

# Documentation technique

## Système de Gestion des Risques de Contrepartie sur Blockchain

Ce projet implémente un système de gestion des risques de contrepartie basé sur la blockchain, en combinant les principes de la gestion des risques financiers avec les avantages offerts par la technologie blockchain, comme la transparence, la sécurité, et l'immutabilité des données.

### Objectif :

- Fournir un système décentralisé pour suivre et gérer les expositions des contreparties.
- Calculer les risques associés aux contreparties de manière précise et automatisée.
- Alerter en cas de dépassement des limites d'exposition prédéfinies, tout en assurant une gestion sécurisée et indépendante.

### Technologie utilisée :

- **Blockchain Ethereum** : Plateforme utilisée pour déployer le smart contract.
- **Solidity** : Langage de programmation utilisé pour développer le smart contract.
- **Remix IDE** : Environnement de développement intégré pour écrire, tester, et déployer le contrat.
- **MetaMask** : Portefeuille numérique pour interagir avec le contrat sur un réseau de test Ethereum.

### Fonctionnalités principales :

1. **Gestion des Contreparties** :
  - Ajout des contreparties avec des informations telles que leur adresse, score de crédit, limites d'exposition, collatéral, et état actif/inactif.
2. **Mise à jour des Expositions** :

- Suivi des expositions longues et courtes avec calcul de l'exposition nette.
- 3. **Calcul des Risques :**
  - Évaluation des risques via des formules basées sur l'exposition courante, les limites, et le score de crédit.
- 4. **Calcul du Ratio de Couverture :**
  - Mesure de la couverture financière des expositions par rapport au collatéral.
- 5. **Alerte de Dépassement de Limite :**
  - Notification automatique si une contrepartie dépasse sa limite d'exposition.
- 6. **Désactivation des Contreparties :**
  - Fonctionnalité permettant de désactiver une contrepartie pour empêcher toute modification ou interaction future.

Ce système offre une solution fiable, transparente et robuste pour la gestion des risques financiers dans un environnement décentralisé.

## ● Architecture du Contrat

### 1. Structure

La structure principale du contrat est Contrepartie, utilisée pour stocker toutes les informations nécessaires concernant chaque contrepartie. Voici les champs détaillés :

- portefeuille (**address**) : Adresse Ethereum de la contrepartie, utilisée comme identifiant unique.
- scoreCredit (**uint256**) : Score de crédit attribué à la contrepartie, représentant sa solvabilité.
- limiteExposition (**uint256**) : Limite maximale d'exposition autorisée pour cette contrepartie.
- expositionLongue (**uint256**) : Somme des positions longues de la contrepartie.
- expositionCourte (**uint256**) : Somme des positions courtes de la contrepartie.
- expositionCourante (**uint256**) : Exposition nette de la contrepartie, calculée comme :

**Exposition Courante=Exposition Longue–Exposition Courte**

- **collaterale (uint256)** : Valeur du collatéral déposé par la contrepartie pour couvrir ses expositions.
- **estActif (bool)** : Statut de la contrepartie (active ou désactivée). Une contrepartie désactivée ne peut plus être mise à jour.

## 2. Variables (Mappings)

Les mappings sont utilisés pour gérer efficacement les contreparties et leurs expositions.

- **mapping(address => Contrepatrie) contrepatries :**

-Ce mapping associe chaque adresse Ethereum à une structure Contrepatrie.

-Il est utilisé pour stocker et accéder rapidement aux informations d'une contrepartie.

- **mapping(address => mapping(address => uint256)) expositions :**

-Ce mapping gère les expositions spécifiques entre deux adresses (contrepartie et une autre entité).

- - Il permet de suivre les interactions financières détaillées si nécessaire.

## 3. Evénements

Les événements sont des notifications déclenchées lorsque certaines actions importantes se produisent. Ils permettent une meilleure traçabilité des actions effectuées sur la blockchain.

### ➔ ContrepatrieAjoutée :

- Déclenché lorsqu'une nouvelle contrepartie est ajoutée.
- Paramètres :
  - **address indexed contrepatrie** : Adresse de la contrepartie ajoutée.
  - **uint256 limiteExposition** : Limite d'exposition assignée à la contrepartie.

### ➔ ExpositionMiseAJour :

- Déclenché à chaque mise à jour de l'exposition d'une contrepartie.
- Paramètres :
  - **address indexed contrepatrie** : Adresse de la contrepartie concernée.
  - **uint256 nouvelleExposition** : Nouvelle valeur de l'exposition courante.

### ➔ LimiteDepassee :

- Déclenché lorsqu'une contrepartie dépasse sa limite d'exposition.

- Paramètres :
  - address indexed contrepartie : Adresse de la contrepartie concernée.
  - uint256 exposition : Valeur de l'exposition courante dépassant la limite.

Cette structure et ces mécanismes permettent une gestion robuste des contreparties et de leurs expositions tout en fournissant des notifications claires pour un suivi précis des événements critiques.

## • Fonctionnalités Principales

### 1. Gestion des Contreparties

#### ➔ Ajouter une contrepartie :

- La fonction ajouterContrepartie permet d'ajouter une nouvelle contrepartie au système avec les informations suivantes :
  - Adresse Ethereum.
  - Score de crédit.
  - Limite d'exposition.
  - Valeur du collatéral.
- Événement déclenché : ContrepartieAjoutée, qui enregistre l'ajout d'une nouvelle contrepartie.

#### ➔ Désactiver une contrepartie :

- La fonction desactiverContrepartie désactive une contrepartie, empêchant toute interaction future avec celle-ci.
- La variable estActif de la structure de la contrepartie est mise à false.

### 2. Gestion des Expositions

#### ➔ Mise à jour des expositions :

- La fonction mettreAJourExposition permet de mettre à jour l'exposition d'une contrepartie, en ajoutant une exposition longue ou courte.
- L'exposition nette (longue - courte) est automatiquement calculée et mise à jour dans expositionCourante.

- Événement déclenché : ExpositionMiseAJour, qui enregistre la nouvelle exposition nette.

➔ Alerte en cas de dépassement de la limite :

- Si l'exposition courante dépasse la limite définie pour une contrepartie, un événement LimiteDepassee est émis pour signaler le dépassement.

### 3. Calcul des Risques

➔ Calculer le risque d'une contrepartie :

La fonction calculerRisque utilise la formule suivante pour évaluer le risque :

$$\text{Score de Risque} = \frac{\text{Exposition Courante}}{\text{Limite d'Exposition}} \times \frac{100}{\text{Score de Crédit}}$$

Retourne un score de risque basé sur les données actuelles de la contrepartie.

➔ Calculer le ratio de couverture :

La fonction calculerRatioCouverture évalue la couverture financière d'une contrepartie en utilisant la formule :

$$\text{Ratio de Couverture} = \frac{\text{Collatéral}}{\text{Exposition Totale}}$$

Ce ratio aide à mesurer si une contrepartie dispose de suffisamment de collatéral pour couvrir son exposition.

### 4. Gestion des Événements

➔ Événements émis :

- Contrep partieAjoutée : Lorsqu'une nouvelle contrepartie est ajoutée.
- ExpositionMiseAJour : Lorsqu'une mise à jour d'exposition est effectuée.
- LimiteDepassee : Lorsqu'une contrepartie dépasse la limite d'exposition autorisée.

#### ➔ Utilisation des événements :

- Les événements permettent une transparence sur les actions exécutées sur le smart contract et facilitent leur suivi dans l'interface utilisateur ou via des outils de gestion blockchain.

## 5. Sécurité et Restrictions

#### ➔ Validation des entrées :

- Les fonctions utilisent des conditions require pour s'assurer que :
  - Les données entrées sont valides (exemple : limite > 0, score de crédit > 0).
  - La contrepartie existe avant de mettre à jour son exposition ou calculer son risque.
  - Les fonctions ne peuvent pas être appelées sur des contreparties désactivées.
- Protection contre les abus :
  - La fonction ajouterContrep partie empêche l'ajout de la même adresse plusieurs fois.
  - Les expositions négatives ou nulles sont bloquées par les types Solidity (uint256).

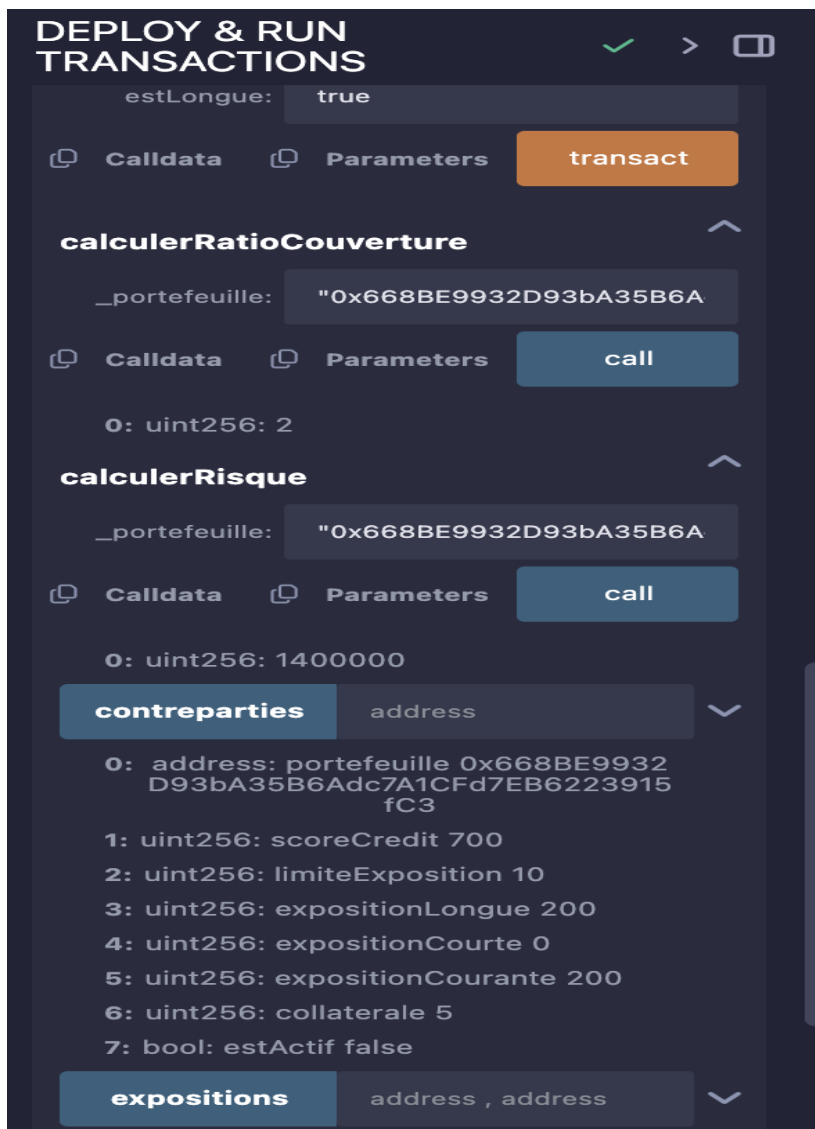
## 6. Transparence et Auditabilité

#### ➔ Consultation publique :

- Le mapping contreparties est public, permettant à tout utilisateur de consulter les données des contreparties (ex : exposition actuelle, limite, collatéral).

- Les calculs de risque et de couverture sont également accessibles publiquement via des appels aux fonctions `calculerRisque` et `calculerRatioCouverture`.
- ➔ Audit grâce aux événements :
- Chaque action (ajout, mise à jour, dépassement de limite) est enregistrée sous forme d'événement, rendant le système entièrement traçable.

## • Tests et résultats



### ➔ Fonction : `ajouterContrepartie`

- Action :
  - Une contrepartie a été ajoutée avec les paramètres suivants :

- **Adresse**  
: 0x668BE9932D93bA35B6Adc7A1CFd7EB6223915fC3
- **Score de crédit : 700**
- **Limite d'exposition : 10**
- **Collatéral : 5**

L'état initial de la contrepartie a été enregistré correctement dans le mapping.

- **Validation :**

- Les données de la contrepartie sont accessibles via le mapping contreparties.
- Toutes les valeurs sont correctes :
  - expositionLongue : 0
  - expositionCourte : 0
  - expositionCourante : 0
  - estActif : true

➔ **Fonction : mettreAJourExposition**

- **Action :**

- Une exposition longue de 200 a été ajoutée à la contrepartie.
- Résultat :
  - expositionLongue mise à jour à 200.
  - expositionCourante calculée comme  $200 - 0 = 200$ .

- **Cas limite validé :**

- La limite d'exposition (10) a été dépassée.
- L'événement LimiteDepassee a été déclenché comme prévu.

➔ **Fonction : calculerRisque**

- **Action :**

- Le risque a été calculé pour cette contrepartie à l'aide de la formule :



$$\text{Score de Risque} = \frac{\text{Exposition Courante}}{\text{Limite d'Exposition}} \times \frac{100}{\text{Score de Crédit}}$$

- **Substitutions :**

Exposition Courante=200

Score de Crédit=700

Limite d'Exposition=10

-Résultat attendu : 1 400 000

- **Validation :**

- Le contrat retourne la valeur correcte : 1400000

➔ **Fonction :** calculerRatioCouverture

- **Action :**

- Le ratio de couverture a été calculé pour cette contrepartie à l'aide de la formule :

$$\text{Ratio de Couverture} = \frac{\text{Collatéral}}{\text{Exposition Totale}}$$

- **Substitutions :**

Collatéral=5

Exposition Courante=200

-Résultat attendu : 2

- **Validation :**

- Le contrat retourne la valeur correcte : 2.

Normalement c'est 2,5 mais on fait l'arrondissement dans les smart contracts.

➔ **Fonction :** desactiverContrepartie

- **Action :**

- La contrepartie a été désactivée avec succès.

- L'état estActif est passé à false.

- **Validation :**

- Toute tentative ultérieure de modifier ou calculer des données pour cette contrepartie échoue conformément aux conditions de validation dans le contrat.

### **Conclusion des Tests**

1. Toutes les fonctions principales ont été testées et validées avec succès.
2. Les résultats des cas limites (dépassement de limite, désactivation) ont été obtenus comme prévu.
3. Les calculs (calculerRisque, calculerRatioCouverture) ont été vérifiés manuellement et donnent des résultats corrects.

Ce projet peut être étendu dans un environnement réel en intégrant des données externes via des oracles, en optimisant la gestion des risques grâce à des algorithmes avancés, et en utilisant des solutions de blockchain évolutives pour réduire les coûts. Il offre un potentiel significatif pour automatiser et sécuriser la gestion des contreparties, tout en répondant aux exigences des institutions financières modernes.

