



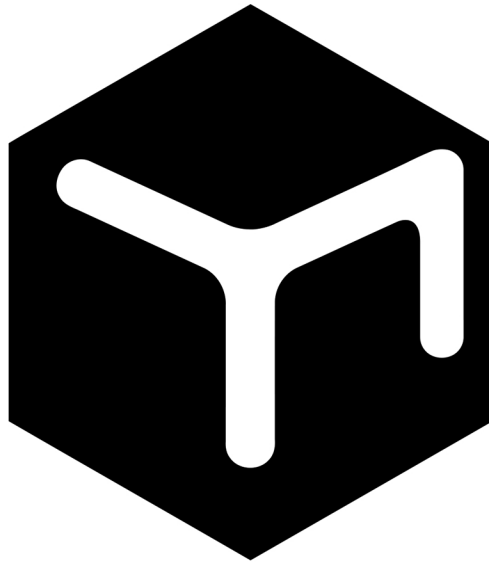
Relictum Pro

Blockchain 5.0

Decentralized Ledger Technology

Yellowpaper

version 2.05.21.ru-2



Relictum Pro

Blockchain 5.0

Global platform covering all the aspects
of human life in a distributed registry

With the use of HYPERNET technology based
on peer-to-peer peering networks

Zusammenfassung

Relictum Pro ist eine Blockchain der neuesten Generation von Blockchain 5.0, die über die notwendigen grundlegenden und ausreichenden Bedingungen verfügt, um den Betrieb der Blockchain der vierten Generation sicherzustellen:

1. HYPERNET - 4-dimensionale Netzwerkvermittlung virtueller Verbindungen.
2. Mehrdimensionale Organisation von Chains (Smart Contracts) mit sofortigem Zugriff.
3. Die globale Plattform zur Formalisierung aller Ereignisse im wirtschaftlichen und täglichen Handeln von Menschen und Staaten.
4. Die Geschwindigkeit der Nachrichtenübermittlung (Block) an jeden Knoten von 100 000 TPS/s.
5. Dynamische Chains.
6. Die Hierarchie der Rollenknoten.
7. Regeneration des Netzwerkes.
8. Beseitigung von Mehrdeutigkeiten.
9. Skalierbarkeit. Es unterscheidet sich darin, dass Binärdateien von Portfolios absolut identisch sind. Bei der Initialisierung ermittelt der Knoten automatisch seinen Status (Rolle), oder der Besitzer kann die Rolle jederzeit manuell ändern.
10. Verteilter Speicher. Es ist unvermeidbar erforderlich, ein verteilter Speicher zu erstellen, da das einfache Hashing von Ereignissen für digitale Signaturen die Speicherung des Inhalts selbst und die anschließende Bereitstellung des Dokuments sowohl in privater als auch in gemeinsam genutzter Form nicht gewährleistet.
11. Zeitsynchronisation nach dem Uhrgang.
12. Verwendung des idealen Zufallszahlengenerators basierend auf der Erfassung von Daten (Abtastwerte – die reduzierte Energie jedes Pixels) des Hintergrunds der Radioemission (Quantisierung).
13. Die Überprüfung der Integrität einer Binärdatei, sowohl im stationären Modus als auch im Speicher, muss von externen Ressourcen durchgeführt werden.
14. Dreistufige Integritätsprüfung der Chains. Es besteht aus dem Empfang eines Hashs, eines vollständigen Hashs, eines Hashs in Intervallen und einer Kontrollsumme.

Inhalt

Einleitung	6
Funktional	7
1. Portfolio	7
2. Chains von Blöcken	8
3. Verteilter Speicher	9
Prinzipien der Vernetzung. Netzwerkprotokoll	11
1. König	11
2. Z level - Generäle	11
3. P level - üblicher Knoten	11
4. L level - lichter Knoten	12
5. S level - privater Knoten	13
6. O level - Knoten von Ruhemodus	13
7. Cloud-Knoten	13
Funktionstabelle der Knotentypen	14
Architektur von Chains	15
Anwendungsgebiet	16
Recht	16
Versicherung	16
Finanzen	16
Medizin	16
Logistik	16
Probleme mit dem Ticketmarkt	16
Kohlenwasserstoffe	17
Token und seine Klassifizierung	17
Zahlungsmittel	17
Utility-Token	17
Digitales Anlagevermögen	17
Tokenized Asset	17
Tokenomics	17
Systemarchitektur (Architekten)	17
Hauptphasen des Netzwerks	18
Kryptoalgorithmus	19
Aufzeichnungsart	19
Blockchain-Netzwerkziele	19



Klärgrube (Funktionen)	19
Formeln und Umsetzung	20
Beispiel für Mining	20
Formel des Zufallszahlengenerators und Erhalten von Hash-Strings	21
1. Der Startwert der Reihe wird generiert	21
2. Die Generierung des nächsten Elements der Serie, GenP[byte], Ergebnis der Generierung	21
3. Herstellung des N-ten Zwischenelements der Hash-Zeichenreihe	21
Vollständige Beseitigung von Blockchain-Kollisionen	22
Gekröpfte Smart Contracts	22
Überprüfung der Knotenintegrität	23
1. Modul zur Überprüfung der Integrität von Blöcken der gesamten Chain	23
2. Modul zur Überprüfung der Integrität der Blöcke der Chain aller eigenen Transaktionen	24
3. Modul zur Überprüfung der Integrität einer lokalen (originalen) ausführbaren Binärdatei	24
4. Einschaltung vom Plugin	25
Herstellung von Hash-String	25
1. Stufe	25
2. Stufe	25
3. Stufe	26
4. Stufe	26
5. Stufe	26
Liste	27

Einleitung

Wir stellen die [IaaSB](#)-Plattform bereit - eine Infrastruktur in Form eines Dienstes, wobei B ein Portfolio ist.

Der Wert sind alle Ereignisse, die eine Person generiert, die durch ihren Wert und Widerstand gegen den unberechtigten Zugriff auf diese Informationen bestimmt sind.

Relictum Pro ist der erste globale Blockchain-Service, der die Möglichkeit bietet, moderne Geräte optimal zukunftssicher zu nutzen.

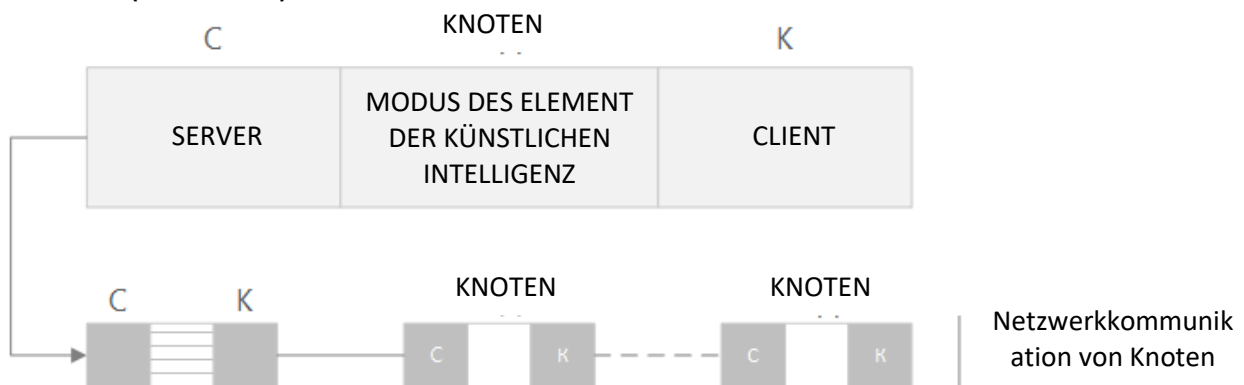
Die Hauptmerkmale umfassen:

1. Unabhängigkeit vom Transport. Das heißt, es ist nicht erforderlich, das Internet zu verwenden. Es ist möglich, WLAN, Bluetooth, Telefonnetzwerke, Glasfaserkommunikation bis hin zur Übertragung von Informationen über ein USB-Speicherstick oder optische, quanten-vielversprechende Netzwerke zu verwenden.
2. Unabhängig von der Art des Clients (Thin Client, Fat Client, Rechenzentren, mobile Drucktastengeräte), auf dem ein Betriebssystem vorhanden ist.
3. TPS liegt in der Nähe der Dicke der Master-Knoten-Kanäle.
4. Blockchain hat eine 100%-ige Verteilung. Für einen begrenzten Zeitraum, innerhalb eines Tages nach dem Betrieb des Netzwerks, werden die Befugnisse bis zu 100.000 Mal an einen der Knoten delegiert, wodurch Entscheidungen für das gesamte Netzwerk getroffen werden. Das Protokoll der Delegierung der Befugnisse ist einem Zufallszahlengenerator zugeordnet, der aus dem Hintergrund der Radioemission abgeleitet ist, da er die Gaußsche Normalverteilungskurve am besten beschreibt.
5. Die Organisation von gekröpftem Smart Contract.
6. Die Verfügbarkeit dynamischer Chains zum Speichern dynamischer Daten, beispielsweise für repetitive logistische gekröpfte Smart Contracts.
7. Die Verwendung der durchgängigen Nummerierung von Chain-Hashes zur garantierten Vermeidung von Kollisionen über einen langen Zeitraum.
8. Verteilter Speicher. Fragmentierung und unterschiedliche Ordnung von digitalen Daten auf Benutzergeräten, die einen Knoten enthalten.

9. Der Algorithmus zum Übertragen von Blöcken über das Netzwerk an jeden Knoten sieht einen vollständigen Zyklus vor, wenn alle Knoten diese Daten empfangen haben: 100% - (minus) Ruhemodus(Sleep) (Formel).
10. Ein Knoten kann den Ruhemodus nur verlassen, wenn die vollständige Initialisierung mit Bestätigung der Bindung an die Hardware, Empfang der Hashes, Überprüfung der Hashes, Überprüfung der Binärdatei mit einer externen Ressource und der vollständigen Chains-Pumpen abgeschlossen ist.

Funktional

Knoten (Portfolio)



ЭИИ - Element der künstlichen Intelligenz.

Ein Knoten besteht aus drei Entitäten:

1. Portfolio.
2. Chains von Blöcken.
3. Verteilter Speicher.

1. Portfolio:

Eine Binärdatei, die die folgenden Elemente enthält:

1. Verwaltung von Knoten.
2. Explorer.
3. Generator von Smart Contracts.
4. Erstellung von IO.
5. Generierung von Tokens.

6. Funktional von Transaktionen jeglicher Art von Währung.
7. Privater Chat.
8. Austausch.
9. Eigendiagnose.
10. Upgrade-Ressource.
11. Störungsbehebung.
12. Bindung an die Hardware.
13. Das Kommunikationsmodul mit einer externen Ressource, um den Ruhemodus zu beenden.
14. API-Initialisierungsmodul.
15. SDK-Initialisierungsmodul.
16. Low-Level-API-Initialisierungsmodul (Socket).
17. Modul für die Arbeit mit verteiltem Speicher.
18. Berechtigungsmodul.
19. Biometrisches Modul. Biohash.
20. Zeitsynchronisationsmodul (Uhr gang).
21. Modul zum Empfangen von Daten von einer externen Quelle der Hintergrundstrahlungsdaten in Form einer dreidimensional normalisierten digitalen Karte.
22. Modul der Status-Blockierung. Es wird verwendet, um den unberechtigten Zugriff auf die Binärdatei und einen Teil der heruntergeladenen Anwendungsdatei zu blockieren.
23. Modus von Tick der Eigendiagnose.
24. Selbstzerstörungsmodul. Wenn kritische Hashes nicht übereinstimmen, wird das eingekapselte Programm in der Anwendung gestartet, wodurch alle Ressourcen einschließlich der Benutzerauthentifizierungsdaten entfernt werden.

2. Chains von Blöcken:

1. Die Chain von Blöcken ist eine n-dimensionale Matrix von Chains von Smart Contracts, die von einem Signaturmechanismus für Hash-Strings mit der obligatorischen Bestätigung und Ordnung jeder Transaktion in der Masterchain gesteuert wird.
2. Die Größe jedes Blocks variiert zwischen 120 und 300 Bytes. Gleichzeitig hat jeder Block eine durchgängige Nummerierung in der n-dimensionalen Matrix.
3. Die Korrelationsinfrastruktur dieser Blöcke ist so organisiert, dass die Datensuche vom Ende bis zum Anfang erfolgt. Dies führt zu einer nahezu



sofortigen Erfassung von Informationen über die Daten in der n-dimensionalen Matrix. Beispiel: Um Listen aller Kryptowährungstransaktionen abzurufen, wenn es sich von dem letzten zu den ersten Abtastwert zu bewegen, wird diese Liste sofort abgerufen, wobei die Zwischenblöcke aufgrund der durchgängigen Nummerierung der Blöcke in der Matrix umgangen werden. Auf diese Weise kann man alle Daten in der Blockchain lokal analysieren; Diagramme erstellen; Wege des Daten-Durchschnitts finden; dynamische Smart Contracts verwalten; interpolierte Datenanalysen berechnen und die Extrapolationsfunktion approximieren, um die Erwartungen für einen ausreichend langen Zeitraum zu berechnen.

4. Das Vorhandensein von dynamischen Blöcken.
5. Jeder Knotenbesitzer kann eine eigene Datenbank für die Interaktion mit dynamischen Blöcken erstellen (Smart Contract DB). Beispielsweise können Kontrahentenprofile usw. in einer solchen Datenbank gespeichert werden. Dynamische Smart Contracts können auch als gekröpfte Smart Contracts verwendet werden.

3. Verteilter Speicher:

1. Sie benötigen einen Smart Contract über die verteilte Speicherung und eine Bestätigung des Besitzers des Portfolios, um die Ressourcen seines Computers nutzen zu können.
2. Die mindestens zugewiesene Datenmenge beträgt 200 MB. Maximum – unbegrenzt.
3. Durch die Initialisierung des verteilten Speichers im Portfolio kann der Besitzer anhand der Anzahl der Anrufe bei der Datenbank des verteilten Speichers Geld verdienen.
4. Verteilter Speicher empfängt Daten wie sie sind (in Form eines Arrays von Bytes) und kontrolliert den Inhalt der Datenbank in keiner Weise. Und für den Inhalt der Datenbank sorgt ein Knoten, der differenzierte Daten des Speichers im gesamten Netzwerk ablegt.
5. Jedes Datenelement kann eine andere Länge haben, von 1 Byte bis zum Maximalwert des Computerregisters (2^{64}).
6. Informationen zu einem Teil von Bytes werden in einem Smart Contract (Chain) mit verteiltem Speicher gespeichert.
7. Das Duplizieren von Bytesequenzteilen erfolgt wiederholt auf verschiedenen Knoten, um Datenverlust zu vermeiden.

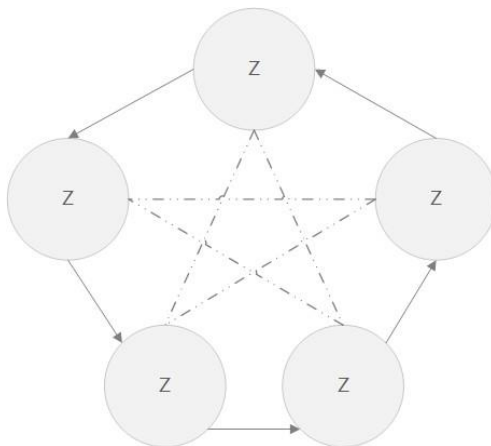
8. Der Besitzer der im verteilten Speicher abgelegten Daten kann die Datei vollständig in seiner eigenen Datenbank in seinem Portfolio ablegen, und zwar sowohl Cryptodatenbank als auch gemeinsam genutzte Datenbank.
9. Die im Speicher abgelegten Daten haben den Status von den offenen Daten zu Mosaik-Mischteilen. Die Daten können Share- oder Private-Share haben.
10. Beim Initialisieren eines neuen Dokuments wird es geprüft, ob die Daten objektiv und zuverlässig sind, um Spam auszuschließen.
11. Das Speichern von Daten in einem verteilten Speicher ist für das Ausschließen von Spam kostenpflichtig.

Prinzipien der Vernetzung. Netzwerkprotokoll:

1. Das Netzwerk hat 5 Stufen (Levels):

1. König.
2. Z level - Generäle.
3. P level - üblicher Knoten.
4. L level - lichter Knoten.
5. S level - privater Knoten.
6. O level - Knoten von Ruhemodus.
7. Cloud-Knoten

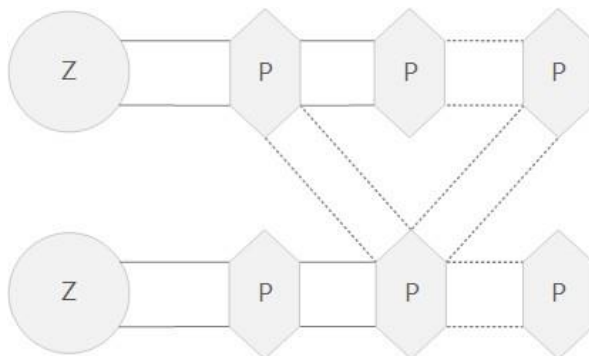
2. Z-level (Generäle) (Zero-Level)



1) Einer von ihnen ist König, aber es ist unbekannt.

2) Alle Z arbeiten wie ein einzelner Computer mit mehreren Prozessoren.

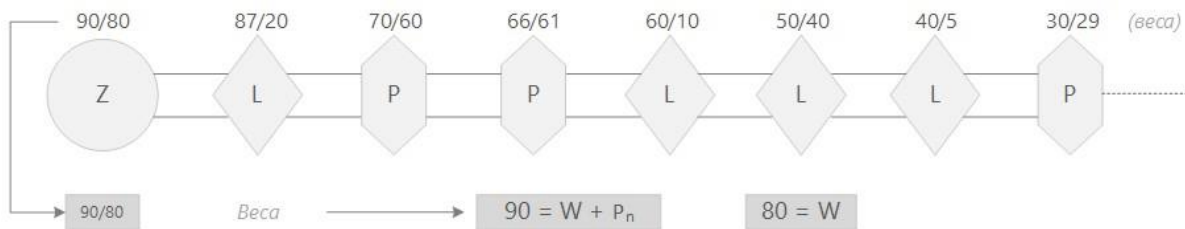
3. P-level (Power level) Knoten von Stufe 1



Chains Z-P.

VK-Transport
(Vermittlungskanäle).

4. L-level (lichter Knoten) von Stufe 2



wo

W - aktuelles Knotengewicht

P_n - Formelergebnis

$P_n = G_p [R]$

wo

G_p - Zufallszahlengenerator

R - Probenahme von Relikt normalisiert auf den Bereich (1-100%)

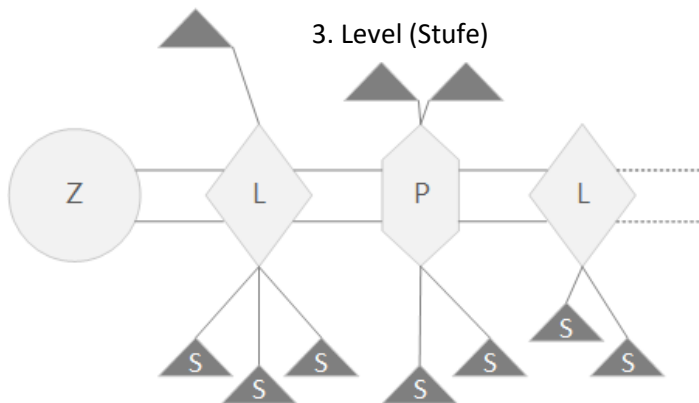
Gewichtsformel (aktuelles + dynamisches)

$$W_d = W + G_p [R_n]$$

$G_p \rightarrow$ Nummer im Array R_n

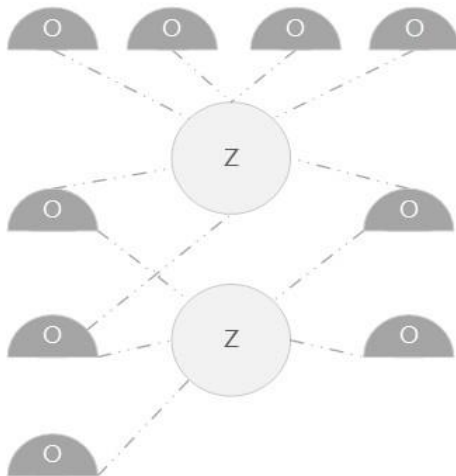
wo n - Netzwerk-Regenerationsnummer.

5. S (Self-Knoten) = privater Knoten









1. Eigene Transaktionen
2. 2. Ausgewählte Knoten (nur ihre Transaktionen)
3. Speicherung verteilter Speicherdaten

6. O - (Offline-Knoten) = Knoten von Ruhemodus



1. Versuch einer Verbindung mit dem Z-Knoten.
2. Abrufen der Adresse der Verbindung (IP).
3. Überprüfung der Integrität der lokalen Ressource.
4. Überprüfung der Hardware.
5. Synchronisierung von Hashes.

Funktionstabelle der Knotentypen

0	1	2 W IP	3	4	5	6	7	8
0		+	+	+	+	+	+	-
1		+	+	+	+	+	+	-
2		±	±	+	+	+	+	-
3		-	±	+	+	+	+	-
4		±	-	-	-	-	-	+
5		-	-	-	+	-	-	+

0. Nummer

1. Thumbs

2. White IP

3. General werden

4. König werden

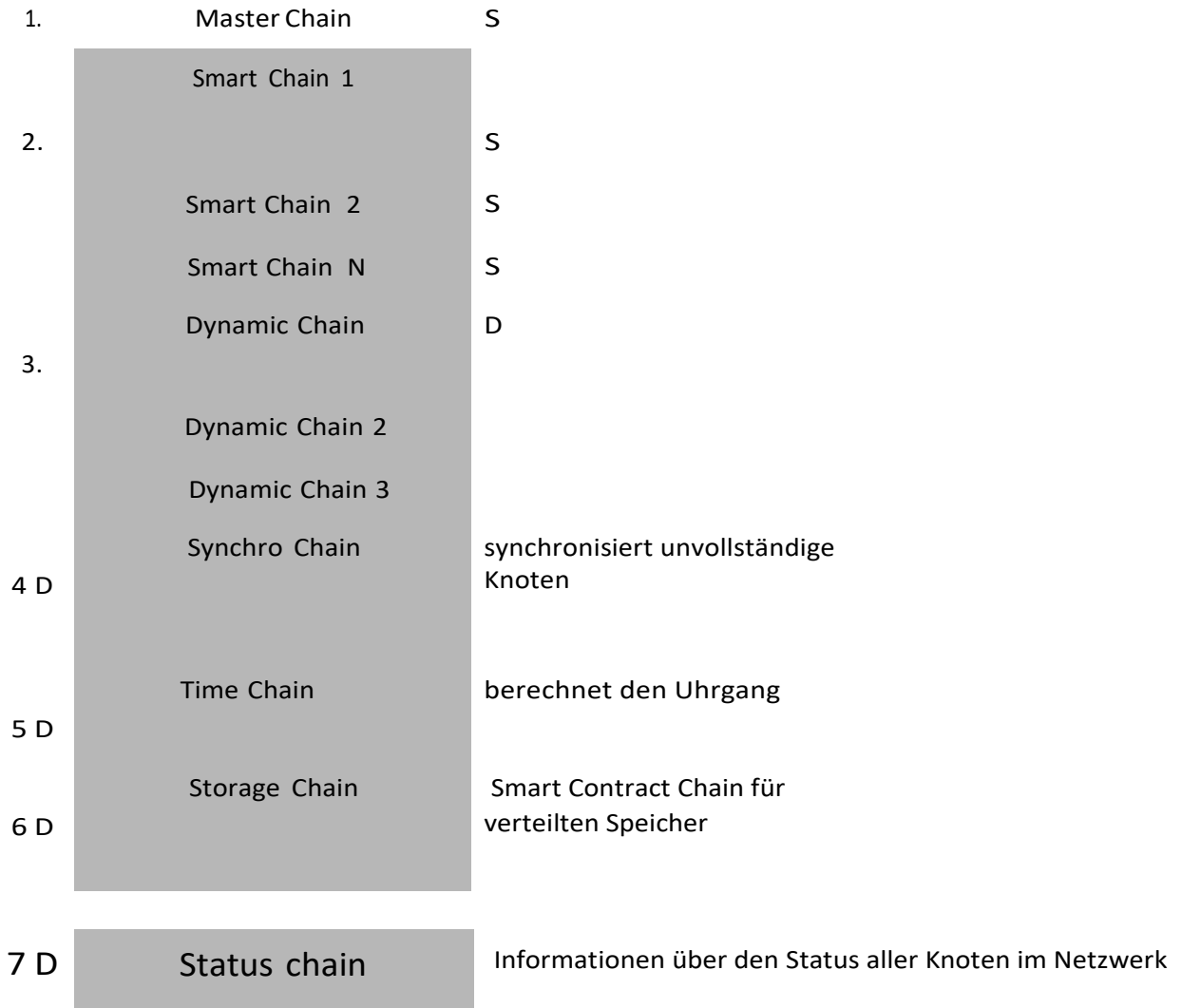
5. Initialisieren der Transaktion

6. Teilnahme an der Abstimmung

7. Ergebnis

8. Bewertungseinschränkung

Architektur von Chains



S - Standardknoten

D - dynamischer Knoten

Anwendungsgebiet

Recht:

Wert, Skalierbarkeit, Widerstandsfähigkeit gegen Inflation und Anreize zu deren Nutzung.

Versicherung:

Optimierung des Zahlungsmodells für Versicherungsbeiträge
Reduzierung oder Beseitigung betrügerischer Risiken
Automatische Zahlung von Versicherungsbeiträgen mit Smart Contracts
Verbesserung der Qualität der Kundenbetreuung

Finanzen:

Verbesserter Mechanismus zur Identifizierung der Identität
Verringerung der Risiken zwischen den Kontrahenten
Volle Transparenz des Leistungserbringungsprozesses
Schnelle grenzüberschreitende Transaktionen

Medizin:

Vereinfachung der elektronischen medizinischen Aufzeichnungen
Effiziente Erfassung und Verwaltung von Daten von Medizinprodukten
Transparenz der Verfolgung der Arzneimittelversorgungskette
Modernisierung der Krankenversicherung
Mehrfach gekröpfter Smart Contract (Lieferung)

Logistik:

Effektive Frachtverfolgung, die die Wahrscheinlichkeit von Verlusten verringert
Verkürzte Lieferzeit
Transparenz logistischer Informationen
Erhöhung der Kapazität der Lieferkette

Probleme mit dem Ticketmarkt:

Fälschungen
Schleichhandel

Kohlenwasserstoffe:

Die Blockchain-Technologie bietet der Öl- und Gasindustrie die Möglichkeit, Probleme mit grenzüberschreitendem Transport, großem Transaktionsvolumen und komplexem Dokumentenfluss zu lösen.

Token und seine Klassifizierung

Token als digitale Verpflichtung kann wie folgt klassifiziert werden:

Zahlungsmittel:

Es ist analog zu traditionellem Bargeld.

Utility-Token:

Es wird als „Kraftstoff“ im Blockchain-System verwendet und gewährleistet dessen Funktionsfähigkeit.

Digitales Anlagevermögen:

Ein digitaler Vermögenswert, der in der realen, nicht digitalen Welt rechtlich abgesichert ist.

Tokenized Asset:

Digitale Verpflichtung für den Austausch gegen ein echtes Produkt oder eine echte Dienstleistung.

Tokenomics:

Eine Reihe wirtschaftlicher Regeln und Modelle, die das Funktionieren der Projektwirtschaft auf der Grundlage von Tokens sicherstellen.

Wirtschaftsmodell:

Formalisierte Beschreibung verschiedener wirtschaftlicher Phänomene und Prozesse.

Systemarchitektur (Architekten)

AGT (Algorithmic Game Theory) wird als Systemarchitekt verwendet, der den bedingungslosen Abschluss von Transaktionen vor dem Start der nächsten Transaktion sicherstellt. Es werden auch Proof-of-Relay und Proof-of-Utility verwendet - Arbeitsnachweise und Utility-Nachweise, die zu einer



bedingungslosen Monetarisierung der Verwendung von Knoten gemäß dem Gesetz der Spieltheorie (Game Theory) führen.

Hauptphasen des Netzwerks

Regenerationsmodus (Änderungen der TOPOLOGIE, Machtwechsel).
HERSTELLUNG DER TOPOLOGIE.

Die Regeneration erfolgt immer oder auf Abruf zur Ablesung oder Aufzeichnung (am Ende des Abrufs zur Ablesung oder Aufzeichnung). Wenn sich das Netzwerk im Leerlauf befindet, wird nach Timeout eine Regeneration durchgeführt.

In diesem Modus stoppt das Netzwerk die Verarbeitung von Abrufen (ABLESUNG, AUFZEICHNUNG) und Netzwerknoten (KNOTEN) analysiert die Arbeit des vorherigen Zyklus - die Verarbeitungszeit von Blöcken, die Zugriffsgeschwindigkeit (GESCHWINDIGKEIT DER KOMMUNIKATIONSKANÄLE), die Anzahl der verarbeiteten Transaktionen. Basierend auf den Ergebnissen dieser Analyse wird eine eindeutige Netzwerktopologie (Hierarchie, Bewertung) für den nächsten Zyklus erstellt. Die Wahl des Königs erfolgt zufällig auf der Grundlage des RCS-Algorithmus (Random Circle Stop, ein EINDEUTIGER Algorithmus) unter den Generälen zu Beginn des Regenerationsmodus. Der RCS-Algorithmus ist ein Spiel wie „Schere, Stein, Papier“. Während des zugewiesenen Zeitraums erstellen die KNOTEN sequentielle Berechnungen (WOBEI ZUFÄLLIGE DATEN AUS DER AKTUELLEN TABELLE DER HINTERGRUNDSTRAHLUNG ABGERUFEN WERDEN). Nach der festgelegten Zeit wird der König ausgewählt.

Die Wahl der GENERÄLE erfolgt aus den SPITZEN DER KNOTEN, DIE DURCH DIE FORMEL (S. 4) $Wd=W+Gp[Rn]$ erhalten werden, deren Anzahl von der Netzwerklast abhängt. In jedem Zyklus werden die Generäle aktualisiert. (Es ist nicht möglich, für zwei Zyklen hintereinander ein General zu sein).

Der König berechnet (überprüft, genehmigt) jeden neuen BLOCK und initiiert Transaktionen, die von den Generälen im vorherigen Zyklus aggregiert wurden. Nachdem die Generäle jeden Block überprüft haben, werden die Blöcke über das Netzwerk an alle Knoten nach unten übertragen.

Die neue Netzwerktopologie (Hierarchie) wird am Ende des aktuellen Zyklus vom König berechnet und genehmigt und von KNOTEN zu KNOTEN im Netz entsprechend der Topologie jedes Generals in diesem Zyklus für den entsprechenden Zweig verteilt.

Auf dieser Plattform mit automatischer Datenverarbeitung gibt es keine Demokratie.

Kryptoalgorithmus

Es handelt sich um einen SHA-1-Algorithmus mit anschließender Konvertierung in die Proprietary Jump-Methode, bei der der nicht diagnostizierte Fehler Overflow für Einweg-Hashing verwendet wird.

Aufzeichnungsart

Jeder Knoten, der Informationen zur Aufzeichnung in die BLOCKCHAIN empfängt, leitet diese über das Netzwerk NACH OBEN. Der KÖNIG bildet aus den erhaltenen Informationen einen Block, zeichnet ihn in die Blockchain auf und übergibt die fertigen Blöcke an alle Generäle.

BLOCKCHAIN-NETZWERKZIELE

- Jeder Knoten kann von überall aus eine Verbindung zu diesem vollständig offenen Netzwerk herstellen.
 - Die Netzwerktopologie ermöglicht eine wirksame Netzwerkfreigabe.
 - Sichere Netzwerkneutralität durch Innovation auf Netzwerkebene.
 - Immer offen und skalierbar.
 - Automatisches wirksames und dynamisches Routing.
 - Tokenization-Mechanismus (Automatisierung des Berechnungsprozesses mithilfe des Token Accounting-Mechanismus als Gebühr für die Wartung und Erweiterung sowie die Optimierung des Netzwerks und der Netzwerkverbindungen, der Datenübertragungsressourcen und die Stimulierung der teilnehmenden Knoten.
- Entwerfen und erstellen das Blockchain-Netzwerk der nächsten Generation.

Klärgrube (Funktionen)

Die KLÄRGRUBE für Knoten (Ruhemodus), die nicht mit der Netzwerkleistung fertig werden oder aus irgendeinem Grund die Verbindung unterbrechen.



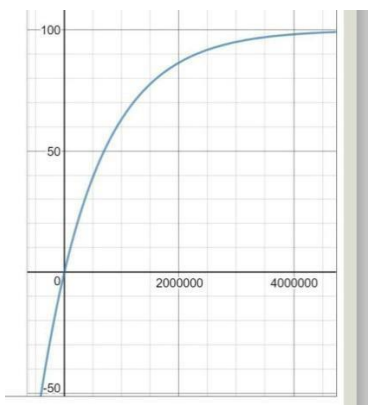
RÜCKKEHR des Knotens AUS DEM KLÄRGRUBE.
DISKREDITIERUNG DES KNOTENS (SUCHE NACH JUDAS).

Formeln und Umsetzung

Die Funktion, die das Wachstum des Gewichts (der Kosten) von Relict Coin im Unendlichen verlangsamt und es ihm nicht ermöglicht, die voreingestellten maximalen Kosten (Emission, Gewicht ...) = 100% zu überschreiten.

Die Formel für den Anstieg der Relict Coin-Kosten auf 100 Einheiten (%) am Beispiel von Smart Contract von Proof-of-Time-Mining:

$$y = -100e^{(-0.00082x)} + 100$$



Unten gibt es den Link zum Überprüfen des Formelgraphen:

http://www.yotx.ru/#11/3_h/ubWwcH@2cHBwf7Rgzhf23/aP9g/2DfT0qt7W9uHRysrW9sHkAODg5gO3u70K2Dg/2DfRiNu7Fzyng83WI8bl1e705v7QMG

Beispiel für Mining

$$\text{TicS} = \text{TicL} + \text{TicD}$$

$$\text{AmC} = -100e^{(-0.00082\text{TicS})} + 100$$

$$\text{AmD} = \text{AmC} - \text{AmL}$$

$$\text{AmD} = \text{SUM}(\text{APn})$$

$$\text{TicD} = \text{SUM}(\text{TDn})$$

$$\text{APn} = (\text{TDn} / \text{TicD}) * \text{Amd}$$

wo:

1 Tic = 24Hour / 24 Stunden

AmL - Last Amount / Letzter Betrag

AmC - Current Amount / Aktueller Betrag



AmD - delta - SUM aller Coins für den Zeitraum {TickL->TikC}

APn - Der Betrag von jedem für den Zeitraum

ADn - delta TicL - Last Tick / Letzter Tick

TDn - delta Tick Nod(jeder Knoten) TSn - Sum All Ticks Nod / Summe aller Ticks Knoten

TicD - delta - SUM aller Ticks für den Zeitraum {TickL->TikC}

Formel des Zufallszahlengenerators und Erhalten von Hash-Strings

1. Der Startwert der Reihe wird generiert:
ep=15436757;
ep=abs(sin(ep));
ep=ep*429496729;
2. Die Generierung des nächsten Elements der Serie, GenP[byte]-Ergebnis der Generierung:
gran=255;
ep=abs(sin(ep));
ep=ln(7)/ln(ep);
ep=ep*10;
ep=ep-trunc(ep);
GenP=trunc(ep*gran);
3. Herstellung des N-ten Zwischenelements der Hash-Zeichenreihe s[byte]. s-Zeichenreihe. Hash [255]- Byte-Array. z-Nummer des Array-Elements [Hesh]. i - Elementnummer des Array-Strings[s]

Im Fall von $\text{length}(s) < \text{Ingth}((\text{Hash}))$ die Werte von 0 [byte] werden zu s addiert, um die Bedingung $\text{length}(s) = \text{Ingth}((\text{Hash}))$ zu erfüllen.

Zyklus

inc(z);

inc(Hash[z], GenP(ep) + ord(s[i]));

Im Fall von $z > \text{Ingth}((\text{Hash}))$, 1 ist z zugeordnet, was zu einer Modifikation des z-ten (Zwischen-) Hash-Element führt.

Die Werte des Zufallszahlengenerators werden verwendet, um absolut zufällige Werte aus der Gewichtungsmatrix der Hintergrundfunkemission zu erhalten.

Vollständige Beseitigung von Blockchain-Kollisionen

Im Hash werden die ersten 4 Bytes durch den Count-Wert der aktuellen Nummer des MasterChain-Blocks ersetzt.

Dies bedeutet, dass bei 2147483647 Transaktionen Kollisionen ausgeschlossen sind.

Für jede Chain eines Smart Contracts wird 1 Token (für jeden Besitzer) mit einem Nennwert von N (max. 9223372036854775807) erstellt.

Gekröpte Smart Contracts

Die Methode zur Berechnung des optimalen gekröpten Smart Contracts (Tokenomics).

Berechnung eines gekröpten Smart Contracts mit Zwischenereignissen (Checkpoints).

Parameter von Smart Contracts:

L - Länge (Meter, Glieder, Parsec, Wellenschwingungen (Anzahl der Perioden), Schritte usw.)

T(L) - Zeitaufwand für den Durchlauf der gesamten Länge unter Berücksichtigung von Haltestellen an den Checkpoints

Ln - Länge des n-ten Abschnitts (Checkpoint)

Tn - Zeit des n-ten Abschnitts

Rn - Bewertung (Zuverlässigkeit des n-ten Punktes)

Gn - Ergebnis eines Zufallszahlengenerators für Bewertungsänderung am n-ten Punkt

Wn - Einfluss externer Parameter auf jede Stufe

Gn - Zuverlässigkeitsfaktor von Checkpoint.

Sn - der berechnete Wert der Geschwindigkeit der Überwindung von Ln. Es ist notwendig, die Korrelationen mit anderen Ereignissen früherer Blöcke zu berechnen und die Extrapolation zukünftiger Ereignisse und Berechnung der Zuverlässigkeit der Knie zu berechnen, um eine andere Flugbahn zu wählen, zum Beispiel, im Fall von Arbeitsbelastung von Checkpoints (Kreuzungsproblem).

Z - Zuverlässigkeit auf ganzer Linie für einen Block

$$T(L) = \sum(T_n * R_n * G_n * W_n * G_n)$$

$$T_n = T_n * R_n * G_n * W_n * G_n$$

$$S_n = L_n / T_n$$

$$S = S_n / n$$

$$Z = S / L ; \text{ bei } Z=1 \text{ (perfekte Zuverlässigkeit).}$$

Es ist möglich auch ein schwaches Knie und andere Parameter für die Analyse berechnen.

Mit der Anzahl der Blöcke > 17 verwenden wir die Methode der kleinsten Module und können einen solchen Smart Contract als ein Element der II (Selbststudium + Selbstanalyse) betrachten.

Beispielsweise können damit Ampeln gesteuert werden.

Überprüfung der Knotenintegrität

1. Modul zur Überprüfung der Integrität von Blöcken der gesamten Chain.

Formel:

Überprüfung der Erhaltung des Zustands in der gesamten Chain ab dem 2. Element

$$h(B_{n-1}) = H_n \text{ bei } n[1..BlockCount-1]$$

wo,

h - Block-Hash

H_n - Hash in Block

B_n - Block N

Wenn die Bedingung nicht erfüllt ist, wird die Chain als beschädigt betrachtet und ein Neustartereignis tritt für die gesamte Chain auf.

2. Modul zur Überprüfung der Integrität der Blöcke der Chain aller eigenen Transaktionen.

Formel:

Überprüfung der Erhaltung des Zustands in der gesamten Chain ab dem 2. Element

$$h(B_{n-1})=H_n$$

wo,

h - Block-Hash

H_n - Hash in Block

B_n - Block N

Wenn die Bedingung nicht erfüllt ist, wird die Chain als beschädigt betrachtet und ein Neustartereignis tritt für die gesamte Chain auf.

3. Modul zur Überprüfung der Integrität einer lokalen (originalen) ausführbaren Binärdatei.

- A1 Überprüfung des Namens (extern)
- A2 Überprüfung des Erstellungsdatums (extern)
- A3 Überprüfung des Startorts (lokal)
- A4 Größenüberprüfung(extern)
- A5 Überprüfung von Hash der Datei selbst (extern)
- A6 Überprüfung der Bindung an die Hardware (lokal)

Formel:

$$h(A1,A2,A4,A5)$$

wo,

h - Hash der Konkatenation von Variablen A

A - Zeichenfolgenvariable vom string-Typ

Das Ergebnis h erfordert auf Abruf des Smart Contracts eine Bestätigung der Integrität der eindeutigen Dateien über ein Netzwerk anderer Knoten

Wenn die min (9)-Bestätigungsbedingung nicht erfüllt ist, werden die Knoten als beschädigt betrachtet, was dazu führt, dass die Anwendung vollständig desavouiert wird und dem Netzwerk der Dienst verweigert wird.

4. Beim Starten einer ausführbaren Binärdatei wird ein Plugin geladen und gestartet, das separat vom Hauptprogramm funktioniert. In diesem Fall wartet das Hauptprogramm auf die Bestätigung der Authentifizierung (Warten auf den Sitzungsschlüssel).

Um den Sitzungsschlüssel zu erhalten, führt das Plugin Schritt 3 aus und sendet $h(A1, A2, A4, A5)$ an das Netzwerk. Nachdem das Netzwerk die min (9)-Bestätigung der dynamischen Chain von zufälligen Knoten überprüft hat, sendet es einen neuen Sitzungsschlüssel an das Hauptprogramm. Der Sitzungsschlüssel wird auch in die dynamische Chain aller Knoten aufgezeichnet.

Somit wechselt der Status des Knotens von SLEEP (Ruhemodus) zu ONLINE.

- Das Plugin selbst ist eine eindeutige Datei, die vom Netzwerk (auf Abruf) beim Start des Hauptprogramms generiert wird.
- Beim Start überprüft das Plugin die Zeitsynchronisation des Netzwerks und des Geräts, auf dem es ausgeführt wird.
- Das Plugin funktioniert 1-2 Sekunden lang. Wenn er in dieser Zeit kein h gesendet hat($A1, A2, A4, A5$), werden alle Programme versehentlich geschlossen.
- Wenn Sie dasselbe Plugin neu starten, reagiert das Netzwerk mit einem Fehler, da das Plugin nicht mehr eindeutig ist.
- Die Eindeutigkeit des Plugins wird durch den Hash der Plugin-Datei bestimmt, der in der dynamischen Chain aller Masterknoten aufgezeichnet ist.

Herstellung von Hash-String

Herstellung von Hash-String $f(X(L))$ (Hash) von String L

1. **Stufe** - Standardfunktion sha1 wird berechnet (mit geringfügigen Änderungen), als Ergebnis erhalten wir $f(X(L))$ mit Dimensionsgrenzen von 20 Bytes.
2. **Stufe** - Konvertierung von $f(X(L))$ in eine 32-Byte-Zahl unter Verwendung der byteseriellen Verschiebung mit Summierung

Byteserielle Verschiebung mit Summierung:

```
Type Tsb32=array[1..32+1] of byte;  
for i:=1 to HashLength do sb[i]:=sb[i]+sb[i+1]+i*i;
```

wo $f(X(L))=sb$

In diesem Fall wird eine Hardware-vorbeugende Blockierung des Überlauffehlers der Prozessorzelle verwendet.

Beispiel:

```
var a,b:byte;  
a:=255;  
b:=1;  
a:=a+b;  
Ergebnis: a=0;
```

Solche Operation spart erheblich Ressourcen, zusätzliche Überprüfungen und Berechnungen sind nicht erforderlich.

3. Stufe

Ein paar Sprünge über die Barriere 255

```
inc(b,sb[i]  
inc(sb[i],b)  
wo b Byte-Zelle ist
```

4. Stufe

Die byteserielle Additionsfunktion des resultierenden Hash $f(X(L))$ mit der Quellzeichenfolge L wird verwendet.

```
inc(sb[k],ord(s[i]));  
wo k(1..32), i(1..length(L))
```

5. Stufe

Es ist notwendig, Stufe 3 auszuführen, um die Rückwirkung in Stufe 4 zu beseitigen.

Liste

(Pascal)

```
function ShaM_Bin(s:string):THash;
const HashLength = 32;
    MagicByte =*****;
var i,k,L:integer;b:byte;
    sb:Tsb32;
    SHA1Digest:TSHA1Digest;
begin
    SHA1(s,SHA1Digest);
    k:=1;
    L:=20;
    for i:=1 to L do sb[i]:=SHA1Digest[i-1];
    for i:=L+1 to HashLength do sb[i]:=$0;
    for i:=HashLength-L to HashLength do inc(sb[i],SHA1Digest[i-HashLength-L]);
    *****=MagicByte;
    for i:=1 to HashLength do sb[i]:=sb[i]+sb[i+1]+i*i;
    b:=0;
    for k:=1 to 4 do begin
        for i:=1 to HashLength do inc(b,sb[i]);
        for i:=1 to HashLength do inc(sb[i],b);
    end;
    k:=1;
    L:=length(s);
    for i:=1 to L do begin
        inc(sb[k],ord(s[i]));
        inc(k);if k>HashLength then k:=1;
    end;
    for i:=1 to HashLength do ShaM_Bin[i]:=sb[i];
    //result:=SHAMDigestToHex(sb);
end;
```