TechnischesWhite Paper

(Einige Detail können hinzugefügt werden)

Dezentrales Finanzsystem

CREDITS

Version 1.6/26 nov 2017

Inhaltsverzeichnis

Abstract 2
Einleitung 3

1. Netzwerk Ledger 4

Definitionen 4
Netzwerkknoten 4
Der letzte gesicherte Block 4
Synchronisation der Netzwerkknoten 5

2. Netzwerk-Konsens 5

Konsensvergleich 5
Das Konzept des Hauptnetzknotens 6
Ausstattung von Netzwerkknoten 7
Konsensbildung 7
Aufbau und Initiierung des Ledgers 8
Nicht im Register erfasste Transaktionen 8

3. Transaktionsabwicklung 8

Transaktionen 8
Konsensbildung 8
Transaktionsabwicklung 9
Ledger-Erfassungsstruktur 9
Struktur des CREDITS-Ledgers 9
Blockgröße 9
Suche nach Transaktionsteilnehmern 10
Datenübertragungskanal 10
Aktionen im System 11
Hinzufügen einer Transaktion zur Validierung 11
Transaktionskosten 11

4. Smart Contracts 12

Einleitung 12
Entitäten 12
Smart Contract Methode 12
Virtuell ausführbare Maschine 13
Value Term 13
Ausführung der Smart Contract Bedingungen 13
Datenquellen 13

5. Implementations-Plan 15

Technischer Plan der Projekt-Implementation 15
CREDITS Kryptowährung 16

Abstract

Die CREDITS-Plattform ist ein dezentrales Finanzsystem für die unmittelbare Interaktion zwischen den Teilnehmern nach dem Prinzip des Peer-to-Peer (P2P). Die Plattform erweitert das Potenzial der Nutzung von Finanzdienstleistungen auf der Basis eines verteilten Ledgers, selbstausführender Smart Contracts und CREDITS Krypto-Währung. Ziel des Systems ist es, alle Beteiligten auf einer Seite zu vereinen und ihnen eine Plattform für die Bereitstellung und Nutzung von Finanzdienstleistungen zu bieten, auf der jeder einen Service anbieten und nutzen kann. Dank eines klar definierten und ausgewogenen technologischen Systems bietet die CREDITS-Plattform eine neue technische Lösung und ein neues konzeptionelles Modell für die Vernetzung der Interaktion der Teilnehmer für die Entwicklung moderner dezentraler Finanzdienstleistungen.

Einleitung

Eine vollständige Peer-to-Peer-Vereinbarung für Dienstleistungserbringungssysteme, die die Bildung von Finanzdienstleistungen ermöglicht: Geldtransfers, Devisen- und Werteaustausch, Kreditvergabe, Finanzierung und andere Dienstleistungen direkt zwischen den Teilnehmern. Alles wird ohne zusätzliche Vermittler bereitgestellt, von einem der gleichberechtigten Teilnehmer - an einen anderen Systemteilnehmer. Das Ergebnis ist, dass jeder billiger, schneller und besser bedient wird.

Die Welt bewegt sich in Richtung der gleichberechtigten, direkten Interaktion zwischen Menschen nach dem Peer-to-Peer-Prinzip. Eine Revolution ist im Gange! Deutlich wird dies an der Umwälzung der Massenmedien: Bis in die 1990er Jahre waren Zeitungen, Zeitschriften und TV die wichtigsten Informationsanbieter. Heute sind sind Blogger, die auf Youtube-Kanälen und sozialen Netzwerken zu finden sind Meinungsbildner, Geld wird in Crowdfunding und ICO investiert, und die Informationen werden in dezentralen Cloud-Systemen gespeichert.

Die Finanzindustrie ist vielleicht eine der wenigen Branchen, die hinterherhinkt, die sich der Einführung von Dezentralisierung und direkter Interaktion zwischen den Teilnehmern widersetzt. Obwohl es technisch gesehen sehr viel leichter ist dezentrale Finanzdienstleistungen zu schaffen als beispielsweise unbemannte Fahrzeuge zu bauen. Ein entsprechendes technologisches Umfeld ist erforderlich, um ein System von dezentralen Finanzprodukte und -dienstleistungen auf Basis des distribute Ledgers zu schaffen:

1. Hohe Ausführungsgeschwindigkeit (in Sekunden), zusammen mit der Fähigkeit, eine große Anzahl von Transaktionen gleichzeitig (Hunderttausende pro Sekunde) zu einem niedrigen Preis für jede Transaktion (für Mikrozahlungen und bargeldlose Transaktionen) abzuwickeln.

2. Entwicklung eines Systems, in dem alle Teilnehmer und Elemente, die für die qualitativen dezentralen Finanzdienstleistungen notwendig sind, kombiniert werden: Personalisierung der Benutzer, KYC, Bonitätsauskunftbüros, Abwicklungszentren für Papiergeld, Abhebung und Einzahlungen von Kryptowährungen usw. Dies sind zwei große und grundlegende Aufgaben, die derzeit die Entwicklung von Peer-to-Peer-Finanzprodukten behindern.

Wir präsentieren Ihnen eine Lösung für diese Aufgaben, die wir mit Hilfe des Finanzsystems CREDITS umgesetzt haben.

CREDITS technologisch dezentrale Plattform kann alle Teilnehmer von Finanzdienstleistungen kombinieren und schnell und sicher alle Transaktionen nach den Prinzipien eines verteilten Ledgers ausführen. Selbstausführende Smart Contracts und die Prinzipien eines föderativen Abstimmungssystems bieten unbegrenzte Möglichkeiten für alle Beteiligten, einzigartige Interaktionen verschiedener Finanzprodukte zu schaffen. Die Plattform eröffnet einen neuen riesigen Markt und ein neues Potenzial für die Nutzung von Blockchain-Projekten und -Dienstleistungen in Finanz- und anderen Sektoren, die bisher aufgrund von Geschwindigkeits- und Transaktionskostenbeschränkungen nicht genutzt werden konnten.

1. Netzwerk Ledger

Definitionen

1. Ein System ist eine Gruppe von dezentralen Netzwerkknoten, die die Verarbeitung, das Speichern von Transaktionen, die Ausführung und Bestätigung von Smart-Contracts, die Bearbeitung von Anfragen aus Drittsystemen und die Bereitstellung von Informationsdatenauf Anfrage ausführen.
2. Ein Netzwerkknoten ist ein Rechner, auf dem ein kompletter Netzwerkclient installiert ist, der mit einem gemeinsamen System verbunden ist. Er verifiziert Transaktionen und schreibt sie auf den Ledger.
3. Ein Ledger ist die Liste der vom System bestätigten Vorgänge, die auf allen Netzknoten gespeichert sind.
4. Eine Transaktion ist die Systemposition, die eine Aufforderung zur Durchführung einer Smart-Contract Methode oder jede Aktion im Netzwerk registriert und die Resultate in einem einem Blockchain-System aufzeichnet.
5. Ein Smart Contract ist das System Element, das Computerprotokolle, die die Einhaltung der Interaktionsbedingungen erleichtern, überprüfen oder sicherstellen. Sie verfügen in der Regel über eine Benutzeroberfläche und emulieren oft die Logik der Vertragsbeziehungen. Die zentrale Eigenschaft eines Smart Contracts ist seine Dezentralisierung und seine Unabhängigkeit von einer zentralen Quelle.
6. Eine Smart Contract-Methode ist der Programmcode, der für die Berechnung des Arbeitsergebnisses von den Smart Contract Bedingungen und die für Verbuchung im Ledger übernimmt.
7. Ein Vertragspartner ist der letzte Netzteilnehmer und der Systembenutzer.

Netzwerkknoten

Wir verwenden verschiedene Arten von Netzwerk-knoten, je nach Zweck, um ein dezentrales Netzwerk aufzubauen.

Basierend auf freiem Zugang und Knotenverbindung:

1. Ein häufiger Knoten (ОУ) ist der Knoten, die Transaktion verifiziert auf Gültigkeit, hat aber einen minimaler Vertrauensfaktor. Es ist auch ein Kandidat für die Rolle eines vertrauenswürdigen Knotens und der Knoten der aktuellen Verarbeitung im nächsten Zyklus der Knotenrollenauswahl im Netzwerk.
2. Ein Trusted Node (ДУ) ist der Knoten, der an der Transaktionsverifikation teilnimmt und den maximalen Vertrauensfaktor (1) hat. Dieser Knoten kann nicht während einer mathematisch berechneten Anzahl von Auswahl- und Abstimmungszyklen zwischen den Knoten vertrauenswürdig werden. Die mathematische Berechnung hängt von der Anzahl der Knoten und der Komplexität des Netzwerks ab.
3. Der Hauptknoten (ГУ) des Netzes ist der Knoten, der an der Verifizierung beteiligt ist und für das Hinzufügen von Transaktionen zum Transaktions-Ledger-Block verantwortlich ist. Dieser Knoten kann nicht während einer mathematisch berechneten Anzahl von Abstimmungszyklen - deren mathematische Berechnung von der Anzahl der Knoten und der Netzwerkkomplexität abhängt - vertrauenswürdig werden.

Das System verwendet einen Vertrauensfaktor (Trust-Factor) - einen absoluten numerischen Wert von 0 bis 1, ausgedrückt in mathematischen Begriffen der Anzahl der vertrauenswürdigen Knoten +1 bis zur Gesamtzahl der Knoten im Netzwerk. Die maximale Anzahl der vertrauenswürdigen Knoten darf 50 % aller Netzwerkknoten nicht überschreiten.

Der letzte gesicherte Block

Das Common Ledger of Blocks (CRB) ist der synchronisierte Zustand des gesamten Common Ledger of Blocks in allen Systemknoten.

Mit dem Inhalt des Ledger-Blocks ist eine Informationseinheit gemeint, die einen Hash-Code des vorherigen Blocks und eine Liste von Daten enthält, die sich auf diesen Ledger beziehen, mit der zugehörige Zahl an Blocks. Mit dem Inhalt des Ledger-Blocks ist eine Informationseinheit gemeint, die

einen Hash-Code des vorherigen Blocks und eine Auflistung von Daten zu diesem Ledger mit der zugehörigen Nummer des vorherigen Blocks. Nach Erhalt des Blocks von einem anderen Knoten nimmt er seinen Platz im gemeinsamen Ledger entsprechend der Nummer ein. Das spart Netzwerkbandbreite.

Bei der Synchronisation wird zunächst nur die Blocknummer geprüft. Fehlt der Block bei einem Knoten wird er heruntergeladen und gespeichert.

Dadurch ist jederzeit die aktuellste Kopie des Ledgers im System vorhanden. Wir nennen es der letzte Ledger (LR). Sie wird automatisch von dem Knoten, der für die Ledgerbildung verantwortlich ist, angelegt, bis ein Konsens erzielt wird. Dieser Block wird an alle Systemknoten gesendet, um die aktuelle Einheitlichkeit des Ledgerstatus in allen Systemknoten aufrechtzuerhalten.

Jeder Knoten ist mit allen anderen Knoten im Netzwerk verbunden und tauscht ständig neue Blöcke mit Transaktionen mit ihnen aus, um stets die relevanten Informationen zu pflegen. Alle Blöcke bilden eine Gruppe von Transaktions-Kandidaten, die darauf warten, in das Ledger aufgenommen zu werden. Gleichzeitig generiert jeder Server angenommene Sätze der Kandidaten für andere Server und die vorgeschlagene Menge von Transaktionen. Bei der Prüfung, ob sie in das Ledger aufgenommen werden sollen, wird eine Entscheidung getroffen.

Dadurch ist es möglich, die Ledgerdaten mehrfach auf mehreren Servern - den Systemknoten - zu speichern, und alle Informationen sind geschützt. Je mehr Knoten im System vorhanden sind, desto zuverlässiger und unabhängig ist das System.

Synchronisation der Netzwerkknoten

Jeder neue Knoten wird gestartet und synchronisiert, nachdem die Bestimmungsdefinition und eine gründliche Vertrauensüberprüfung durchgeführt wurden. Um die Geschwindigkeit der Informationsverarbeitung zu verbessern, werden alle Prozesse gleichzeitig und unabhängig voneinander abgewickelt. Wenn es keine Eingangsvariablen gibt, wird eine leere Ledgerablage angelegt - ein Platz im RAM wird für einen weiteren vereinfachten Zugriff reserviert. Falls das gewünschte Ledger nicht verfügbar ist, wird eine Anforderung an Vertrauensknoten (trusted nodes) gesendet, um alle Transaktionen für das synchronisierte Konto zu empfangen.

Wenn der Eingabeparameter ein Objekt ist, das die Transaktion charakterisiert, dann wird die Suche in allen laufenden Synchronisations-Threads gestartet. Die Operation ergibt einen numerischen Code - die Positionsnummer im Trusted Node Ledger für den aktuellen Thread oder die Fehlernummer, wenn der Wert kleiner als Null ist. Wenn die Thread-Methode mit einem Verbindungsfehler endet, dann endet der Thread komplett.

2. Netzwerk-Konsens

Der Konsens in CREDITS ist eine Methode der Gruppenentscheidung. Mit dem Ziel, Lösungen zu entwickeln, die für alle Netzwerkknoten akzeptabel sind.

Konsensvergleich

Die Definition der Prinzipien des dezentralen CREDITS-Ledgers, um verschiedene Arten von Konsens zu vergleichen:

- Ledger-Verfügbarkeit (Knoten können jederzeit Daten in das Ledger schreiben und daraus lesen);
- Änderbarkeit durch alle beteiligten Netzwerkknoten;
- Konsistenz aller Systemknoten (alle Knoten sehen eine absolut identische Version des Ledgers, die nach allen Änderungen aktualisiert wird);
- Separations-widerständig (wenn ein Knoten nicht mehr funktionsfähig ist, hat dies keinen Einfluss auf den Betrieb des gesamten Ledgers)

Vergleichsparameter	Credits spezifisch dPoS und BFT	PoW	PoS
Das Prinzip der Identifizierung des Knotens, der den Block erzeugt hat.	Berechnung der mathematischen Funktion. Bestätigung der Speicherung der letzten Ledgerkopie und BFT	Durchführung einer iterativen Berechnung der mathematischen Funktion mit unterschiedlicher Komplexität.	Suche nach dem maximalen Stack unter den Teilnehmern (konkurrierende Knoten).
Angriff 51 %.	Unwahrscheinlich, da es notwendig ist, ein vollständiges Ledger in den Ressourcen zu haben und die Rechenleistung zu berechnen; und da die vertrauenswürdigen Knoten dynamisch ausgewählt werden.	Wahrscheinlich, aber sehr teuer in Bezug auf die Nutzung der Ressourcen.	Wahrscheinlich, aber teuer, wegen der Notwendigkeit, den eigenen Stack zu erhöhen.
Kompensation für die die auf der Seite erbrachten Leistungen für Ledger/Blockchain.	Automatisch berechnet, abhängig von der Kommission pro Vorgang.	Fix Angeboten für das Block-Mining	Fix Angeboten für das Block-Mining

Das Konzept des Hauptnetzknotens

Alle Netzwerkknoten sind dezentralisiert und haben keiner davon hat Vorrang. Es ist erforderlich, einen Netzwerkknoten zu definieren, der die Warteschlange von Transaktionen verarbeitet, die auf verschiedenen Netzwerkknoten gespeichert sind. Danach muss er einen neu generierten Transaktionsblock in das Ledger eintragen.

Die CREDITS-Plattform verwendet ein eigenes kombiniertes Protokoll, um die Geschwindigkeit der Transaktionsverarbeitung zu erhöhen und die vollständige Sicherheit der Datenspeicherung, Verarbeitung und Übertragung von Transaktionen zu gewährleisten. Das Protokoll basiert auf der Berechnung der mathematischen Funktion aller Ledger-Transaktionen unter Anwendung der Proof-of-Work-Prinzipien. Es bestimmt genau die Speicherung der aktuellsten Kopie des Ledgers und der Software an diesem Knotenpunkt (Proof of Capacity), indem es die Prüfsumme der Werte des gesamten Inhalts - den Hash-Code - berechnet. Die Größe der Dateien wird ebenfalls bestimmt, als Nachweis dafür, dass es sich um die neueste Version handelt, eine aktuelle Kopie und ein Hash-Code der letzten im System aufgezeichneten Transaktion.

Um den Hauptnetzknoten zu werden, sucht der Knoten nach dem Wert der Hash-Funktion, die er auf der Grundlage des zuletzt gespeicherten Ledgers berechnet. Wir organisieren ein gesundes Wettbewerbsumfeld zwischen den

Netzwerkknoten für die Möglichkeit, zum Hauptknoten zu werden, ein neues Ledger zu erzeugen und zu speichern.

Nachdem die Funktion berechnet und das Ergebnis erzielt wurde, wird sie zur Überprüfung an alle Netzwerkknoten gesendet. Das Ergebnis enthält einen Zeitstempel der Berechnung und einen Wert, der auf der Berechnung der Funktion der Ledger-Dateien und der Software basiert. Alle Knoten erhalten den berechneten Wert, vergleichen die für die Suche nach dem Hauptnetzwerkserver zugewiesene Rechenzeit, überprüfen und bestätigen den Vertrauensfaktor des Knotens und bestätigen auch seine Möglichkeit, am Wettbewerb teilzunehmen - zum Hauptnetzknoten zu werden.

Nach der Freigabe aller Netzwerkknoten (BFT Principals) wird eine Liste von Knoten gebildet, die den Wert der Funktion korrekt berechnet haben und einen Zeitstempel enthalten. Der Knoten, der das korrektes Ergebnis und es in der schnellsten Zeit genehmigt, wird zum Hauptnetzknoten des Momentes.

Das Konzept des SHA-2-Algorithmus wird verwendet, um die Hash-Summe der Datei zu berechnen. Die Hash-Funktionen der SHA-2-Familie sind auf der Basis der Merkle-Damgard-Struktur aufgebaut.

Die Ausgangsnachricht nach der Addition ist in Blöcke aufgeteilt, jeder Block ist in 16 Worte aufgeteilt. Der Algorithmus durchläuft jeden Nachrichtenblock durch einen Zyklus mit 64 oder 80 Iterationen (Runden). Bei jeder Iteration werden 2 Wörter konvertiert, und der Rest der Wörter definiert die Konvertierungsfunktion. Die Ergebnisse der einzelnen Blockprozesse werden zusammengefasst. Die Summe ist der Wert der Hash-Funktion. Der interne Zustand wird jedoch aufgrund der Ergebnisse der vorangegangenen Blockprozessierung initialisiert. Daher ist es unmöglich, dass man

Blöcke selbständig bearbeiten und die Ergebnisse zusammenfassen kann.

Ausstattung von Netzwerkknoten

Wir sind bestrebt, eine Plattform mit den schnellstmöglichen Transaktionsverarbeitungsmerkmalen zu schaffen, und schlagen daher vor, einen materiellen Anreiz zu nutzen, um die Netzwerkknoten in bestem Zustand zu halten: leistungsfähige Serverausrüstung und hohe Internetbandbreite.

Als materielle Kompensation erhält der Eigentümer des Hauptnetzknotens eine Vergütung in Form von CREDITS Währungseinheiten aus einer Anzahl von Provisionen pro Transaktion dieses bearbeiteten Ledgers. Der Rest (½) ist für die trusted Nodes gedacht, die am BFT-Konsens teilnehmen. Der Prozentsatz kann nach dem ersten initialen Coin-Angebot für mindestens drei Jahre angepasst und vom Quotenbildungssystem getrennt werden durch ein föderatives Voting aller Netzwerkknoten.

Daher empfehlen wir den Besitzern von Servern, die Hardware der Server auf der höchsten Performance zu halten und ein qualitativ hochwertigen, schnellen Kommunikationskanal aufrecht zu erhalten.

Konsensbildung

Als Ergebnis haben wir den Hauptnetzknoten durch alle Knoten selektiert. Die Hauptaufgaben des Hauptknotens bestehen darin, Transaktionen im Kandidatenstatus zu erhalten, um das Ledger von allen Knoten aus zu ergänzen, sie zu bearbeiten, das letzte relevante Ledger aufzubauen und ein neu aufgebautes Ledger an alle Netzknoten zu senden. Der Prozess der Transaktionsabwicklung und des Aufbaus des letzten relevanten Ledgers ist genau die Suche nach einer geeigneten Konsenslösung. Das Resultat aus dem Aufbau des letzten relevanten Ledgers ist die Konsens-Lösung.

Der gesamte Prozess kann in die folgenden Schritte unterteilt werden:

1. Suche nach dem Hauptnetzknoten;;
2. Aufbau von Trusted Nodes;
3. Empfang der Liste der Transaktionen und Aufbau einer Liste von Kandidaten für die Aufnahme in das Ledger;
4. Bearbeitung der Kandidatenliste, Abstimmung der Knoten (Trusted und Common Nodes haben unterschiedliche Gewichtsfaktoren (Trust Factor);
5. Streichung aus der Liste der Kandidaten für unbestätigte Transaktionen, die nicht verifiziert wurden oder eine negative Bestätigung haben;
6. Erstellung einer Liste der bestätigten Transaktionen, die dem Ledger hinzugefügt werden sollen;
7. Hinzufügen von Transaktionen zum Ledger mit dem Zeitstempel und dem Hash-Code des Blocks, der die Transaktion enthielt;
8. Senden des Blocks mit Transaktionen an alle Netzwerkknoten. Wenn es empfangen wird, wird es zu den Registrierungsdatenbanken aller Knoten hinzugefügt.

Aufbau und Initiierung des Ledgers

Der gesamte Prozess kann in der folgenden Reihenfolge beschrieben werden:

1. Der Endnutzer des Netzwerks im System erzeugt eine Transaktion.
2. Wenn alle Bedingungen des darin festgelegten Smart-Contracts erfüllt sind, leitet der Benutzer die Aktion (Transaktion) ein, indem er die gewünschte Methode mit Hilfe der Plattform-Software aufruft.
3. Um den Grundprinzipien der Blockchain zu folgen, verfolgt der Validiererkern die Synchronisation und Invarianz der neuesten Ledger-Version.
4. Zum Zeitpunkt der Konsensbildung werden alle Transaktionen, die während des Zyklus empfangen werden, im Block gesammelt.
5. Dem Block wird eine Nummer zugewiesen, die aus einem Zeitstempel und einem in einen Hash-Code konvertierten Node-Identifier besteht, und dann wird der Block in das Konsensus-Modul gestellt.
6. Nach der Erstellung der White List der Kandidaten wird nicht nur der Hash der Transaktion in das Ledger geschrieben, sondern auch der Hash des Blocks, um die Quelle immer darauf basierend zu zertifizieren.
7. Dieser Hash ist eine Art Signatur des Blocks und desjenigen, der diesen Block mit Transaktionen angelegt hat.
8. Nach der Konsensbildung mit Hilfe eines föderativen Suchalgorithmus werden die dem Block hinzugefügten Transaktionen an den Kernel des Prüfers übergeben und in das Ledger geschrieben.

Nicht im Register erfasste Transaktionen

Transaktionen, die nicht in der Liste der fertigen Transaktionen enthalten sind, werden als abgelehnt markiert. Eine Information dazu wird sofort beim Sender (Initiator) der Transaktion angezeigt. Transaktionen, die nicht im Ledger enthalten sind, verbleiben in der Kandidatenliste und werden in den Netzknoten gespeichert. Dort kommen auch alle neuen Transaktionen an, die der Server zum Zeitpunkt des Konsenses erhalten hat, und dann beginnt der Suchprozess von neuem. Ein solcher kontinuierlicher zyklischer Betrieb des Netzes ermöglicht es, Transaktionen für einen relativ kurzen Zeitraum durchzuführen und dabei einen hohen Grad an Zuverlässigkeit und Relevanz der Informationen zu halten.

3. Transaktionsabwicklung

Transaktionen

Eine Transaktion ist die Mindesteinheit des Systems, die die Plattform über die Ausführung von Vertragsmethoden oder direkte Übertragungen zwischen Konten informiert, ohne einen Smart-Contract zu erstellen, gefolgt von der Einordnung des Ergebnisses in das Peer-to-Peer-Netzwerk.

Konsensbildung

Das System verwendet ein föderatives Modell, um einen Konsens zu bilden - die Abstimmung über vertrauenswürdige Validatorknoten und auch den Konsensbildungsalgorithmus - einen Algorithmus für die Passage eines endlichen Automaten. Consensus arbeitet nach Zyklen (Zeitschritten), pro Zeitschritt werden Transaktionen extrahiert und in einen Pool (eindimensionales Array) gestellt. Nach der Platzierung im Pool werden alle Transaktionen an vertrauenswürdige Knoten gesendet, um eine Antwort zu erhalten. Wenn der Response empfangen wird, dann kann die Transaktion, für die die Anforderung

zum Hinzufügen gesendet wurde, in das Ledger dieses Prüfers aufgenommen werden. Danach wird es an den nächsten Prüfer im Netzwerk gesendet. Wenn ein Konsens zustande kommt - am Ende der Kette, wo die Rechtmäßigkeit der Übertragung voll und ganz gegeben ist, wird der Vorgang zur Validierung mit einer Markierung zum Schreiben und Sichern in das Ledger gesendet.

Transaktionsabwicklung

Um die Dezentralität des Systems zu erreichen, muss jeder Server über Ledgerspeicher verfügen und gleichzeitig ein vollwertiger Handler aller Transaktionen sein.

Das System verwendet das Konzept der Systemkerne. Unter Kernel versteht man einen Data Handler, der eine bestimmte Produktionsaufgabe ausführt, unabhängig von der Verfügbarkeit und Funktionsfähigkeit der übrigen Systemkomponenten. Jeder Kernel erhält bei der Eingabe, zum Zeitpunkt der Ausführung der Aufgabe, eine Liste von Variablen zur Verarbeitung. Dabei wird am Ausgang immer ein Ergebnis ausgegeben - ein positives, irgendein anderes oder ein Fehler. Daher enthält der Systemkern neben dem Hauptdatensatz immer auch den Antwortcode. Diese Struktur ist

Voraussetzung für die höchstmögliche Geschwindigkeit eines jeden Prozesses, der unabhängig voneinander arbeiten muss.

Ledger-Erfassungsstruktur

Um eine signifikante Performance des Ledgers zu erreichen, aber gleichzeitig, ohne die Sicherheit zu beeinträchtigen, schlagen wir vor, eine Ledger-Datenbank zu verwenden, ohne den Merkle-Baum aus dem Hash-Code des vorherigen Blocks und dem Transaktionsergebnis zu erstellen.

Merkle tree (TTH - Tiger Tree Hashing) ist eine Art Hash-Funktion, die dazu dient, die Integrität von Daten zu überprüfen, einen eindeutigen Identifikator der Kette zu erhalten und die Sequenz wiederherzustellen. Die Daten werden in kleine Teile aufgeteilt - Blöcke, die einzeln mit Leaf Tiger Hash gehasht werden, dann wird der Internal Tiger Hash aus jedem Paar von Hashes einzeln berechnet. Wenn der Hash kein Paar hat, dann wird er unverändert in die neue Kette übernommen. Als nächstes wird der Internal Tiger Hash in der Kette für jedes Paar neu berechnet. Dieser Vorgang wird solange wiederholt, bis noch ein Hash übrig ist.

Wenn das Ledger mit Merkle-trees betrieben wird, ist die Transaktionsgeschwindigkeit sehr gering und die Rechenleistung sehr hoch. Unserer Meinung nach handelt es sich hierbei nicht um eine rationelle Nutzung der Datenspeicherung.

Struktur des CREDITS-Ledgers

Wir bieten an, Merkle-trees aufzugeben und das Transaktions-Ledger im CREDITS-System zu verwenden; jeder Eintrag besteht aus einem Hash-Code des Transaktionsblocks, der zusätzlich zum Ledger in die Kandidatenliste aufgenommen wird. Außerdem hat der Eintrag den Node-Identifier und den Zeitstempel, wann er generiert wurde. Der Ledger-Eintrag enthält die Transaktionsrichtung, die Anfangs- und Endkonten, die Art der Ausbuchung, die Anzahl der Ausbuchungseinheiten, die Art des Deposits und die Anzahl der Depositeneinheiten. Dieses Prinzip

erhöht die Geschwindigkeit der Vorgangsbearbeitung, erhöht die Komplexität des unzulässigen Ledgerwechsels und schließt mögliche Änderungen der Ledgerbuchung im Nachhinein aus.

Blockgröße

Die Zeiteinheit ist der Zyklus der Suche nach den Haupt- und Vertrauensknoten, und die Zykluszeit wird in Abhängigkeit von der Komplexität des Netzwerks berechnet. Pro Zeiteinheit enthält der Netzplan N Transaktionen, die vom Ende des vorhergehenden Zyklus bis zum Beginn des nächsten Zyklus generiert und zur Bearbeitung an das Netzwerk übergeben werden, um den Status "Kandidat, der dem Ledger hinzugefügt werden soll" zu erhalten. Die aus dem Netzplan N ausgewählten Transaktionen werden auf dem Block platziert. Die Blockgröße hängt von der Anzahl der Transaktionen darin ab.

Suche nach Transaktionsteilnehmern

CREDITS Peer-to-Peer-Netzwerk kann als Graph dargestellt werden, mit Benutzerkonten in Form von Eckpunkten und einer Vielzahl von möglichen Transaktionen in Form von Richtkanten, die zwei Eckpunkte (Accounts) verbinden. Da alle Kanten einen Anfangs- und einen Endpunkt haben, können Sie jederzeit einen orientierten Graphen (orgraph) konstruieren.

Wenn wir die folgenden Voraussetzungen für die Identifizierung annehme:

- Jede Transaktion hat immer einen Sender und einen Empfänger;
- Jeder Eckpunkt (Account) kann immer mit einem anderen Eckpunkt mit einer ausgerichteten Flanke verbunden werden (Transaktion);
- Jeder Scheitelpunkt des Graphen (Kontos) hat eine endliche Anzahl gerichteter Kanten (eingehende und ausgehende Transaktionen).

Im Zusammenhang mit dem Vorstehenden kann man sagen, dass der orgraph den erforderlichen Weg enthält, um die notwendigen Transaktionsbedingungen zu erfüllen und eine einfache Kette zu konstruieren. Da es sich um eine endliche Folge von Eckpunkten handelt, bei der jeder Eckpunkt (außer dem letzten) durch eine Kante mit dem nächsten Eckpunkt in der Folge verbunden ist.

Datenübertragungskanal

Jeder Kommunikationskanal zwischen dem Hauptnetzknoten und dem common Knoten des CREDITS-Netzwerks ist ein separater Thread (Multithreading), innerhalb dessen die Daten bei der Ausführung der Transaktion verschlüsselt übertragen werden.

Um die Netzwerksicherheit zu gewährleisten, werden alle Daten zwischen den Validierungsknoten verschlüsselt übertragen und jede Verbindung zwischen den Knoten ist Low-Level basierend auf der Netzwerkbibliothek. Wenn die Datenübertragung mit einem Fehler erfolgt, sollte der Thread automatisch unterbrochen werden, der entsprechende Eintrag wird zum Schreiben in das Logging-System und dann in die Log-Datei gestellt. Die Datenübertragung erfolgt über typisierte Variablen. Die übertragenen Daten werden mit dem symmetrischen RC4-Algorithmus verschlüsselt. Da dieser Algorithmus unter einem geheimen Schlüssel funktioniert, wird dieser Schlüssel beim Verbindungsaufbau zwischen den Knoten übertragen und verschlüsselt nach dem Diffie-Hellman-Algorithmus übertragen.

Der RC4-Algorithmus ist, wie jede Stream-Chiffre, auf der Basis eines Pseudozufallsbitgenerators aufgebaut. Der Schlüssel wird in den Generator-Eingang geschrieben, und am Ausgang werden Pseudozufallsbits gelesen. Die Schlüssellänge kann von 40 bis 2048 Bit betragen. Generierte Bits haben eine gleichmäßige Verteilung.

Der Diffie-Hellman-Algorithmus ermöglicht es zwei Parteien, einen gemeinsamen geheimen Key zu erhalten, indem sie einen Kanal verwenden, der ungeschützt vor dem Abhören, aber geschützt vor einem Wechsel des Kommunikationskanals ist. Der empfangene Schlüssel kann verwendet werden, um Nachrichten mit symmetrischer Verschlüsselung auszutauschen. Der Algorithmus basiert auf der Komplexität der Berechnung diskreter Logarithmen. In ihm - wie in vielen anderen Algorithmen mit einem öffentlichen Schlüssel - werden die Berechnungen modulo zu einer bestimmten großen Primzahl P durchgeführt. Zunächst wird eine bestimmte natürliche Zahl A, die kleiner als P ist, auf bestimmte Weise ausgewählt. Wenn wir den Wert X verschlüsseln wollen, dann berechnen wir Y=AX mod. P.

Und es ist einfach, Y mit X zu berechnen. Das umgekehrte Problem der Berechnung von X aus Y ist ziemlich kompliziert. Exponent X wird genau als diskreten Logarithmus Y bezeichnet. Wenn man also die Komplexität der Berechnung des diskreten Logarithmus kennt, kann die Zahl Y auf jedem Kommunikationskanal öffentlich übertragen werden, da bei einem großen Modul P der Anfangswert X fast unmöglich zu wählen ist.

Der Diffie-Hellman-Algorithmus zur Erzeugung eines Schlüssels basiert auf dieser mathematischen Tatsache. Alle Aktionen im System sind mit dem Zeitstempel, der Nummer des vorherigen Blocks, dem Login des Benutzers und der Smart Contract ID verknüpft. Dies ermöglicht das

Auffinden von Duplikaten bei der Ausführung. Wenn ein Duplikat gefunden wird, dann nehmen wir die erste Transaktion aus dem Pool, der Rest gilt als unzulässig.

Aktionen im System

Eine Aktion im System ist eine Transaktion, die die einfachste Übertragung des Wertes von Konto zu Konto oder die Übertragung des Ergebnisses der Contract-Methode an den Validator für die anschließende Suche nach einer Lösung im Konsensus-Such-Subsystem charakterisiert..

Um eine Verdoppelung der Transaktion im selben Block mit dem gleichen Identifikator zu vermeiden, akzeptiert das System eine Vereinbarung, dass die einzig wahre und korrekte Transaktion diejenige ist, die zuerst zum Validator-Subsystem zur Verarbeitung gelangt ist. Da bereits im Prüfsystem vermerkt ist, dass eine Transaktion bereits vom aktuellen Account aus durchgeführt wurde und keine Werte mehr im Konto für die Durchführung der Transaktion vorhanden sind, kann kein Konsens gefunden werden. Damit ist das Problem des doppelten Waste gelöst.

Wenn die Transaktion ausgeführt wird, werden Informationen an den Validator gesendet und bestätigt. Die Informationen über die Änderung des Ledger-Status werden automatisch an alle Knoten aus der Trusted List verteilt, woraufhin das Ledger synchronisiert wird.

Um immer ein aktuelles Transaktions-Ledger unter allen Trusted Nodes für den aktuellen Validator-Knoten zu haben, ist es notwendig, die neu eingetroffene Transaktion jedes Mal im Ledger aller Knoten zu synchronisieren. Um dieses Problem zu lösen, sollte ein separater Port für die Synchronisation verwendet werden (wenn es eine solche Möglichkeit gibt). Diese Möglichkeit erhöht die Geschwindigkeit der Verarbeitung der Informationen, die an den Validator-Kernel eingehen, aufgrund der Verteilung der Last auf den Port. Der Synchronisations-Thread wird immer ausgeführt, er ist zyklisch. Die Priorität für die Zuweisung von RAM und CPU-Last (unter Verwendung von CPU-Zyklen) ist niedriger als der Durchschnitt. Der Speicher speichert die letzten 1000 Operationen und den Status der Konten für sie (in verschlüsselter Form unter Verwendung eines synchronen Algorithmus), was die Reaktionsgeschwindigkeit auf Anfragen von anderen Prüfknoten erhöht.

Hinzufügen einer Transaktion zur Validierung

Das Hinzufügen von Transaktionen zum Ledger wird nur aus dem Validator-Subsystem unmittelbar nach der Konsensbildung und der Erstellung einer Whitelist mit dem Ergebnis der Speicherung von Transaktionen im Ledger aufgerufen. Um die Sicherheit zu erhöhen, sind Aufrufe von Drittsystemen sind nicht möglich. Eingehende Parameter - das Objekt, das die Transaktion charakterisiert. Der Ergebniswert kann ein möglicher Fehlercode sein "/ 0 < ResultValue". Wurde die Funktion wurde fehlerfrei ausgeführt, ist das Ergebnis die Nummer des Eintrags im Ledger. Eingangsparameter - das Objekt, das die eindeutige Bezeichnung der Transaktion, den Absender, den Empfänger, den übergebenen Wert, die Wertkorrespondenz, den gewünschten Wert, den Betrag des transferierten Wertes, den Betrag des gewünschten Wertes und andere Systeminformationen enthält, die bei Bedarf geändert werden können.

Transaktionskosten

Das System verwendet die Währung CREDITS. Sie dient folgendermassen:

- Als internes Zahlungsmittel für die Systemnutzung;
- Um verschiedene Währungen innerhalb des Systems auszutauschen;
- Um verschiedene Werte innerhalb des Systems auszutauschen;
- Erstellen und Bearbeiten von Vorgängen im Rahmen von Smart Contracts;
- Um Informationen von Drittanbietern für Dienstleistungen innerhalb des Systems einzukaufen.

Die Kosten für eine Transaktion können je nach Netzlast und Benutzer des Systems variieren, was theoretisch einen riesigen Transaktionsfluss zu einer bestimmten Spitzenzeit steuern kann. Wir schlagen

vor, die Materialmethode und die Auswirkungen auf die Systembenutzer zu verwenden, um die Netzwerklast zu kontrollieren.

Die Kosten für die Durchführung von Transaktionen in den ersten drei Jahren des Systembetriebs werden individuell für verschiedene Arten von Transaktionen und Operationen festgelegt. Zukünftig wird ein Algorithmus für den Automatische Generierung der Transaktionskosten wird entwickelt..

4. Smart Contracts

Einleitung

Ein Smart Contract im CREDITS-System ist ein elektronischer Algorithmus, der eine Reihe von Bedingungen beschreibt, unter denen Aktionen und Ereignisse in der realen Welt oder in digitalen Systemen assoziiert werden können.

Um selbstgesteuerte Smart-Contracts zu implementieren, ist eine dezentrale Umgebung erforderlich, die den Faktor Mensch vollständig ausschließt. Und um die Übertragung der Kosten eines Smart-Contracts zu nutzen, ist eine von der zentralen Instanz unabhängige Krypto-Währung erforderlich ist.

Entitäten

Ein Smart-Vertrag in CREDITS besteht aus folgenden Einheiten:

1. Property (öffentliche Variablen) - die Systemeinheit, die öffentliche Daten speichert, die für die Arbeit des Vertrages im CREDITS-System notwendig sind.
2. Methode ist die CREDITS-System-Einheit, die für die Einhaltung der Logik und Reihenfolge der Aktionen bei der Durchführung der Transaktion verantwortlich ist (Aktionen im Rahmen des Vertrags).

Die Teilnehmer des CREDITS-Systems unterzeichnen die Smart Contracts mit Hilfe des Methodenaufrufs, der die Vertragseigenschaften modifiziert, indem sie die Prozesse zur Überprüfung der Einhaltung der Bedingungen und der Koordination starten.

Ein Smart-Contract tritt nach der Unterzeichnung durch die Parteien in Kraft. Um eine automatisierte Erfüllung der Verpflichtungen zu gewährleisten, ist ein Umfeld der Existenz erforderlich, das die Ausführung der Vertragsbedingungen vollständig automatisiert. Das bedeutet, dass Smart Contracts nur in einer Umgebung existieren können, die ungehinderten Zugriff auf den ausführbaren Code der Smart Contracts hat.

Alle Vertragsbedingungen müssen eine mathematische Beschreibung und eine klare Ausführungslogik haben. Das Grundprinzip eines Smart Contracts ist also die vollständige Automatisierung und Zuverlässigkeit der Vertragsbeziehungen zwischen den Parteien.

Smart Contract Methode

Die CREDITS smart contract-Methode ist die Systeminstanz, die für die Einhaltung der Logik und Reihenfolge der Aktionen während der Transaktion verantwortlich ist (Aktionen unter dem Vertrag).

Die Logik und die Reihenfolge der Aktionen werden durch einen Programmcode (Modul) beschrieben, der Befehle enthält; ihre sequentielle Ausführung ermöglicht es, das gewünschte Ergebnis zu erzielen. Dieser Code kann Systemkommandos (z.B. den Zuweisungsbefehl), Benutzerkommandos (separat geschriebene Funktionen), Vertragseigenschaften (statisch oder dynamisch initialisierte Variablen, die in jeder beliebigen Vertragsmethode verfügbar sind) und Methoden aus jedem anderen Drittvertrag behandeln, die nur dem Eigentümer des verbundenen Vertrags zur Verfügung stehen. Zur weiteren Verbreitung wird die Entwicklung in Skriptsprachen (z.B. JavaScript) bereitgestellt.

Die Methode (Programmcode) erlaubt die Verwendung aller weit verbreiteten Skriptsprachenoperatoren (Befehle) (Zuweisung, bedingte und unbedingte Sprünge), die Erstellung von Funktionen und Prozeduren (Unterprogramme), Anbindung von Drittanbieter-Bibliotheken.

Virtuell ausführbare Maschine

Die Vertragsmethode des CREDITS-Systems wird in der virtuellen Umgebung des Systems (Virtual Machine, im Folgenden VM genannt) ausgeführt. Wenn eine Methode für einen bestimmten Vertrag aufgerufen wird, weist VM einen Speicherbereich zu und lädt den Vertrags-bytecode, der die Methoden und die initialisierten Variablen enthält (oder beim Aufruf anderer Vertragsmethoden neu definiert). VM beginnt mit der Verarbeitung des Methodenbytecodes, zur Laufzeit werden Variablen und Code in seinen Speicherbereich geladen, und Befehle werden nacheinander ausgeführt, ihr Ergebnis wird an das Peer-to-Peer-Netzwerk zur späteren Platzierung im Ledger übergeben..

Der Initiator der Ausführungsmethode ist der Benutzer des Systems, in dessen Namen diese Methode wird gestartet.

Value Term

CREDITS Kryptowährung ist auch ein Indikator für die Werthaltigkeit einer Vertragseinheit, um zwei völlig unterschiedliche Einheiten zu vergleichen und einen Konsens bei der Ausführung oder Annahme des Vertrages durch die Parteien zu erzielen. Anstatt jede einzelne Wert/Gateway-Kombination zu registrieren, dient die CREDITS Kryptowährung als Bündel für die Durchführung von Werttransfers. Dies ist möglich, da jeder Wert in Bezug auf die Währung CREDITS liquide ist, d.h. jeder Wert kann in Bezug auf jeden anderen Wert liquide sein.

Ausführung der Smart Contract Bedingungen

Die Vertragsterms im CREDITS-System sind die Werte der auslösenden (geprüften) Felder, die zum Abschließen (Vervollständigen) des Vertrags erforderlich sind.

Die Erfüllung der Smart-Vertragsbedingungen ist ein Verfahren, bei dem die Trigger-(Wunsch) Felder auf einen äquivalenten Sollwert geprüft werden. Es gibt drei Möglichkeiten, eine Lösung zur Erfüllung der Vertragsbedingungen zu finden:

1. Der Vertrag kommt zwischen zwei oder mehreren Parteien zur Wertübertragung zustande. In diesem Fall ist die Vertragserfüllung die Bereitstellung des Kostenäquivalents in Höhe des Wertes für die übertragende Partei von der empfangenden Partei.
2. Der Vertrag kommt zwischen den Parteien über die Wertübertragung zustande, wobei die Zahlung jedoch erst nach Erfüllung einer bestimmten Anzahl von Bedingungen (z.B. Lieferung des Wertes an die empfangende Partei) erfolgen muss.
3. Im System wird ein Vertrag über die Umrechnung von einem Wert in einen anderen mit einem Kostenäquivalent in Form von CREDITS abgeschlossen. In diesem Fall sucht die Plattform nach dem kürzestmöglichen Weg, einen Wert gegen einen anderen auszutauschen, indem sie in andere Kontrakte konvertiert. Jede Vertragserfüllung kann für eine Transaktion oder für mehrere Transaktionen erbracht werden, was die Möglichkeit bietet, die erforderliche Menge an Werteinheiten zu sammeln, um den Vertrag abzuschließen..

Datenquellen

Für die korrekte und vollständige Arbeit, die Prüfung und Bereitstellung zusätzlicher Informationen, um eine ausgewogenere und optimale Lösung zu finden, nutzt CREDITS Drittanbieter im Datenbereich. Die Notwendigkeit, zusätzliche Datenquellen in das System einzubinden, ist auf die Unzulänglichkeit der öffentlichen Informationen über eine oder mehrere Vertragsparteien zurückzuführen (z.B. Erlangung der Kreditwürdigkeit des Kreditnehmers für die Entscheidung über die Kreditvergabe).

Um mit Informationssystemen von Drittanbietern zu arbeiten, kann die Plattform einen Integrationsbus aufrufen, der per Fernzugriff eine Anfrage an ein Drittsystem (Site) in einem Format für die kostenpflichtige Datenpräsentation für die Systemteilnehmer mit Bezahlung in CREDITS generiert.

Die Anfrage wird in verschlüsselter Form an Ports und Adressen gesendet, die von anderen Informationssystemen als den Standard-Informationssystemen zur Verfügung gestellt werden. Das Ergebnis der Anfrage kann jede Antwort auf den Dienst sein, die die notwendigen Informationen enthält, um eine Entscheidung zu treffen, oder ein Fehlercode, der die Unmöglichkeit kennzeichnet, die erforderliche Antwort zu erhalten sowie mögliche Schritte zur Behebung des Fehlers enthält.

5. Implementations-Plan

Technischer Plan der Projekt-Implementation

	S1	S2	S3	S4	S5
	Pre-Alpha	Alpha	Beta	Release candidate	Release
Storage, Consensus mFA Consensus	FA : Key Design Implementation	mFA : Key Design Implementation PoW	mFA Optimierung	–	–
		(Proof-of-Work) und PoC (Proof-of-Capacity)			
Data Store	Decentralization Ledger, NoSQL Store Implementation	MessagePack History	–	Blockchain backup	–
CVM (Credits virtual machine)	Design und Implementation	Integration mit ecosystem	Optimierung	Errors checken	–
Drittanbieter-System	–	Design und Implementation	Im vollen System integrieren	–	Optimierung
Inference Engine	Formale Spezifikation und Key Design Elemente	Reasoner Integration mit Blockchain	Reasoner Optimierung	–	–
User interface	Implementation	Web UX Design	–	–	–
Wallet	Wallet Formal Spezifikation		UX Design Application Test	–	Android, iOS, Desktop Wallets
RPC & REST API	Formal Spezifikation	Blockchain Explorer	–	Drittanbieter-Explorer	–

CREDITS Kryptowährung

Nach dem Start der Release-Version des Systems wird ein fester Betrag von 1,000,000,000 CREDITS ausgegeben. Sie werden gegen ERC20-Standard-Token eingetauscht, die beim ersten Tokenverkauf ausgegeben werden. Sie werden zu einem festen Wechselkurs getauscht: 1 ERC20 Standard-Token = 1 CREDITS Geldeinheiten.