Loopring:
Ein dezentrales Token-Austauschprotokoll

Daniel Wang
daniel@loopring.org

Jay Zhou
jay@loopring.org

Alex Wang
alex@loopring.org

Matthew Finestone
matt.finestone@gmail.com

https://loopring.org

19. März 2018

Zusammenfassung

Loopring ist ein offenes Protokoll um dezentralisierte Handelsplattformen zu realisieren. Loopring operiert über einen Zusammenschluss von mehreren Smart Contracts welche für den Handel und die Regulierung verantwortlich sind, über eine Gruppe von Akteuren, die fernab der Blockchain agiert und die Aufträge zusammenfasst und verteilt. Loopring ist ein freies, erweiterbares Protokoll und agiert als standardisiertes Framework für dezentralisierte Applikationen (dApps), die die Möglichkeiten einer Handelsplattform beinhalten. Seine interoperablen Standards ermöglichen einen vertrauensfreien, anonymen Handel. Eine wichtige Verbesserung gegenüber den derzeitigen dezentralen Austauschprotokollen ist die Möglichkeit, dass Aufträge mit anderen, unterschiedlichen Aufträgen gemischt werden können. Dadurch werden die Zwänge von Zwei-Token-Handelspaaren vermieden werden und die Liquidität wird drastisch verbessert. Loopring verwendet auch eine einzigartige und robuste Lösung, um ein Front-Running zu verhindern: den unfairen Versuch, Transaktionen schneller als der ursprüngliche Lösungsanbieter in einem Block einzureichen. Loopring ist Blockchain-agnostisch und in jeder Blockchain mit Smart-Contract-Funktionalität einsetzbar. Zum jetzigen Zeitpunkt ist es auf Ethereum [1] [2], Qtum [3] und NEO [4] in Entwicklung.

1 Einleitung

Mit der Verbreitung von Blockchain-basierten Vermögenswerten hat die Notwendigkeit diese Vermögenswerte untereinander auszutauschen deutlich zugenommen. Da tausende neue Token eingeführt werden, einschließlich traditionelle Vermögenswerte, ist dies ein wachsendes Bedürfnis. Unabhängig davon, ob Token für spekulatives Handeln ausgetauscht werden oder über ihre nativen Utility-Token in Access-Netzwerke umgewandelt werden, ist die Fähigkeit, ein Cryptoasset gegen ein anderes auszutauschen für das größere Ökosystem grundlegend. In der Tat gibt es eine potentielle Energie in Vermögenswerten [5] und um diese Energie zu realisieren - das Entsperren von Kapital - erfordert nicht nur die Eigentumsrechte, sondern auch die Fähigkeit diese Vermögenswerte frei zu transferieren und zu transformieren.

Daher ist der vertrauenslose Austausch von Token (Wert) ein überzeugender Anwendungsfall für die Blockchain-Technologie. Bis jetzt haben sich Krypto-Enthusiasten jedoch weitgehend für Token an traditionellen zentralisierten Handelsplattformen entschieden. Das Loopring-Protokoll wird benötigt weil, genau wie Bitcoin pflichtbewusst betonte, dass in Bezug auf elektronisches Geld, "Die Hauptvorteile gehen verloren, wenn noch eine vertrauenswürdige dritte Partei benötigt wird, um Doppelausgaben zu vermeiden". Die Hauptvorteile der dezentralen Vermögenswerte gehen auch verloren, wenn sie über eine zentralisierte Handelsplattform laufen müssen.

Der Handel dezentraler Token an zentralisierten Börsen ist auch aus philosophischer Sicht nicht adäquat, da die Tugenden, die diese dezentralisierten Projekte unterstützen, nicht eingehalten werden können. Es gibt auch zahlreiche praktische Risiken und Einschränkungen bei der Verwendung zentralisierter Vermittlungsstellen, die nachstehend beschrieben werden. Dezentralisierte Handelsplattformen (DEXs) [6] [7] [8] haben versucht, diese Probleme zu Lösen, und in vielen Fällen ist es gelungen Sicherheitsrisiken durch die Verwendung von Blockchains zur Disintermediation zu mindern. Da die Fähigkeiten dezentralisierter Handelsplattformen jedoch zu einer entscheidenden Infrastruktur für die New Economy wird, gibt es erheblichen Spielraum für Leistungsverbesserungen. Loopring möchte Module für diese Infrastruktur mit seinem offenen Protokoll dApp bereitstellen.

2 Aktuelle Handelsplattform-Landschaft

2.1 Unzulänglichkeiten von Zentralen Börsen

Die drei Hauptrisiken zentraler Handelsplattformen sind; 1) Mangel an Sicherheit, 2) Mangel an Transparenz, und 3) Mangel an Liquidität.

Ein Mangel an Sicherheit ergibt sich daraus, dass Benutzer typischerweise die Kontrolle über ihre privaten Schlüssel (Mittel) an eine zentralisierte Einheit abgeben. Dies setzt die Benutzer der Gefahr aus, dass zentralisierte Austauschvorgänge böswilligen Hackern zum Opfer fallen. Die Sicherheits- und Hacking-Risiken [9] [10], mit denen alle zentralisierten Börsen konfrontiert sind, sind allgemein bekannt, werden aber oft als Einsatz für den Token-Handel akzeptiert. Zentralisierte Plattformen sind nach wie vor Honeypots für Hackangriffe, da ihre Server über Millionen von Dollar an Benutzergeldern verfügen. Zusätzlich besteht immer die Möglichkeit, dass Angestellte einer Handelsplattform Fehler machen was zum Verlust der Gelder führt. Einfach gesagt, Benutzer haben nicht die Kontrolle über ihre eigenen Token, wenn sie an einer zentralisierten Handelsplattform hinterlegt werden.

Ein Mangel an Transparenz führt zu der Gefahr, dass unehrliche Handelsplattformen unfair agieren. Hier muss unterschieden werden da der Anwender nicht wirklich mit seinen eigenen Vermögenswerten handelt sondern sozusagen einen Schuldschein ausgestellt bekommt. Wenn die Token auf das Wallet der Handelsplattform gesendet werden, übernimmt die Handelsplattform die Verwahrung und bietet stattdessen einen Schuldschein an. Alle Tauschgeschäfte finden demnach zwischen den Schuldscheinen der Anwender statt. Um seine Token abheben zu können wird der Schuldschein eingelöst und man bekommt seine Token auf sein externes Wallet überwiesen. Während dieses Prozesses besteht ein Mangel an Transparenz und der Austausch kann zum Stillstand kommen, ihr Konto könnte eingefroren werden, die Handelsplattform könnte bankrott gehen, usw. Es ist auch möglich, dass sie Anwendervermögen für andere Zwecke verwenden, wie z. B. Ausleihen an Dritte. Ein Mangel an Transparenz kann Benutzer zusätzliche Kosten bescheren, ohne dass ein Totalverlust der Mittel eintritt, wie etwa höhere Handelsgebühren, Verzögerungen bei zu Spitzenzeiten, regulatorische Risiken und Aufträge die im Vordergrund stehen.

Der Mangel an Liquidität. Aus der Sicht der Betreiber einer Handelsplattform verhindert eine fragmentierte Liquidität den Einstieg neuer Börsen aufgrund von zwei Szenarien. Zum einen gewinnt die Handelsplattform mit der größten Anzahl von Handelspaaren, weil es für die Nutzer wünschenswert ist, alle ihre Geschäfte an einer Börse abzuwickeln. Zum anderen gewinnt die Handelsplattform mit dem größten Auftragsbuch aufgrund günstiger Bid-Ask Spannen für jedes Handelspaar. Dies erschwert den Wettbewerb durch Neulinge, da es für sie schwierig ist, eine initiale Liquidität aufzubauen. Infolgedessen haben viele Börsen einen hohen Marktanteil, trotz vieler Beschwerden von Benutzern und sogar größeren Hacking-Vorfällen. Es ist erwähnenswert, dass zentralisierte Börsen, welche schnell wachsen, zu einem immer größeren Angriffsziel für Hacker werden.

Aus der Sicht der Benutzer reduziert eine fragmentierte Liquidität die Benutzerfreundlichkeit erheblich. In einer zentralisierten Börse können Benutzer nur innerhalb der eigenen Liquiditätspools der Handelsplattform, gegen ihr eigenes Auftragsbuch und zwischen ihren unterstützten Token-Paaren handeln. Um ein Token für ein Token zu handeln, müssen Anwender an einer Handelsplattform handeln welche beide Token unterstützt oder sich an verschiedenen Handelsplattformen registrieren, wobei sie immer persönliche Informationen offenlegen. Benutzer müssen häufig vorläufige Transaktionen durchführen, in der Regel gegen BTC oder ETH, und dabei Bid-Ask-Spannen zahlen. Schließlich sind die Auftragsbücher möglicherweise nicht groß genug, um den Handel ohne einen niedrigeren initialen Preis abschließen zu können. Selbst wenn die Handelsplattform ein großes Volumen verarbeiten möchte, gibt es keine Garantie dafür, dass dieses Volumen und diese Liquidität nicht gefälscht sind [11].

Das Ergebnis sind getrennte Liquiditätsspeicher und ein fragmentiertes Ökosystem, das dem alten Finanzsystem ähnelt, mit bedeutendem Handelsvolumen, das an wenigen Handelsplattformen zentralisiert ist. Die globalen Liquiditätsversprechungen von Blockchains sind im zentralisierten Austausch nicht sinnvoll.

2.2 Unzulänglichkeiten von dezentralisierten Handelsplattformen

Dezentralisierte Handelsplattformen unterscheiden sich von zentralisierten Handelsplattformen zum Teil dadurch, dass Benutzer die Kontrolle über ihre privaten Schlüssel (Vermögenswerte) behalten indem sie Geschäfte direkt mit der zugrunde liegenden Blockchain abwickeln. Durch die Nutzung der Trustless-Technologie von Kryptowährungen selbst können viele der oben genannten Sicherheitsrisiken erfolgreich gemildert werden. Probleme bestehen jedoch hinsichtlich der Leistung und der strukturellen Einschränkungen.

Die Liquidität bleibt häufig ein Problem, da die Benutzer nach Gegenparteien in unterschiedlichen Liquiditätspools und -standards suchen müssen. Fragmentierte Liquiditätseffekte treten auf, wenn DEXs oder dApps insgesamt keine konsistenten Standards für die Interoperabilität verwenden und wenn Aufträge nicht über ein weites Netzwerk verteilt / verbreitet werden. Die Liquidität von Limit-Auftragsbüchern und insbesondere deren Ausfallsicherheit - wie schnell erfüllte Limit-Aufträgen neu generiert werden - können die optimalen Handelsstrategien erheblich beeinflussen [12]. Das Fehlen solcher Standards hat nicht nur zu einer

verringerten Liquidität geführt, sondern auch zu einer Reihe von potenziell unsicheren proprietären Smart-Contracts.

Da Transaktionen auf der Blockchain ausgeführt werden, erben DEXs außerdem die Beschränkungen der zugrunde liegenden Blockchain, nämlich: Skalierbarkeit, Verzögerungen bei der Ausführung (Mining) und kostspielige Änderungen an Aufträgen. Daher skalieren Blockchain-Auftragsbücher nicht besonders gut, da das Ausführen von Code auf der Blockchain Kosten (Gas) verursacht, wodurch mehrere Auftragsabbruch-Kadenzen unerschwinglich teuer werden.

Da die Auftragsbücher der Blockchain öffentlich sind, ist die Transaktion, um einen Kauf in Auftrag zugeben, für die Miner sichtbar, da die Transaktion abgearbeitet werden muss um danach in einem Auftragsbuch plaziert zu werden. Diese Verzögerung setzt den Benutzer dem Risiko aus, in Front zu laufen und den Preis oder die Ausführung gegen ihn zu bewegen.

2.3 Hybride Lösungen

Aus den obigen Gründen haben rein Blockchain-basierte Handelsplattformen Beschränkungen, die sie gegen zentralisierten Austauschen nicht wettbewerbsfähig machen. Es gibt einen Kompromiss zwischen der inhärenten Vertrauenswürdigkeit der Blockchain und der zentralisierten Austauschgeschwindigkeit und Auftragsflexibilität. Protokolle wie Loopring und 0x [13] erweitern eine Lösung der On-Chain-Abwicklung mit Off-Chain-Order-Management. Diese Lösungen drehen sich um offene Smart Contracts, navigieren jedoch durch Skalierbarkeitsbeschränkungen indem sie mehrere Funktionen fernab der Blockchain ausführen und den Knoten Flexibilität bei der Erfüllung wichtiger Rollen für das Netzwerk geben. Die Nachteile bleiben jedoch auch für das Hybridmodell bestehen [14]. Das Loopring-Protokoll schlägt in diesem Whitepaper sinnvolle Unterschiede in unserem Ansatz für eine hybride Lösung vor.

3 Loopring Protokoll

Loopring ist kein DEX, sondern ein modulares Protokoll zum Aufbau von DEXs auf mehreren Blockchains. Wir zerlegen die Bestandteile einer traditionellen Handelsplattform und bieten stattdessen eine Reihe von offenen Smart-Contracts und dezentralen Akteuren an. Zu den Rollen im Netzwerk gehören Wallets, Relais, Blockchains für Konsortien mit Liquiditätsteilung, Auftragsbuch-Browser, Ring-Miners und Vermögenswert-Tokenisierungsdienste. Bevor wir die einzelnen definieren, sollten wir zuerst die Loopring-Aufträge verstehen.

3.1 Auftrags Ring

Loopring-Aufträge werden in einem sogenannten unidirektionalen Ordermodell (UDOM)[15] ausgedrückt. UDOM drückt Aufträge als Token-Austausch-Anfragen aus, `BetragS`/`BetragB`, (Betrag zu verkaufen/kaufen) statt Bids und Asks. Da jede Bestellung nur eine Austauschrate zwischen zwei Token ist, ist eine mächtige Eigenschaft des Protokolls das Mischen und Abgleichen mehrerer Bestellungen im Umlaufhandel. Durch den Einsatz von bis zu 16 Aufträgen anstelle eines einzelnen Handelspaars steigen die Liquidität und das Potenzial für eine Preisverbesserung dramatisch an.

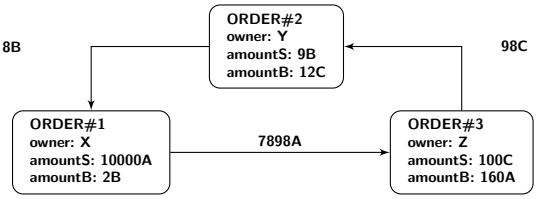

Abbildung 1: A order-ring of 3 Orders

Die obige Abbildung zeigt einen Auftrags Ring mit drei Aufträgen. Jeder Auftrags Token der zum Verkauf steht (`tokenS`), steht bei einem anderen Auftrag als Token zum Kauf (`tokenB`). Es entsteht eine Schleife, die es jeder Bestellung erlaubt, ihre gewünschten Token zu tauschen, ohne eine gegenläufige Reihenfolge für ihr Paar zu benötigen. Herkömmliche Auftrags-Paar-Abwicklungen können natürlich noch ausgeführt werden, was im Wesentlichen ein Sonderfall eines Auftrags-Rings ist.

Definition 3.1 (order-ring) *C_0, C_1, $\cdots$, C_{n-1} sind n unterschiedliche Token, $O_{0\to1}$, $\cdots$, $O_{i\to i\oplus1}$, $\cdots$, $O_{n-1\to0}$ sind n Aufträge. Diese Aufträge können einen Auftrags Ring formen:*

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

wobei n die Länge des Auftrags-Ringes ist, und $i \oplus 1 \equiv i+1 \mod n$.

Ein Auftrags-Ring ist gültig, wenn alle Komponententransaktionen zu einem Wechselkurs ausgeführt werden können, der gleich oder besser ist als der ursprünglich implizit vom Benutzer festgelegte Kurs. Um die Gültigkeit des Auftrags-Rings zu überprüfen, müssen Smart-Contracts mit Loopring-Protokoll Auftrags-Ringe von Ring-Minern erhalten, wenn das Produkt der ursprünglichen Wechselkurse aller Orders gleich oder größer als 1 ist.

Nehmen wir an, Alice und Bob wollen ihr Token `A` und `B` tauschen. Alice hat 15 Token `A` und sie möchte 4 Token `B` dafür; Bob hat 10 Token `B` und möchte 30 Token `A` dafür.

Wer kauft und wer verkauft? Dies hängt nur von dem Token ab welcher als Referenz hergenommen wird. Wenn Token `A` die Referenz ist, dann kauft Alice Token `B` für den Preis von $\frac{15}{4} = 3.75$`A`, während Bob 10 Token `B` für den Preis von $\frac{30}{10} = 3.00$`A` verkauft. Für den Fall das Token `B` die Referenz ist, Alice verkauft 15 Token `A` für den Preis von $\frac{4}{15} = 0.26666667$`B` und Bob kauft 10 Token `A` für den Preis von $\frac{10}{30} = 0.33333334$`B`. Deshalb ist der Käufer oder Verkäufer willkürlich.

Im ersten Fall ist Alice bereit einen höheren Preis (3.75`A`) zu bezahlen als Bob (3.00`A`) seine Token verkaufen würde, während im zweiten Fall Bob bereit ist einen höheren

Preis (0.33333334B) zu bezahlen als Alice (0.26666667B) ihre Token verkaufen würde. Es ist klar, dass ein Handel möglich ist, wann immer der Käufer bereit ist, einen gleichen oder höheren Preis als den Preis des Verkäufers zu zahlen.

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

Damit ein Satz von n Aufträgen ganz oder teilweise erfüllt werden kann, müssen wir also wissen, ob das Produkt eines jeden der Wechselkurse als Kaufaufträge zu einer Zahl größer oder gleich 1 führt. Wenn ja, können alle n Aufträge entweder teilweise oder vollständig erfüllt werden.

Wenn wir eine dritte Partei Charlie hinzufügen, so dass Alice x_1 Token `A` geben und dafür y_1 Token `B` erhalten möchte, Bob möchte x_2 Token `B` geben und dafür y_2 Token `C` erhalten möchte und Charlie möchte x_3 Token `C` geben und dafür y_3 Token `A` erhalten. Die erforderlichen Token sind vorhanden und der Handel ist möglich, wenn:

$$\frac{x1 \cdot x_2 \cdot x_3}{y_1 \cdot y_2 \cdot y_3} \geq 1 \quad (2)$$

Siehe Punkt 7.1 für detailliertere Informationen über Loopring Aufträge.

4 Teilnehmer im Ökosystem

Die folgenden Ökosystemteilnehmer stellen gemeinsam alle Funktionalitäten bereit, die ein zentralisierter Austausch bieten kann.

- **Wallets**: Ein allgemeiner Wallet-Service oder eine Schnittstelle, die Benutzern Zugriff auf ihre Token und eine Möglichkeit zum Senden von Aufträgen an das Loopring-Netzwerk bietet. Wallets werden durch das Teilen von Gebühren mit Ring-Minern angespornt Aufträge zu erstellen (siehe Abschnitt **??**). Mit der Überzeugung, dass die Zukunft des Handels innerhalb der sicheren Wallets einzelner Benutzers stattfinden wird, hat die Verbindung dieser Liquiditätspools durch unser Protokoll große Bedeutung.

- **Konsortium liquiditätsteilende Blockchain/Relais-Netzwerke**: Ein Relais Netzwerk für Auftrags und Liquiditäts Teilung. Wenn einzelne Knoten die Loopring Relais Software verwenden, sind diese in der Lage einem bereits existierenden Netzwerk beizutreten und seine Liquidität über eine Konsortium Blockchain zu teilen. Die Konsortium Blockchain, die wir als erst Implementierung erstellten, hat nahezu Echtzeit-Auftragsteilung (1-2 Sekunden-Blöcke) und trimmt alte Historien, um ein schnelleres Herunterladen durch neue Knoten zu ermöglichen.

- **Relais/Ring-Miner**: Relais sind Knoten welche Aufträge von Wallets oder Relais-Netzwerken erhalten, öffentliche Auftragsbücher und Auftrags Historien verwalten und optional Aufträge zu anderen Relais' übertragen (über irgendein beliebiges off-chain Medium). Ring-Minen ist ein Feature – und keine Anforderung – von Relais. Es ist rechenintensiv und erfolgt komplett fernab der Blockchain. Relais welche dieses Feature aktiviert haben werden "Ring-Miner" genannt, die Order-Ringe produzieren, indem sie unterschiedliche Aufträge zusammenfügen. Relays können frei wählen (1) wie sie sich untereinander verständigen, (2) wie sie ihre Auftragsbücher aufbauen und (3) wie sie Auftrags-Ringe abbauen (Mining-Algorithmen).

- **Loopring Protokoll Smart Contracts (LPSC)**: Ein Set von öffentlich frei verfügbaren Smart Contracts welche Auftrags-Ringe von Ring-Minern überprüfen, Token im Namen der Anwender transferieren, den Ring-Minern und Wallets Anreize durch Gebühren geben und Events übertragen. Relays/Auftrags-Browser überwachen diese Ereignisse, um ihre Auftragsbücher und Handelshistorien aktuell zu halten. Siehe Anhang A für Details.

- **Vermögenswert-Tokenisierungs-Service(VTS)**: Sie schlagen eine Brücke zwischen Vermögenswerten die nicht direkt über Loopring gehandelt werden können. Diese bestehen aus zentralisierten Diensten welche von vertrauenswürdigen Unternehmen oder Organisationen betrieben werden. Der Anwender deponiert Vermögenswerte (Real, Fiat, Token anderen Blockchains) und bekommt Token ausgestellt, die für eine Auszahlung eingelöst werden können. Loopring ist kein Blockchain übergreifendes Austauschprotokoll (bis eine geeignete Lösung existiert), jedoch ermöglicht VTS sowohl das Handeln von ERC20 Token [16] mit physikalischen Vermögenswerten als auch Vermögenswerte anderer Blockchains.

5 Austauschprozess

1. **Protokollautorisierung**: In Abbildung 2 autorisiert, Inhaber `Y`, der Token handeln möchte, autorisiert das LPSC `BetragS` von Token `B` zu verwalten, welche der Inhaber verkaufen möchte. Dies sperrt die Token des Inhabers nicht und ihm steht es frei die Token zu verschieben solange der Auftrag bearbeitet wird.

2. **Auftragserstellung**: Der aktuelle Preis und das Auftragsbuch für den Token `B` gegen Token `C` werden von Relais oder anderen Agenten, die mit dem Netzwerk verbunden sind,zur Verfügung gestellt, wie auch die Auftragsbuch-Browser. Inhaber `Y` erstellt einen Auftrag (Limit Auftrag) spezifiziert `BetragS` und `BetragB` sowie andere Parameter durch jede mögliche Wallet Schnittstelle. Der Betrag an LRx kann dem Auftrag als Gebühr für Ring-Miner hinzugefügt werden; Höhere LRx Gebühren bedeuten eine höhere Chance früher

von Ring-Minern bearbeitet zu werden. Der Hash des Auftrags wird durch den privaten Schlüssel von Inhaber `Y` signiert.

3. **Auftragsverteilung**: Das Wallet überträgt den Auftrag und seine Signatur an ein oder mehrere Relais. Die/Das Relais aktualisieren/aktualisiert ihr Auftragsbuch. Das Protokoll schreibt nicht vor wie diese Auftragsbücher auszusehen haben, wie z.B. dass der erste Auftrag als erstes bearbeitet wird. Stattdessen haben Relais die Freiheit ihre eigenen Entscheidungen beim Aufbauen eines Auftragsbuches zu treffen.

4. **Liquidität teilen**: Relais verteilen den Auftrag über ein beliebiges Übertragungsmedium mit anderen Relais. Auch hier ist es flexibel wie Knoten miteinander interagieren. Um ein gewisses Maß an Netzwerkkonnektivität zu ermöglichen, gibt es ein integriertes Liquiditäts-Sharing-Relais-Netzwerk, das eine Konsortium-Blockchain verwendet. Wie im vorherigen Abschnitt erwähnt, ist dieses Relay-Netzwerk für Geschwindigkeit und Inklusivität optimiert.

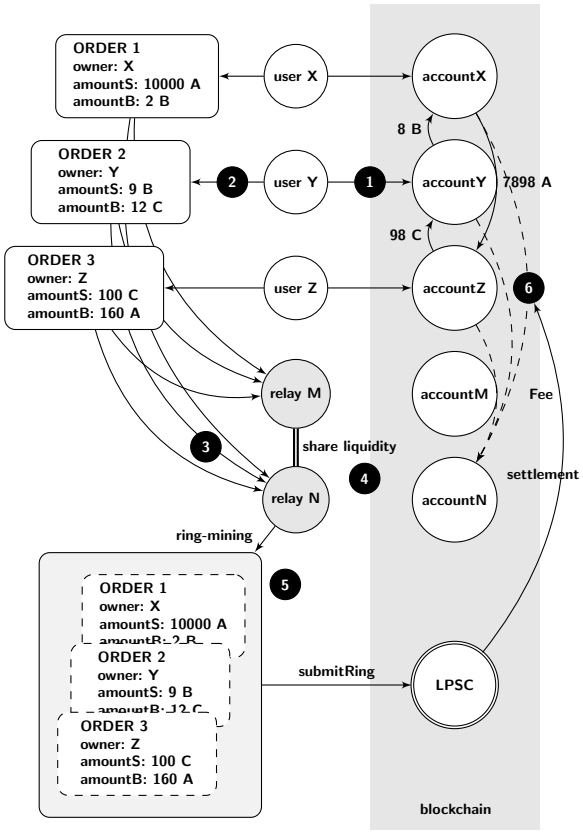

Abbildung 2: Loopring Exchange Process

5. **Ring-Mining (Auftragsabgleich)**: Ring-Miner versuchen den Auftrag vollständig oder teilweise zu dem geforderten Preis oder besser zu erfüllen indem er mit mehreren anderen Aufträgen abgeglichen wird. Ring-Mining ist der Hauptgrund durch den das Protokoll hohe Liquidität zu jedem Paar bereitstellen kann. Wenn der ausgeführte Kurs besser ist als der von Benutzer Y angegebene Wert, wird die Marge unter allen Orders im Order-Ring geteilt. Als Belohnung wählt der Ring-Miner, ob er einen Teil der Marge (Margen-Split dem Benutzer die LRx zurückgeben) beansprucht oder einfach die LRx-Gebühr einbehält.

6. **Überprüfung & Abrechnung**: Der Auftragsring wird von LPSC erhalten. Es führt mehrere Überprüfungen durch, um die von Ring-Minern gelieferten Daten zu verifizieren und stellt fest, ob der Bestellungsring ganz oder teilweise ausgeglichen werden kann (abhängig von der Füllrate von In-Ring-Bestellungen und Token in Benutzerportfolios). Wenn alle Überprüfungen erfolgreich sind, überträgt der Vertrag die Token automatisch an die Benutzer und zahlt gleichzeitig die Ring-Miner und Wallet-Gebühren. Wenn Inhabers `Y` Saldo, nicht ausreicht, entschiden vom LPSC, wird dieser Auftrag als verkleinert angesehen: Ein verkleinerter Auftrag wird automatisch auf die initiale Größe hochgestuft wenn ausreichende Mittel zur Verfügung stehen, indem sie überwiesen werden, nicht wie ein Abbruch, welche nur in eine Richtung funktioniert und nicht rückgängig gemacht werden kann.

6 Operative Flexibilität

Es ist wichtig zu beachten, dass der freie Loopring Standard den Teilnehmern eine große Flexibilität bei der Bedienung ermöglicht. Akteuren steht es frei, neue Geschäftsmodelle zu implementieren und den Nutzern einen Mehrwert zu bieten, indem sie LRx-Gebühren für das Volumen oder andere Kennzahlen im Prozess verdienen (wenn sie dies wünschen). Das Ökosystem ist modular aufgebaut und soll die Teilnahme an einer Vielzahl von Anwendungen unterstützen.

6.1 Auftragsbuch

Relays können ihre Auftragsbücher auf verschiedene Arten gestalten, um die Bestellungen der Benutzer anzuzeigen und abzugleichen. Eine erste Implementierung unseres eigenen Auftragsbuches folgt einem OTC-Modell, bei dem Limit Aufträge nur auf Basis des Preises positioniert werden. Zeitstempel von Aufträgen haben also keine Auswirkungen auf das Auftragsbuch, jedoch steht es dem Relay frei, sein Auftragsbuch so zu gestalten, wie ein typisches Auftragsbuches einer zentralisierten Handelsplattform bei dem Zeitstempel ebenso berücksichtigt werden. Wenn ein Relay geneigt ist, diese Art von Auftragsbuch anzubieten, kann es ein Wallet integrieren und diese Wallet-Aufträge ausschließlich an dieses einzeln Relay schicken, welches dann in der Lage sein würde, die Aufträge zeitabhängig abzugleichen. Eine solche Konfiguration ist möglich.

Während andere DEX-Protokolle zeitweise Ressourcen benötigen - anfängliche Token-Salden, um Taker-Aufträge zu platzieren - müssen Loopring-Relays nur nach zusammenpassenden Aufträgen suchen, um einen Handel zu vollenden. Sie können dies auch ohne initiale Token tun.

6.2 Liquiditätsteilung

Relays können frei entscheiden, wie sie Liquidität (Aufträge) miteinander teilen. Unser Konsortium Blockchain ist nur eine Lösung um dies zu erreichen und dem Ökosystem steht es frei wie es vernetzt oder kommuniziert. Sie können sich nicht nur einer Blockchain eines Konsortiums anschließen, sondern auch eigene Regeln erstellen und verwalten und Regeln / Anreize erstellen, wenn sie es für richtig halten. Relays können auch alleine arbeiten, wie in der zeitsensitiven Wallet-Implementierung zu sehen ist. Natürlich gibt es klare Vorteile bei der Kommunikation mit anderen Relays bei der Verfolgung von Netzwerkeffekten, jedoch können verschiedene Business Modelle andere Verteilmöglichkeiten und Verteilen der Gebühren in Betracht ziehen.

7 Protokoll Spezifikation

7.1 Anatomie eines Auftrags

Ein Auftrag ist ein Datenpaket, das die Absicht des Handels des Benutzers beschreibt. Eine Loopring-Auftrag wird mithilfe des unidirektionalen Ordermodells (UDOM) wie folgt definiert:

```
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;
}
```

Um den Ursprung des Auftrags sicherzustellen, wird er gegen den Hash seiner Parameter signiert ohne `authAddr`, mit dem privaten Schlüssel des Anwenders. Der `authAddr` Parameter wird für das Signieren von Auftragsringen, dessen Teil dieser Auftrag ist um Front-Running zu vermeiden, verwendet. Weitere Informationen finden Sie in Abschnitt 9.1. Die Signatur wird durch `v`, `r`, und `s` Felder dargestellt und wird zusammen mit den Auftragsparametern über das Netzwerk gesendet. Dies garantiert, dass die Bestellung während ihrer gesamten Lebensdauer unveränderbar bleibt. Auch wenn sich der Auftrag niemals ändert, kann das Protokoll seinen aktuellen Status basierend auf dem Saldo seiner Adresse zusammen mit anderen Variablen berechnen.

UDOM enthält keinen Preis (der von Natur aus eine Gleitkommazahl sein muss), sondern verwendet stattdessen den Begriff `rate` oder r, welcher als `amountS`/`amountB` ausgedrückt wird. Der Preis ist keine Gleitkommazahl sondern ein Ausdruck, der nur mit anderen unsignierten Integer zahlen bei Bedarf evaluiert wird um alle Zwischenergebnisse als vorzeichenlose Ganzzahlen zu behalten und die Rechengenauigkeit zu erhöhen.

7.1.1 Kaufbeträge

Wenn ein Ring-Miner Aufträge abgleicht, besteht die Möglichkeit das eine bessere Rate ausführbar ist. Dies erlaubt es dem Anwender mehr `tokenB` zu erhalten als `amountB` spezifiziert war. Sollte jedoch `buyNoMoreThanAmountB` auf `True` gesetzt sein, dann stellt das Protokoll sicher dass der Anwendet nicht mehr als `amountB` von `tokenB` erhält. Somit bestimmt UDOM's `buyNoMoreThantokenB` Parameter, wann ein Auftrag als vollständig erfüllt angesehen wird. `buyNoMoreThantokenB` gilt als eine Obergrenze für entweder `amountS` oder `amountB` und erlaubt es dem Anwender, detailliertere Handelsabsichten auszudrücken als traditionelle Kauf- / Verkaufsaufträge.

Zum Beispiel: mit `amountS` $= 10$ und `amountB` $= 2$, wäre der Kurs $r = 10/2 = 5$. Also möchte der der Anwender 5 `tokenS` für jeden `tokenB`. Der Ring-Miner gleicht ab und findet einen Anwender mit dem Kurs 4 welcher es dem Anwender erlaubt 2.5 `tokenB` anstatt 2 zu erhalten. Sollte der Anwender jedoch nur 2 `tokenB` wollen und hat den Parameter `buyNoMoreThanAmountB` auf `True` gesetzt, wird die Transaktion mit dem Kurs 4 und verkauft 4 `tokenS` für jeden `tokenB` und spart effektiv 2 `tokenS`. Es muss jedoch beachtet werden das die Mining-Gebühren dabei nicht in Betracht gezogen werden (siehe Abschnitt 8.1).

Wenn wir Folgendes verwenden

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

um einen Auftrag vereinfacht darzustellen, dann können traditionelle Kauf-Verkauf-Modelle für die ETH / USD-Märkte an einer traditionellen Handelsplattform die erste

und die dritte Order unten ausdrücken, aber nicht die anderen beiden:

1. Verkaufe 10 ETH für 300 USD/ETH. Dieser Auftrag kann wie folgt dargestellt werden: `Order(10, ETH, 3000, USD, False)`.
2. Verkaufe ETH für 300 USD/ETH um 3000 USD zu erhalten. Dieser Auftrag kann wie folgt dargestellt werden: `Order(10, ETH, 3000, USD, True)`.
3. Kaufe 10 ETH um 300 USD/ETH, dieser Auftrag kann wie folgt dargestellt werden: `Order(3000, USD, 10, ETH, True)`.
4. Gebe 3000 USD aus um so viele ETH zu kaufen wie möglich für einen Preis von 300 USD/ETH, dieser Auftrag kann wie folgt dargestellt werden: `Order(3000, USD, 10, ETH, False)`.

7.2 Ring Verifizierung

Die Loopring Smart Contracts führen keine Wechselkurs- oder Mengenberechnungen durch, sondern müssen empfangen und überprüfen, was die Ring-Miner für diese Werte liefern. Diese Berechnungen werden aus zwei Gründen von Ring-Minern durchgeführt: (1) die Programmiersprache für Smart-Contracts, wie Solidity[17] auf Ethereum, unterstützt keine Gleitkomma-Berechnung, speziell $pow(x, 1/n)$ (Berechnung der n-ten Wurzel einer Gleitkommazahl) und (2) Es ist wünschenswert, dass die Berechnung außerhalb der Blockchain erfolgt, um die Blockchain-Berechnungskosten zu reduzieren.

7.2.1 Sub-Ring Überprüfung

Dieser Schritt verhindert, dass Arbitrageure die gesamte Marge in einem Auftragsring unfair realisieren, indem sie neue Aufträge implementieren. Sobald ein Ring-Miner einen gültigen Auftrags-Ring gefunden hat, könnte es verlockend sein, dem Auftrags-Ring weitere Aufträge hinzuzufügen, um die Marge des Benutzers vollständig zu absorbieren (Preisnachlässe). Wie unten in Abbildung 3 dargestellt, zusammengezählt ergeben $x1$, $y1$, $x2$ und $y2$ das Produkt aller Kurse, genau 1, somit wird es keinen Preisnachlass geben.

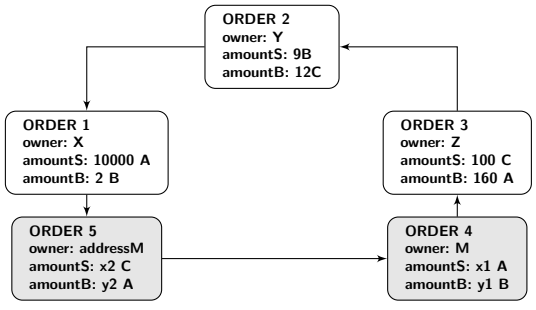

Abbildung 3: Ein Auftrags-ring mit Sub-Ring

Dies ist ein Null-Risiko, Null-Wert-Zusatz zum Netzwerk und wird als unfaires Verhalten des Ring-Miner angesehen. Um dies zu verhindern spezifiziert Loopring, dass eine gültige Schleife keine Sub-Ringe enthalten darf. Um dies zu überprüfen, stellt die LPSC sicher, dass ein Token nicht zweimal in einer Kauf- oder Verkaufsposition sein kann. Im obigen Diagramm können wir sehen, dass Token `A` zweifach als Kauf und Verkauf verwendet wird. Dies wäre nicht erlaubt.

7.2.2 Füllratenprüfung

Die Wechselkursberechnungen im Auftrags-Ring werden von Ring-Miner aus den oben genannten Gründen vorgenommen. Es ist der LPSC, der sicherstellen muss, dass sie korrekt sind. Erstens überprüft es, ob die Kaufrate, die der Ring-Miner für jede Bestellung ausführen kann, gleich oder kleiner als die ursprüngliche Kaufrate ist, die vom Benutzer festgelegt wurde. Dies stellt sicher, dass der Benutzer mindestens den Wechselkurs erhält, den er verlangt oder besser. Sobald die Wechselkurse bestätigt sind, stellt der LPSC sicher, dass jede Bestellung im Auftrags-Ring den gleichen Preisnachlass teilt. Zum Beispiel, wenn der ermäßigte Kurs γ ist, dann ist der Preis für jeden Auftrag:

$r_{0\to1} \cdot (1-\gamma)$, $r_{1\to2} \cdot (1-\gamma)$, $r_{2\to0} \cdot (1-\gamma)$ und erfüllt:

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

deshalb:

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

Wenn die Transaktion n Aufträge überschreitet, der `Diskont`:

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

wobei r^i ist die Auftragsfluktuationsrate von i-ten Auftrag. Offensichtlich nur, wenn der Diskontsatz ist $\gamma \geq 0$, können diese Aufträge erfüllt werden; und der i-ten Auftrag (O^i)'s wirklicher Kurs ist $\hat{r^i} = r^i \cdot (1-\gamma)$, $\hat{r^i} \leq r^i$.

7.2.3 Erfüllungsverfolgung & Abbruch

Ein Benutzer kann einen Auftrag teilweise oder komplett abbrechen, indem er eine spezielle Transaktion an den LPSC sendet, die Informationen über den Auftrag und den abzubrechenden Betrag enthält. Der LPSC beachtet dies, speichert den abzubrechenden Betrag und sendet ein `OrderCancelled` Event an das Netzwerk. Der LPSC verfolgt gefüllte und stornierte Beträge, indem sie ihre Werte mit dem Hash der Bestellung als Kennung speichert. Diese Daten sind offen einsehbar und `OrderCancelled` / `OrderFilled` Events werden gesendet wenn sie sich verändern. Das Verfolgen dieser Werte ist für den LPSC während des Auftragsring-Abrechnungsschritts kritisch.

LPSC unterstützt ebenso das Abbrechen aller Aufträge für ein Handelspaar mit dem `OrdersCancelled` Event und

das Abbrechen aller Aufträge für eine Adresse mit dem `AllOrdersCancelled` Event.

7.2.4 Auftragsskalierung

Aufträge werden nach der Historie der erfüllten und stornierten Beträge und dem aktuellen Kontostand der Absender skaliert. Der Prozess findet den Auftrag mit dem kleinsten zu erfüllenden Betrag gemäß den obigen Merkmalen und verwendet ihn als Referenz für die Skalierung aller Transaktionen im Auftragsring. Wenn der niedrigste Auftrag nach Wert gefunden wurde, kann das helfen das Füllvolumen für jede Bestellung zu ermitteln. Wenn zum Beispiel der i-te Auftrag der niedrigste wäre, dann kann die Anzahl der von jeder Bestellung $\hat{s}$ verkauften Token und die Anzahl der von jeder Bestellung gekauften Token $\hat{b}$ berechnet werden:

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

wobei $\overline{s}_i$ das Saldo ist, welches übrig bleibt nachdem die Aufträge teilweise erfüllt wurden.

Während der Implementierung können wir sicher annehmen, dass jeder Auftrag im Auftrags-Ring den niedrigsten Wert hat, und durchläuft dann den Auftrags-Ring höchstens zweimal um das Füllvolumen jeder Order zu berechnen.

Beispiel: Wenn der kleinste zu füllende Betrag im Vergleich zur ursprünglichen Reihenfolge 5% ist, werden alle Transaktionen im Auftrags-Ring auf 5% herunterskaliert. Sobald die Transaktionen abgeschlossen sind, sollte die Bestellung, die den geringsten noch zu füllenden Betrag aufweist, vollständig erfüllt werden.

7.3 Ring Regelung

Wenn der Auftrags-Ring alle vorherigen Prüfungen erfüllt hat, kann der Auftrags-Ring geschlossen und Transaktionen können durchgeführt werden. Dies bedeutet, dass alle n Aufträge einen geschlossenen Auftrags-Ring bilden und verbunden sind wie in Abbildung 4:

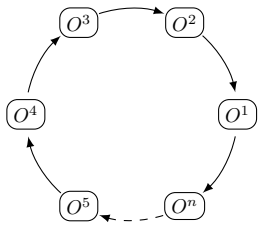

Abbildung 4: Ring Regelung

Um die Transaktion auszuführen verwendet der LPSC den `TokenTransferDelegate` Smart-Contract. Die Einführung eines solchen Delegaten erleichtert die Aktualisierung des Protokoll-Smart-Contracts, da alle Aufträge nur diesen Delegaten anstelle verschiedener Versionen des Protokolls autorisieren müssen.

Für jeden Auftrag im Auftrags-Ring wird eine Zahlung von `tokenS` für den nächsten oder vorhergehenden, abhängig von der Implementierung, Auftrag gemacht. Dann wird die Ring-Miner Gebühr abhängig vom Gebühren Modell, welches von dem Ring-Miner ausgewählt wurde bezahlt. Wurden alle Transaktionen ausgeführt wird ein `RingMined` Event gesendet.

7.3.1 Events

Das Protokoll sendet Events, welche es Relays, Order-Browsern und anderen Teilnehmern erlauben Auftragsbuch-Aktualisierungen so effizient wie möglich zu erhalten. Die gesendeten Events sind:

- **OrderCancelled**: Ein spezieller Auftrag wurde abgebrochen.
- **OrdersCancelled**: Alle Aufträge aller Handelspaare einer besitzenden Adresse wurden abgebrochen.
- **AllOrdersCancelled**: Alle Aufträge aller Handelspaare einer besitzenden Adresse wurden abgebrochen.
- **RingMined**: Ein Auftragsring wurde erfolgreich abgearbeitet. Dieses Event enthält Daten für jeden im Ring Token Transfer.

8 LRx Token

LRx ist unsere verallgemeinerte Tokennotation. LRC ist der Loopring Token auf Ethereum, LRQ auf Qtum und LRN auf NEO usw. Andere LRx-Typen werden in Zukunft eingeführt, wenn Loopring auf anderen öffentlichen Blockchains eingesetzt wird.

8.1 Gebühren Modell

Wenn ein Benutzer einen Auftrag erstellt, wird ein Betrag angegeben, der an die Ring-Miner als Gebühr bezahlt wird, in Verbindung mit einem Prozentsatz der Marge (`marginSplitPercentage`) auf dem Auftrag, welche der Ring-Miner beanspruchen kann. Dies wird als Margensplit bezeichnet. Die Entscheidung welche (Gebühren- oder Margensplit), bleibt dem Ring-Miner überlassen.

Eine Darstellung des Margensplit:

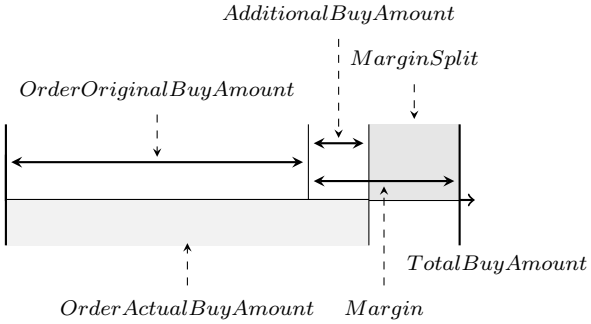

Abbildung 5: A 60% Margin Split

Wenn die Marge auf dem Auftrags-Ring zu klein ist, wählt ein Ring-Miner die LRx-Gebühr. Wenn die Marge im Gegensatz dazu so hoch ist, sodass der daraus resultierende Margensplit weit über die LRx-Gebühr hinausgeht, wählt ein Ring-Miner den Margensplit. Es gibt noch einen weiteren Vorbehalt: Wenn der Ring-Miner den Margensplit wählt, muss er dem Benutzer (Auftragserzeuger) eine Gebühr zahlen welche den LRx entspricht, die der Benutzer dem Ring-Miner als Gebühr gezahlt hätte. Dies erhöht den Schwellenwert für den Fall, in dem der Ring-Miner den Margensplit auf das Doppelte der LRx-Gebühr des Auftrags wählt. Dies macht die LRx-Gebühre attraktiver. Es ermöglicht den Ring-Minern, ein konstantes Einkommen auf Auftrags-Ringen mit niedriger Marge zu erhalten, um bei höheren Auftrags-Ringen weniger Erträge zu erhalten. Unser Gebühren Modell basiert auf der Erwartung, dass mit zunehmendem und reifendem Markt weniger Auftrags-Ringe mit hohen Margen auftreten und somit fixe LRx-Gebühren als Anreiz nötig sind.

Am Ende steht folgender Graph:

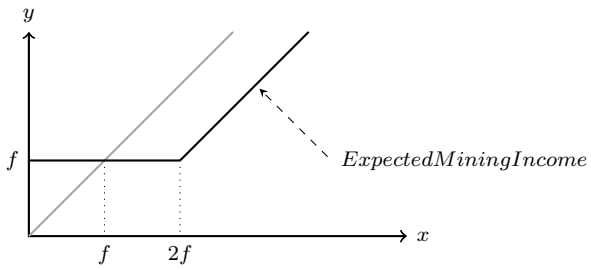

Abbildung 6: Loopring's Fee Model

wobei f die LRx Gebühr ist, x ist der Margensplit, y ist das Mining Einkommen. $y = max(f, x - f)$ wie durch die durchgezogene Linie angezeigt; wenn die LRx Gebühr für den Auftrag 0 ist, die Gleichung ist$y = max(0, x - 0)$ vereinfacht zu $y = x$ wie von der grauen Linie angezeigt wird.

Die Konsequenzen sind:

1. Wenn der Margensplit 0 ist werden Ring-Miner die Flache LRx Gebührt wählen und sind immer noch motiviert.
2. Wenn die LRx-Gebühr 0 ist, ergibt sich die graue Linie und das Einkommen basiert auf einem allgemeinen linearen Modell.
3. Wenn das Margensplit-Einkommen größer als 2x ist (LRx-Gebühr), wählen die Ring-Miner den Margin-Split und zahlen LRx an den Nutzer.

Es sollte angemerkt werden, dass, wenn die LRx Gebühr nicht Null ist, egal welche Option der Ring-Miner wählt, es immer eine Übertragung von LRx zwischen dem Ring-Miner und dem Absender des Auftrags geben wird. Entweder erhält der Ring-Miner die LRx-Gebühr oder zahlt die LRx-Gebühr an den Absender zurück, um den Margin-Split zu übernehmen.

Ring-Miner teilen einen bestimmten Prozentsatz der Gebühren mit Wallets. Wenn ein Benutzer einen Auftrag über ein Wallet abgibt und dieser erfüllt wird, wird das Wallet mit einem Teil der Gebühren oder der Margenaufteilung belohnt.Obwohl dies modular ist und einzigartige Geschäftsmodelle oder Implementierungen möglich sind, neigen wir dazu, dass Wallets etwa 20% - 25% der verdienten Gebühren erhalten. Wallets stellen ein Hauptziel für die Loopring-Protokollintegration dar, da sie die Benutzerbasis sind, aber wenig oder keine Einkommensquelle haben.

8.2 Dezentralisierte Steuerung

Das Loopring-Protokoll ist ein soziales Protokoll in dem Sinne, dass es auf Koordination zwischen den Mitgliedern angewiesen ist, um effektiv auf ein Ziel hin zu arbeiten. Dies ist den kryptoökonomischen Protokollen insgesamt nicht unähnlich, und in der Tat wird seine Nützlichkeit weitgehend durch die gleichen Mechanismen der Koordinationsprobleme [18], des grimmigen Triggergleichgewichts und der beschränkten Rationalität geschützt. Zu diesem Zweck werden LRx-Token nicht nur zur Gebührenzahlung verwendet, sondern auch, um die finanziellen Anreize der verschiedenen Netzwerkteilnehmer in Einklang zu bringen. Eine solche Anpassung ist für die breite Verabschiedung eines Protokolls erforderlich und besonders wichtig für Austauschprotokolle, da der Erfolg weitgehend auf der Verbesserung der Liquidität in einem robusten dezentralen Ökosystem beruht.

LRx-Token werden verwendet, um Protokoll-Updates durch dezentrale Steuerung zu bewirken. Intelligente Vertragsaktualisierungen werden von Token-Inhabern geregelt, um Kontinuität und Sicherheit zu gewährleisten und die Risiken der Liquidität durch Inkompatibilität zu mindern. Angesichts der Tatsache, dass intelligente Verträge nach der Bereitstellung nicht mehr geändert werden können, besteht das Risiko, dass dApps oder Endbenutzer weiterhin mit veralteten Versionen interagieren und sich von aktualisierten Verträgen ausschließen. Die Aktualisierbarkeit ist entscheidend für den Erfolg des Protokolls, da es sich an die Marktanforderungen und die zugrunde liegenden Blockchains anpassen muss. Eine dezentrale Steuerung durch die LRx-Stakeholder ermöglicht Protokoll-Smart-Contract-Updates, ohne dApps

oder Endbenutzer zu stören oder sich zu sehr auf die intelligente Vertragsabstraktion zu verlassen. Zunächst wird dies durch einen einfachen Multisignatur-Smart-Contract geschehen, um auf einen DAO-Typ-Mechanismus hinzuarbeiten.

9 Betrugs- und Angriffsschutz

9.1 Front-Running-Prävention

In dezentralisierten Handelsplattformen bezeichnet man als Front-Running, wenn jemand versucht eine Handelsstrategie eines anderen Konten zu kopieren und zu verarbeiten bevor die ursprüngliche Transaktion welche sich noch im Pool befindet bearbeitet wird. Dies kann durch Angabe einer höheren Transaktionsgebühr (Gaspreis) erreicht werden. Das Hauptschema von Front-Running in Loopring (und jedes Protokoll mit Auftragsabgleichung) sind Auftrags-Filch: wenn ein Front-Runner einen oder mehrere Aufträge von einem ausstehenden Auftrags-Pool stiehlt und speziell im Fall von Loopring, wenn ein Front-Runner einen kompletten Auftrags-Ring einer ausstehenden Transaktion stiehlt.

Wenn eine submitRing Transaktion noch nicht bestätigt wurde und noch immer im ausstehenden Auftrags Pool ist. Diese kann von jedem einfach gefunden werden und `minerAddress` kann mit der eigenen Adresse (`filcherAddress`) ersetzt werden, danach können die Nutzdaten mit `filcherAddress` erneut signiert werden. Der Fincher kann einen höheren Gas Preis setzen und eine neue Transaktion erstellen welche dann vor der ursprünglichen Transaktion von den Minern bearbeitet wird.

Frühere Lösungen für dieses Problem hatten wichtige Nachteile: sie erforderten mehr Transaktionen und somit kosten die Miner mehr Gas; benötigen doppelt so viele Blöcke um einen Auftrags-Ring abzuschließen. Unsere neue Lösung Dual-Authoring[19], umfasst den Mechanismus der Einrichtung von zwei Autorisierungsstufen für Aufträge - einen für die Abwicklung und einen für die Ring-Auftrags Abarbeitung..

Dual Authoring Prozess:

1. Für jeden Auftrag generiert die Wallet-Software ein zufälliges öffentliches / privates Schlüsselpaar und fügt das Schlüsselpaar in das JSON-Snippet des Auftrags ein. (Eine Alternative besteht darin, die vom öffentlichen Schlüssel abgeleitete Adresse anstelle des öffentlichen Schlüssels selbst zu verwenden, um die Bytegröße zu reduzieren. Wir verwenden `authAddr` um eine solche Adresse darzustellen und `authKey` anstatt `authAddr`'s passenden privaten Schlüssel).

2. Berechne den Auftrags Hash mit allen Feldern außer `r`, `v`, `s` und `authKey`, signiere den Hash mit dem `owner`'s privaten Schlüssel (nicht `authKey`).

3. Das Wallet sendet den Auftrag zusammen mit dem `authKey` zu den Relays um Ring-Mining zu ermöglichen. Ring-Miner überprüfen ob `authKey` und `authAddr` korrekt gepaart wurden und die Auftragssignatur mit der `owner` Adresse übereinstimmt.

4. Wenn ein Auftrags-Ring identifiziert wurde verwenden die Ring-Miner jedes Auftrags `authKey` um den Ring Hash zu signieren, `minerAddress` und alle anderen Mining Parameter. Enthält ein Auftrags-Ring n Aufträge, so gibt es auch n Signaturen von n `authKey`s. Wir nennen diese Signaturen `authSignature`s. Der Ring-Miner muss möglicherweise auch den Hash des Rings zusammen mit allen Mining-Parametern unter Verwendung von`minerAddress`'s privaten Schlüssels signieren.

5. Der Ring-Miner ruft die submitRing Funktion mit allen Parametern und zusätzlich allen `authSignature`s. `authKey`s sind NICHT Teil der Blockchain Transaktion und bleiben deshalb für jeden außerhalb des Rings unbekannt.

6. Das Loopring Protokoll wird nun alle `authSignature` gegen die zugehörige `authAddr` von jedem Auftrag verifizieren und den Ring ablehnen sollten `authSignature`en fehlen oder ungültig sein..

Das Ergebnis ist:

- Die Auftragssignatur (mit dem privaten Schlüssel `owner`s Adresse) garantiert, dass der Auftrag inklusive der `authAddr` nicht verändert werden kann.

- Die Ring-Miner Signature (mit dem privaten Schlüssel `minerAddress`), wenn mitgegeben, garantiert, dass niemand mit der gleichen Identität einen Auftrags-Ring abarbeiten kann.

- Die `authSignature`en versichern dass der gesammte Auftragsring nicht verändert werden kann, inklusive `minerAddress` und Aufträge nicht gestohlen werden können.

Dual Authoring verhindert ring-Filch und order-Filch und stellt gleichzeitig sicher, dass die Abwicklung von Auftrags-Ringen in einer einzigen Transaktion durchgeführt werden kann. Darüber hinaus öffnet Dual Authoring Türen für Relays, um Aufträge auf zwei Arten zu teilen: nicht passende Freigabe und passende Freigabe. Standardmäßig arbeitet Loopring mit einem OTC-Modell und unterstützt nur Limit-Preis-Aufträge, d.h. die Zeitstempel der Aufträge werden ignoriert. Dies bedeutet, dass ein Front-Running eines Handels keine Auswirkungen auf den tatsächlichen Preis dieses Handels hat, sondern sich darauf auswirkt, ob er ausgeführt wird oder nicht.

10 Andere Angriffsszenarien

10.1 Sybil oder DOS Attacke

Böswillige Benutzer – handeln als sie selbst oder über gefälschte Identitäten – könnte eine große Anzahl kleiner Aufträge senden, um Loopring-Knoten anzugreifen. Da wir es jedoch zulassen, dass Knoten Aufträge aufgrund eigener Kriterien ablehnen – welche sie verstecken oder offenbaren können – werden die meisten dieser Aufträge abgelehnt, wenn sie bei der Abstimmung keinen zufriedenstellenden Gewinn erzielen. Indem wir Relays befähigen, selbst zu entscheiden, wie sie Aufträge verwalten, sehen wir einen Angriff durch viele winzige Aufträge nicht als Bedrohung.

10.2 Unzureichendes Guthaben

Böswillige Benutzer können Aufträge bearbeiten und verteilen, deren Wert nicht Null ist, deren Adresse aber tatsächlich kein Guthaben hat. Knoten können überwachen und bemerken, dass der tatsächliche Saldo einiger Aufträge Null ist, sie aktualisieren diese Auftragszustände entsprechend und verwerfen Sie sie anschließend. Knoten müssen Zeit investieren, um den Status eines Auftrags zu aktualisieren, können aber den Aufwand minimieren, indem sie beispielsweise Adressen auf die schwarze Liste setzen und zugehörige Bestellungen automatisch löschen.

11 Conclusio

Das Loopring-Protokoll ist eine grundlegende Schicht für den dezentralen Austausch. Dadurch hat es tiefgreifende Rückwirkungen auf den Austausch von Vermögenswerten und Werten. Geld, als eine Zwischenware, erleichtert oder ersetzt den Tauschhandel und löst das doppelte Zusammentreffen von Wunschproblemen[20], wobei zwei Kontrahenten das unterschiedliche Gut oder den Dienst des anderen begehren müssen. In ähnlicher Weise zielt das Loopring-Protokoll darauf ab, unsere Abhängigkeiten vom Zufall der Wünsche in Handelspaaren aufzuheben, indem Ring-Abgleich verwendet wird, um einen Handel leichter durchzuführen. Es ist bedeutsam wie die Gesellschaft und die Märkte Token, traditionelle Vermögenswerte und darüber hinaus austauschen. So wie dezentralisierte Kryptowährungen eine Bedrohung für die Kontrolle eines Landes über Geld darstellen, so ist ein kombinatorisches Protokoll, das Händler (Konsumenten / Produzenten) maßstabsgetreu zusammenbringen kann, eine theoretische Bedrohung für das Konzept des Geldes selbst.

Die Protokoll Vorteile umfassen:

- Off-Chain-Auftragsverwaltung und On-Chain-Abwicklung bedeutet keine Leistungseinbußen für die Sicherheit.
- Höhere Liquidität durch Ring-Mining und Auftragsteilung.
- Dual Authoring löst das gefährliche Problem des Front Runnings, das heute von allen DEXs und ihren Benutzern bewältigt werden muss.
- Mit kostenlosen, öffentlichen Smart-Verträgen kann jede dApp das Protokoll erstellen oder mit diesem interagieren.
- Die Standardisierung unter den Operatoren ermöglicht Netzwerkeffekte und eine verbesserte Endbenutzererfahrung.
- Das Netzwerk wird mit Flexibilität in der Ausführung der Auftragsbücher und der Kommunikation aufrechterhalten.
- Geringere Eintrittsbarrieren bedeuten niedrigere Kosten für Knoten, die dem Netzwerk und den Endnutzern beitreten.
- Anonymes Handeln direkt vom privaten Wallet.

12 Danksagungen

Wir möchten unseren Mentoren, Beratern und den vielen Menschen in der Gemeinschaft, die so herzlich und großzügig mit ihrem Wissen waren, unseren Dank aussprechen. Insbesondere möchten wir Shuo Bai (von ChinaLedger); Professor 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 und Encephalo Path für die Überprüfung des Projekts sowie das Feedback danken.

Literatur

[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] Fabian Schuh and Daniel Larimer. Bitshares 2.0: Financial smart contract platform, 2015.

[7] Bancor protocol. *URL https://bancor.network/*, 2017.

[8] Yaron Velner Loi Luu. Kybernetwork: A trustless decentralized exchange and payment service. https://kr.kyber.network/assets/KyberNetworkWhitepaper.pdf, Accessed: 2018-03-05.

[9] Reuters. Coincheck. https://www.reuters.com/article/us-japan-cryptocurrency-q-a/the-coincheck-hack-and-the-issue-with-crypto-assets-on-centralized-exchanges-idUSKBN1FI0K4, Accessed: 2018-03-05.

[10] Robert McMillan. The inside story of mt. gox, bitcoin's 460 dollar million disaster. 2014.

[11] Sylvain Ribes. Chasing fake volume: a crypto-plague, Accessed: 2018-03-10.

[12] Rossella Agliardi and Ramazan Gençay. Hedging through a limit order book with varying liquidity. 2014.

[13] Will Warren and Amir Bandeali. 0x: An open protocol for decentralized exchange on the ethereum blockchain, 2017.

[14] Iddo Bentov and Lorenz Breidenbach. The cost of decentralization. http://hackingdistributed.com/2017/08/13/cost-of-decent/, Accessed: 2018-03-05.

[15] 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.

[16] Fabian Vogelsteller. Erc: Token standard. *URL https://github.com/ethereum/EIPs/issues/20*, 2015.

[17] Chris Dannen. *Introducing Ethereum and Solidity*. Springer, 2017.

[18] Vitalik Buterin. Notes on blockchain governance, Accessed: 2018-03-05.

[19] 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.

[20] 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

Anhang A Loopring auf Ethereum

A.1 Smart Contracts

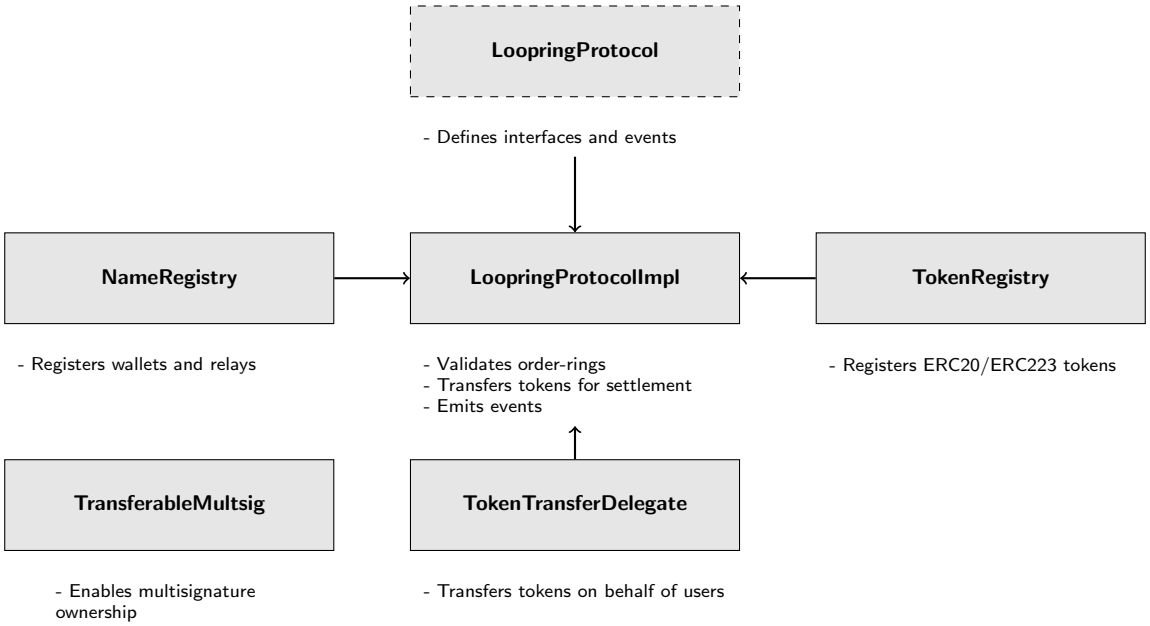

Abbildung 7: Smart Contracts

A.2 Verteilung

Die folgenden Smart Contracts wurden auf Ethereum Mainnet implementiert:

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