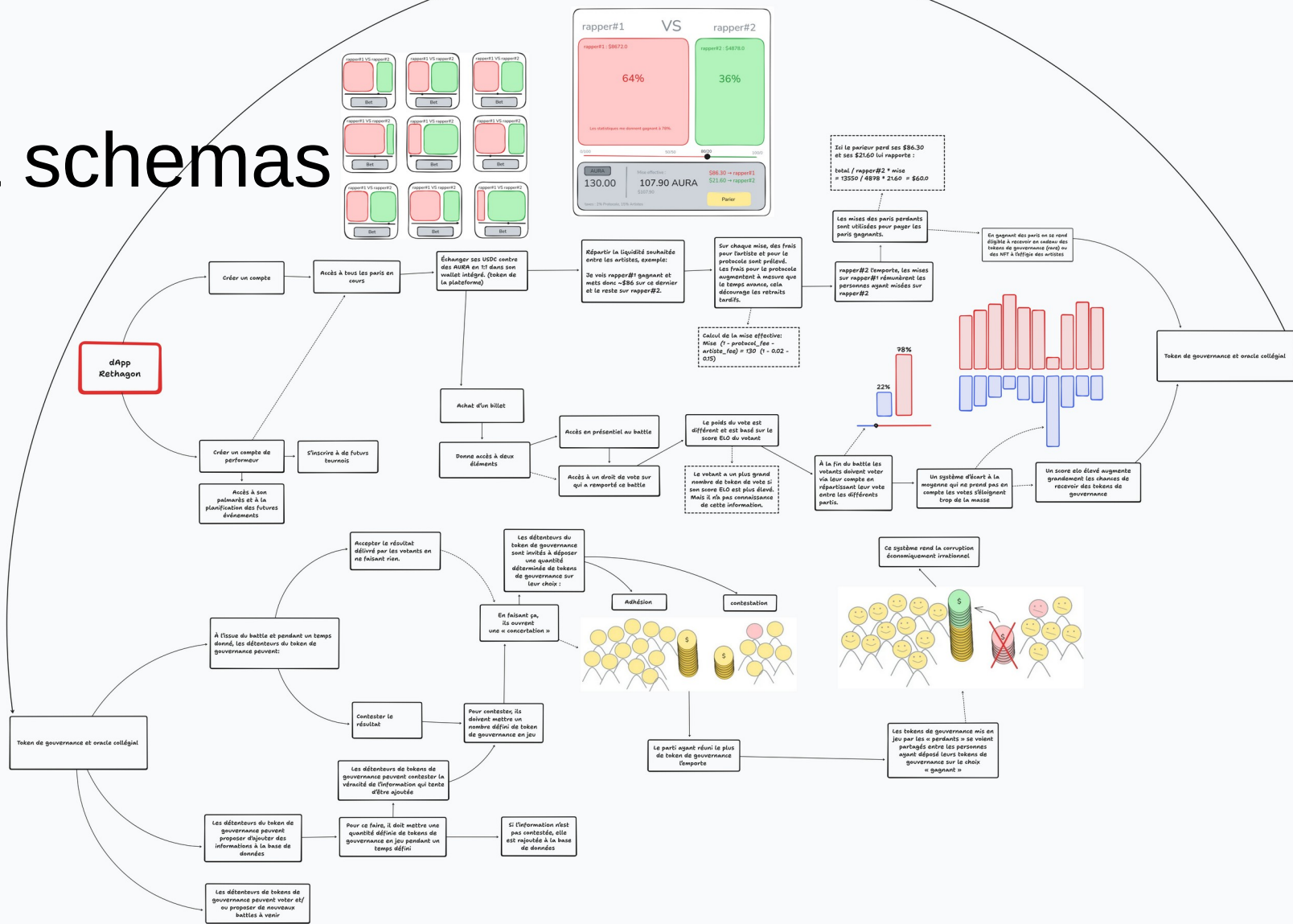
- Légal : Obtention Licence iGaming + Activation KYC.
- Produit : Ouverture Mode "Pari" (AMM & Liquidité USDC).
- Tech : Activation du off-ramp (Settlement of Winnings).

Quarter 4 : Mass Adoption & US

- UX : Connexion Google/Apple (Account Abstraction).
- Tech : Migration Layer 2 (Réduction frais de gaz).



Annexes & schemas



```

1 // === Paramètres ===
2 base_protocol_fee // frais de base fixé avant le début du battle
3 protocol_fee // frais évoluant dans le temps (protocol_fee à temps 0 = base_protocol_fee)
4 fee_change_factor // facteur d'évolution des frais en fonction du temps
5 time // temps écoulé depuis le début du battle
6 time_limit // limite de temps pour plafonner les frais
7 stake // somme mise par l'utilisateur
8 outcomes_liquidity // tableau des liquidités {rapper#1, rapper#2, ...}
9 MCs_fee // frais pour rémunération des MCs
10
11 // === Calcul des frais du protocole ===
12 function protocol_fee(fee_change_factor, time, base_protocol_fee, time_limit) {
13     if (time < time_limit)
14         return base_protocol_fee + fee_change_factor * time;
15     else
16         return base_protocol_fee + fee_change_factor * time_limit;
17 }
18
19 // === Frais prélevés par le protocole ===
20 function protocol_amount(protocol_fee, stake) {
21     return stake * protocol_fee;
22 }
23
24 // === Frais pour les MCs ===
25 function MCs_amount(MCs_fee, stake) {
26     return stake * MCs_fee;
27 }
28
29 // === Mise après prélèvement des frais ===
30 function stake_after_fee(MCs_fee, protocol_fee, stake) {
31     return stake * (1 - MCs_fee - protocol_fee);
32 }
33
34 // === Liquidité totale pour un battle ===
35 function total_liquidity(outcomes_liquidity) {
36     let total = 0;
37     for (let i = 0; i < outcomes_liquidity.length; i++) {
38         total += outcomes_liquidity[i];
39     }
40     return total;
41 }
42

```