Loopring:
Ein dezentrales Protokoll zum Tokenhandel

Daniel Wang
daniel@loopring.org

Jay Zhou
jay@loopring.org

Alex Wang
alex@loopring.org

Matthew Finestone
matt.finestone@gmail.com

https://loopring.org

9. Mai 2018

Kurzdarstellung

Loopring ist ein offenes Protokoll, um dezentrale Handelsplattformen zu vernetzen. Mittels öffentlicher Smart Contracts zur Ausführung des Tokenhandels werden Aufträge mehrerer Akteure gesammelt, ausgewertet und unabhängig der Blockchain durchgeführt. Es handelt sich um ein freies, erweiterbares Protokoll und dient als standardisierter Grundbaustein für dezentrale Applikationen (dApps), die den Handel von Token umfassen. Interoperable Standards ermöglichen einen anonymen Handel ohne Dritte. Eine wichtige Verbesserung gegenüber den derzeitigen dezentralen Handelsprotokollen ist die Möglichkeit, dass Aufträge mit anderen, unterschiedlichen Bestellungen gemischt werden können. Dadurch entfällt die Notwendigkeit von Zwei-Token-Handelspaaren und die Liquidität wird drastisch verbessert. Loopring verwendet eine einzigartige und solide Lösung, um ein Front-Running zu verhindern: der unfaire Versuch, Transaktionen schneller als der ursprüngliche Tauschanbieter in einen Block einzureichen. Loopring ist Blockchain-unabhängig und auf jede Blockchain mit Smart-Contract-Funktionalität anwendbar. Derzeit ist Loopring auf Ethereum [1] [2] und Qtum [3] nutzbar und befindet sich für NEO [4] in der Entwicklung.

1 Einleitung

Mit der Verbreitung von Blockchain-basierten Vermögenswerten hat die Notwendigkeit, diese Güter untereinander auszutauschen, deutlich zugenommen. Da tausende neue Token zusätzlich zu traditionellen Vermögenswerten eingeführt werden, gibt es eine steigende Nachfrage. Unabhängig davon, ob Token zum spekulativen Handel oder für den Einsatz als Utility-Token auf Plattformen erworben werden, ist es essentiell für ein größeres Ökosystem, Krypto-Vermögenswerte untereinander austauschen zu können. Derzeit herrscht eine wachsende Dynamik unter Vermögenswerten [5]. Um diese zu ermöglichen und Kapital freizugeben, ist nicht nur die Beanspruchung von Eigentum erforderlich, was Blockchains nach wie vor erlauben, sondern auch die Möglichkeit, diese Vermögenswerte frei zu transferieren und zu transformieren.

Der Austausch von Token (Werten) ohne Zwischenhändler ist eine überzeugende Anwendung der Blockchain-Technologie. Dennoch haben sich Krypto-Enthusiasten bislang weitgehend für den Tokenhandel auf traditionellen, zentralisierten Handelsplattformen entschieden. Das Loopring-Protokoll wird benötigt, da - wie Bitcoin [6] bereits pflichtgemäß betonte - in Bezug auf elektronisches Geld "entscheidende Vorteile verloren gehen, wenn noch eine vertrauenswürdige dritte Partei benötigt wird, um doppelte Ausgaben zu vermeiden". Von daher gehen die Hauptvorteile von dezentralisierten Vermögenswerten verloren, wenn sie über zentrale Handelsplattformen laufen müssen.

Der Handel dezentraler Token an zentralisierten Börsen ist auch aus philosophischer Sicht nicht nachvollziehbar, da das zugrundeliegende Konzept von dezentralisierten Projekten keine Anwendung findet. Es gibt auch zahlreiche praktische Risiken und Einschränkungen bei der Verwendung zentralisierter Tauschbörsen, die nachstehend beschrieben werden. Dezentralisierte Handelsplattformen (DEXs) [7] [8] [9] haben versucht, diese Probleme zu lösen und in vielen Fällen ist es gelungen, Sicherheitsrisiken durch die Verwendung von Blockchains zur Einlagenumschichtung zu mindern. Da die Möglichkeiten dezentraler Handelsplattformen zu einer entscheidenden Infrastruktur für die neue Wirtschaft beitragen, gibt es erheblichen Spielraum für Leistungsverbesserungen. Loopring möchte Module für diese Infrastruktur mittels offenem Protokoll und der Erweiterbarkeit durch dApps bereitstellen.

2 Aktuelle Handelsplattformen

2.1 Unzulänglichkeiten zentraler 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 Nutzer typischerweise die Kontrolle über die privaten Schlüssel ihrer Adressen bzw. Konten an eine zentrale Einheit abgeben. Dies setzt Nutzer der Gefahr aus, dass jene zentralisierte Handelsplattformen kriminellen Hackern zum Opfer fallen können. Die Sicherheits- und Hacking-Risiken [10] [11], mit denen sich alle zentralisierten Börsen konfrontiert sehen, sind allgemein bekannt, werden beim Tokenhandel aber oft als gegeben hingenommen. Zentrale Plattformen sind nach wie vor lukrative Ziele für Hackerangriffe, da ihre Server Millionen von Dollar an Vermögenswerten verwalten. Zusätzlich besteht immer die Möglichkeit, dass Programmierern einer Handelsplattform Fehler unterlaufen können, die womöglich zum Verlust von Anlagen führen. Daraus folgt, dass Nutzer nicht die Kontrolle über ihre Vermögenswerte haben, wenn sie an einer zentralisierten Handelsplattform hinterlegt werden.

Ein Mangel an Transparenz führt zum Risiko, dass unseriöse Handelsplattformen unfair agieren. Der Grund liegt darin, dass der Anwender nicht mit seinen tatsächlichen Vermögenswerten handelt, sondern lediglich einen Schuldschein ausgestellt bekommt. Wenn Token in die Geldbörse (Wallet) einer Handelsplattform eingezahlt werden, übernimmt diese die Verwahrung und stellt anstelle der Token einen Schuldschein aus. Alle Tauschgeschäfte finden nunmehr zwischen den Schuldscheinen der Nutzer statt. Um seine Token abzuheben, wird der Schuldschein eingelöst und die Token an eine externe Wallet überwiesen. Während dieses Prozesses besteht ein Mangel an Transparenz und der Austausch kann zum Stillstand kommen, das Konto könnte eingefroren werden, die Handelsplattform könnte pleitegehen usw. Ebenfalls ist es möglich, dass die Vermögen der Anleger in der Zeit der Hinterlegung für andere Zwecke verwendet werden, wie z.B. der Ausleihe an Dritte. Ein Mangel an Transparenz kann Nutzern zusätzliche Kosten verursachen, ohne dass ein Totalverlust der Anlage eintritt, wie etwa höhere Handelsgebühren, Verzögerungen bei hoher Nachfrage, regulatorische Risiken und die Ausführung bevorzugter Aufträge.

Der Mangel an Liquidität. Aus Sicht der Betreiber einer Handelsplattform verhindert eine fragmentierte Liquidität den Einstieg neuer Börsen aufgrund von zwei "Der Gewinner bekommt alles"-Szenarien. Zum einen gewinnt die Handelsplattform mit der größten Anzahl von Handelspaaren, weil es für die Nutzer vorteilhaft ist, alle ihre Geschäfte an einer Börse abzuwickeln. Zum anderen gewinnt die Handelsplattform mit dem größten Auftragsbuch aufgrund günstiger Ask-Bid-Spannen für jedes Handelspaar. Dies erschwert den Wettbewerb durch Neueinsteiger, da es für sie schwierig ist, eine initiale Liquidität aufzubauen. Infolgedessen haben einige Börsen einen hohen Marktanteil trotz vielfacher Nutzerbeschwerden und teilweise sogar größeren Hackerangriffen. Es ist erwähnenswert, dass schnell wachsende, zentralisierte Börsen zu einem immer größeren Angriffsziel für Hacker werden.

Aus Sicht der Nutzer reduziert eine fragmentierte Liquidität die Benutzerfreundlichkeit erheblich. In einer zentralisierten Börse können Nutzer nur innerhalb der eigenen Liquiditätsspanne und gebunden ans Auftragsbuch zwischen unterstützten Token-Paaren handeln. Um Token A für Token B zu tauschen, müssen sich Nutzer an eine Handelsplattform wenden, welche beide Token unterstützt oder sich an verschiedenen Handelsplattformen registrieren, wobei sie immer persönliche Daten offenlegen. Nutzer müssen häufig vorbereitende Transaktionen durchführen, in der Regel gegen BTC oder ETH, wobei Ask-Bid-Spannen anfallen. Schlussendlich sind die Auftragsbücher möglicherweise nicht groß genug, um den Handel ohne signifikante Preisabweichung abschließen zu können. Selbst wenn die Handelsplattform vorgibt ein großes Volumen zu verarbeiten, gibt es keine Garantie, dass Volumen und Liquidität nicht gefälscht sind [12].

Das Ergebnis sind getrennte Liquiditätsmengen und ein fragmentiertes Ökosystem, das dem alten Finanzsystem ähnelt, mit einem bedeutenden und zugleich auf wenige zentrale Handelsplattformen konzentrierten Handelsvolumen. Die globalen Liquiditätsversprechungen der Blockchains bewähren sich nicht im zentralisierten Austausch.

2.2 Unzulänglichkeiten von dezentralisierten Handelsplattformen

Dezentrale Handelsplattformen unterscheiden sich von zentralen Tauschbörsen zum Teil dadurch, dass Nutzer die Kontrolle über die privaten Schlüssel ihrer Vermögenswerte behalten, indem sie Transaktionen direkt mit der entsprechenden Blockchain abwickeln. Durch die Nutzung der zugrundeliegenden Trustless-Technologie von Kryptowährungen können viele der oben genannten Sicherheitsrisiken erfolgreich gemindert werden. Probleme bestehen jedoch hinsichtlich Leistung und struktureller Einschränkungen.

Die Liquidität bleibt häufig ein Problem, da Nutzer nach Tauschpartnern bei unterschiedlichen Verfügbarkeiten und Standards suchen müssen. Eine fragmentierte Liquidität tritt dann auf, wenn dezentrale Handelsplattformen oder dApps keine weitläufigen Standards für eine Zusammenarbeit durchsetzen und Aufträge nicht über ein umfangreiches Netzwerk verteilt werden. Die Liquidität von Limit-Auftragsbüchern und insbesondere deren Flexibilität – wie schnell ausgeführte Limit-Aufträge aktualisiert werden – können Handelsstrategien erheblich beeinflussen [13]. Das Fehlen solcher Standards hat nicht nur zu einer verringerten Liquidität geführt, sondern auch zu einer Reihe von eigenentwickelten, potenziell unsicheren Smart Contracts.

Da Transaktionen auf der Blockchain ausgeführt

werden, leiden dezentrale Handelsplattformen an den Einschränkungen der zugrundeliegenden Blockchains: Skalierbarkeit, Verzögerungen bei der Ausführung (Mining) und kostspielige Änderungen von Aufträgen. Blockchain-Auftragsbücher skalieren nur unzureichend, da das Ausführen von Code auf der Blockchain Kosten (Gas) verursacht, wodurch mehrere Auftragsabbrüche unverhältnismäßig teuer werden.

Da die Auftragsbücher der Blockchain öffentlich sind, ist die Transaktion eines neu erstellten Auftrags für die Miner sichtbar, während sie in den nächsten Block geschrieben wird, um im Auftragsbuch zu erscheinen. Diese Verzögerung setzt den Nutzer dem Risiko aus, von anderen überholt und zu seinen Ungunsten unter- bzw. überboten zu werden.

2.3 Hybride Lösungen

Aus den obigen Gründen haben rein Blockchain-basierte Handelsplattformen Beschränkungen, die sie gegenüber dem zentralisierten Handel nicht wettbewerbsfähig machen. Es besteht jedoch eine Lösung, um Transaktionen auf der Blockchain, die Geschwindigkeit von zentralen Handelsplattformen und die Tauschflexibilität zu vereinen. Protokolle wie Loopring und 0x [14] erweitern eine Lösung der On-Chain-Abwicklung mit Off-Chain-Bestellmanagement. Dieses Konzept agiert mit offenen Smart Contracts, vermeidet Skalierbarkeitsbeschränkungen, indem es mehrere Funktionen unabhängig der Blockchain ausführt und den Schnittstellen ausreichende Flexibilität in der Erfüllung von Netzwerkentscheidungen einräumt. Nachteile bleiben jedoch auch im Hybridmodell bestehen [15]. Das Loopring-Protokoll stellt mit dieser Abhandlung sinnvolle Lösungsansätze für eine hybride Technik vor.

3 Loopring-Protokoll

Loopring ist keine dezentrale Handelsplattform, sondern ein modulares Protokoll zum Aufbau von DEXs auf mehreren Blockchains. Loopring teilt die Bausteine einer traditionellen Handelsplattform auf und setzt stattdessen auf eine Kombination aus offenen Smart Contracts und dezentralen Akteuren. Zu den Bestandteilen im Netzwerk zählen Wallets, Netzwerkknoten, Liquidität-teilende Blockchains, Browser für Auftragsbücher, Ring-Miner und Dienste zur Tokenisierung von Vermögenswerten. Bevor darauf näher eingegangen wird, sollte man zuerst die Loopring-Aufträge verstehen.

3.1 Bestellkreislauf

Loopring-Aufträge werden in einem multidimensionalen Bestellmodell (Unidirectional Order Model) [16] verarbeitet. Das UDOM stellt Aufträge zum Tokentausch aus, `amountS`/`amountB`, (Menge zu verkaufen/kaufen) anstelle von Ask-Bid-Preisen. Da jede Bestellung nur eine Austauschrate zwischen zwei Token ist, ist eine tragende Rolle des Protokolls das Mischen und Abgleichen mehrerer Bestellungen im Handelskreislauf. Durch den Einsatz von bis zu 16 Aufträgen anstelle eines einzelnen Handelspaars steigt die Liquidität rapide an und bietet Potenzial zur Preisverbesserung.

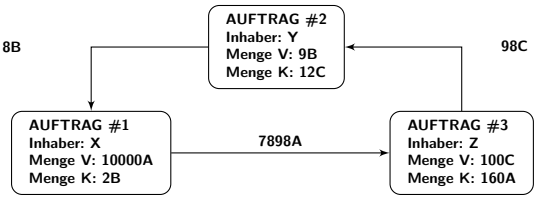

Abbildung 1: Auftragsring von drei Bestellungen

Die obige Abbildung zeigt einen Handelskreislauf mit drei Aufträgen. Jeder Token, der zum Verkauf steht (`tokenS`), wird in einem anderen Auftrag zum Kauf (`tokenB`) angefragt. Es entsteht ein Ring, der es jeder Bestellung erlaubt, ihre gewünschten Token zu tauschen ohne auf ein weiteres Handelspaar angewiesen zu sein. Herkömmliche Abwicklungen von Auftragspaaren können dennoch ausgeführt werden, was jedoch einen Sonderfall im Bestellkreislauf darstellt.

Definition 3.1 (Bestellkreislauf) *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 Bestellkreislauf formen:*

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

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

Ein Auftragsring ist gültig, wenn alle Token-Transaktionen zu einem Wechselkurs ausgeführt werden können, der gleich oder besser ist als der ursprünglich vom Nutzer festgelegte Kurs. Um die Gültigkeit des Handels zu überprüfen, beziehen Smart Contracts des Loopring-Protokolls den Bestellkreislauf von sogenannten Ring-Minern, wobei das Produkt der ursprünglichen Wechselkurse aller Bestellungen gleich oder größer als 1 sein muss.

Nehmen wir an, Alice und Bob wollen ihren Token `A` und `B` tauschen. Alice hat 15 Token `A` und möchte 4 Token `B` erhalten. Bob hat 10 Token `B` und möchte 30 Token `A`.

Wer kauft und wer verkauft? Dies hängt allein von dem Token ab, welcher als Referenz herangezogen 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, dass Token `B` die Referenz ist, verkauft Alice 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`. Von daher können Käufer und Verkäufer willkürlich gewählt werden.

Im ersten Fall ist Alice bereit einen höheren Preis (3.75`A`) zu bezahlen als Bob für seine Token verlangen würde (3.00`A`), wobei im zweiten Fall Bob bereit ist einen höheren Preis (0.33333334`B`) für Alice's Token zu geben (0.26666667`B`). Es ist offensichtlich, dass ein Handel möglich ist, wann immer

der Käufer bereit ist, einen gleichen oder höheren Preis als den 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 wissen, ob das Produkt eines jeden Wechselkurses als Kaufauftrag zu einer Zahl größer oder gleich 1 führt. Wenn ja, können alle n Aufträge entweder teilweise oder vollständig abgewickelt werden.

Wir ziehen eine dritte Partei – Charlie – hinzu, so dass Alice x_1 Token `A` verkaufen und dafür y_1 Token `B` erhalten möchte, Bob will x_2 Token `B` geben und dafür y_2 Token `C` erhalten 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:

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

Siehe Punkt 7.1 für detaillierte Informationen über Loopring-Aufträge.

4 Bestandteile des Ökosystems

Die folgenden Parteien stellen gemeinsam alle Funktionalitäten, die auch ein zentralisierter Austausch bietet.

- **Wallets**: Ein gewöhnlicher Kontoservice bzw. Bedienoberfläche, der Nutzern den 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 8). Mit der Absicht, dass die Zukunft des Handels durch die Sicherheit von individuellen Wallets erfolgt, hat die Verbindung dieser Liquiditätsmengen durch unser Protokoll oberste Priorität.

- **Konsortium von Liquidität fördernden Blockchains und Netzwerkknoten**: Ein Netzwerk zur Auftrags- und Liquiditätssteigerung. Wenn einzelne Netzwerkknoten die Loopring-Vermittlungssoftware anwenden, können sie einem bereits existierenden Netzwerk beitreten und die Liquidität auf Grundlage der vereinigten Blockchain-Interessengemeinschaft vorantreiben. Die erste vereinigte Blockchain, die wir erschaffen, hat nahezu eine Auftragsteilung in Echtzeit (1-2 Sekunden-Blöcke) und setzt neue Maßstäbe, um ein schnelleres Herunterladen durch neue Netzwerkknoten zu ermöglichen.

- **Netzwerkknoten/Ring-Miner**: Netzwerkknoten sind Schnittstellen, die Aufträge von Wallets oder anderen Knoten erhalten, öffentliche Auftragsbücher und Auftragshistorien verwalten und optional Aufträge zu anderen Schnittstellen übertragen (via beliebigem Off-Chain-Medium). Ring-Mining ist eine Option für Schnittstellen, keine Anforderung. Es ist rechenintensiv und erfolgt vollständig off-chain. Netzwerkknoten, welche diese Funktion aktiviert haben, werden "Ring-Miner" genannt und Bestellkreisläufe ermöglichen, indem sie unterschiedliche Aufträge zusammenführen. Netzwerkknoten können frei wählen (1) wie sie sich untereinander verständigen, (2) wie sie ihre Auftragsbücher aufbauen und (3) wie sie Auftragsringe abbauen (Mining-Algorithmen).

- **Loopring Protocol Smart Contracts (LPSC)**: Ein Zusammenschluss öffentlicher Smart Contracts, der Auftragsringe von Ring-Minern überprüft, Token im Namen der Nutzer transferiert, den Ring-Minern und Wallets Anreize durch Gebühren bietet und Ereignisse kommuniziert. Schnittstellen bzw. Auftragsbuch-Browser orientieren sich an den Informationen, um ihre Auftragsbücher und Handelshistorie aktuell zu halten. Siehe Anhang **??** für Details.

- **Asset Tokenization Services (ATS)**: Angebote zur Tokenisierung, die den Tausch von nicht über Loopring handelbaren Gütern ermöglichen. Diese bestehen aus zentralen Diensten, welche von vertrauenswürdigen Unternehmen oder Organisationen bereitgestellt werden. Nutzer hinterlegen Werte (physisch, Fiat, Token anderer Blockchains) und bekommen Token ausgestellt, die zukünftig wieder eingetauscht werden können. Loopring ist kein Blockchain-übergreifendes Handelsprotokoll (bis eine geeignete Lösung existiert), jedoch ermöglicht ATS sowohl den Handel von ERC20-Token [17] mit physikalischen Gütern als auch mit Vermögenswerten anderer Blockchains.

5 Handelsprozess

1. **Protokollautorisierung**: In Abbildung 2 autorisiert Inhaber `Y`, der Token handeln möchte, den LPSC `amountS` seiner Token `B` zu verwalten um sie zu verkaufen. Der LPSC sperrt nicht die Token des Inhabers und es steht dem Inhaber frei die Token zu nutzen, solange der Auftrag bearbeitet wird.

2. **Auftragserstellung**: Der aktuelle Preis und das Auftragsbuch für den Token `B` gegen Token `C` wird von Netzwerkknoten oder anderen Agenten wie Auftragsbuch-Browsern zur Verfügung gestellt, die mit dem Netzwerk verbunden sind. Inhaber `Y` erstellt einen Auftrag (Limit-Auftrag), indem er `amountS` und `amountB` sowie andere Parameter mithilfe einer eingebauten Wallet-Oberfläche festlegt. Der LRx-Betrag kann dem Auftrag als Gebühr für die 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 des Inhabers `Y` signiert.

3. **Auftragsverteilung**: Das Wallet überträgt den Auftrag und seine Signatur an eine oder mehrere Netzwerkknoten. Diese aktualisieren ihr Auftragsbuch. Das Protokoll schreibt nicht vor, wie diese Auftragsbücher auszusehen haben z.B. dass der erste Auftrag bevorzugt abgewickelt werden muss. Stattdessen sind Knoten frei in der Gestaltung ihres Auftragsbuches.

4. **Liquidität teilen**: Netzwerkknoten verteilen den Auftrag mittels beliebigem Medium an andere Schnittstellen. Auch hier sind Netzwerkknoten flexibel, ob und wie sie interagieren. Um ein gewisses Maß an Netzwerkkonnektivität zu gewährleisten, gibt es eine integrierte Liquiditätsteilung unter den Schnittstellen auf Grundlage einer vereinigenden Blockchain. Wie im vorherigen Abschnitt erwähnt, ist dieses Netzwerk auf Geschwindigkeit und Inklusivität optimiert.

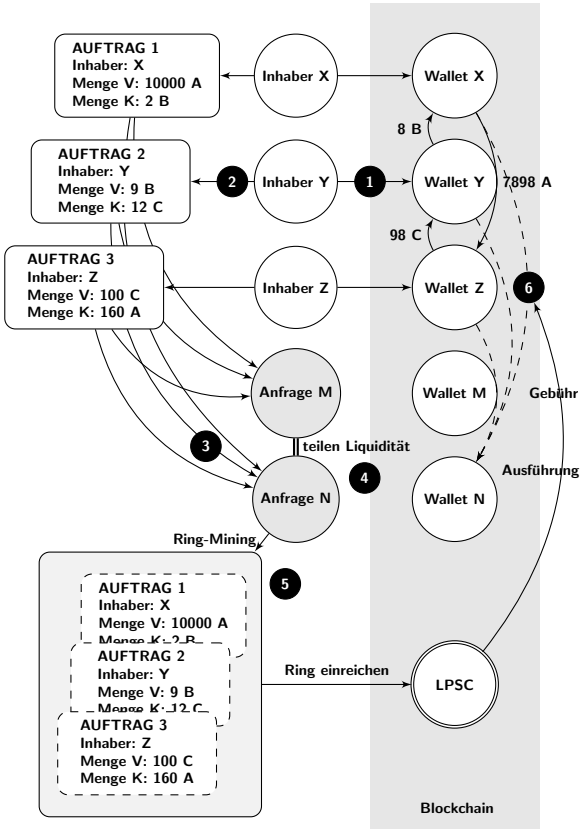

Abbildung 2: Handelsprozess von Loopring

5. **Ring-Mining (Auftragsabgleich)**: Ring-Miner versuchen den Auftrag vollständig oder teilweise zu dem geforderten Preis oder besser zu erfüllen, indem sie ihn mit mehreren anderen Aufträgen abgleichen. Ring-Mining ist der Hauptfaktor, durch den das Protokoll eine hohe Liquidität zu jedem Paar bereitstellen kann. Wenn der ausgeführte Kurs besser ist als der von Nutzer Y angegebene Wert, wird die Marge unter allen Aufträgen im Bestellkreislauf aufgeteilt. Als Belohnung kann der Ring-Miner zwischen einer einfachen LRx-Gebühr oder der Beteiligung an der Marge (Margin Split) wählen, wobei er in diesem Fall die LRx-Gebühr zurückzahlen muss.

6. **Überprüfung & Abrechnung**: Der LPSC erhält den Auftragsring. Er führt mehrere Überprüfungen durch, um die vom Ring-Miner gelieferten Daten zu verifizieren und stellt fest, ob der Bestellkreislauf ganz oder teilweise erfüllt werden kann (abhängig von der Füllrate der Ring-Aufträge und den Token im Nutzerkonto). Wenn alle Überprüfungen erfolgreich sind, überträgt der Smart Contract die Token automatisch an die Nutzer und zahlt gleichzeitig die Ring-Miner und Wallet-Gebühren. Wenn das Kontoguthaben des Inhabers `Y` vom LPSC als nicht ausreicht erkannt wird, wird dieser Auftrag verkleinert: Ein reduzierter Auftrag wird automatisch auf die ursprüngliche Größe skaliert, sobald ausreichende Mittel dem Konto hinzugefügt werden, anders als ein Abbruch, der nur manuell in eine Richtung verläuft und nicht rückgängig gemacht werden kann.

6 Operative Flexibilität

Es ist wichtig zu beachten, dass der freie Loopring-Standard den Teilnehmern eine große Flexibilität in der Ausführung 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 Maße im Prozess verdienen (wenn sie dies wünschen). Das Ökosystem ist modular aufgebaut und soll die Teilnahme einer Vielzahl von Anwendungen unterstützen.

6.1 Auftragsbuch

Netzwerkknoten können ihre Auftragsbücher auf verschiedene Arten gestalten, um die Bestellungen der Nutzer anzuzeigen und abzugleichen. Eine erste Implementierung unseres eigenen Auftragsbuches folgt einem "Over the counter"-Modell (OTC), bei dem Limit-Aufträge nur auf Basis des Preises positioniert werden. Zeitstempel von Aufträgen haben demnach keine Auswirkungen auf das Auftragsbuch, jedoch steht es den Netzwerkknoten frei, ihr Auftragsbuch so zu gestalten, dass Zeitstempel wie ein typisches Auftragsbuch einer zentralisierten Handelsplattform ebenso berücksichtigt werden. Wenn ein Knoten diese Art von Auftragsbuch anbieten möchte, kann er ein Wallet vernetzen und dessen Aufträge alleinig über seine Schnittstelle abwickeln, um sie zeitabhängig zu verteilen. Derartige Konfigurationen sind möglich.

Während andere DEX-Protokolle Anforderungen an ein Tokenguthaben seitens der Netzwerkknoten stellen, damit diese Taker-Aufträge veranlassen können, müssen Loopring-Schnittstellen nur passende Aufträge finden um einen Handel einzuleiten. Dieser Schritt bedarf keiner eigenen Token.

6.2 Liquiditätsteilung

Netzwerkknoten können frei entscheiden, wie sie Liquidität (Aufträge) miteinander teilen. Unsere Konsortium-Blockchain ist nur eine Lösung um dies zu erreichen. Dem Ökosystem steht es frei, sich nach eigenem Belieben zu vernetzen und zu kommunizieren. Knoten können sich nicht nur der Blockchain eines Konsortiums anschließen, sondern auch ihre eigene Blockchain mit Regeln und Besonderheiten entwerfen. Sie können ebenfalls alleine agieren, wie mit der zeitbasierten Wallet-Implementierung beschrieben. Wenngleich unterschiedliche Geschäftsmodelle besondere Verteilstrategien berücksichtigen und die Vergabe von Gebühren beliebig gestalten können, existieren offensichtliche Vorteile in der Kommunikation mit anderen Schnittstellen hinsichtlich Netzwerkeffekte.

7 Spezifikation des Protokolls

7.1 Anatomie eines Auftrags

Ein Auftrag ist ein Datenpaket, das die Absicht des Nutzers zum Handel ausdrückt. Ein Loopring-Auftrag wird mithilfe des multidimensionalen Bestellmodells (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;
  uint256 nonce;
}
```

Um den Ursprung des Auftrags sicherzustellen, wird er gegen den Hash seiner Parameter mittels privatem Schlüssel des Nutzers signiert (nicht `authAddr`). Der `authAddr` Parameter wird für das Signieren des zugehörigen Auftragsrings verwendet, um ein Front-Running zu verhindern. Weitere Informationen finden Sie in Abschnitt 9.1. Die Signatur wird durch die Felder `v`, `r`, und `s` repräsentiert und 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 Guthaben seiner Adresse zusammen mit anderen Variablen berechnen.

UDOM enthält keinen Preis (der naturgemäß eine Gleitkommazahl sein muss), sondern verwendet stattdessen den Begriff `rate` oder r, welcher als `amountS`/`amountB` (Menge V/K) ausgedrückt wird. `rate` ist keine Gleitkommazahl, sondern eine Form, die nur mit anderen unsignierten ganzen Zahlen bei Bedarf ausgewertet wird, um alle zwischenzeitlichen Ergebnisse als ganze Zahlen während der Kalkulation beizubehalten.

7.1.1 Kaufbeträge

Wenn ein Ring-Miner Aufträge abgleicht, besteht die Möglichkeit, dass ein besserer Wechselkurs erzielbar ist. Dies erlaubt es dem Nutzer mehr `tokenB` zu erhalten als die Summe, die mittels `amountB` spezifiziert wurde. Sollte jedoch `buyNoMoreThanAmountB` auf `True` gesetzt sein, stellt das Protokoll sicher, dass der Nutzer 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 Nutzer, detailliertere Handelsabsichten auszudrücken als traditionelle Kauf- und Verkaufsaufträge.

Beispiel: mit `amountS` $= 10$ und `amountB` $= 2$, wäre der Kurs $r = 10/2 = 5$. Also möchte der Nutzer 5 `tokenS` für jeden `tokenB`. Der Ring-Miner vergleicht und findet einen Auftrag mit dem Kurs 4, welcher es dem Nutzer erlaubt, 2.5 `tokenB` anstatt 2 zu erhalten. Möchte der Nutzer jedoch nur 2 `tokenB` und hat den Parameter `buyNoMoreThanAmountB` auf `True` gesetzt, wird die Transaktion mit dem Kurs 4 ausgeführt, wodurch er 4 `tokenS` für jeden `tokenB` erhält und folgerichtig 2 `tokenS` spart. Es muss beachtet werden, dass hierbei Mining-Gebühren nicht berücksichtigt werden (siehe Abschnitt 8.1).

Wenn wir Folgendes verwenden

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

um einen Auftrag vereinfacht darzustellen, können populäre Kauf-/Verkauf-Modelle für ETH/USD-Märkte an einer traditionellen Handelsplattform den unten abgebildeten ersten und dritten Auftrag ausdrücken, jedoch 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 für 300 USD/ETH. Dieser Auftrag kann wie folgt dargestellt werden: `Order(3000, USD, 10, ETH, True)`.
4. Gebe 3000 USD aus, um so viel ETH wie möglich zu kaufen zum Preis von 300 USD/ETH. Dieser Auftrag kann wie folgt dargestellt werden: `Order(3000, USD, 10, ETH, False)`.

7.2 Ring-Verifizierung

Loopring Smart Contracts führen keine Wechselkurs- oder Mengenberechnungen durch, sondern empfangen und überprüfen lediglich die Auswertung der Ring-Miner. Diese Berechnungen werden aus zwei Gründen von Ring-Minern durchgeführt: (1) Die Programmiersprache für Smart Contracts wie Solidity [18] auf Ethereum unterstützt keine Gleitkomma-Berechnung, speziell $pow(x, 1/n)$ (Berechnung der n-ten Wurzel einer Gleitkommazahl) und (2) ist es wünschenswert, dass die Berechnung außerhalb der Blockchain erfolgt, um die Rechenleistung und Kosten der Blockchain zu reduzieren.

7.2.1 Subring-Überprüfung

Dieser Schritt verhindert, dass Preisunterschiede innerhalb des Rings ausgenutzt werden können, indem neue Aufträge platziert werden. Sobald ein Ring-Miner einen gültigen Auftragsring gefunden hat, könnte es lukrativ sein, dem Auftragsring weitere Aufträge hinzuzufügen, um die Marge der Nutzer vollständig zu absorbieren. Wie unten in Abbildung 3 dargestellt, ergibt $x1$, $y1$, $x2$ und $y2$ das Produkt aller Kurse genau 1, somit wird es keinen Preisnachlass geben.

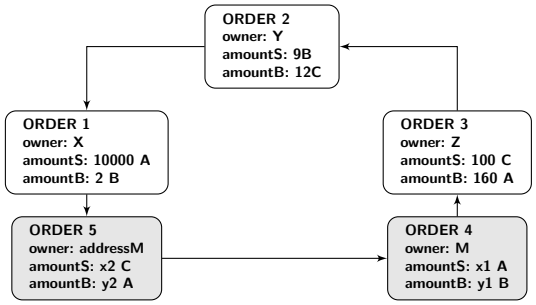

Abbildung 3: Ein Auftragsring mit Subring

Es würde sich um ein risikofreies Geschäft für den Ring-Miner handeln ohne Wertzuwachs für das Netzwerk, was als unfaire Handhabe gewertet wird. Um dies zu verhindern, legt Loopring fest, dass ein gültiger Ring keine Subringe enthalten darf. Zur Überprüfung stellt der LPSC sicher, dass ein Token nicht sowohl in einer Kauf-, als auch in einer Verkaufsposition sein kann. Im obigen Diagramm ist zu sehen, dass Token `A` zweifach als Kauf- und als Verkaufsposten auftritt. Dies wäre nicht zulässig.

7.2.2 Füllratenprüfung

Die Wechselkursberechnungen im Auftragsring werden von Ring-Minern aus den vorherigen Gründen vorgenommen. Es ist der LPSC, der ihre Korrektheit sicherstellen muss. Als erstes muss überprüft werden, ob die Kaufrate, die der Ring-Miner für jede Bestellung ausführen kann, gleich oder kleiner als der ursprüngliche Wechselkurs ist, der vom Nutzer festgelegt wurde. Dies stellt sicher, dass der Nutzer mindestens den Kurs erhält, den er verlangt oder besser. Sobald die Raten bestätigt sind, stellt der LPSC sicher, dass jede Bestellung im Auftragsring den gleichen Preisnachlass erhält. 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:

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

deshalb:

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

Wenn die Transaktion n Aufträge überschreitet, beträgt der `discount`:

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

wobei r^i der Umschlagsfaktor des i-ten Auftrags ist. Nur wenn die Ermäßigung $\gamma \geq 0$ ausfällt, können diese Aufträge ausgeführt werden; wobei vom i-ten Auftrag (O^i) der tatsächliche Kurs $\hat{r^i} = r^i \cdot (1-\gamma)$, $\hat{r^i} \leq r^i$ ist.

7.2.3 Statusnachverfolgung & Abbruch

Ein Nutzer kann einen Auftrag teilweise oder komplett abbrechen, indem er eine spezielle Transaktion an den LPSC sendet, die Informationen über den Auftrag und die zu stornierende Summe enthält. Der LPSC berücksichtigt und speichert den widerrufenen Betrag und sendet ein `OrderCancelled`-Ereignis an das Netzwerk. Der LPSC registriert gefüllte und stornierte Beträge, indem er ihre Werte mit dem Hash der Bestellung als Kennung speichert. Diese Daten sind offen einsehbar und `OrderCancelled`- sowie `OrderFilled`-Ereignisse werden kommuniziert, sobald sie sich verändern. Das Verfolgen dieser Werte ist für den LPSC während der Ausführung des Auftragsrings obligatorisch.

Der LPSC unterstützt auch das Abbrechen sämtlicher Aufträge eines Handelspaares mit dem `OrdersCancelled`-Befehl; alle Aufträge einer Adresse können mit dem Ereignis `AllOrdersCancelled` storniert werden.

7.2.4 Auftragsskalierung

Aufträge werden gemäß der Historie von ausgeführten und stornierten Beträgen sowie dem aktuellen Kontostand des Inhabers 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, hilft dies 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 Guthaben ist, welches übrig bleibt nachdem die Aufträge teilweise erfüllt wurden.

Während der Durchführung können wir sicher annehmen, dass jeder Auftrag im Auftragsring den niedrigsten Wert hat und den Auftragsring höchstens zweimal durchläuft, um das Füllvolumen jeden Auftrags zu berechnen.

Beispiel: Wenn der kleinste zu füllende Betrag im Vergleich zur ursprünglichen Bestellung 5% ist, werden alle Transaktionen im Auftragsring auf 5% herunterskaliert. Sobald die Transaktionen abgeschlossen sind, sollte der Auftrag, der die geringste zu füllende Menge aufwies, vollständig abgeschlossen sein.

7.3 Ausführung des Auftragsrings

Wenn der Auftragsring alle vorangegangenen Prüfungen erfüllt hat, kann er geschlossen und die Transaktionen durchgeführt werden. Dies bedeutet, dass alle n Aufträge einen geschlossenen Auftragsring bilden und verbunden sind, veranschaulicht in Abbildung 4:

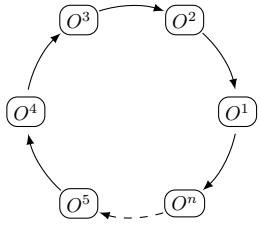

Abbildung 4: Durchführung des Auftragsrings

Um die Transaktionen durchzuführen, verwendet der LPSC den `TokenTransferDelegate` Smart Contract. Die Einführung eines solchen Delegierten erleichtert die Aktualisierung des Smart Contracts, da alle Aufträge nur diesen Delegierten anstelle verschiedener Versionen des Protokolls autorisieren müssen.

Für jeden Auftrag im Auftragsring wird eine Zahlung von `tokenS` an den nächsten oder vorhergehenden Auftrag veranlasst, je nach Durchführung. Anschließend wird die Ring-Miner-Gebühr gemäß dem vom Ring-Miner festgelegten Gebührenmodell bezahlt. Wurden alle Transaktionen ausgeführt, wird ein `RingMined`-Ereignis kommuniziert.

7.3.1 Ereignisse

Das Protokoll kommuniziert Ereignisse, um Aktualisierungen bezüglich des Auftragsbuches so effizient wie möglich unter den Netzwerkknoten, Auftragsbuch-Browsern und anderen Parteien zu verbreiten. Die gesendeten Ereignisse sind:

- **OrderCancelled**: Ein spezieller Auftrag wurde abgebrochen.
- **OrdersCancelled**: Alle Aufträge eines Handelspaares einer bestimmten Adresse wurden abgebrochen.
- **AllOrdersCancelled**: Alle Aufträge aller Handelspaare einer bestimmten Adresse wurden abgebrochen.
- **RingMined**: Ein Auftragsring wurde erfolgreich abgearbeitet. Dieses Ereignis enthält Daten jeder einzelnen Transaktion des Auftragsrings.

8 LRx-Token

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

8.1 Gebührenmodell

Wenn ein Nutzer einen Auftrag erstellt, wird ein Betrag angegeben, der als mögliche Gebühr an den Ring-Miner gezahlt wird in Verbindung mit einem Prozentsatz der Marge (`marginSplitPercentage`) des Auftrags, die der Ring-Miner erzielen kann. Dies wird als Margenaufteilung (Margin Split) bezeichnet. Die Wahl zwischen einer Gebühr und der Margenaufteilung obliegt dem Ring-Miner.

Eine Darstellung der Margenaufteilung:

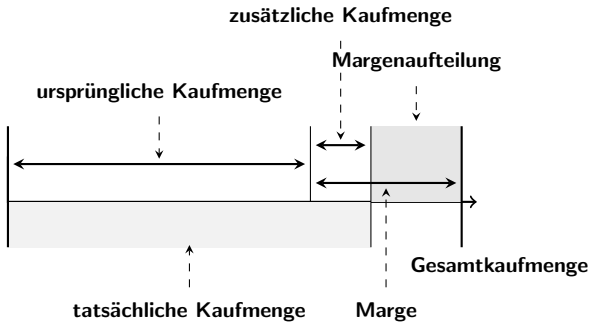

Abbildung 5: Eine 60%-ige Margenaufteilung

Wenn die Marge des Auftragsrings zu klein ist, wählt ein Ring-Miner die LRx-Gebühr. Wenn die Marge im Gegensatz dazu so hoch ist, dass die daraus resultierende Margenaufteilung weit über die LRx-Gebühr hinausgeht, wird der Ring-Miner die Margenaufteilung bevorzugen. Es gilt zu beachten: Wenn der Ring-Miner die Margenaufteilung wählt, muss er dem Nutzer die Summe zahlen, welche der Höhe in LRx

entspricht, die dieser dem Ring-Miner als Gebühr gezahlt hätte. Dies erhöht den Schwellenwert einer rentablen Margenaufteilung auf das Doppelte der LRx-Gebühr, wodurch sich die Nachfrage von LRx erhöht. Es ermöglicht den Ring-Minern ein konstantes Einkommen bei Auftragsringen mit niedriger Marge und reduziert ihre Gewinne bei hohen Margen. Unser Gebührenmodell basiert auf der Erwartung, dass mit zunehmendem und reifendem Markt geringere Margen erzielt werden können und somit fixe LRx-Gebühren als Anreiz nötig sind.

Am Ende ergibt sich folgendes Diagramm:

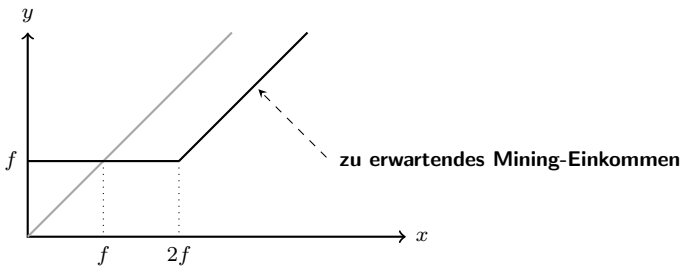

Abbildung 6: Looprings Gebührenmodell

wobei f die LRx-Gebühr ist, x ist die Margenaufteilung, y bezeichnet das Mining-Einkommen. $y = max(f, x - f)$ abgebildet durch die gerade Linie. Wenn die LRx-Gebühr für den Auftrag 0 ist, beträgt die Gleichung $y = max(0, x - 0)$ vereinfacht zu $y = x$ wie von der grauen Linie dargestellt.

Die Konsequenzen sind:

1. Wenn die Margenaufteilung 0 ist, werden Ring-Miner die LRx-Gebühr wählen und sind weiterhin 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 Einkommen der Margenaufteilung größer als 2x die LRx-Gebühr ist, wählen die Ring-Miner die Marge und zahlen LRx an den Nutzer.

Es sollte beachtet werden, dass wenn die LRx-Gebühr ungleich 0 ist, unabhängig von der Gebührenwahl des Ring-Miners, immer eine Übertragung von LRx zwischen dem Ring-Miner und dem Auftraggeber stattfindet. Entweder erhält der Ring-Miner die LRx-Gebühr oder zahlt diese an den Nutzer zurück, um die Margenaufteilung zu wählen.

Ring-Miner teilen einen bestimmten Prozentsatz der Gebühren mit Wallets. Wenn ein Nutzer einen Auftrag über ein Wallet veranlasst und dieser ausgeführt wird, wird das Wallet mit einem Teil der Gebühren oder der Margenaufteilung belohnt. Obwohl dies modular ist und besondere Geschäftsmodelle oder Implementierungen möglich sind, möchten wir Wallets mit etwa 20% - 25% der verdienten Gebühren entlohnen. Sie stellen die Nutzerbasis dar und bilden somit das Fundament einer erfolgreichen Integration des Loopring-Protokolls, wenngleich sie eine geringe oder keine Einkommensquelle haben.

8.2 Dezentrale Struktur

Das Loopring-Protokoll ist von Grund auf ein soziales Protokoll, da es auf die Koordination zwischen den Mitgliedern angewiesen ist, um effektiv auf ein Ziel hinzuarbeiten. Dies ist kryptoökonomischen Protokollen insgesamt nicht unähnlich und ihre Notwendigkeit wird weitgehend durch die gleichen Gegebenheiten geschützt von Koordinationsproblemen [19] über das Grim-Trigger-Gleichgewicht bis zur beschränkten Rationalität. Vorausschauend werden LRx-Token nicht nur zur Gebührenzahlung verwendet, sondern auch, um finanzielle Anreize der verschiedenen Netzwerkteilnehmer in Einklang zu bringen. Eine solche Anpassung ist im Allgemeinen für die Verbreitung von Protokollen erforderlich, jedoch besonders wichtig für Handelsprotokolle, da der Erfolg weitgehend auf der Verbesserung der Liquidität in einem soliden, dezentralen Ökosystem beruht.

LRx-Token werden verwendet, um Protokollaktualisierungen in einer dezentralen Struktur zu erhalten. Intelligente Vertragserneuerungen 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 Endnutzer weiterhin mit veralteten Versionen arbeiten 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-Eigner ermöglicht Updates von Smart Contracts auf dem Protokoll ohne dApps oder Endnutzer zu stören oder sich zu sehr an den Gegebenheiten von Smart Contracts orientieren zu müssen. Dies wird in erster Linie durch einen mehrheitlich signierenden Smart Contract erzielt mit Ausblick auf einen künftigen DAO-Typ-Mechanismus.

9 Betrugs- und Angriffsschutz

9.1 Front-Running-Prävention

Auf dezentralen Handelsplattformen bezeichnet man Front-Running als den Versuch, die Handelsstrategie einer Schnittstelle zu kopieren und früher als die ursprüngliche Transaktion zu verarbeiten, welche sich noch in Bearbeitung (Mempool) befindet. Dies kann durch Zahlung einer höheren Transaktionsgebühr (Gas-Preis) erreicht werden. Das Vorgehen bei Front-Running in Loopring (und bei jedem anderen Protokoll zur Auftragszusammenführung) ist die Imitation des Auftrags: Ein Front-Runner greift ein oder mehrere Bestellungen von der Auftragsmenge ab oder – im Fall von Loopring – kopiert einen kompletten Auftragsring von ausstehenden Transaktionen.

Wenn eine “submitRing”-Transaktion noch nicht bestätigt wurde und sich noch in der Warteschleife befindet, kann diese von jedem gefunden und `minerAddress` durch

die eigene Adresse (`filcherAddress`) ersetzt werden. Anschließend könnte die Kopie mit `filcherAddress` erneut signiert werden. Der Imitator kann einen höheren Gas-Preis setzen und eine neue Transaktion erstellen mit der Absicht, dass sein Auftrag früher als die ursprüngliche "submitRing"-Transaktion von den Minern in den nächsten Block geschrieben wird.

Bisherige Lösungen für dieses Problem hatten große Nachteile: sie erforderten mehr Transaktionen und kosteten daher die Ring-Miner mehr Gas. Zudem benötigten sie doppelt so viele Blöcke, um einen Auftragsring abzuschließen. Unsere neue Lösung – die duale Autorisierung (Dual-Authoring) [20] – umfasst den Mechanismus von zwei Autorisierungsstufen für Aufträge: eine zur Abwicklung und eine zum Ring-Mining.

Doppelter Autorisierungsprozess:

1. Für jeden Auftrag generiert die Wallet-Software ein zufälliges öffentlich/privates Schlüsselpaar und fügt dieses dem JSON-Ausschnitt des Auftrags hinzu. (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` um den passenden privaten Schlüssel aus `authAddr` abzuleiten.)

2. Berechnung des Auftrag-Hashs mit allen Feldern außer `r`, `v`, `s` und `authKey`. Signieren des Hashs mit dem privaten Schlüssel des `owner` (nicht `authKey`).

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

4. Wenn ein Auftragsring gefunden wurde, verwenden die Ring-Miner `authKey`, um für jeden Auftrag den Ring-Hash und alle anderen Mining-Parameter mit `minerAddress` zu signieren. Enthält ein Auftragsring n Aufträge, so gibt es auch n Signaturen von n `authKey`s. Diese Signaturen nennen wir `authSignature`. Der Ring-Miner wird zudem den Hash des Rings zusammen mit allen Mining-Parametern unter Verwendung seines privaten Schlüssels `minerAddress` signieren.

5. Der Ring-Miner ruft die "submitRing"-Funktion mit allen Parametern und `authSignature`s ab. `authKey` ist NICHT Teil der Blockchain-Transaktion und bleibt daher für jeden außer dem Ring-Miner unbekannt.

6. Das Loopring-Protokoll verifiziert nun mittels `authSignature` die korrespondierende `authAddr` eines jeden Auftrags und lehnt den Auftragsring ab, sofern `authSignature` einen fehlenden oder ungültigen Schlüssel erkennt.

Das Resultat ist:

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

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

- Die `authSignature` versichert, dass der gesamte Auftragsring nicht verändert oder gestohlen werden kann, inklusive `minerAddress`.

Die duale Autorisierung verhindert die Kopie des Auftragsrings und stellt gleichzeitig sicher, dass die Abwicklung von Auftragsringen in einer einzigen Transaktion durchgeführt werden kann. Darüber hinaus ermöglicht die duale Autorisierung Netzwerkknoten, 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 des Tauschs hat, sondern sich darauf auswirkt, ob er ausgeführt wird oder nicht.

10 Andere Angriffsszenarien

10.1 Sybil- oder DoS-Attacke

Schadhafte Nutzer, die als sie selbst oder mit gefälschter Identität agieren, könnten eine große Anzahl kleiner Aufträge senden, um Loopring-Knoten anzugreifen. Da Schnittstellen jedoch Aufträge ablehnen können, je nach eigenen Kriterien, die sie mitteilen oder zurückhalten dürfen, werden die meisten dieser Aufträge abgelehnt, da sie im Vergleich keinen zufriedenstellenden Gewinn abwerfen. Da Netzwerkknoten selbst entscheiden können, wie sie Aufträge verwalten, sehen wir den Angriff durch viele Kleinaufträge nicht als Bedrohung.

10.2 Unzureichendes Guthaben

Schadhafte Nutzer könnten Aufträge signieren und verteilen, deren Werte nicht null ist, wohl aber der Kontostand ihrer Adresse. Netzwerkknoten könnten bemerken, dass das tatsächliche Guthaben einiger Aufträge null ist, deren Auftragsstadien aktualisieren und entsprechend verwerfen. Schnittstellen müssen Zeit investieren, um den Status eines Auftrags zu erneuern, können aber den Aufwand minimieren, indem sie beispielsweise Adressen blockieren und zugehörige Bestellungen automatisch abweisen.

11 Zusammenfassung

Das Loopring-Protokoll ist eine wichtige Ebene des dezentralen Handels. Dadurch hat es tiefgreifende Auswirkungen auf den Tausch von Finanzmitteln und Werten. Geld, als eine Zwischenware, erleichtert oder ersetzt den Tauschhandel und löst das Aufeinandertreffen eines gegenteiligen Wunsches [21], wobei zwei Akteure das Gut oder den Dienst des jeweils anderen begehren. In ähnlicher Weise zielt das Loopring-Protokoll darauf ab, unsere Abhängigkeiten von zufälliger Nachfrage von Handelspaaren aufzuheben, indem ein Ringabgleich einen Tausch vereinfacht. Es ist bedeutsam, in welcher Weise die Gesellschaft und Märkte ihre Token, traditionellen Vermögenswerte etc. tauschen. So wie dezentrale Kryptowährungen eine Bedrohung für die Kontrolle eines Landes über ihre Geldmittel darstellen, so ist ein kombinatorisches Protokoll, das Händler (Konsumenten/Produzenten) in großen Mengen zusammenbringt, eine theoretische Bedrohung für das Konzept von Geld selbst.

Die Vorteile des Protokolls umfassen:

- Off-Chain-Auftragsverwaltung und On-Chain-Abwicklung bedeutet keine Leistungseinbußen zugunsten von Sicherheit.
- Höhere Liquidität durch Ring-Mining und Auftragsteilung.
- Duale Autorisierung löst das schädliche Problem des Front Runnings, welches heutzutage von allen DEXs und ihren Nutzern bewältigt werden muss.
- Mit kostenlosen, öffentlichen Smart Contracts kann sich jede dApp mit dem Protokoll verbinden und mit diesem interagieren.
- Standardisierung unter den Akteuren ermöglicht Netzwerkeffekte und eine verbesserte Endnutzererfahrung.
- Das Netzwerk wird mit Flexibilität in der Ausführung der Auftragsbücher und der Kommunikation aufrechterhalten.
- Geringere Eintrittsbarrieren bedeuten niedrigere Kosten für Schnittstellen und Endnutzer, die dem Netzwerk beitreten.
- Anonymer Handel ausgehend vom privaten Wallet.

12 Danksagungen

Wir möchten unseren Mentoren, Beratern und den vielen Menschen in der Gemeinschaft unseren Dank aussprechen, die so herzlich und großzügig mit ihrem Wissen waren. Insbesondere möchten wir danken: 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, Encephalo Path, Milan Hoppe und Mathias Enzensberger für die Überprüfung und Übersetzung des Projekts sowie das Feedback.

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

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

[19] Vitalik Buterin. Notes on blockchain governance. https://vitalik.ca/general/2017/12/17/voting.html, Accessed: 2018-03-05.

[20] 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.

[21] 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.