

Loopring: Un protocole pour marchés d'échange décentralisés

Daniel Wang
daniel@loopring.org

Jay Zhou
jay@loopring.org

Alex Wang
alex@loopring.org

Matthew Finestone
matt.finstone@gmail.com

<https://loopring.org>

May 10, 2018

Abstract

Loopring est un protocole ouvert pour la mise en place de marchés d'échange décentralisés. Loopring fonctionne comme un ensemble de smart contracts publics pour les échanges et les règlements, avec un groupe d'acteurs off-chain agréant et communiquant les ordres. Le protocole est libre, extensible, et sert de composant logiciel standardisé pour applications décentralisées (dApps) qui incorporent des fonctionnalités de marchés d'échange. Ses normes ouvertes d'interopérabilité facilitent des échanges anonymes et sans tiers de confiance. Un avantage notable de Loopring par rapport aux protocoles d'échanges décentralisés actuels est la possibilité pour les ordres d'être panachés avec des ordres d'autres paires, ce qui permet de se libérer des contraintes liées aux paires limitées à deux tokens, mais aussi d'augmenter considérablement la liquidité. Loopring emploie aussi une solution unique et robuste pour se prémunir du front-running (les tentatives, injustes, de soumettre des transactions dans un bloc plus rapidement que celui ayant fourni la solution originelle). Loopring est agnostique quant aux blockchains mises en jeu, et deployable sur n'importe quelle blockchain supportant les smart contracts. Au moment de la rédaction de ce livre blanc, Loopring peut opérer sur Ethereum [1] [2] et Qtum [3] et prochainement NEO [4].

1 Introduction

Avec la prolifération des actifs basés sur la technologie blockchain, le besoin d'échanger ces derniers entre diverses parties a augmenté de manière significative. Alors que des milliers de nouveaux tokens font leur apparition, dont la tokenisation d'actifs traditionnels, ce besoin est amplifié. Que l'échange de token soit à des fins spéculatives ou pour l'accès à des réseaux via leur token natif, la capacité d'échanger des actifs cryptographiques est fondamentale à la croissance de l'écosystème. En effet, il y a une énergie potentielle dans les actifs [5], et réaliser cette énergie - libérer du capital - requiert non seulement d'assurer la propriété de ces actifs, ce que la blockchain permet, mais aussi la capacité de transférer et de transformer ces actifs.

Ainsi l'échange de tokens (valeur) sans tiers de confiance est un cas d'utilisation convaincant de la technologie blockchain. Jusqu'à présent, cependant, les crypto-enthousiastes ont largement accepté d'échanger des tokens sur les marchés centralisés traditionnels. Le protocole Loopring est nécessaire parce que, comme Bitcoin l'a scrupuleusement souligné [6] à propos du cash électronique en peer-to-peer, "les avantages sont perdus si le tiers de confiance est encore requis pour empêcher les doubles

dépenses", tout comme sont perdus les avantages des actifs décentralisés si ces derniers doivent transiter par des marchés de confiance, fermés, et centralisés.

Echanger des tokens décentralisés sur des marchés centralisés n'a que peu de sens philosophiquement parlant, car cela s'oppose aux valeurs soutenues par les projets décentralisés. De nombreux risques et limitations associés à l'utilisation de marchés centralisés sont décrits ci-dessous. Certains marchés d'échange décentralisés (DEX ci-après) [7] [8] [9] ont tenté de répondre à ces problèmes, et ont dans de nombreux cas réussi, en utilisant la blockchain pour la désintermédiation. Cependant, dès lors que l'utilisation des DEX devient un pilier dans l'infrastructure de cette nouvelle économie, il y a matière à s'améliorer. Loopring a pour objectif de fournir des outils modulaire pour cette infrastructure, grâce à son protocole ouvert et agnostique pour applications décentralisées (dApp).

2 Tour d’horizon des marchés centralisés

2.1 Les insuffisances des marchés centralisés

Les trois risques principaux des marchés centralisés sont : un manque de sécurité, un manque de transparence, et un manque de liquidité.

Un manque de sécurité survient dès lors que les utilisateurs cèdent le contrôle de leur clé privée, et donc de leurs fonds, à une seule entité centralisée. Les utilisateurs sont exposés au risque qu’un marché centralisé soit la proie de hackers malveillants. Les risques de sécurité et de piratage auxquels les échanges centralisés font face sont bien connus [10] [11], mais sont pourtant acceptés comme un “pré-requis” pour l’échange de tokens. Les échanges centralisés restent des cibles privilégiées pour les hackers, car leurs serveurs sont en possession de millions de dollars de fonds appartenant aux utilisateurs. Les développeurs des marchés peuvent aussi commettre des erreurs innocentes ou des maladresses avec les fonds des utilisateurs. Tout simplement, les utilisateurs ne sont plus maîtres de leurs tokens une fois déposés sur un marché centralisé.

Un manque de transparence expose aussi les utilisateurs au risque qu’un échange se comporte de manière injuste voire frauduleuse. La différence ici sont les intentions malveillantes du marché, puisque les utilisateurs ne tradent pas directement leurs propres actifs, mais plutôt des reconnaissances de dette. Lorsque les tokens sont déposés sur le portefeuille du marché d’échange, ce dernier en prend possession, et fournit une reconnaissance de dette à la place. Tous les échanges sont ensuite effectués entre les reconnaissances de dette des utilisateurs. Lors d’un retrait, les utilisateurs font valoir leur reconnaissance de dette, et reçoivent leur tokens sur leur portefeuille externe. Tout au long de ce processus, on constate un manque de transparence, alors que l’échange peut fermer, geler votre compte, faire faillite etc. Il est aussi possible qu’ils utilisent les actifs des utilisateurs à d’autres fins, comme les prêter à des tiers. Ce manque de transparence peut avoir un coût comme des commissions élevées, des délais aux heures de pointe, une exposition à des risques réglementaires, ou des ordres frontrunnés.

Le manque de liquidité Pour un opérateur de marché d’échange, grâce à la position dominante conférée par la fragmentation de la liquidité, les chances sont réduites pour la concurrence. Tout d’abord, le marché offrant le plus de paires d’échange gagne, car les utilisateurs préfèrent pouvoir effectuer tous leurs échanges au même endroit. Ensuite, le marché avec les carnets d’ordres les plus fournis gagne aussi, en raison d’écarts entre l’offre et la demande plus favorable. Ceci décourage les nouveaux arrivants parce qu’il leur est difficile d’offrir de la liquidité d’entrée de jeu. Ainsi, plusieurs marchés préservent une part de marché élevée, malgré les réclamations des utilisateurs ou même après un piratage. A

noter que plus un échange concentre de parts de marché, plus il sera une cible intéressante à pirater.

Pour les utilisateurs, la fragmentation de la liquidité affecte leur confort. Sur un marché centralisé, les utilisateurs ne peuvent qu’échanger avec les réserves de liquidité propres au marché, avec ses propres carnets d’ordres, et dans la limite des paires proposées. Pour échanger un token A contre un token B, les utilisateurs doivent trouver un marché supportant A et B, ou s’inscrire sur plusieurs échanges, en fournissant à chaque fois leurs données personnelles. Ils doivent aussi souvent effectuer des transactions intermédiaires via BTC ou ETH, . Finally, the order books may not be deep enough to complete the trade without material slippage. Et même dans le cas où un marché prétend traiter de larges volumes, il n’y a aucune garantie quant à l’authenticité de ces volumes et de cette liquidité [12].

Il en résulte des réserves de liquidité déconnectées les unes des autres, et un écosystème fragmenté rappelant les systèmes financiers traditionnels, avec un volume d’échange important sur une poignée de marchés. La promesse de liquidité globale offerte par la technologie blockchain n’est pas tenue avec les marchés centralisés.

2.2 Les insuffisances des marchés décentralisés

L’un des points sur lesquels les marchés décentralisés diffèrent des marchés centralisés est que les utilisateurs conservent le contrôle de leur clés privées (donc de leurs actifs) en effectuant des échanges directement sur la blockchain sous-jacente. En tirant profit de la technologie permettant de se passer de tiers de confiance, les marchés décentralisés atténuent un certains nombres des risques liés à la sécurité mentionnés plus haut. Cependant, de nombreux problèmes persistent en termes de performance.

Le manque de liquidité reste un problème car les utilisateurs doivent trouver des contreparties à travers des réserves de liquidité et des normes disparates. Les effets de cet éclatement de la liquidité se font sentir si les DEXs ou les dApps n’emploient pas de normes constantes pour interagir, et si les ordres ne sont pas partagés ou diffusés à travers un large réseau. La liquidité des carnets d’ordres à cours limités, et plus précisément, leur souplesse – la vitesse dont les ordres à cours limités sont régénérés – peut affecter de manière significative une stratégie d’échange optimale [13]. L’absence de telles normes ne n’entraîne non seulement une réduction de la liquidité, mais expose aussi les utilisateurs à un éventail de smart contracts potentiellement non-sécurisés.

De plus, puisque les échanges sont effectués directement sur la blockchain concernée, les DEXs héritent des limitations de cette dernière, notamment : mise à l’échelle, délais dans l’exécution (minage), et modifications des ordres coûteuses. Ainsi, les carnets d’ordres sur la blockchain ne peuvent pas monter en capacité facilement, puisque

l'exécution du code sur la blockchain a un coût (gas), ce qui rend l'annulation de nombreux ordres extrêmement coûteuse.

Enfin, comme les carnets d'ordres sur la blockchain sont publics, la transaction correspondant au placement de l'ordre est visible par les mineurs avant même qu'elle soit incluse dans le prochain bloc et effectivement placée dans le carnet d'ordres. Ce délai expose l'utilisateur au risque de se faire "frontrunner" et de voir le prix ou l'exécution altérée défavorablement.

2.3 Les solutions hybrides

Pour les raisons mentionnées ci-dessus, les marchés purement basés sur la blockchain ont des limitations les rendant peu compétitifs avec les marchés centralisés. La confiance conférée par les échanges on-chain est sacrifiée au profit de la vitesse et de la flexibilité des ordres offerte par les marchés centralisés. Des protocoles tels que Loopring ou 0x [14] prolongent une solution de règlements on-chain par une gestion des ordres off-chain. Cette solution se base sur les smart contracts, mais connaît des difficultés d'évolutivité notamment en terme de performance, en exécutant plusieurs fonction off-chain, et en donnant aux noeuds de la flexibilité dans la manière d'accomplir certains rôles critiques pour le réseau. Cela dit, des inconvénients continuent d'exister avec le modèle hybride [15]. Le protocole Loopring propose des différences notables dans notre approche vers une solution hybrides, qui sont décrites dans ce document.

3 Le protocole Loopring

Loopring n'est pas un DEX, mais un protocole modulaire pour construire des DEXs sur plusieurs blockchains. Nous décomposons les éléments d'un marché traditionnel et offrons à leur place un ensemble de smart contracts et d'acteurs décentralisés. Les différents rôles dans le réseau comprennent les portefeuilles, les relays, les blockchains de partage de liquidité, les explorateurs de carnets d'ordres, les mineurs d'anneaux, et les services de tokenisation d'actifs. Avant de définir chacun d'entre eux, comprenons d'abord comment fonctionnent les ordres via le protocole Loopring.

3.1 Anneau d'ordres

Avec Loopring, les ordres sont exprimés sous la forme d'un Modèle d'Ordre Unidirectionnel (MOUD), "Unidirectional Order Model (UDOM)" en anglais [16]. Le MOUD exprime un ordre comme une demande d'échange de tokens, $\text{montantV}/\text{montantA}$, montant à vendre/acheter). Comme les ordres ne sont rien de plus qu'un taux de change entre deux tokens, une des fonctions prépondérantes du protocole est de pouvoir *pana of the protocol is the mixing and matching of multiple orders in circular trade*. En utilisant jusqu'à 16 ordres au lieu d'une simple paire d'échange, on

observe une forte augmentation de la liquidité et une chance d'obtenir un prix plus intéressant.

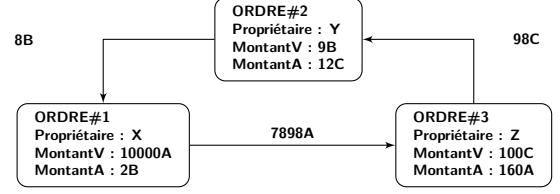


Figure 1: Un anneau d'ordres de 3 ordres

La figure ci-dessus illustre un anneau de 3 ordres. Dans un anneau, le token à vendre (**tokenV**) d'un ordre est le token à acheter (**tokenA**) d'un autre ordre, et ce pour chaque ordre. Une boucle est créée, permettant à chaque token d'être échangé sans avoir besoin d'avoir un ordre opposé pour la paire désirée. Bien entendu, des ordres d'une seule paire peuvent toujours être exécutés, mais c'est en fin de compte un cas particulier d'un anneau d'ordres (anneau à deux ordres).

Définition 3.1 (order-ring) Soient C_0, C_1, \dots, C_{n-1} n tokens distincts, soient $O_{0 \rightarrow 1}, \dots, O_{i \rightarrow i \oplus 1}, \dots, O_{n-1 \rightarrow 0}$ n ordres. Ces ordres peuvent former un anneau d'ordres pour un échange :

$$O_{0 \rightarrow 1} \rightarrow \dots \rightarrow O_{i \rightarrow i \oplus 1} \rightarrow \dots \rightarrow O_{n-1 \rightarrow 0},$$

avec n la longueur de l'anneau, et $i \oplus 1 \equiv i + 1 \pmod n$.

Un anneau d'ordres est considéré valide lorsque toutes les transactions qui le composent peuvent être exécutées à un taux de change au moins aussi avantageux que celui demandé implicitement par l'utilisateur dans son MOUD. Pour vérifier la validité d'un anneau d'ordres, les smart contracts du protocole Loopring doivent recevoir l'anneau de la page d'un mineur d'anneau must receive order-rings from ring-miners where the product of the original exchange rates of all orders is equal to or greater than 1.

Supposons que Alice et Bob veulent échanger leurs tokens A et B. Alice a 15 tokens A et elle aimerait en échange 4 tokens B; Bob a 10 tokens B et aimerait 30 tokens A en échange.

Qui vend et qui achète ? Cela va seulement dépendre de l'actif sur lequel nous allons nous baser pour déterminer un prix. Si le token A sert de référence, alors Alice achète B au prix de $\frac{15}{4} = 3.75A$, tandis que Bob vend 10 tokens B au prix de $\frac{30}{10} = 3.00A$. Inversement, si on prend le token B pour référence, alors on peut dire qu'Alice vend A au prix de $\frac{4}{15} = 0.26666667B$ et Bob achète 10 tokens A à $\frac{10}{30} = 0.33333334B$. Ainsi, qui est l'acheteur ou le vendeur est purement arbitraire.

Dans la première situation, Alice est prête à payer un prix plus élevé (3.75A) que celui auquel Bob vend ses tokens (3.00A), alors que dans le second cas, Bob est prêt à payer un prix plus élevé (0.33333334B) que celui auquel Alice vend les siens (0.26666667B). Il est évident qu'un échange peut être conclu dès lors que l'acheteur est prêt à payer un prix au moins aussi élevé que celui du prix de vente du vendeur.

$$\frac{\frac{15}{4}}{\frac{30}{10}} = \frac{\frac{10}{30}}{\frac{4}{15}} = \frac{15}{4} \cdot \frac{10}{30} = 1.25 > 1 \quad (1)$$

Par conséquent, pour un ensemble n d'ordres à exécuter, entièrement ou partiellement, nous avons besoin de déterminer si le produit de chacun des taux de changes des ordres d'achat est supérieur à 1. Si tel est le cas, tous les ordres n peuvent être partiellement ou totalement exécutés [17].

Si on ajoute un troisième acteur, Charlie, dans le contexte suivant : Alice veut échanger x_1 tokens A contre y_1 token B, Bob x_2 tokens B contre y_2 token C, et Charlie x_3 tokens C contre y_3 tokens A. The necessary tokens are present, and the trade is possible if:

$$\frac{x_1 \cdot x_2 \cdot x_3}{y_1 \cdot y_2 \cdot y_3} \geq 1 \quad (2)$$

La rubrique 7.1 fournit davantage de détails sur les ordres Loopring.

4 Participants à l'Ecosystème

Ensemble, les participants à l'écosystème fournissent toutes les fonctionnalités qu'un marché centralisé peut offrir.

- **Portefeuilles** : Un service ou une interface de portefeuille qui donne aux utilisateurs l'accès à leurs tokens, et un moyen de passer des ordres sur le réseau Loopring. Les portefeuilles sont récompensés pour leur passage d'ordres grâce à un partage des frais avec les mineurs d'anneaux (voir section 8). En sachant que le futur de l'échange au lieu de manière sécurisé directement via les portefeuilles, connecter ces réserves de liquidité avec notre protocole est primordial.
- **Blockchain de regroupement pour le partage de liquidité/maille de relais**: Une maille de relais pour les ordres et le partage de liquidités. Lorsque les noeuds exécutent le code des relais Loopring, ils ont la possibilité de rejoindre un réseau existant et de partager la liquidité sur une blockchain de regroupement. La blockchain de regroupement que nous sommes en train de construire permet un partage des ordres quasiment en temps réel (blocs de 1 à 2 secondes) dès la première implémentation, et ordre l'historique trop ancien pour permettre un téléchargement plus rapide par les nouveaux noeuds. A noter que les relais ne sont pas contraints de se joindre à ce regroupement; ils peuvent agir seuls et ne pas partager la liquidité avec les autres, ou, ils peuvent décider de créer et gérer leur propre réseau de partage de liquidité.
- **Relais/Mineurs d'anneaux** : Les relais sont des noeuds qui reçoivent des ordres des portefeuilles ou du relay-mess, et maintiennent des carnets d'ordres

publics et l'historique des échanges, et facultativement, publient les ordres aux autres relais (via n'importe quel support off-chain) et/ou noeuds de la maille de relais. Miner des anneaux d'ordres est une fonctionnalité – et non une nécessité – pour les relais. C'est une tâche conséquente en terme de calculs, et est effectuée entièrement off-chain. Nous appelons les relais avec la fonctionnalité de minage d'anneaux activée les "Mineurs d'Anneaux". Ils produisent les anneaux d'ordre en rapiécant des ordres disparates. Les relais sont libres de choisir (1) comment ils souhaitent communiquer avec les autres, (2) comment ils construisent leurs carnets d'ordres, et (3) comment ils minent leurs anneaux (algorithmes de minage).

- **Smart Contracts du Protocole Loopring (SCPL)** : Un ensemble de smart contracts public et gratuits qui vérifient les anneaux d'ordres reçus des mineurs d'anneaux, règlent et transfèrent sans tiers de confiance les tokens de la part des utilisateurs, récompensent les mineurs et les portefeuilles avec des frais, et publient des événements. Les explorateurs de relais et d'ordres écoutent ces événements pour maintenir leur carnets d'ordres et l'historique des échanges à jour. Voir annexe A pour plus de détails.
- **Services de tokenisation d'actifs (STA)** : Une passerelle pour les actifs qui ne peuvent être échangés directement avec Loopring. Ce sont des services centralisés gérés par des entreprises ou des organisations de confiance. Les utilisateurs déposent des actifs (réels, fiat, ou des tokens d'autres blockchains) et en échange obtiennent des tokens qui peuvent être réclamés lors du dépôt. Loopring n'est pas un protocole d'échange cross-chain (jusqu'à ce qu'une telle solution existe), mais les STA permettent l'échange de tokens ERC20 [18] avec des actifs physique aussi bien qu'avec des tokens issues d'autres blockchains.

5 Processus d'échange

1. **Protocole d'autorisation**: Sur la figure 2, l'utilisateur Y qui veut échanger des tokens autorise le SCPL à traiter un `montantV` du token B que l'utilisateur veut vendre. Cela ne verrouille pas les tokens de l'utilisateur, qui reste libre de les transférer tant que l'ordre est exécuté.
2. **Création de l'ordre** : Le taux actuel et le carnet d'ordres pour le token B contre le token C, sont fournis par les relais ou d'autres agents connectés au réseau, tels que les explorateurs de carnets d'ordres. L'utilisateur Y passe un ordre (ordre à cours limité) en spécifiant le `montantV` et le `montantA` et autres paramètres via n'importe quelle interface de portefeuille intégrée. Un montant de `LRx` peut être adjointe à l'ordre comme frais pour les mineurs d'anneaux; plus

les frais en LRx sont élevés, plus la chance d'être traité tôt par les mineurs d'anneaux est élevée. Le hachage de l'ordre est signé avec la clé privée de Y.

3. **Publication d'ordre** : Le portefeuille envoie l'ordre et sa signature à au moins un relais. Les relais mettent à jour leurs carnets d'ordres. Le protocole ne requière pas que les carnets d'ordres soient construits d'une manière particulière, de type premier arrivé premier servi. A la place, les relais ont le pouvoir de prendre leurs propres décisions de design dans la construction de leurs carnets d'ordres.
4. **Partage de liquidité** : Les relais publient les ordres aux autres relais through par n'importe quel moyen de communication. Encore une fois, une certaine flexibilité est offerte quant à la manière dont les noeuds interagissent. Pour permettre un certain niveau d'interconnectivité, une maille de noeuds de partage de liquidité utilisant une blockchain de regroupement est intégrée par défaut. Comme mentionné dans la section précédente, cette maille de relais est optimisé pour la vitesse et l'intégration.

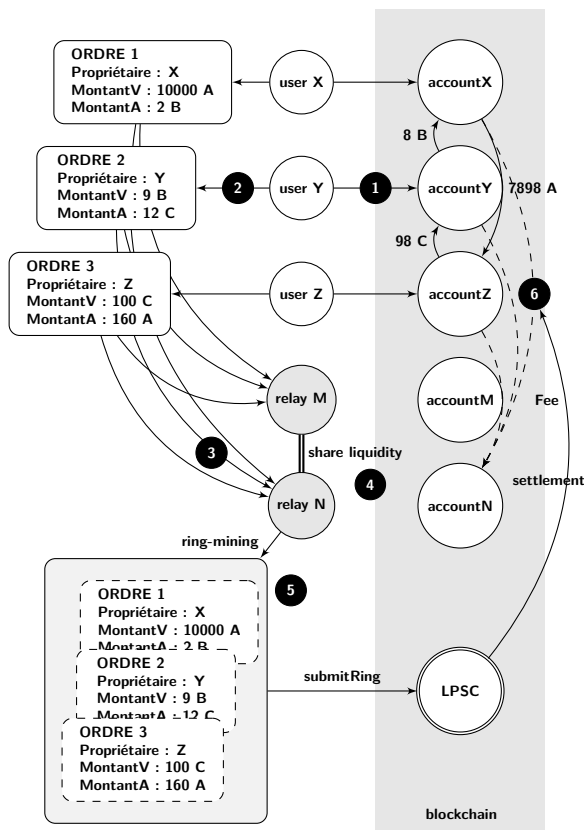


Figure 2: Processus d'échange avec Loopring

5. **Ring-Mining (Order Matching)** : Les mineurs d'anneaux tentent d'exécuter les ordres intégralement ou partiellement au cours actuel ou à un meilleurs cours en les appariant avec plusieurs ordres. Les

mineurs d'anneaux sont la raisons principale pour laquelle le protocole est capable de fournir une liquidité élevée sur n'importe quelle paire. Si le cours exécuté est meilleurs que celui que l'utilisateur Y avait spécifié, la marge est partagée par tous les ordres de l'anneau. A titre de récompense, le mineur d'anneau choisit entre obtenir une part de la marge (partage de marge avec une retour des LRx à l'utilisateur), ou garder la commission fixe en LRx.

6. **Vérification & Règlement** : L'anneau d'ordres est reçu par le SCPL. Il effectue plusieurs vérifications sur les données fournies par le mineur d'anneaux et détermine si l'anneau d'ordres peut être réglé intégralement ou partiellement (selon le taux d'exécution des ordres dans l'anneau et les tokens dans les portefeuilles des utilisateurs). Si tous les critères sont remplis, le contrat effectue le transfert de chaque token aux utilisateurs et paie le mineur d'anneau et les frais du portefeuille à la fois. Si de solde de l'utilisateur Y tel que déterminé par le LPSC n'est pas suffisant, l'ordre sera considéré comme "réduit" : un ordre réduit sera automatiquement remplacé à sa taille originale si les fonds suffisant sont déposés sur son adresse, contrairement à une annulation qui est une opération manuelle et qui n'est pas réversible.

6 Flexibilité opérationnelle

Il est important de souligner que les normes ouvertes de Loopring donnent aux participants une flexibilité important dans leur manière d'opérer. Les différents acteurs sont libres d'implémenter de nouveaux modèles économiques et de créer de la valeur supplémentaire pour les utilisateurs, en engrangeant des commissions sous forme de LRx basées sur le volume d'échange ou d'autres métriques de leur choix. L'écosystème Loopring est modulaire et prévu pour supporter une multitude d'applications.

6.1 Carnet d'ordres

Les relais peuvent concevoir leur carnet d'ordres de n'importe quelle façon tant qu'ils affichent et appariant les ordres des utilisateurs. La première implémentation de notre propre carnet d'ordres suit le modèle OTC (over-the-counter, hors cote en français), dans lequel les ordres à cours limité sont ordonnés uniquement selon leur prix. En d'autres termes, la date et l'heure à laquelle l'ordre a été passé n'a pas d'importance. Cependant, un relay est libre de concevoir ses carnets d'ordres de manière à émuler le comportement typique d'un marché d'échange centralisé, où les ordres sont ordonnés par prix et pour un prix donné par ordre chronologique. If a relay was inclined to offer this type of order book, they can own/integrate with a wallet, and have those wallet orders sent solely to the single relay,

who would then be able to match orders based on time. Toutes les configurations sont possibles.

Alors que d'autres protocoles de DEX nécessitent que les relais aient des ressources propres (un solde initial de tokens pour placer les ordres de l'acheteur), les relais de Loopring ont seulement besoin d'ordres correspondants pour exécuter un échange, et ce sans le moindre token initial.

6.2 Partage de Liquidité

Les relais sont libres de décider comment se partager la liquidité entre eux. Notre blockchain de regroupement n'est que l'une des solutions possible, et l'écosystème peut choisir ses propres moyens de communication. Hormis se joindre à une blockchain de regroupement, ils peuvent aussi construire et gérer la leur, créer des règles et des incitations monétaires comme bon leur semble. Les relais peuvent aussi fonctionner seuls, as seen in the time-sensitive wallet implementation. Of course, there are clear advantages in communicating with other Relays in pursuit of network effects, however, different business models could merit peculiar sharing designs and split fees in any number of ways.

7 Spécifications du Protocole

7.1 Anatomie d'un ordre

Un ordre est un ensemble de données décrivant les intentions d'échange d'un utilisateur. Un ordre Loopring est donné selon le Modèle Uni-Directionnel d'Ordre, ou MOUD, sous la forme suivante :

```
message Order {
    address protocol;
    address owner;
    address tokenS;
    address tokenB;
    uint256 amountS;
    uint256 amountB;
    unit256 lrcFee
    unit256 validSince; // Seconds since epoch
    unit256 validUntil; // Seconds since epoch
    uint8 marginSplitPercentage; // [1-100]
    bool buyNoMoreThanAmountB;
    uint256 walletId;
    // Dual-Authoring address
    address authAddr;
    // v, r, s are parts of the signature
    uint8 v;
    bytes32 r;
    bytes32 s;
    // Dual-Authoring private-key,
    // not used for calculating order's hash,
    // thus it is NOT signed.
    string authKey;
}
```

Pour s'assurer de l'origine d'un ordre, ce dernier est signé avec le hash de ses paramètres, en excluant `authAddr`, et avec la clé privée de l'utilisateur. Le paramètre `authAddr` est utilisé pour signer des anneaux d'ordres dont cet ordre fait partie, ce qui empêche les tentatives de front-running. Veuillez vous référer à la section 9.1 pour plus de détails. La signature est représentée avec les champs `v`, `r`, et `s`, et est envoyée sur le réseau aux côtés des paramètres de l'ordre. Cela garantit l'immutabilité de l'ordre tout au long de son existence. Bien que l'ordre ne change jamais, le protocole peut toujours déterminer son état actuel d'après le solde de son adresse ainsi que d'autres variables.

Le MOUD n'inclut pas de prix (qui est par nature un nombre à virgule flottante), mais utilise plutôt le terme `taux` ou `r`, égal à `MontantV/MontantA`. Le taux n'est pas un nombre à virgule flottante, mais une formule qui sera seulement évaluée à la demande avec d'autres entiers non-signés pour pouvoir garder tous les résultats intermédiaires sous la forme d'entiers non signés et améliorer ainsi la précision des calculs.

7.1.1 Montants d'achat

Lorsqu'un mineur d'anneau arrive à compléter un anneau d'ordres, il est possible qu'un meilleur taux soit exécutable, permettant ainsi aux utilisateurs de recevoir plus de `TokenB` que le `montantA` qu'ils avaient demandé. Cependant, si `buyNoMoreThanAmountB` est défini à `True`, le protocole s'assure que les utilisateurs ne reçoivent pas plus que `montantA` de `tokenB`. Ainsi, le paramètre `buyNoMoreThanAmountB` du MOUD détermine lorsqu'un ordre est considéré comme entièrement exécuté. Le paramètre `buyNoMoreThanAmountB` applique un plafond soit sur `montantV` soit sur `montantA`, et permet aux utilisateurs plus de finesse dans l'expression de leurs intentions d'échange que les ordres traditionnels de type achat/vente.

Par exemple : avec `montantV` = 10 et `montantA` = 2, le taux est $r = 10/2 = 5$. Par conséquent, l'utilisateur est prêt à vendre 5 `TokenV` par `TokenA`. Le mineur d'anneau trouve un anneau avec un taux de 4, permettant à l'utilisateur de recevoir 2.5 `tokenB` au lieu de 2. Mais si l'utilisateur souhaite seulement 2 `tokenB` et définit `buyNoMoreThanAmountB` à `True`, le LPSC effectue une transaction à un taux de 4 `tokenV` pour chaque `tokenA` pour un total de 8 `tokenV`, économisant effectivement 2 `tokenV`. Notons au passage que ce modèle ne prend pas en compte les frais (voir section 8.1).

En effet, si on utilise:

```
Order(montantV,tokenV,
      amountA,tokenA,
      pasplusquetantdetokenA)
```

pour exprimer un ordre sous une forme simplifiée. Avec une paire ETH/USD sur un marché centralisé, le modèle achat/vente traditionnel peut exprimer les première et troisième ordres ci-dessous mais pas les deux autres :

1. Vendre 10 ETH à 300 USD/ETH. Cet ordre peut être exprimé comme: `Order(10, ETH, 3000, USD, Faux)`.
2. Vendre ETH à 300 USD/ETH to get 3000 USD. Cet ordre peut être exprimé de la manière suivante: `Order(10, ETH, 3000, USD, Vrai)`.
3. Acheter 10 ETH à 300 USD/ETH, cet ordre peut être exprimé de la manière suivante: `Order(3000, USD, 10, ETH, Vrai)`.
4. Dépenser 3000 USD pour acheter d'autant d'ETH que possible à 300 USD/ETH, cet ordre peut être exprimé de la manière suivante: `Order(3000, USD, 10, ETH, Faux)`.

7.2 Vérification d'un anneau

Les smart contracts de Loopring n'effectuent pas les calculs de taux ou de montants, mais doivent recevoir et vérifier les valeurs que les mineurs d'anneaux. Ces calculs sont faits par les mineurs d'anneaux pour deux raisons majeures: (1) le langage de programmation utilisé pour l'écriture des smart contracts, comme solidity[19] sur Ethereum, les calculs à virgule flottante, surtout pour $pow(x, 1/n)$ (calcul de la racine n-ième d'un nombre à virgule flottante), et (2) il est souhaitable que le calcul soit effectué off-chain pour alléger le coût et l'effort sur la blockchain.

7.2.1 Validation de sous-anneaux

Cette étape empêche les arbitrageurs de réaliser une marge injustement sur un anneau d'ordres en implémentant de nouveaux ordres en son sein. L'idée est que dès qu'un ordre valide est trouvé par un mineur d'anneau, il pourrait être tentant d'ajouter de nouveaux ordres à l'anneau pour absorber toute marge des utilisateurs (rate discounts). Comme illustré sur la figure 3 ci-dessus, un calcul précis de x_1 , y_1 , x_2 and y_2 rendra le produit du taux de tous les ordres exactement égal à 1 ce qui empêche tout .

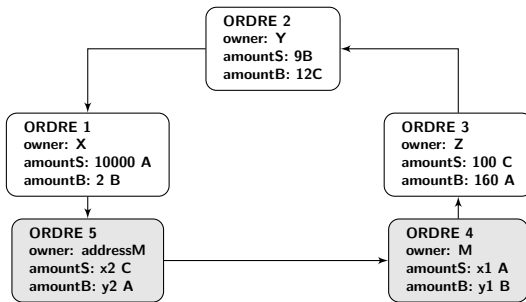


Figure 3: Un anneau avec un sous-anneau

Ce mouvement n'implique aucun risque et n'apporte aucune valeur au réseau, et est considéré comme un comportement injuste par le mineur d'anneau. Pour empêcher ceci, Loopring nécessite qu'un anneau valide ne peut contenir aucun sous-anneau. Pour ce faire, le LPSC s'assure qu'un

token ne peut être plus d'une fois un token à vendre ou à acheter. Dans le diagramme ci-dessus, on voit que le token A est vendu et acheté deux fois, ce qui devrait être rejeté.

7.2.2 Vérification du taux d'exécution

Le calcul des taux de change au sein de l'anneau d'ordres est effectué par les mineurs d'anneaux pour les raisons citées plus haut. C'est au LPSC de s'assurer qu'ils sont corrects. D'abord, il vérifie que le taux auquel le mineur d'anneau peut exécuter chaque ordre est inférieur ou égal au taux initialement demandé par l'utilisateur. Cela apporte à l'utilisateur la garantie d'obtenir un taux au moins aussi bon que celui demandé. Dès que les taux de change sont confirmés, dans le cas où un taux meilleur que celui demandé est possible (il en résulte un rabais sur le taux demandé), le LPSC s'assure que tous les ordres composant l'anneau partagent le même taux réduit. Par exemple, si le rabais est de γ , alors le prix pour chaque ordre est de :

$r_{0 \rightarrow 1} \cdot (1 - \gamma)$, $r_{1 \rightarrow 2} \cdot (1 - \gamma)$, $r_{2 \rightarrow 0} \cdot (1 - \gamma)$, et satisfait à l'équation suivante :

$$r_{0 \rightarrow 1} \cdot (1 - \gamma) \cdot r_{1 \rightarrow 2} \cdot (1 - \gamma) \cdot r_{2 \rightarrow 0} \cdot (1 - \gamma) = 1 \quad (3)$$

ainsi :

$$\gamma = 1 - \frac{1}{\sqrt[3]{r_{0 \rightarrow 1} \cdot r_{1 \rightarrow 2} \cdot r_{2 \rightarrow 0}}} \quad (4)$$

Pour une transaction comprenant n ordres, la réduction est égale à :

$$\gamma = 1 - \frac{1}{\sqrt[n]{\prod_{i=0}^{n-1} r^i}} \quad (5)$$

où r^i est le taux d'exécution du i -ième ordre.

Retournons à notre exemple précédent dans lequel Alice a 15 tokens A and veut 4 tokens B en échange, Bob a 10 tokens B et veut 30 tokens A en échange. Si le token A sert de référence, alors Alice achète le token B au cours $\frac{15}{4} = 3.75A$, tandis que Bob vend le token B pour $\frac{30}{10} = 3.00A$. Pour calculer le t: $\frac{150}{120} = 1.25$ thus $\frac{1}{1.25} = 0.8 = (1 - \gamma)^2$. Thus the exchange rate that renders the trade equitable for both parties is $\sqrt{0.8} \cdot 3.75 \approx 3.3541$ token A per token B.

Bob gives 4 token B and receives 13.4164 token A, more than the 12 he was expecting for those 4 tokens. Alice receives 4 token B as intended but gives only 13.4164 token A in exchange, less than the 15 she was willing to give for those 4 tokens. Note, a portion of this margin will go towards paying fees to incentivize miners (and wallets). (See section 8.1).

7.2.3 Taux de remplissage et annulation

A user can partially or fully cancel an order by sending a special transaction to the LPSC, containing the details about the order and the amounts to cancel. The LPSC takes that into account, stores the amounts to cancel, and emits an `OrderCancelled` event to the network. The LPSC keeps track of filled and cancelled amounts by storing their values

using the order's hash as an identifier. This data is publicly accessible and **OrderCancelled** / **OrderFilled** events are emitted when it changes. Tracking these values is critical for the LPSC during the order-ring settlement step.

LPSC also supports cancelling all orders for any trading pair with the **OrdersCancelled** event and cancelling all orders for an address with the **AllOrdersCancelled** event.

7.2.4 Order Scaling

Orders are scaled according to the history of filled and cancelled amounts and the current balance of the senders' accounts. The process finds the order with the smallest amount to be filled according to the above characteristics and uses it as a reference for scaling all transactions in the order-ring.

Finding the lowest value order can help to figure out the fill volume for each order. For instance, if the i -th order is the lowest value order, then the number of tokens sold from each order \hat{s} and number of tokens purchased \hat{b} from each order can be calculated as:

$$\begin{aligned}\hat{s}^i &= \bar{s}_i, \hat{b}^i = \hat{s}^i / \hat{r}^i, ; \\ \hat{s}^{i+1} &= \hat{b}^i, \hat{b}^{i+1} = \hat{s}^{i+1} / \hat{r}^{i+1}; \\ \hat{s}^{i+2} &= \hat{b}^{i+1}, \hat{b}^{i+2} = \hat{s}^{i+2} / \hat{r}^{i+2}; \\ &\dots\end{aligned}$$

avec \bar{s}_i représentant le solde une fois que les ordres ont été partiellement exécutés.

Lors de l'implémentation, on peut sans aucun souci supposer que n'importe quel ordre dans l'anneau d'ordres a la valeur la plus faible, et ensuite itérer sur l'anneau d'ordres deux fois au plus pour calculer chaque volume d'exécution.

Exemple : Si le plus petit montant à être exécuté par rapport à l'ordre original est de 5%, toutes les transactions de l'anneau d'ordres seront réduites à 5%. Une fois les transactions exécutées, l'ordre qui était considéré comme ayant le plus faible montant restant à exécuter devrait être complètement exécuté.

7.3 Règlement d'anneau

Si un anneau d'ordres remplit toutes les conditions requises, l'anneau d'ordres peut être fermé, et les transaction effectuées. Cela signifie que n ordres forment un anneau d'ordres fermé, tel qu'illustré sur la figure 4:

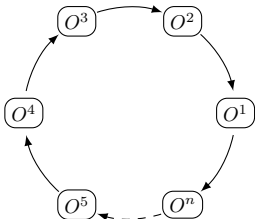


Figure 4: Ring Settlement

Pour exécuter une transaction, le LPSC utilise le smart contract **TokenTransferDelegate** ("délégation du transfert de tokens" en français). L'inclusion d'un tel système de délégation simplifie le protocole puisque tous les ordres ont seulement besoin d'autoriser le délégué au lieu des différentes versions potentielles du protocole.

Pour chaque ordre de l'anneau, un règlement de **tokenV** est versé à l'ordre suivant ou précédant selon le type d'implémentation. Ensuite, la commission du mineur d'anneau est versée selon le modèle de rémunération choisi par le mineur d'anneau. Enfin, une fois toutes transactions effectuées, un événement de type **RingMined** ("Anneau-Miné" en français) est émis.

7.3.1 Evènements émis

Le protocole émet des événements qui permettent aux relais, aux explorateurs de carnets d'ordres, et aux autres acteurs de recevoir des mises à jour des carnets d'ordres de la manière la plus efficace possible. Les événements émis sont les suivants :

- **OrderCancelled** : Un ordre a été annulé.
- **OrdersCancelled** : Tous les ordres d'une paire d'une adresse sont annulés.
- **AllOrdersCancelled** : Tous les ordres de toutes les paires d'une adresse ont été annulés
- **RingMined** : Un anneau-order a été exécuté. Cet événement contient toutes les données relatives au transfert de chaque anneau intérieur.

8 Les Tokens LRx

LRx est la notation générique de nos tokens. LRC est le token Loopring sur Ethereum, LRQ sur Qtum, et LRN sur NEO, etc. D'autres types de tokens LRx seront ajoutés à l'avenir lorsque Loopring sera déployé sur d'autres blockchains publiques.

8.1 Modèle de frais

Lorsqu'un utilisateur passe un ordre, il précise une commission en LRx à payer au mineur d'anneau, ainsi qu'un pourcentage de la marge effectuée sur l'ordre (**marginSplitPercentage**) à laquelle le mineur d'anneau peut prétendre. C'est ce qu'on appelle le partage de marge. La décision de choisir le type de commission (commission fixe ou marge partagée) appartient au mineur d'anneau.

Le partage de marge peut être illustré de la manière suivante :

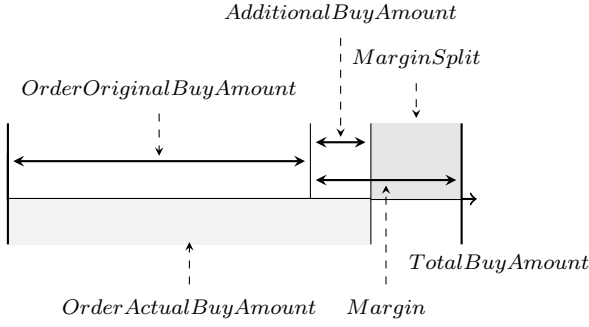


Figure 5: Une marge partagée de 60%

Si la marge d'un anneau d'ordres est trop faible, un mineur d'anneau choisira la commission fixe en LRx. Si au contraire, la marge est conséquente au point que la marge partagée vaille plus que la commission fixe, alors le mineur d'anneau choisira la marge partagée. Il existe une clause supplémentaire cependant : lorsque le mineur d'anneau choisit la marge partagée, il doit payer une commission à l'utilisateur (le créateur de l'ordre) when the ring-miner chooses the margin split, they must pay the user (order creator) a fee, qui est égale au montant en LRx que l'utilisateur aurait dû payer au mineur d'anneau comme commission. Cela porte le seuil à partir duquel le mineur d'anneau choisira le partage de marge, au double de la commission fixe de l'ordre, ce qui tend à rendre plus fréquent le choix d'une commission fixe en LRx. Cela permet au mineur d'anneau de lisser son revenu sur des anneaux d'ordres à faible marge, au détriment d'une marge partagée sur les anneaux d'ordres à marge élevée. Notre modèle se base sur notre prévision qu'avec la croissance du marché, les anneaux d'ordres à marge élevée seront de plus en plus rares, ce qui incitera des commissions fixes en LRx.

Nous nous retrouvons avec le graphe suivant :

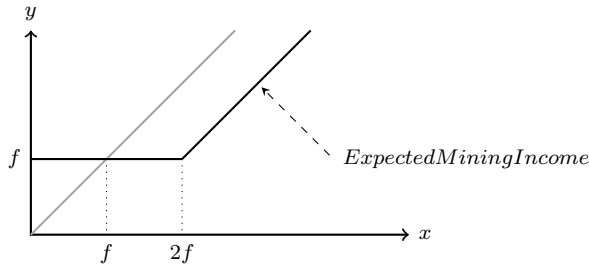


Figure 6: Le modèle Loopring's Fee Model

où f est la commission fixe en LRx, x est la marge partagée, y est le revenu du minage. $y = \max(f, x - f)$ comme indiqué par la ligne pleine; si la commission fixe pour l'ordre est de 0, l'équation devient $y = \max(0, x - 0)$ ce qui peut être simplifié en $y = x$ tel qu'indiqué par la ligne grise.

On peut en déduire les points suivants :

1. Si la marge partagée est de 0, les mineurs d'anneaux choisiront la commission fixe et sont toujours récompensés.

2. Si la commission fixe est de 0, la ligne grise indique un revenu selon un modèle linéaire.
3. Dès lorsque la marge partagée est supérieure à deux fois la commission fixe, le mineur d'anneau choisit la marge partagée, et paie un montant en LRx à l'utilisateur.

Il faut garder à l'esprit que si la commission fixe est supérieure à zéro, quelle que soit l'option choisie par le mineur, il y aura toujours un transfert de LRx entre le mineur d'anneau et l'émetteur de l'ordre. En effet, soit le mineur reçoit une commission en LRx, soit il paie la commission en retour à l'émetteur pour prendre la marge partagée.

Les mineurs d'anneaux partageront un certain pourcentage des commissions avec les portefeuilles. Lorsqu'un utilisateur place un ordre via un portefeuille et que cet ordre est exécuté, le portefeuille est récompensé par une portion des commissions ou du partage de marge. Bien que ce soit modulaire, et que des modèles économique et implémentations variés sont possibles, nous recommandons qu'un portefeuille reçoive environ 20 25% des commissions de l'ordre. Les portefeuilles sont une cible importante du protocole Loopring parce qu'ils sont nécessaires pour les utilisateurs, mais n'ont habituellement pas ou peu de sources de revenus.

8.2 Gouvernance décentralisée

Le protocole Loopring est un protocole social dans le sens où il dépend de la coopération des membres pour fonctionner efficacement dans un but commun. Ce n'est pas sans rappeler les autres protocoles cryptos en général, et en pratique, son utilité repose sur les mêmes mécanismes de coopération [20], de stratégie dite "Grim Trigger equilibrium", et de rationalité limitée. A cette fin, les tokens LRx ne sont pas seulement au paiement des commissions liées aux échanges, mais aussi de récompenser l'ensemble des participants au réseau. Un tel alignement est nécessaire à l'adoption de n'importe quel protocole, mais l'est d'autant plus pour les protocoles d'échange, étant donné que le succès est principalement liée à l'amélioration de la liquidité dans une écosystème décentralisé robuste.

Les tokens LRx seront aussi utilisés pour permettre des mises à jour du protocole via une gouvernance décentralisée. Les mises à jour des smart contracts seront gouvernées par les détenteurs de tokens LRx pour assurer la continuité et la sécurité du réseau, et aussi éviter les baisses de liquidité causées par les problèmes d'incompatibilité. Etant donné que les smart contracts ne peuvent être modifiés après leur déploiement, il existe un risque que les dApps ou les utilisateurs continuent d'interagir avec des versions obsolètes, s'excluant ainsi de l'utilisation des smart contracts à jour. La capacité d'être mis à jour est crucial dans le succès du protocole car il n'a d'autre choix que de s'adapter aux demandes du marché et aux blockchains sous-jacentes. La

gouvernance décentralisée par les détenteurs de LRx permettra de mettre à jour les smart contracts sans interrompre l'utilisation des dApps ou des utilisateurs ni dépendre excessivement de l'abstraction de smart contracts. Au départ, cela se déroulera grâce à une simple multisignature des smart contracts pour à terme évoluer vers un mécanisme de gouvernance de type DAO.

9 Protection contre les fraudes et attaques

9.1 Protection contre le front-running

Dans le contexte des échanges décentralisés, le front-running correspond à la tentative pour quelqu'un de copier la solution d'un anneau d'un autre noeud, et de miner le bloc correspondant avant la transaction originale qui se trouve encore dans le pool de transactions en attente (mempool). Cela peut se réaliser en offrant une commission plus élevée (prix du gas). Le principal procédé de front-running avec Loopring (et de n'importe quel protocole d'appariement d'ordres) est le vol d'ordre : lorsqu'un front-runner vole un ordre ou plus d'un anneau d'ordres en attente de validation de transaction; et plus spécifique à Loopring, lorsqu'un front-runner vole l'anneau d'ordres entier d'une transaction en attente.

Lorsqu'une transaction de type "submitRing" ("soumission d'anneau" en français) n'est pas encore confirmée et toujours dans le pool de transactions en attente, n'importe qui peut facilement détecter une telle transaction et remplacer le champ "minerAddress" ("adresse du mineur" en français) avec leur propre adresse, la **filcherAddress** (adresse du voleur en français), il est alors possible que le voleur puisse resigner l'ensemble avec l'adresse **filcherAddress** pour remplacer la signature de l'anneau d'ordres original. Le voleur peut définir un prix du gas plus élevé, et soumettre une nouvelle transaction en espérant que les mineurs l'incluent dans le premier bloc suivant au lieu de la transaction submitRing originale.

Les précédentes solutions pour à ce problème comptaient des inconvénients importants : nécessiter plus de transaction et ainsi coûter plus de gas aux mineurs d'anneaux, et prendre au moins le double de blocs pour régler un anneau d'ordres. Notre nouvelle solution, le "Dual Authoring" [21] ("double paternité" en français), implique un mécanisme à double niveau d'autorisation pour les ordres : un pour le règlement, un pour le minage d'anneaux.

Processus de "Dual Authoring" :

1. Pour chaque ordre, le code du portefeuille va générer un couple clé publique/clé privée aléatoire, et placer cette clé dans le JSON de l'ordre. Une alternative peut consister à prendre une adresse dérivée de la clé publique au lieu de la clé publique elle-même afin de réduire le nombre d'octets requis. Nous utilisons le champ **authAddr** (adresse d'autorisation en français)

pour stocker cette adresse, et **authKey** pour stocker la clé privée correspondant à **authAddr**).

2. Le traitement du hash de l'ordre avec tous ses champs a lieu, à part **r**, **v**, **s**, et **authKey**, et le hash est signé en utilisant la clé privée de l'émetteur (et non pas avec **authKey**).
3. Le portefeuille envoie l'ordre avec la clé dans **authKey** aux relais pour être miné. Les mineurs d'anneaux vérifient que **authKey** et **authAddr** correspondent, et que la signature de l'ordre est valide pour l'adresse de l'émetteur.
4. Lorsqu'un anneau d'ordre est identifié, le mineur d'anneaux utilise chacune des clés **authKey** pour signer le hash de l'anneau, l'adresse **minerAddress**, et tous les autres paramètres de minage. Si un anneau d'ordres contient n ordres, il y a alors n signatures pour chacune des n clés **authKey**. Nous appelons ces signatures les **authSignatures**. Le mineur peut aussi avoir besoin de signer le hash de l'anneau avec tous les paramètres associés en utilisant la clé privée de son adresse **minerAddress**.
5. Le mineur d'anneaux appelle la fonction **submitRing** avec tous ses paramètres, ainsi que les signatures **authSignature** additionnelles. À noter que les clés dans le champ **authKey** ne font PAS partie de la transaction on-chain, et restent donc inconnues des participants autres que le mineur d'anneaux.
6. Le protocole Loopring vérifie maintenant chaque signature **authSignature** avec l'adresse **authAddr** de chaque ordre, et rejette l'anneau si **authSignature** est manquante ou invalide.

On se retrouve dans la situation suivante :

- La signature de l'ordre (par la clé privée de l'adresse de l'émetteur de l'ordre) garantit que l'ordre ne peut être modifié, y compris l'adresse **authAddr**.
- La signature du mineur d'anneau (par la **minerAddress**), si fournie, garantit que personne ne peut se faire passer pour lui pour miner l'anneau d'ordres.
- La signature **authSignature** garantit que l'intégralité de l'anneau d'ordres ne peut être modifié, y compris le champ **minerAddress**, et qu'aucun ordre ne peut être volé.

Le "Dual Authoring" empêche le vol d'anneaux et d'ordres tout en s'assurant que le règlement des anneaux d'ordres peut être effectué en une seule transaction. De plus, le "Dual Authoring" offre l'opportunité aux relais de partager des ordres de deux façons : , Dual Authoring opens doors for relays to share orders in two ways: non-matchable sharing and matchable sharing. By default,

Loopring operates an OTC model and only supports limit-price orders, meaning that orders' timestamps are ignored. This implies that front-running a trade has no impact on the actual price of that trade, but does impact whether it gets executed or not.

10 Autres types d'attaques

10.1 Attaque de type Sybil ou DoS

Des utilisateurs malveillants, sous une identité pourraient envoyer un très grand nombre d'ordres pour tenter de saturer les noeuds de Loopring. Mais puisque le protocole autorise les noeuds à accepter ou rejeter les ordres selon leurs propres critères, qui ne sont pas forcément publics, la plupart de ces ordres seraient rejetés faute de générer un profit lorsqu'appariés. En laissant la possibilité aux relais de gérer les ordres comme bon leur semble, une attaque de ce type n'est pas considérée comme une menace pour le protocole Loopring.

10.2 Solde insuffisant

D'autres utilisateurs malveillants peuvent signer des ordres dont la valeur serait non-nulle, mais dont les adresses auraient un solde nul. Les noeuds pourraient surveiller et remarquer que le solde de ces ordres est en réalité nul ou insuffisant, et simplement mettre à jour le statut de ces ordres afin de s'en débarrasser. Les noeuds doivent en effet consacrer un certain temps pour effectuer cette mise à jour, mais cet effort peut être minimisé en mettant sur une liste noires les adresses et en ignorant tous les ordres associés.

11 Résumé

The Loopring protocol sets out to be a foundational layer for decentralized exchange. In so doing, it has profound repercussions in how people exchange assets and value. Money, as an intermediate commodity, facilitates or replaces barter exchange and solves the double coincidence of wants problem [22], whereby two counterparties must desire each other's distinct good or service. Similarly, Loopring protocol aims to dispense of our dependencies on coincidence of wants in trading pairs, by using ring matching to more easily consummate trades. This is meaningful for how society and markets exchange tokens, traditional assets, and beyond. Indeed, just as decentralized cryptocurrencies pose threat to a nation's control over money, a combinatorial protocol that can match traders (consumers/producers) at scale, is a theoretical threat to the concept of money itself.

Les principaux avantages du protocole sont les suivants :

- Une gestion des ordres off-chain et des règlements on-chain, ce qui permet de procurer une sécurité efficace sans compromis sur la performance.

- Une amélioration de la liquidité grâce aux minage d'anneaux et aux partage d'ordres.
- Une double signature des ordres qui permet d'éviter le risque de front running rencontré par tous les DEX et leurs utilisateurs aujourd'hui.
- Un ensemble de smart contracts gratuits, publics permettant à n'importe quelle dApp de utiliser ou interagir avec le protocole.
- Une standardization au sein des opérateurs qui permet de profiter d'effets de réseaux et d'une amélioration de l'expérience utilisateur.
- Un réseau flexible dans sa manière de communiquer et de gérer des carnets d'ordres.
- Une réduction des barrières à l'entrée signifiant une réduction des coûts pour les noeuds rejoignant le réseau et pour les utilisateurs finaux.
- Un trading anonyme directement à partir du portefeuille des utilisateurs.

12 Remerciements

Nous aimerions exprimer notre gratitude envers nos mentors et conseillers, et aux nombreuses personnes de la communauté qui ont été si accueillants et généreux lorsqu'il a été question de partager leurs connaissances. Nous aimerions plus particulièrement remercier Shuo Bai (de ChinaLedger), le Professeur Haibin Kan, Alex Cheng, Hongfei Da, Yin Cao, Xiaochuan Wu, Zhen Wang, Wei Yu, Nian Duan, Jun Xiao, Jiang Qian, Jiangxu Xiang, Yipeng Guo, Dahai Li, Kelvin Long, Huaxia Xia, Jun Ma, et Encephalo Path pour avoir examiné et critiqué notre projet.

References

- [1] Vitalik Buterin. Ethereum: a next generation smart contract and decentralized application platform (2013). URL {<http://ethereum.org/ethereum.html>}, 2017.
- [2] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151, 2014.
- [3] Patrick Dai, Neil Mahi, Jordan Earls, and Alex Norta. Smart-contract value-transfer protocols on a distributed mobile application platform. URL: <https://qtum.org/uploads/files/cf6d69348ca50dd985b60425ccf282f3.pdf>, 2017.
- [4] Viktor Atterlön. A distributed ledger for gamification of pro-bono time, 2018.
- [5] Hernando de Soto. *The Mystery Of Capital*. Basic Books, 2000.

- [6] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
- [7] Fabian Schuh and Daniel Larimer. Bitshares 2.0: Financial smart contract platform, 2015.
- [8] Bancor protocol. URL <https://bancor.network/>, 2017.
- [9] Yaron Velner Loi Luu. Kybernetwork: A trustless decentralized exchange and payment service. <https://kr.kyber.network/assets/KyberNetworkWhitepaper.pdf>, Accessed: 2018-03-05.
- [10] Fortune. How to steal \$500 million in cryptocurrency. <http://fortune.com/2018/01/31/coincheck-hack-how>, Accessed: 2018-03-30.
- [11] Robert McMillan. The inside story of mt. gox, bitcoin’s 460 dollar million disaster. 2014.
- [12] Sylvain Ribes. Chasing fake volume: a crypto-plague. <https://medium.com/@sylvainartplayribes/chasing-fake-volume-a-crypto-plague-ea1a3c1e0b5e>, Accessed: 2018-03-10.
- [13] Rossella Agliardi and Ramazan Gen?ay. Hedging through a limit order book with varying liquidity. 2014.
- [14] Will Warren and Amir Bandeali. 0x: An open protocol for decentralized exchange on the ethereum blockchain, 2017.
- [15] Iddo Bentov and Lorenz Breidenbach. The cost of decentralization. <http://hackingdistributed.com/2017/08/13/cost-of-decent/>, Accessed: 2018-03-05.
- [16] Daniel Wang. Coinport’s implemenation of udom. <https://github.com/dong77/backcore/blob/master/coinex/coinex-backend/src/main/scala/com/coinport/coinex/markets/MarketManager.scala>, Accessed: 2018-03-05.
- [17] Supersymmetry. Remarks on loopring. <https://docs.loopring.org/pdf/supersimmetry-loopring-remark.pdf>, Accessed: 2018-03-05.
- [18] Fabian Vogelsteller. Erc: Token standard. URL <https://github.com/ethereum/EIPs/issues/20>, 2015.
- [19] Chris Dannen. *Introducing Ethereum and Solidity*. Springer, 2017.
- [20] Vitalik Buterin. Notes on blockchain governance. <https://vitalik.ca/general/2017/12/17/voting.html>, Accessed: 2018-03-05.
- [21] Daniel Wang. Dual authoring i^a loopringi’s solution to front-running. URL <https://medium.com/loopring-protocol/dual-authoring-looprings-solution-to-front-running-d0fc9c348ef1>, 2018.
- [22] Nick Szabo. Menger on money: right and wrong. <http://unenumerated.blogspot.ca/2006/06/menger-on-money-right-and-wrong.html>, Accessed: 2018-03-05.

Appendices

Appendix A Loopring sur Ethereum

A.1 Smart Contracts

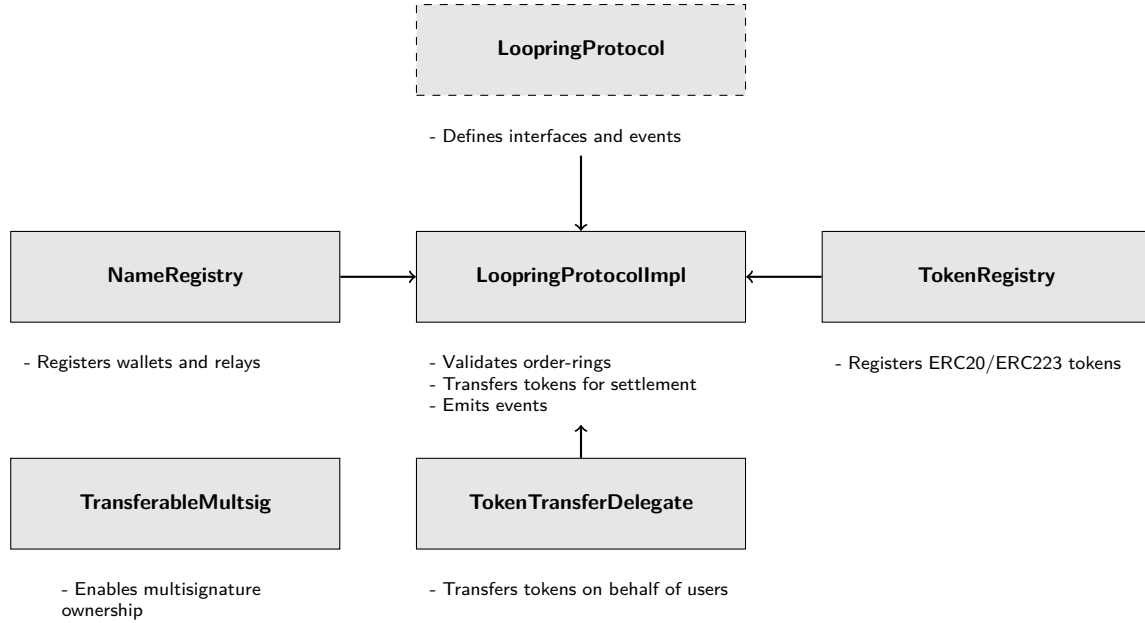


Figure 7: Smart Contracts

A.2 Deployment

Les smart contracts suivant ont été déployés sur le mainnet Ethereum :

- LRC: 0xEF68e7C694F40c8202821eDF525dE3782458639f
- TokenRegistry: 0xa21c1f2AE7f721aE77b1204A4f0811c642638da9
- TokenTransferDelegate: 0xc787aE8D6560FB77B82F42CED8eD39f94961e304
- NameRegistry: 0x0f3Dce8560a6010DE119396af005552B7983b7e7
- LoopringProtocolImpl: 0xc80BbAb86cED62CF795619A357581FaF0cB46511
- TransferableMultisig: 0x7421ad9C880eDF007a122f119AD12dEd5f7C123B