Loopring:
分散型トークン取引所プロトコル

Daniel Wang
daniel@loopring.org

Jay Zhou
jay@loopring.org

Alex Wang
alex@loopring.org

Matthew Finestone
matt.finestone@gmail.com

https://loopring.org

April 6, 2018

Abstract

Loopring は分散型取引所を構築するためのオープンプロトコルです。Loopring は、注文を集約し伝達するオフ・チェーンのアクターグループが集まり、注文の取引と決済の責任を負う一連のパブリックなスマートコントラクトとして機能します。プロトコルは自由で拡張性があり、交換機能を組み込んだ分散アプリケーション（dApp）用の標準化された構築用ブロックとして機能します。その相互運用可能な標準は、トラストレスな匿名取引を容易にします。現在の分散型取引所プロトコル上の重要な改善点は、2 つのトークン取引ペアの制限を取り除き、異なるオーダー同士をミックスして一致させる能力であり、流動性を大幅に改善することです。Loopring ではフロントランニング（オリジナルのソリューションプロバイダよりも早くブロックにトランザクションを登録する不公平な試行）を防止するための独自で頑強なソリューションを使っています。Loopring はブロックチェーンに依存せず、スマートコントラクト機能を備えたいかなるブロックチェーンにもデプロイすることができます。執筆時点では、イーサリアム [1] [2] で動作可能であり、Qtum[3]、NEO[4] は準備中です。

1 導入

ブロックチェーンベースの資産が急増したため、取引相手との間でこれらの資産を交換する必要性が大幅に増加しました。従来の資産のトークン化を含む、何千もの新しいトークンが導入されているため、このニーズは拡大しています。投機的な取引を目的としたトークン交換、または、ネットワークへアクセスするためのネイティブなユーティリティトークンへの交換にかかわらず、1 つの暗号資産を別の資産に交換する能力は、より大きなエコシステムの基礎になります。資産には潜在的なエネルギーが確かに存在 [5] しており、（資本を開放する）このエネルギーを実現するには、ブロックチェーンが永久に認めている所有権を主張するだけでなく、これらの資産を自由に移転し変換する能力が必要です。

そのため、トラストレスなトークン（価格）の交換は、ブロックチェーン技術の魅力的な使用例です。しかしながら、今では暗号通貨の熱心家は、伝統的な中央集権型取引所でのトークン取引で大部分を清算しています。Loopring が必要とされる理由として、Bitcoin[6] がピアツーピアの電子キャッシュに関して、「もし信頼できる第三者が以前として二重支払を防止する必要がある場合多くの利点が失われる」ことを忠実に強調したことと同様です。分散型資産もまた、信頼された、閉鎖的な中央集権型の取引所を通らなければならないのであれば、多くの利点が失われます。

中央集権型取引所上での分散型トークンの取引は、哲学的観点からは意味をなさない。なぜなら、これらの分散型プロジェクトが支持する長所を支えることができないからである。中央集権型取引所を使用する際には以下に説明されるように、多くの実用的なリスクと限界がある。分散型取引所（DEX）[7] [8] [9] は、これらの問題に取り組むことを模索しており、多くの場合、直接取引のためにブロックチェーンを使用することによってセキュリティリスクを軽減することに成功している。”しかしながら、DEX の特性が新しい経済にとって重要なインフラストラクチャーになるにつれ、パフォーマンスの改善の余地はかなりあります。Loopring は dApp に依存しないオープンなプロトコルで、インフラストラクチャのための組み立て式ツールを提供することを目指しています。

2 現在の取引所状況

2.1 中央集権型取引所の不十分さ

中央集権型取引所の 3 つの主要なリスクは、1）安全性の欠如、2）透明性の欠如、3）流動性の欠如です。

セキュリティの欠如は、概して、ユーザーが自分の秘密鍵（資産）の制御をある中央集権型のエンティティに委譲することから発生します。これにより、ユーザーは中央集権型取引所が悪意のあるハッカーの餌食となる可能性にさらされます。すべての中央集権型取引所が直面するセキュリティとハッキングのリスクはよく知られていますが [10] [11]、未だにトークン取引において「当たり前のこと」としてよく受け入れられています。中央集権型取引所は、自身のサーバーに何百万ドルものユーザー資

産を保有しているため、ハッカーにとって格好の的となり続けています。取引所の開発者は、ユーザーの資産で、過失的なエラーを引き起こすこともあります。簡単に言えば、ユーザーは中央集権型取引所に入金したときに自分のトークンの管理権限を失うことになります。

透明性の欠如ユーザーを不正交換取引のリスクに不当にさらします。ここにおける特異性は取引所の悪意ある意図によるものであり、つまり中央集権型取引所においてユーザーは自身の資産を取引するというよりも、IOU を取引するということです。トークンが取引所のウォレットに送られると、その取引所に保管され、代わりに IOU として提供します。そして、すべての取引は実質的にユーザーの IOU の間で行われます。出金するには、ユーザーは自身の IOU を償還し、トークンを外部ウォレットアドレスで受け取ります。これらの一連のプロセスは透明性が欠如しており、取引所が口座の停止、凍結、破産などを引き起こす可能性があります。また、ユーザーの資産が取引所の管理下にある間、それらの資産を第三者に貸し出すなど、他の目的で使用される可能性もあります。透明性の欠如は、取引手数料の増加、需要のピーク時の遅延、規制上のリスク、フロントランニング取引など、資産額の損失を生じさせることなく利用者を犠牲にする可能性があります。

流動性の欠如。取引所の観点から、2 つの勝者総取りのシナリオにより、断片化された流動性は新規取引所の参入を抑制します。一つ目のシナリオは、最も多くの取引ペアをもつ取引所が勝利するということです。なぜなら、ユーザーがすべての取引を一つの取引所で行うことが望ましいと感じるからです。二つ目のシナリオは、最も大きいオーダーブックを抱える取引所が勝利するということです。なぜなら、それぞれの取引ペアにおいて有利な売買スプレッドになるためです。これは新規参入者にとって初期の流動性を構築することが困難であるため、競争意欲を削いでしまいます。結果として、ユーザーからの苦情や大きなハッキング事件が起きてもなお、多くの取引所が高いマーケットシェアを持つこととなります。中央集権型の取引所は市場シェアを獲得するにつれて大きな価値を生むため、これまで以上に大きなハッキングの標的になります。

ユーザーの観点からは、断片化された流動性・換金能力はユーザー体験を大幅に低下させます。中央集権型の取引所では、ユーザーは取引所の流動性プールおよびオーダーブックの中でサポートされているトークンペア間でのみ取引することができます。トークン`A` をトークン`B` に交換するには、トークンを両方ともサポートする取引所に行くか、個人情報を開示して異なる取引所に登録する必要があります。ユーザは、通常、BTC または ETH に対して予備的または中間的な取引をする必要があり、そのプロセスでは売買のスプレッドを支払う必要があります。最後に、オーダーブックは、注文した価格と実際に約定された価格差なしに取引を完了するのに十分な量がないかもしれません。たとえ取引所が大量処理を目的としていたとしても、この量と流動性が偽ではないという保証はありません [12]。

その結果は古びた金融システムのように、流動性のないサイロと断片化されたエコシステムとなり、大きなボリュームが少数の取引所に集中します。ブロックチェーンのグローバルな流動性の保証は中央集権型取引所においてはメリットがありません。

2.2 分散型取引所の不十分さ

分散型取引所は中央集権型取引所とは異なり、ユーザーは基盤となるブロックチェーン上で直接的に取引を実行することによって自身の秘密鍵（資産）の管理を維持するからです。トラストレスな暗号化技術を活用することで、セキュリティを取り巻く上述のリスクの多くを首尾よく緩和します。しかしながら、性能及び構造上の限界に関して問題が依然として存在します。

流動性は、ユーザーが異なる流動性プールや標準を超えて取引相手を探す必要があるため、多くの場合問題となります。もし大規模な DEX や dApps が相互運用するための一貫した標準を採用していない場合や、注文が幅広いネットワーク全体で共有/伝播されていない場合、断片的な流動性の影響は存在します。指値注文の流動性、具体的には、指値注文の再生成ーいかに早く指値注文を再注文できるかーは、最適なトレーディング戦略に大きな影響を及ぼしかねません [13]。そのような標準が存在しないことは、流動性の低下だけでなく、潜在的に安全性に欠ける独自のスマートコントラクトにさらされていることにもつながります。

さらに、オンチェーンで取引が行われるため、DEX はスケーラビリティ、実行の遅延（マイニング）、取引注文の変更に対するコストなど、基盤となるブロックチェーンの制限を継承します。したがって、ブロックチェーンのコードを実行するとコスト（ガス）が発生し、複数の注文取消は非常に高価になるため、ブロックチェーンのオーダーブックは特にうまく拡張できません。

最後に、ブロックチェーン上のオーダーブックは公開されているため、注文を行うトランザクションは、次のブロックで採掘されてオーダーブックに置かれるのを待っているので、マイナーから見えるようになります。この遅れは、ユーザーがフロントランニングの状態となるリスクと、思惑とは反する価格や取引執行が行われてしまうリスクにさらされます。

2.3 ハイブリッドな解決策

上記の理由から、純粋なブロックチェーンベースでの取引は制限を抱えており、中央集権型取引所と対抗させることはできません。オンチェーン固有のトラストレスであることと、中央集権型取引所の速さと注文の柔軟性との間にはトレードオフがあります。Loopring や 0x[14] のようなプロトコルは、オフチェーンでの注文管理とともにオンチェーンでの決済のソリューションを拡張しています。これらのソリューションは、オープンなスマートコントラクトを中心に展開されていますが、いくつかの機能をオフチェーンで実行し、ネットワークにとって非常に重要な役割を果たすノードに柔軟性を与えることによって、スケーラビリティの制限を回避します。しかし、ハイブリッドモデルにも欠点が残っています [15]。Loopring プロトコルは、このホワイトペーパーを通じて我々のハイブリッドソリューションに対するアプローチにおける重要な違いを提案しています。

3 Loopring プロトコル

Loopring は DEX ではなく、複数のブロックチェーンに DEX を構築するためのモジュラーなプロトコルです。我々は従来の取引所の構成部分を分解し、代わりにパブリックなスマートコントラクト群と分散したアクターを提供します。ネットワークの役割は、ウォレット、リレー、流動性共有コンソーシアムブロックチェー

ン、オーダーブックブラウザ、リングマイナー、アセットのトークン化サービスがあります。それぞれを定義する前に、我々は先ず Loopring における注文を理解する必要があります。

3.1 オーダーリング

Loopring の注文は、一方向オーダーモデル（Unidirectional Order Model: UDOM）[16] と呼ばれるもので説明されます。UDOM は、注文をトークンの交換リクエストとして、売り（Bid）と買い（Ask）ではなく、amountS/amountB（売る量/買う量）として表します。注文とはすべて、2 つのトークン間の交換レートにすぎないため、サーキュラートレードでの複数の注文を混ぜ合わせマッチングさせることが、プロトコルの強力な特徴となります。一組の取引ペアの代わりに、最大 16 の注文を使うことで、流動性と価格上昇の可能性が劇的に増大します。

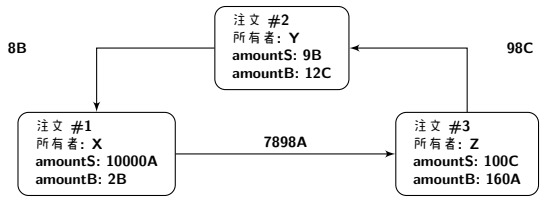

Figure 1: 3 つの注文のオーダーリング

上の図は 3 つの注文のオーダーリングです。各注文の売却するトークン（tokenS）は、別の注文の購入するトークン（tokenB）です。これは、各注文が、対となるトークンのペアを必要とせずとも、欲しいトークンと交換することを可能にするループを作成します。本質的にはオーダーリングの特別なケースとなるが、従来のペアトレードの注文も、もちろん実行可能です。

Definition 3.1 (オーダーリング) $C_0, C_1, \cdots, C_{n-1}$ が、n 種類のトークンとし、$O_{0\to1}, \cdots, O_{i\to i\oplus1}, \cdots, O_{n-1\to0}$ が n 種類の注文とします。このとき、注文はオーダーリングを作り、取引可能になります。

$$O_{0\to1} \to \cdots \to O_{i\to i\oplus1} \to \cdots \to O_{n-1\to0},$$

n がオーダーリングの長さだとすると、$i \oplus 1 \equiv i + 1$ を n で割った余りは等しくなります。

オーダーリングが成立するのは、すべての構成トランザクションが、ユーザーが明示的に指定した元々のレートと同等、もしくはより良いレートで取引されるときです。オーダーリングの成立を確認するためには、Loopring スマートコントラクトは、すべての注文の元々の交換レートが、1 に等しいか 1 より大きいオーダーリングを、リングマイナーから受け取らなければなりません。

アリスとボブがトークンA とトークンB を交換したがっているとしましょう。アリスは 15 トークンA をもっていて、その対価に 4 トークンB が欲しく、ボブは 10 トークンB もっていて、その対価に 30 トークンA が欲しいとします。

誰が買い、誰が売るのでしょうか？これは価格相場によってのみ決定されます。トークンA を固定して考えるのであれば、アリスはトークンB を、$\frac{15}{4} = 3.75$A の価格で買い、ボブは 10 トークンB を、$\frac{30}{10} = 3.00$A で売ります。トークンB を固定して考えるのであれば、アリスは 15 トークンA を、$\frac{4}{15} = 0.26666667$B の価格で売り、ボブはトークンA を $\frac{10}{30} = 0.33333334$B の価格で買います。つまり、誰が買って誰が売るかは恣意的なものなのです。

最初のシチュエーションでは、アリスは、ボブが売る価格（3.00A）よりも高い価格（3.75A）で買おうとしています。次のシチュエーションでは、ボブは、アリスが売る価格（0.26666667B）よりも高い価格（0.33333334B）で買おうとしています。買う側が、売値と同じか高い価格で買おうとするとき、取引は常に成立するのです。

$$\frac{\frac{15}{4}}{\frac{30}{10}} = \frac{\frac{10}{30}}{\frac{4}{15}} = \frac{15}{4} \cdot \frac{10}{30} = 1.25 > 1 \tag{1}$$

したがって、n 個の注文が処理されるには、完全であれ、部分的であれ、買い注文の各交換レートの結果が 1 と同じ、或いはそれを上回っているかを知る必要があります。もしそうであれば、n 個の注文は完全に、もしくは部分的に処理がなされます。[17]

仮に第三の人物であるチャーリーを持ち出してみると、アリスは x_1 トークンA を手放し y_1 トークンB を受け取り、ボブは x_2 トークンB を手放し y_2 トークンC を受け取り、チャーリーは x_3 トークンC を手放し y_3 トークンA を受け取ることになります。必要なトークンが存在するとした場合、取引は次のとき成立します。

$$\frac{x_1 \cdot x_2 \cdot x_3}{y_1 \cdot y_2 \cdot y_3} \geq 1 \tag{2}$$

Loopring の注文の詳細についてはセクション 7.1を参照してください。

4 エコシステム参加者

以下のエコシステム参加者は、協同することで、中央集権の取引所が備える全ての機能を提供します。

- ウォレット: トークンへのアクセスや、Loopring ネットワークへの注文の送信を、ユーザーが行えるようにする一般のウォレットサービスおよびインターフェースです。ウォレットには、リングマイナーと手数料を分け合うことで、注文を生成するインセンティブが生まれます。（セクション 8を参照。）取引の将来は、個々のユーザーのウォレットの安全性が確保された中で行われるという考えに立つと、これらの流動性プールを我々のプロトコルで繋げることが最良となるのです。

- コンソーシアム型流動性共有ブロックチェーン/リレーメッシュ: 注文と流動性共有のためのリレーメッシュネットワークです。ノードが Loopring のリレーソフトウェアを実行すると、既存のネットワークに接続し、コンソーシアム型のブロックチェーンを介して他のリレーと流動性を共有することができます。最初の実装として我々が構築しているコンソーシアム型ブロックチェーンは、ほぼリアルタイムのオーダー共有機能（1-2 秒のブロック）を持ち、古い記録を削除することで新たなノードの高速なダウンロードを可能にします。ノードはこのコンソーシアムに接続しないこともできます。流動性を共有せず独立を保ったり、あるいは独自の流動性共有ネットワークを立ち上げることもできます。

- リレー／リングマイナー: リレーはウォレットやリレーメッシュから注文を受け取るノードで、公のオーダーブックと取引履歴を維持し、任意に他のリレー (オフチェーン媒体を介して) および/またはリレーメッシュノードに注文をブロードキャストします。リングマイニングはリレーの特徴であり、必要条件ではありません。リングマイニングは計算負担が高く、完全にオフチェーンで行われます。私達はリングマイニングの特徴を有するリレーを「リングマイナー」と呼び、リングマイナーは全く異なる注文をつなぎ合わせてオーダーリングを作ります。リレーは、(1) それらが相互に通信する方法、(2) オーダーブックを作成する方法、(3) オーダーリングをマイニング (マイニングアルゴリズム) する方法は自由です。

- **Loopring** プロトコルスマートコントラクト **(LPSC)**: リングマイナーから受信したオーダーリングをチェックし、ユーザーに代わりトークンをトラストレスに決済および交換し、リングマイナーとウォレットへ報酬を与え、イベントを発生させる、公開された自由なスマートコントラクトのセットです。リレー/オーダーブラウザは、これらのイベントを監視し、オーダーブックと取引履歴を最新の状態に保ちます。詳細については、付録 Aを参照してください。

- 資産トークン化サービス **(ATS)**: Loopring 上で直接取引することが出来ない資産の橋渡し。これらは信頼のできる会社もしくは組織によって中央集権化されたサービスです。ユーザーは資産 (現物、通貨あるいは他のチェーンからのトークン) を預金し、トークンを発行します。これは将来、預けた資産と引き換えられます。Loopring は（適切な解決策が存在するまで）クロスチェーン取引プロトコルではありませんが、ATS は ERC20 トークン [18] の取引を、他のブロックチェーン上の資産だけでなく物理資産でも可能にします。

5 取引プロセス

1. プロトコル権限: 図 2において、トークンを交換したいユーザー`Y` は自身が売却したいトークン`B` の`amountS` を処理するよう LPSC に許可します。これにより、ユーザーのトークンはロックされず、注文が処理されるまでトークンを自由に動かすことができます。

2. 注文作成: トークン`B` とトークン`C` の現在のレートとオーダーブックは、リレーもしくはオーダーブックブラウザのようなネットワークに繋がっている他のエージェントによって提供されます。ユーザー`Y` は、任意の統合されたウォレットのインターフェースを介して特定の`amountS`、`amountB`、または他のパラメーターの注文（指値）を置きます。LRx はリングマイナーへの手数料として注文に加えることができます。より高い LRx 手数料はリングマイナーによって早く処理されるより良い機会を与えます。注文のハッシュはユーザー`Y` の秘密鍵で署名されます。

3. 注文のブロードキャスト: ウォレットは一つあるいは複数のリレーに注文と署名を送信します。リレーは公開されているオーダーブックを更新します。プロトコルは、先入れ先出しのように、ある一定の方法でオーダーブックを作成することを要求しません。代わりに、リレーはオーダーブックを作成する際に、独自の設計上の決定を下す権限を持っています。

4. 流動性の共有: リレーは任意の通信媒体を介して他のリレーにオーダーをブロードキャストします。また、ノードがどのように相互作用するかについては柔軟性があります。一定レベルのネットワーク接続を容易にするために、コンソーシアムブロックチェーンを使用した組み込みの流動性共有のためのリレーメッシュが存在します。前のセクションで述べたように、このリレーメッシュは速度と包括性のために最適化されています。

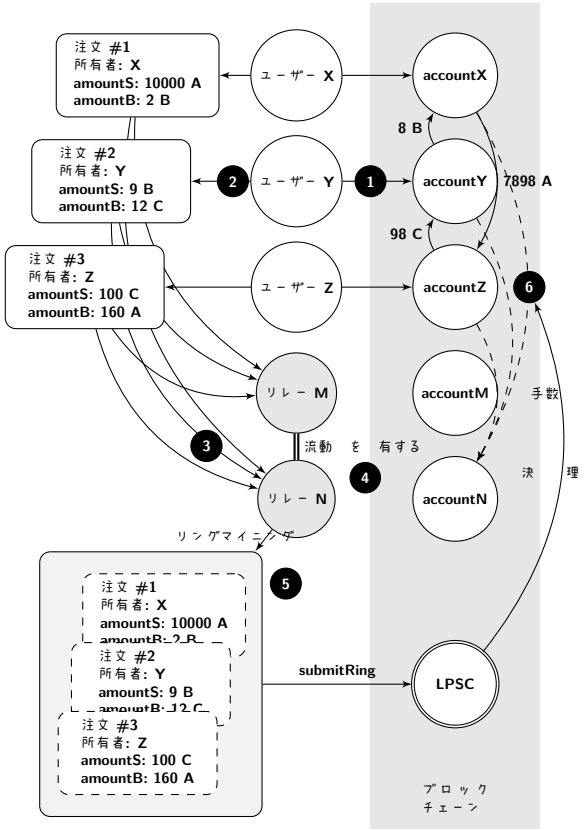

Figure 2: Loopring 取引プロセス

5. リングマイニング（注文マッチング）: リングマイナーは、注文を複数の他の注文と照合することで、指定された為替レートまたは、より良いレートで完全または部分的に記入しようとします。プロトコルがどんなペアよりも高い流動性を提供することができる主な理由はリングマイニングです。もし実行されたレートがユーザ Y が指定したレートよりも良い場合は、オーダーリング内のすべての注文間でマージンが共有されます。報酬としてリングマイナーは、マージンの部分（マージンスプリット、そして LRx トークンをユーザーに返す）を請求するか、単に LRx 料金を維持するかを選択します。

6. 検証 & 決済: オーダーリングは LPSC によって受け取られます。リングマイナーが提供するデータを検証するための複数のチェックを行い、オーダーリングが完全にまた

は部分的に決済することができるかどうかを決定します(ユーザーウォレット内のトークンおよびリング内の注文の記入割合によって決定します)。もしすべてのチェックが成功している場合、コントラクトはユーザーにアトミック(完全に行なわれるか全く行なわれないかのいずれか)にトークンを転送し、リングマイナーとウォレットに手数料を同時に支払います。LPSC によって決定されたユーザY の残高が不十分な場合、注文自体がスケールダウンされたものとみなされます:キャンセルはされず、アドレスに十分な資金が入金された場合、スケールダウンした注文は自動的に元のサイズに拡大されます。これは一方通行の手動操作であり、元に戻すことはできません。

6 オペレーションの柔軟性

Loopring のオープンスタンダードにより、参加者のオペレーションには大きな柔軟性が提供されることに注意することが重要です。参加者は新しいビジネスモデルを自由に構築してユーザーに価値を提供し、取引量やその他の基準で LRx 手数料を得ることも、(もしそう選べば)可能なのです。エコシステムはモジュール式になっており、多様なアプリケーションからの参加者をサポートするようにできています。

6.1 オーダーブック

リレーは、ユーザーの注文を表示して一致させるために、多数の方法でオーダーブックを設計することができます。我々自身の最初のオーダーブックは OTC モデルを採用しており、価格のみに基づいて指値のポジションがとられます。言い換えれば、注文のタイムスタンプは、オーダーブックとは関係がありません。しかしながらリレーは、自由にオーダーブックを設計することができるため、典型的な中央集権型の取引所のマッチングエンジンをまねて注文を価格順に並べる一方、同時にタイムスタンプも考慮する、といったことも可能です。リレーがこのようなオーダーブックを提供する場合、内部にウォレットを保持あるいは組み込み、ウォレットの注文を単独のリレーに送り、タイムスタンプに基づいて注文をマッチさせます。このような構成も可能となっています。

他の DEX プロトコルの中には、リレーにリソース(テイカーが注文を出すための初期トークン残高)が必要なものがあるのに対し、Loopring リレーは、マッチして取引を成立させる注文を見つけるのみでよいため、初期トークンなしで行うことができます。

6.2 流動性の共有

リレー同士は、いかに流動性(注文)を互いに共有するかの設計も自在にできます。我々のコンソーシアム型ブロックチェーンは、これを成し遂げるソリューションであり、エコシステムは望むままにネットワークに接続し交流することが可能です。コンソーシアム型ブロックチェーンに接続する以外にも、独自のものを構築・管理することができ、各々に合うルール/インセンティブを加えることができます。リレーは独立して稼働することもできます。時間が重要な要素となるウォレットの実装などがその例です。もちろん、ネットワーク効果を追求する上では、他のリレーと交流することに明確な優位性がありますが、異なるビジネスモデル間では、特定の共有設計において手数料を共有するほうがメリットが大きいこともありえます。

7 プロトコルの特徴

7.1 注文の詳細

注文とは、ユーザーの取引の意図を示すデータの集まりです。Loopring での注文は、一方向オーダーモデル、すなわち UDOM を使うことで、以下のように定義されます。

```
message Order {
  address protocol;
  address owner;
  address tokenS;
  address tokenB;
  uint256 amountS;
  uint256 amountB;
  unit256 lrcFee
  unit256 validSince; //      時点からの通算秒
  unit256 validUntil; //      時点からの通算秒
  uint8   marginSplitPercentage;  // [1-100]
  bool    buyNoMoreThanAmountB;
  uint256 walletId;
  // ニ重     アドレス
  address authAddr;
  // v, r, sは   名のペーツ
  uint8   v;
  bytes32 r;
  bytes32 s;
  // ニ重     するプライベートキー。
 // オーダーのハッシュ計算には使用されないため、
  //    名されない。
  string  authKey;
}
```

注文の発信元を保証するため、authAddr を除くパラメーターのハッシュに対してユーザーの秘密鍵で署名します。authAddr パラメーターは、この注文の一部であるオーダーリングに署名するために使用され、フロントランニングを防止します。詳細はセクション 9.1 を参照してください。署名は、v、r、およびs フィールドで表され、ネットワークを介して注文パラメータとともに送信されます。これにより、注文は完了するまで不変であることが保証されます。注文が不変であっても、プロトコルは他の変数と一緒にアドレスの残高に基づいて現在の状態を計算することが可能です。

UDOM は(性質的に浮動小数点数でなければならない)価格を含んでいませんが、代わりに、amountS/amountB で表されるrate または r という用語が使用されます。レートは浮動小数点数ではなく、要求により他の符号なし整数で評価されるものですが、これにより、中間結果が符号なし整数のままとなり、計算の正確性が増します。

7.1.1 購入量

リングマイナーが注文をリングマッチさせるとき、より良いレートで取引が行われ、ユーザーが指定したamountB よりも多く

のtokenB を手にすることは起こりえます。しかしながら、もしも「amountB 以上は買わない」が真と設定されれば、ユーザーがamountB 以上のtokenB を手にしないよう、プロトコルは保証します。このように、UDOM のbuyNoMoreThantokenB パラメータにより、注文が完全に処理されたと考えられる時期が決まるのです。buyNoMoreThantokenB はamountS にもamountB の上限にもあてはまり、伝統的な買い/売りの注文よりもきめ細かいユーザーの取引の意図を反映可能になります。

例えば、amountS が 10 でamountB が 2 の場合、レート $r = 10/2 = 5$ となります。ユーザーは各tokenB に対し 5 つのtokenS を売るつもりということになります。リングマイナーがレート 4 のユーザーを見つけてマッチすれば、そのユーザーは 2 ではなく 2.5 のtokenB を得ることができます。しかしながら、そのユーザーが 2 つのtokenB しか欲しくなく、buyNoMoreThanAmountB フラグをTrue と設定していた場合、LPSC はレート 4 にて作用し、ユーザーは各tokenB に対し 4 つのtokenS を売り、2 つのtokenS を節約します。これは、マイニング手数料を考慮していないことに留意してください。(セクション 8.1を参照してください。)

実際に、

```
Order(amountS,tokenS,
      amountB,tokenB,
      buyNoMoreThantokenB)
```

を使用して注文を簡素化した形式で表すと、従来の取引所での ETH/USD マーケットにおいて、従来の購入-売却モデルでは下の 1 番目と 3 番目の注文は表現できますが、他の 2 つは表現することができません。

1. 10 ETH を 300 USD/ETH の価格で売る。この注文は次のように表されます: `Order(10, ETH, 3000, USD, False)`。
2. 1 ETH を 300 USD/ETH の価格で売り 3000 USD を得る。この注文は次のように表されます: `Order(10, ETH, 3000, USD, True)`。
3. 10 ETH を 300 USD/ETH の価格で買う。この注文は次のように表されます: `Order(3000, USD, 10, ETH, True)`。
4. 3000 USD で買える限りの ETH を 300 USD/ETH の価格で買う。この注文は次のように表されます: `Order(3000, USD, 10, ETH, False)`。

7.2 リング検証

Loopring のスマートコントラクトは、交換レートや取引量の計算は行いませんが、リングマイナーが取引で提示するこれらの情報を受け取り、検証しなければなりません。これらの計算は、以下の 2 つの理由により、リングマイナーの手でなされます。(1) イーサリアムでの solidity[19] のような、スマートコントラクト用のプログラミング言語は、浮動小数点数、$pow(x, 1/n)$(浮動小数点数の n 乗根の計算)をサポートしておらず、(2) ブロックチェーンの処理量とコストを削減するには、処理がオフチェーンで行われたほうが望ましいためです。

7.2.1 サブリングチェック

このステップにより、アービトラージをする取引者が、新規の注文を実行してオーダーリング内のマージンを不当に得ることを防止します。リングマイナーが、オーダーリングが有効となったのを確認すると、他の注文をオーダーリングに加えてユーザーのマージン(レートディスカウント)を全て得ようという誘因が本質的に生まれます。下の図 3で表されるように、注意深く計算された場合には、$x1$, $y1$, $x2$, $y2$ は注文のレート結果はちょうど 1 となり、レートディスカウントは発生しません。

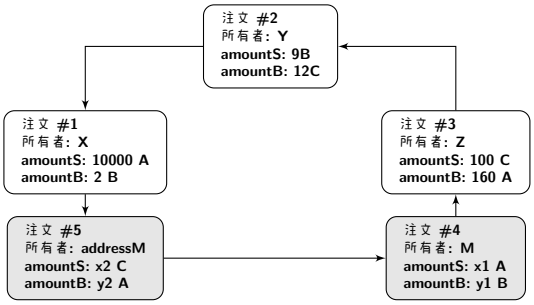

Figure 3: サブリングのあるオーダーリング

これはゼロリスクかつ、ネットワークに価値は付加されず、リングマイナーの不公平な行動とみなされます。これを防ぐために、Loopring は、有効なループが追加のサブリングを含められないことを要件としています。このチェックとして、トークンの売買ポジションを 2 回置くことが不可能なように LPSC が保証します。上図では、トークンA が 2 回売り、2 回買いとなっており、これは禁止されます。

7.2.2 フィルレートチェック

オーダーリング内の交換レート計算は、上記の理由によりリングマイナーによって行われます。その計算が正しいことを証明しなければならないのは LPSC です。第一に、リングマイナーのそれぞれの注文に対する実際の買いレートが、ユーザーによって設定された元々の買いレートと同じかそれより低いことを検証します。これにより、ユーザーが指定したレートかそれより良いレートで買えることが保証されます。交換レートが確認されると、オーダーリング内のそれぞれの注文が同じだけのレートディスカウントを得ることを LPSC が保証します。例えば、ディスカウント後のレートが γ のとき、各注文の価格は以下のようになり、

$r_{0\to1}\cdot(1-\gamma)$, $r_{1\to2}\cdot(1-\gamma)$, $r_{2\to0}\cdot(1-\gamma)$、以下を満たします。

$$r_{0\to1}\cdot(1-\gamma)\cdot r_{1\to2}\cdot(1-\gamma)\cdot r_{2\to0}\cdot(1-\gamma)=1 \qquad (3)$$

したがって、

$$\gamma = 1-\frac{1}{\sqrt[3]{r_{0\to1}\cdot r_{1\to2}\cdot r_{2\to0}}}. \qquad (4)$$

n 個のオーダーが取引で処理されるとき、ディスカウントは、

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

となり、i 番目の注文の注文回転率は r^i となります。明らかに、ディスカウントレートが $\gamma \geq 0$ のときのみ、これらの注文

は処理されます。そして、i 番目の注文 (O^i) の交換レートは $\hat{r^i} = r^i \cdot (1-\gamma)$, $\hat{r^i} \leq r^i$ となります。

アリスは 15 トークンA で 4 トークンB を欲しがっている、ボブは 10 トークンB で 30 トークンA を欲しがっているという前述の例を思い出してください。トークンA を固定して考えるのであれば、アリスはトークンB を、$\frac{15}{4} = 3.75$A の価格で買い、ボブは 10 トークンB を、$\frac{30}{10} = 3.00$A で売ります。割引を計算するには: $\frac{150}{120} = 1.25$ であるので $\frac{1}{1.25} = 0.8 = (1--\gamma)^2$。従って、両者にとって公平な取引レートは、トークンB 当たり $\sqrt{0.8} \cdot 3.75 \approx 3.3541$ トークンA となります。

ボブは 4 トークンB を売り 12 トークンA を受け取る予定でしたが、実際にはより多い 13.4164 トークンA を受け取ります。アリスは予定通り 4 トークンB を受け取りますが、予定していた 15 より少ない 13.4164 トークンA のみを売ります。このマージンの一部は、マイナー(およびウォレット)にインセンティブを与えるための手数料をとなることに注意してください。(セクション 8.1参照)。

7.2.3 追跡とキャンセルの実施

ユーザーは注文の詳細とキャンセルの量を含む LPSC に特別なトランザクションを送ることで、一部もしくは注文の全体をキャンセルすることができます。LPSC はそれを考慮に入れて、キャンセル料を記録、ネットワークに`OrderCancelled` イベントを送信。LPSC は識別として注文のハッシュを用いてその価値を記録することで、注文とキャンセルされた量の記録を残します。このデータは誰でもアクセス可能であり、`OrderCancelled` / `OrderFilled` イベントが生成されます。これらの値を追跡することは、オーダーリングの決済ステップ中の LPSC にとって重要なものになります。

また、LPSC は`OrdersCancelled` イベントで取引ペアのすべての注文をキャンセルし、`AllOrdersCancelled` イベントで特定アドレスのすべての注文をキャンセルすることもサポートしています。

7.2.4 注文のサイズ調整

執行済みの注文量とキャンセルされた注文量の履歴と、注文送信者の現在の残高に応じて注文はサイズ調整されます。このプロセスでは、上記の条件から最低注文量が提示され、オーダーリングにおけるすべてのトランザクションのサイズ調整が行われます。

最も小さい値の注文を見つけることで、各注文の注文量を把握することができます。例えば、i 番目の注文が最も小さい値の注文の場合、各注文から売却されるトークン量 $\hat{s}$ と購入されるトークン量 $\hat{b}$ は以下のように計算されます。

$$\hat{s^i} = \bar{s_i}, \hat{b^i} = \hat{s^i}/\hat{r^i}, ;$$
$$\hat{s^{i\oplus 1}} = \hat{b^i}, \hat{b^{i\oplus 1}} = \hat{s^{i\oplus 1}}/\hat{r^{i\oplus 1}};$$
$$\hat{s^{i\oplus 2}} = \hat{b^{i\oplus 1}}, \hat{b^{i\oplus 2}} = \hat{s^{i\oplus 2}}/\hat{r^{i\oplus 2}};$$
$$...$$

注文が部分的に執行された後に残る残高が $\bar{s_i}$ です。

履行中は、安全にオーダーリング内のどの注文も最低値を得られるようになっており、オーダーリングを最大 2 回反復することにより、各注文の量が計算できます。

例:元の注文と比較して最も少ない金額を 5% とすると、注文リング内のすべての取引は 5% に縮小されます。すべての取引が完了すると、最小の未執行分を持つと想定される注文が完全に執行されます。

7.3 リング決済

オーダーリングがすべてのチェックを終えると、オーダーリングは閉じられ、トランザクションが生成されます。これは、すべてのオーダーが閉じたオーダーリングを形成し、図 4 のように繋がれるということを意味します。

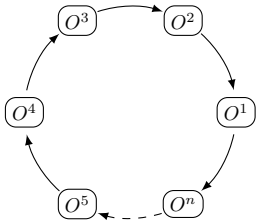

Figure 4: リング決済

トランザクションを生成するために、LPSC は`TokenTransferDelegate` というスマートコントラクトを使用します。すべての注文が異なるバージョンのプロトコルの代わりに上記のスマートコントラクトを許可する必要があるため、このスマートコントラクトを導入することで、プロトコル内のスマートコントラクトのアップグレードが容易になります。

オーダーリングの各注文について、実装に応じて後続もしくは先行する注文に`tokenS` の支払いが行われます。そして、リングマイナーの手数料は、リングマイナーが選択する手数料モデルによって選ばれます。最終的に、すべてのトランザクションが生成されたら、`RingMined` イベントが送出されます。

7.3.1 発生イベント

プロトコルは、リレー、オーダーブラウザや、その他のアクターがオーダーブックの更新を受け取れるようにするイベントを発行します。発行されるイベントは次のとおりです。

- **OrderCancelled**: 特定の注文がキャンセルされた。
- **OrdersCancelled**: アドレスが所有する 1 トレードペアのすべての注文がキャンセルされた。
- **AllOrdersCancelled**: アドレスが所有するすべてのトレードペアのすべての注文がキャンセルされた。
- **RingMined**: オーダーリングが正常に完了。このイベントにはリング内での各トークン移動に関する情報が含まれます。

8 LRx トークン

LRx とは我々のトークンの一般的な表記法です。LRC は Ethreum での Loopring トークンを指し、Qtum では LRQ、NEO では LRN などになります。他の LRx タイプは、今後 Loopring が他のパブリックブロックチェーンに展開される際に導入されます。

8.1 手数料モデル

ユーザーが注文を行うとき、リングマイナーが要求できる注文に設定されたマージン（marginSplitPercentage）の割合とともに、手数料としてリングマイナーに支払われる LRx の数量を指定します。これはマージンスプリットと呼ばれます。手数料またはマージンスプリットを選ぶかはリングマイナーに選択権があります。

マージンスプリットの説明

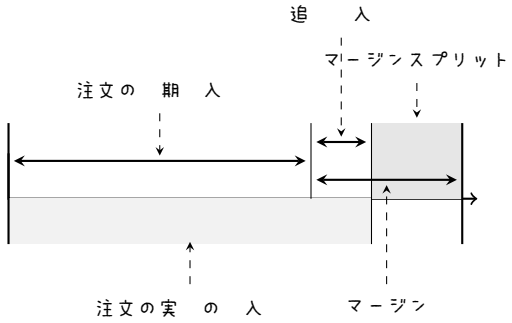

Figure 5: 60% のマージンスプリット

オーダーリングにおけるマージンが小さすぎる場合、リングマイナーは手数料 (LRx) を選択します。逆に、マージンに余裕があり、マージンスプリットの結果がが手数料 (LRx) よりも価値が大きい場合は、マージンスプリットを選択します。しかしながら、もう一つ条件があります。リングマイナーがマージンスプリットを選択した場合、リングマイナーは、ユーザー (注文作成者) がリングマイナーに支払うべき手数料 (LRx) と同額を、ユーザーに支払わなければならないため、リングマイナーがマージンスプリットを選択するための閾値が、注文手数料（LRx）の 2 倍となり、手数料（LRx）を選択する傾向が高まります。これにより、リングマイナーは、マージンの高いオーダーリングで収入が少なくなる反面、低マージンのオーダーリングでは安定した収入を得ることができます。我々の手数料モデルは、マーケットが成長・成熟するにつれて、高いマージンのオーダーリングが少なくなり、インセンティブとして固定された LRx 手数料を必要とするという予想に基づいています。

グラフは以下のようになります。

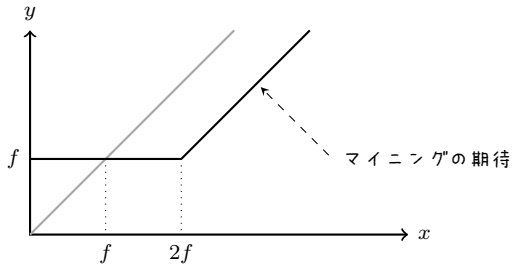

Figure 6: Loopring の手数料モデル

f が LRx 手数料、x がマージンスプリット、y がマイニング報酬です。実線で示されているように、$y = max(f, x - f)$ となります。もしも注文の LRx 手数料が 0 ならば、方程式は $y = max(0, x - 0)$ であり、グレーの線で示されるように、単に $y = x$ となります。

結果は以下のようになります。

1. マージンスプリットが 0 の場合、リングマイナーは、均一の LRx 手数料を選択し、インセンティブを受けます。
2. LRx 手数料が 0 の場合、グレー線の結果となり、報酬は通常の線形モデルに基づくことになります。
3. マージンスプリット報酬が 2x（LRx 手数料）よりも大きい場合、リングマイナーはマージンスプリットを選択し、LRx をユーザーに支払います。

LRx 手数料が 0 でない場合、リングマイナーがどちらの選択をしようとも、リングマイナーと注文送信者との間には常に LRx の送金が発生することに注意してください。リングマイナーは LRx 手数料を得るか、マージンスプリットを得るために LRx 手数料を送信者に返金します。

リングマイナーは、ウォレットと一定のパーセンテージの手数料を共有します。ユーザーがウォレットを通じて注文を行い、処理がなされると、ウォレットは手数料またはマージンスプリットの一部を報酬として受けます。これが基本的な形式ですが、独自のビジネスモデルや実施様態を構築することも可能です。獲得される手数料の約 20% から 25% をウォレットが受け取るのが我々の目指す方向です。ユーザーベースはあるものの、現状ではそれが収入源となっていることは殆どないため、ウォレットは Loopring プロトコル統合の主要ターゲットとなります。

8.2 分散ガバナンス

Loopring プロトコルは、メンバー間の協調に基づいて効果的に目標を達成するという意味でのソーシャルプロトコルです。これは暗号通貨経済のプロトコルと概して似ており、実際その利便性は、ゲーム理論の協力問題 [20]、グリムトリガー均衡、限定合理性などと同じ仕組みによって守られているのです。この目的のために、LRx トークンは手数料支払いのためだけでなく、様々なネットワーク参加者の金銭的インセンティブを協調させるためにも用いられます。このような協調は、幅広く取り入れられるためにはどんなプロトコルにも必要なことですが、堅固で分散したエコシステムにおいて、流動性を向上させることが成功への大きな鍵となる取引所プロトコルでは殊更に重要となります。

LRx トークンは、分散ガバナンスでのプロトコルのアップデートを実現するために使われます。スマートコントラクトのアップデートは、連続性と安全性を保ち、不整合性による流動性の低下リスクを少なくするために、トークンホルダーによるガバナンスが部分的に行われます。スマートコントラクトは一度デプロイされると変更ができないため、dApps やエンドユーザーが廃止予定のバージョンと相互作用し続け、アップデートされたコントラクトから適合外となるリスクがあります。市場の需要と新たなブロックチェーンに適応しなければならないため、アップグレード可能であることがプロトコルの成功には不可欠です。LRx ホルダーによる分散ガバナンスは、dApps やエンドユーザーに影響を与えたり、スマートコントラクトの抽象化に過度に依存することなく、プロトコルのアップデートを可能にします。LRx トークンは供給量が固定されており、LRC の場合には、一定割合が Loopring 財団によって凍結されており、コミュニティ指向のファンドに割り当てられます [21]。

しかしながら、LRx のトークン所有者のみが、プロトコルの方向性を左右するステークホルダーなのではありません。リレー、リングマイナー、ウォレット、開発者等は皆、エコシステムに不可欠な存在であり、その声は聞かれなければなりません。実際、これ

らの存在は、それぞれの役割を果たすために LRx を保持する必要はなく（従来のメーカー/テイカーとマーケットメーカーは存在しないため、初期のトークンリザーブは必須ではありません）他の方法によって、彼らの利益を尊重しなければなりません。さらに言えば、「単純な」トークンベースの投票は、低い投票率とトークン保持者の偏在性がリスクを生むため、オンチェーン・オフチェーンを問わず、合意不成立への対処としては不完全です。したがって、最終的なゴールはレイヤー上にガバナンスモデルを構築することであり、何らかの意思決定プロセスを規範とする共有知識に、この成否がかかっています。これは、幅広い参加者からのシグナルと、ことによればまだ確立していないプロトコルの焦点からのシグナルを提供する協力機関によって促進されます。分散ガバナンスが実現するにつれて、Loopring 財団は必然的に、プロトコルの開発者から、プロトコルの世話役となるでしょう。

9 詐欺と攻撃への耐性

9.1 フロントランニング防止

分散型取引所においてのフロントランニングとは、他のノードのトレードソリューションをコピーし、未処理のトランザクションプール（メモリプール）にある元々のトランザクションよりも前にマイニング承認を済ませることです。これは、より高い取引手数料（ガス価格）を指定することで起こりえます。Loopring（といかなるオーダーマッチングのプロトコル）でのフロントランニングの主な手法は、オーダーフィルチ（フロントランナーが未処理のオーダーリング決済トランザクションから、1 つあるいは複数の注文を盗むこと。Loopring に限れば、フロントランナーが未処理のオーダーリング全体を盗むこと）です。

submitRing トランザクションが承認されないまま、未処理のトランザクションプールに残っている場合、誰でもそのトランザクションを見つけ、`minerAddress` を自分のアドレス（`filcherAddress`）に変更し、`filcherAddress` でペイロードを放棄させ、オーダーリングの署名を書き換えることができます。フィルチャーは、ブロックマイナーが元の submitRing トランザクションではなく、フィルチャーのトランザクションを次のブロックに記録することを期待して、より高いガス価格を設定したトランザクションを新たに送信することができます。

この問題への従来の解決策には重大な弱点がありました。追加のトランザクションを必要とするため、リングマイナーに多くのガス負担をかけ、オーダーリングの決済に最低でも 2 倍のブロックを使用することです。我々の新たな解決策であるデュアル認証 [22] では、注文に 2 段階の認証（決済段階とリングマイニング段階）を設定する仕組みを採用しています。

デュアル認証プロセス

1. 各注文に対し、ウォレットソフトウェアはランダムな公開鍵/秘密鍵のペアを生成し、注文の JSON スニペット内に配置します。(あるいは、バイトサイズを小さくするために、公開鍵自体ではなく公開鍵に紐づいたアドレスを使用できます。このようなアドレスを表すのに`authAddr` を使用し、対応する秘密鍵を表すのに`authAddr` を使用します）
2. 注文に含まれる全てのフィールド（`r`, `v`, `s` および`authKey` を除く）で注文のハッシュを計算し、所有者の秘密鍵（`authKey` ではなく）を使ってハッシュに署名します。
3. ウォレットは`authKey` と合わせて注文をリングマイニング用のリレーに送ります。リングマイナーは、`authKey` と`authAddr` が正しくペアになっていることと、オーダーの署名が所有者アドレスに対して有効であることを検証します。
4. オーダーリングが識別されると、リングマイナーは各注文の`authKey` を使用して、リングのハッシュ、`minerAddress`、その他全てのマイニングパラメータに署名します。オーダーリングに n 個の注文が含まれている場合、n 個の`authKey` による n 個の署名が存在し、これらの署名を`authSignature` と呼びます。リングマイナーは`minerAddress` の秘密鍵を使用して、全てのマイニングパラメータとともにリングのハッシュに署名をする必要があります。
5. リングマイナーは全てのパラメータと全ての余分な`authSignature` を使用して submitRing 関数を呼び出します。`authKey` はオンチェーンのトランザクションの一部分ではなく、リングマイナー以外の第三者には明かされないことに注意してください。
6. Loopring プロトコルは各注文の対応する`authAddr` に対して各`authSignature` を検証し、`authSignature` が不足または無効なオーダーリングを拒否します。

その結果は以下のようになります。

- 注文の（所有者アドレスの秘密鍵による）署名は、`authAddr` を含め、注文の変更が不可能であることを保証します。
- リングマイナーの（`minerAddress` の秘密鍵による）署名が提供された場合は、マイナーのアイデンティティを使用してのオーダーリングのマイニング承認が不可能であることを保証します。
- `authSignature` は、`minerAddress` を含め、オーダーリング全体の変更、および注文の盗難が不可能であることを保証します。

デュアル認証はリングフィルチとオーダーフィルチを防止し、オーダーリングの決済がひとつのトランザクション内で行われることを保証します。さらに、デュアル認証は、非マッチ共有・マッチ共有という 2 つの方法でリレーが注文を共有することを可能にします。デフォルトでは、Loopring は OTC モデルで稼働し、指値注文のみをサポートし、注文のタイムスタンプは無視されます。よって、フロントランニングは、実際の取引価格には影響しませんが、取引が処理されるか否かには影響を及ぼすことを意味します。

10 オーダー攻撃

10.1 シビル攻撃および DOS 攻撃

悪意あるユーザー（当人としてふるまおうと別人になりすまそうと）は、規模の小さい注文を大量に送り、Loopring ノードへの攻撃を試みるかもしれません。しかしながら、ノードはそれぞれの基準（公開および非公開）に基づいて注文を拒否することがで

きるため、このような注文のほとんどは、マッチしたとしても、十分な利益を生みださないとして拒否されます。注文を管理するリレーの権限を強化することで、小規模かつ多量のオーダー攻撃は脅威となりません。

10.2 不十分な残高

悪意のあるユーザーが、アドレスの実際の残高がゼロにもかかわらず、注文価値がゼロではない注文に署名をして広めることができます。ノードは、実際の残高がゼロの注文を監視して発見することができ、それらの注文ステータスを更新し破棄することができます。ノードは注文ステータスを更新するために時間を消費しますが、アドレスのブラックリスト化や関連する注文の破棄を行うことで、コストを最小限に抑えることもできます。

11 要約

Loopring プロトコルは分散型取引所のための基盤となることを目指しています。そうすることで、人々が資産や価値をどのように交換するかに大きな影響を与えるでしょう。中間商品として、お金は物々交換を容易にしたり、または置き換え、二重の要求の一致問題 [23] を解決します、これは二つのグループがお互い別々のモノを求めなければならないという状況による問題です。同じように、Loopring プロトコルは、より簡単に取引を簡潔させるリングマッチングを用いて、取引ペアにおける要求の一致の信頼度を分配することを狙っています。これは、社会やマーケットがトークンや従来の資産などを、どのように交換するかについて意味のあるものです。確かに、分散型の暗号通貨は国家のお金に対する支配権を脅かすように、トレーダー (消費者と生産者) 同士を結び付ける複合的なプロトコルは、お金の概念そのものへの理論上の脅威となります。

プロトコルは以下の利点をユーザーに与えます。

- オフチェーンのオーダー管理とオンチェーンでの決済は安全性を犠牲にすることはありません。
- リングマイニングとオーダーシェアリングによるより高い流動性。
- 二重認証は現在の全ての DEX とそのユーザーが直面するフロントランニングの悪質な問題を解決します。
- 自由かつパブリックなスマートコントラクトはいかなる dApps でも、プロトコル上に構築または操作することを可能にします。
- 事業者間の標準化により、ネットワーク効果とエンドユーザー体験の向上が可能になります。
- ネットワークはオーダーブックの運用と伝達での柔軟性とともに維持されます。
- 参加への障壁が減るということは、ノードとエンドユーザがネットワークに参加するためのコストが低いということを意味します。
- ユーザーウォレットからの直接の匿名取引。

12 謝辞

メンターやアドバイザー、そして私たちを歓迎してくれた有識なたくさんのコミュニティメンバーに感謝いたします。特に、我々のプロジェクトに関してのレビューをいただいた Shuo Bai 氏 (ChinaLedger)、Habin Kan 教授、Alex Cheng 氏、Hongfei Da 氏、Yin Cao 氏、Xiaochuan Wu 氏、Zhen Wang 氏、Wei Yu 氏、Nian Duan 氏、Jun Xiao 氏、Jiang Qian 氏、Jiangxu Xiang 氏、Yipeng Guo 氏、Dahai Li 氏、Kelvin Long 氏、Huaxia Xia 氏、Jun Ma 氏、Encephalo Path 氏に感謝申し上げます。

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 Atterlönn. 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 Gençay. 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] Loopring Foundation. Lrc token documents. https://docs.loopring.org/English/token/, Accessed: 2018-03-05.

[22] Daniel Wang. Dual authoring —loopring's solution to front-running. *URL https://medium.com/loopring-protocol/dual-authoring-looprings-solution-to-front-running-d0fc9c348ef1*, 2018.

[23] 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　イーサリウム上の Loopring

A.1　スマートコントラクト

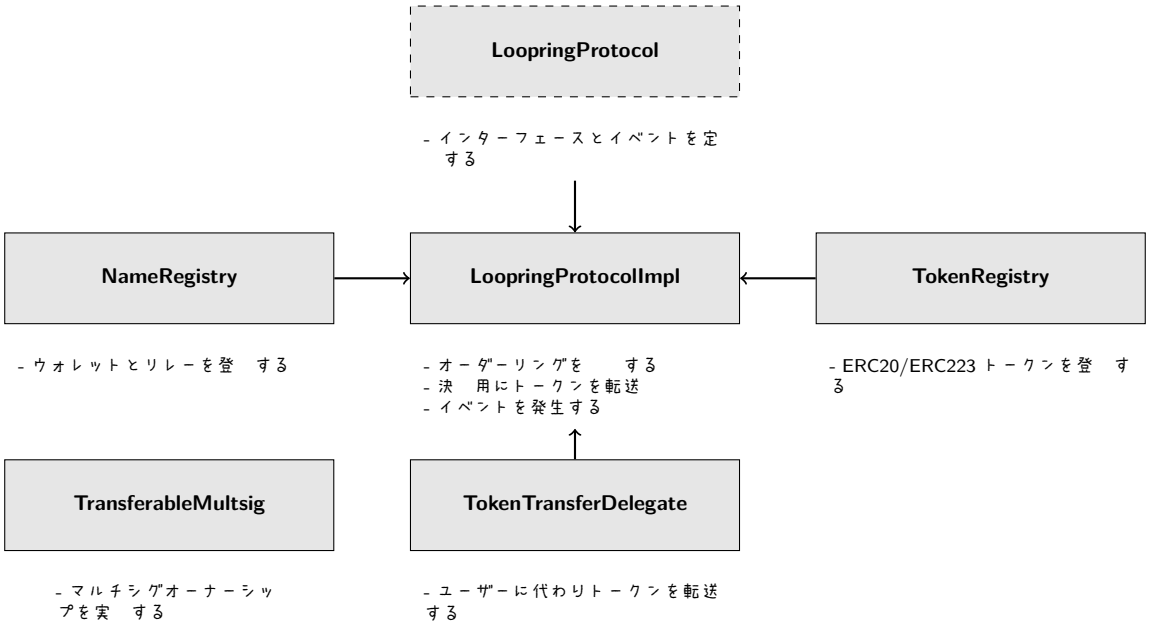

Figure 7: スマートコントラクト

A.2　デプロイメント

以下のスマートコントラクトはイーサリアムメインネット上にデプロイされています。

- LRC: 0xEF68e7C694F40c8202821eDF525dE3782458639f
- TokenRegistry: 0xa21c1f2AE7f721aE77b1204A4f0811c642638da9

- TokenTransferDelegate: `0x7b126ab811f278f288bf1d62d47334351dA20d1d`
- NameRegistry: `0xd181c1808e3f010F0F0aABc6Fe1bcE2025DB7Bb7`
- LoopringProtocolImpl: `0x0B48b747436f10c846696e889e66425e05CD740f`

A.3 Qtum

以下のスマートコントラクトはクアンタムメインネット上にデプロイされています。

- LRQ: `2eb2a66afd4e465fb06d8b71f30fb1b93e18788d`
- TokenRegistry: `c89ea34360258917daf3655f8bec5550923509b3`
- TokenTransferDelegate: `60b3fa7f461664e4dafb621a36ac2722cc680f10`
- NameRegistry: `e26a27d92181069b25bc7283e03722f6ce7678bb`
- LoopringProtocolImpl: `5180bb56b696d16635abd8dc235e0ee432abf25d`