Technisch White Paper

(Sommige details kunnen toegevoegd worden)

Gedecentraliseerd financieel systeem

CREDITS

Versie 1.5/12.09.2017

Inhoud

Algemeen

CREDITS platform is een gedecentraliseerd financieel system voor de directe interactie tussen participanten op basis van peer-to-peer (P2P) principes. Het platform breidt het potentieel uit van bestaande financiële diensten op basis van gedistribueerde zelfuitvoerende grootboek- smart contracts (slimme contracten) en CREDITS cryptogeld. Het systeem is erop gericht om alle participanten te verenigen op één site, hen voorziend van een platform voor het maken en gebruiken van financiële diensten waar iedereen zowel een dienst kan aanbieden en gebruiken. Dankzij een goed gedefinieerd en gebalanceerd systeem biedt het CREDITS platform een nieuwe technologische oplossing en een nieuw conceptueel model van het netwerken van de interactie van participanten voor de ontwikkeling van moderne gedecentraliseerde financiële diensten.

Introductie

Een volledig peer-to-peer arrangement voor dienstleverende systemen dat het vormen toestaat van financiële diensten: geldoverboekingen, geldeenheid- en waarde-omwisselingen, creditering, fondswerving en andere diensten direct tussen participanten. Alles wordt geleverd zonder additionele bemiddelaars, volgens een principe – één van de gelijke participanten – naar andere systeemparticipanten. Als resultaat krijg iedereen goedkopere, snellere en betere diensten.

De wereld beweegt naar de directe interactie tussen mensen op basis van peer-to-peer principes – gelijke naar gelijke. Een revolutie vond plaats! Dit wordt duidelijk gezien door de wenteling an de massamedia: tot de 1990's waren kranten, magazines en TV de belangrijkste voorzieners van informatie. Vandaag zijn opiniemakers bloggers, gevonden op Youtube kanalen en social netwerken, geld wordt geïnvesteerd in crowdfunding en ICO, en informatie wordt opgeslagen in gedecentraliseerde cloud-systemen.

De financiële industrie is, misschien, een van de weinige industrieën die achterloopt, die de introductie van decentralisatie en directe interactie tussen participanten afweert. Hoewel het technisch veel gemakkelijker is om gedecentraliseerde financiële diensten te creëren dan het is om onbemande voortuigen te creëren.

Een corresponderend technologische omgeving is nodig om een systeem te creëren van gedecentraliseerde financiële producten en diensten gebaseerd op een verdeeld grootboek:

1. Hoge uitvoeringssnelhuid (in seconden), samen met de mogelijkheid om een enorme hoeveelheid van transacties tegelijkertijd af te handelen (honderdduizenden per seconde) met lage kosten voor iedere transactie (voor microbetalingen en non-contante transacties).
2. Ontwikkeling van een systeem waar alle participanten en voorwerpen, noodzakelijk voor de kwalitatieve gedecentraliseerde diensten, worden gecombineerd: personalisatie van gebruikers, KYC (=Ken Je Klant), kredietgeschiedenisbureau, vestigingscentrum van fiduciair geld, opname en contant maken van cryptogeldeenheden, enzovoort.

Dit zijn twee grote en basale taken die heden de ontwikkeling van peer-to-peer financiële producten tegenhouden.

We presenteren je een oplossing voor deze taken, welke we implementeerden met behulp van het CREDITS financiële systeem.

Het CREDITS enkele technologisch gedecentraliseerde platform kan alle participanten van financiele diensten combineren en veilig en snel alle transacties uitvoeren gebruikmakend van de principes van een gedistribueerd grootboek. Zelfuitvoerende smart contracts (slimme contracten) en de principes van een federatief stemsysteem voorzien in ongelimiteerde mogelijkheden voor alle participanten om unieke interacties te creëren van diverse financiële producten. Het platform opent een nieuwe enorme markt en een nieuw potentieel voor het gebruik van blockchainprojecten en –diensten in financiële en andere sectoren die voorheen niet gebruikt konden worden vanwege limitatie van snelheid en transactiekosten.

1. Netwerkgrootboek

Definities

1. Een system dat is opgebouwd uit gedecentraliseerde netwerknodes [-knooppunten] die processen uitvoeren, transacties op slaan, de voorwaarden van smart contracts confirmeren en uitvoeren, aanvragen verwerken van derde-partij-systemen, informatiedata geven wanneer gevraagd.

2. Een netwerknode is een computer waar een complete netwerkcliënt is geïnstalleerd, verbonden aan een gemeenschappelijk system, transacties verifiërend en deze schrijvend naar het grootboek.

3. Een grootboek is de lijst van transacties geconfirmeerd door het system en opgeslagen op alle netwerknodes.

4. Een transactie is het systeemvoorwerp, een verzoek aanduidend om een smart contractmethode uit te voeren of iedere actie op het netwerk en de resultaten opslaand in een blockchainsysteem.

5. Een smart contract (slim contract) is het systeemvoorwerp, computerprotocollen die het nakomen van de voorwaarden van interactie verifiëren of verzekeren. Ze hebben normaliter een gebruikersinterface en emuleren vaak de logische of contractuele relaties. De sleuteleigenschap van een smart contract is zijn decentralisatie en zijn onafhankelijkheid van een centrale bron.

6. Een smart contractmethode is de programmeringscode die verantwoordelijk is voor het berekenen van de resultaten van werk of de smart contractvoorwaarden en het opnemen daarvan in het grootboek.

7. Een contracterende partij is de uiteindelijke netwerkparticipanten en de systeemgebruiker.

Netwerknodes

We gebruiken diverse typen nodes, afhankelijk van hun doel om een gedecentraliseerd netwerk te bouwen gebaseerd op vrije toegang en node verbinding:

1. Een algemene node (ОУ) is de node participerend in transactieverificatie voor validiteit maar heeft een minimum vertrouwensfactor. Hij is ook kandidaat voor de rol van een vertrouwde node en de node van het huidige verwerken in de volgende cyclus van rol selectie in het netwerk.

2. Een vertrouwde node (ДУ) is de node die participeert in transactieverificatie en heeft de maximum vertrouwensfactor (1), is een kandidaat voor de rol van de node van de huidige verwerkende en algemene node. Deze node kan niet vertrouwd worden gedurende het mathematisch gecalculeerde aantal van selectie- en stemcycli onder nodes. De mathematische calculatie hangt af van het aantal nodes en de netwerkcomplexiteit.

3. De hoofdnode (ГУ) van het netwerk is de node participerend in verificatie en is verantwoordelijk voor het toevoegen van transacties aan de transactiegrootboekblock. Deze node kan niet vertrouwd worden of de node van de huidige verwerking gedurende een mathematisch gecalculeerd aantal stemcycli, waarvan de mathematische calculatie afhang van het aantal nodes en de netwerkcomplexiteit. Het systeem gebruikt een vertrouwensfactor – een absoluut gebroken numerieke waarde tussen 0 en 1, uitgedrukt in mathematische termen van het aantal van vertrouwde nodes +1 gedeeld door het totaal aantal van nodes in het netwerk. Het maximum aantal vertrouwde nodes kan 50% van de netwerknodes niet overschrijden.

De Laatst Opgeslagen Block

Het Algemene Grootboek van Bocks (CRB) [The Common ledger of Blocks] is de gesynchroniseerde staat van het totale algemene grootboek van blocks in alle systeemnodes.

Met de grootboekblockinhouden bedoelen we een eenheid van opgeslagen informatie die een hashcode bevat van het vorige block en een lijst van data gerelateerd aan het grootboek met het geassocieerde aantal van het vorige block. Bij ontvangst van het block van een andere node neemt het plaats in het algemene grootboek van blocks overeenstemmend met het nummer. Dit bespaart netwerkbandbreedte.

Gedurende de synchronisatie wordt alleen het blocknummer eerst gecheckt. Als de block mist bij deze node wordt deze gedownload en bewaard.

Als resultaat bevat het systeem op ieder moment de laatste up-to-date kopie van het grootboek. We noemen het het laatste grootboek (LR) [last ledger]. Het wordt automatisch gecreëerd door de node die verantwoordelijk is

voor de grootboekformatie wanneer deze consensus bereikt. Dit block wordt verzonden naar alle systeemnodes om de up-to-date uniformiteit van de grootboekstaat te behouden in alle systeemnodes.

Iedere node wordt geassocieerd met alle andere nodes in het netwerk en wisselt constant nieuwe blocks met transacties met deze uit, om altijd de relevante informatie te behouden. Alle blocks vormen een set van transactiekandidaten wachtend om toegevoegd te worden aan het grootboek. Tegelijkertijd genereert iedere server aangenomen sets van de kandidaten voor andere servers en de voorgestelde set van transacties. Een beslissing wordt gemaakt bij het checken of ze aan het grootboek worden toegevoegd.

Als resultaat is het mogelijk om de grootboekdata meerdere keren op te slaan op meerdere servers – de systeemnodes en alle informatie is beschermd. Hoe meer nodes in het systeem hoe meer betrouwbaar en onafhankelijk dit is.

Synchronisatie van Nodes

Iedere nieuwe node wordt gelanceerd en gesynchroniseerd na determinatiedefenitie en grondige betrouwbaarheidsverificatie. Om de informatiebewerkingsratio te verbeteren worden alle processen tegelijkertijd behandeld, onafhankelijk van elkaar. Als er geen inkomende variabelen zijn wordt een lege grootboekopslagplaats gecreëerd – een ruimte wordt gereserveerd in RAM voor verdere vereenvoudigde toegang. In het geval het benodigde grootboek niet beschikbaar is wordt een verzoek verzonden aan vertrouwde nodes om alle transacties gemaakt voor de gesynchroniseerde account te ontvangen.

Als de invoerparameter een object is welk de transactie karakteriseert dan wordt het zoeken in alle lopende gesynchroniseerde draden gestart. De bewerking resulteert in een numerieke code – het positienummer in het vertrouwde nodegrootboek voor de komende draad of het foutnummer als de waarde minder dan nul is. Als de draadmethode afsluit met een verbindingsfout eindigt de draad volledig.

2. Netwerkconsensus

De consensus in CREDITS is een methode van groepsbeslissing. Met het doel een eindoplossing te ontwikkelen die aanvaardbaar is voor alle netwerknodes.

Consensusvergelijking

De definitie van het principe van het gedecentraliseerde CREDITS grootboek om verschillende typen van consensus te vergelijken:

- Grootboekbeschikbaarheid (nodes kunnen data schrijven naar het grootboek en deze op ieder tijdstip lezen);
- Aanpasbaarheid bij alle participerende netwerknodes;
- Consistentie van alle systeemnodes (alle nodes zien een absoluut identieke versie van het grootboek, welk na veranderingen wordt geüpdatet;
- Weerstand tegen afscheiding (als één node onbedienbaar wordt beïnvloed dit de werking van het gehele grootboek niet).

Vergeleken parameter	Kredietspecifieke PoW en PoC	PoW	PoS
Het principe van het identificeren van de node die de block genereerde.	Calculatie van de mathematische functie. Confirmatie van opslag van de laatste grootboekkopie.	Uitvoeren van een herhalende calculatie van de mathematische functie met variërende complexiteit.	Zoektocht naar maximum stapel onder participanten (wedijverende nodes).

Aanpak 51%.	Onwaarschijnlijk omdat het noodzakelijk os om een compleet grootboek te hebben in bronnen en een berekenbare kracht om te calculeren, en de vertrouwde nodes worden dynamisch geselecteerd.	Waarschijnlijk maar zal erg duur zijn waar het het gebruik van de bronnen betreft..	Waarschijnlijk maar erg duur vanwege de nood om je eigen stapel te vergroten.
Compensatie voor het werk gedaan p de site voor het toevoegen aan het grootboek / de blockchain.	Automatisch berekend, hangt af van de commissie per operatie.	Aangeboden aanpassing voor het blockmijnen.	Aangeboden aanpassing voor het blockmijnen.

Het Concept van de Hoofdnetwerknode

Alle netwerknodes zijn gedecentraliseerd en geen daarvan heeft prioriteit. Het is vereist een netwerknode te definiëren die de rij van transacties, opgeslagen bij verschillende netwerknodes, zal verwerken. Daarna moet het een nieuw gegenereerd transactieblock in het grootboek invoeren.

Het CREDITS platform gebruikt zijn eigen gecombineerde protocol om de snelheid van transacties te vergroten, een volledige veiligheid van data-opslag te bieden, transacties verwerkend en overschrijvend. Het protocol is gebaseerd op de calculatie van de mathematische functie van alle grootboektransacties, de Proof of Work [Bewijs van Werk] toepassend. Het bepaalt nauwkeurig de opslag van de laatste up-to-date kopie van het grootboek en software op deze node (Proof of Capacity [Bewijs van Capaciteit]), door de checksom van de waardes van het gehele inhouden te calculeren – de hashcode. De grootte van bestanden wordt ook bepaald, als het bewijs dat dit de laatste up-to-date kopie is, en een hashcode van de laatste transactie opgenomen in het systeem.

Om de hoofdnetwerknode te worden zoekt de node naar de waarde van de hashfunctie die deze berekend op basis van het laatst opgeslagen grootboek. We organiseren een gezonde competitieve omgeving tussen de netwerknodes voor de gelegenheid om de hoofdnode te worden; om een nieuw grootboek te genereren en op te slaan.

Na calculatie van de functie en het resultaat verkrijgend wordt deze verzonden aan alle netwerknodes voor verificatie. Het resultaat bevat een tijdsmarkering van de calculatie en een waarde gebaseerd op de calculatie van de functie van de grootboekbestanden en software. Alle nodes ontvangen de gecalculeerde waarde, vergelijken de calculatietijd toegekend voor het zoeken naar de hoofdnetwerkserver, verifiëren deze en confirmeren de betrouwbaarheidsfactor van de node, en confirmeren ook zijn gelegenheid om te participeren in de competitie – om de hoofdnetwerknode te worden.

Na goedkeuring ontvangen te hebben van alle netwerknodes wordt een lijst gevormd van nodes die correct de waarde calculeerden van de functie en bevat een tijdsmarkering. De node die het correcte resultaat ontving en deze het snelste goedkeurde wordt de hoofdnetwerknode van dat moment.

Het SHA-2 algoritmeconcept wordt gebruikt om de hashsom te calculeren van het bestand.

Hash-functies van de SHA-2 familie worden gebouwd op basis van de Merkle-Damgard-structuur.

De initiële boodschap na de toevoeging wordt verdeeld in blocks, elke block wordt verdeeld in 16 woorden. Het algoritme passeert ieder boodschapsblock via een cyclus met 64 of 80 herhalingen (ronden). Bij iedere herhaling worden 2 woorden geconverteerd, en de rest van de woorden definiëren de conversiefunctie. De resultaten van ieder blockproces worden samengevat. De som is de hashfunctiewaarde. Echter de interne staat wordt geïnitieerd op basis van de resultaten van de vorige blockverwerking. Daarom is het onmogelijk om onafhankelijk blocks te verwerking en de resultaten op de sommen.

Gereedschap van Netwerknodes

We streven ernaar om een platform te bouwen met de snelst mogelijke transactieverwerkende

karakteristieken, dus stellen we voor een materiaalpremie te gebruiken om netwerknodes te onderhouden in de beste conditie: hoogpresterend servermateriaal en hoge internetbandbreedte.

Als materiaalcompensatie zal de eigenaar van de hoofdnetwerknode een vergoeding ontvangen in de CREDITS geldeenheid vanuit een aantal commissies per transacties van zijn verwerkte grootboek. De rest (½) is bedoeld voor het algemene projectontwikkelingsbudget voor gebruikersondersteuning, huidige kenmerken en ontwikkeling van nieuwe producten. Het percentage kan veranderd worden, alsook verdeeld worden naar het beoordelingsformatiesysteem middels federatief stemmen door de netwerknodes, na de initiële coinaanbieding voor ten minste drie jaar.

Als resultaat moedigen we servereigenaren aan deze server of hardware op de hoogste prestaties te houden om een hoogkwalitatief, zeer snel communicatiekanaal te behouden.

Consensusopbouw

Als resultaat is de hoofdnetwerknode geselecteerd door alle nodes. De hoofdtaken van de hoofdnode zijn: transacties verkrijgen in de kandidaatstatus om ze aan het grootboek van alle nodes toe te voegen, ze te verwerken, het laatste relevante grootboek op te bouwen en een nieuw opgebouwd grootboek aan alle netwerknodes te sturen. Het proces van transactieverwerking en opbouwen van het laatste relevante grootboek is exact het zoeken naar een consensusoplossing. Het resultaat van het opbouwen van het laatste relevante grootboek is de consensusoplossing.

Het hele proces kan verdeeld worden in de volgende stadia:

1. Zoeken naar de hoofdnetwerknode;
2. Opbouwen van vertrouwde nodes;
3. De lijst van transacties ontvangen en opbouwen van een lijst van kandidaten voor het toevoegen aan het grootboek;
4. De lijst van kandidaten verwerken, stemmen van nodes (vertrouwde en algemene nodes hebben verschillende afwegingsfactoren (vertrouwensfactor));
5. Verwijdering van de lijst van kandidaten van onbevestigde transacties die niet geverifieerd zijn of een negatieve bevestiging hebben;
6. Opbouwen van een lijst van bevestigde transacties om toe te voegen aan het grootboek;
7. Transacties toevoegen aan het grootboek met de tijdsmarkering en hashcode van het block dat de transactie bevatte;
8. Versturen van de block met transacties aan alle netwerknodes. Wanneer ontvangen wordt deze toegevoegd aan de registers van alle nodes.

Het Grootboek Opbouwen en Initiëren

Het hele proces kan beschreven worden in de volgende volgorde:

1. De eindgebruiker van het netwerk in het system genereert een transactie.
2. Wanneer aan alle voorwaarden van het smart contract die daarin gespecificeerd zijn wordt voldaan initieert de gebruiker de actie (transactie) via het aanroepen van de benodigde methode, gebruikmakend van de software van het platform.
3. Om de fundamentele principes van de blockchain te volgen houdt de kern van valideerders de synchronisatie en onveranderlijkheid van de laatste grootboekversie bij.
4. Op het moment van het opbouwen van de consensus worden alle transacties ontvangen gedurende de cyclus verzameld in het block.
5. Er wordt een nummer aan het block toegeschreven, bestaande uit een tijdsmarkering en een node-identificeerder omgezet in een hashcode, en dan wordt het block geplaatst in de consensusmodule.
6. Na compilatie van de witte lijst van kandidaten wordt niet alleen de hash van de transactie naar het grootboek geschreven maar ook de hash van het block, om altijd de bron die daarop is gebaseerd te certificeren.
7. Deze hash is een soort handtekening van het block en degene die dit block creëerde met transacties.
8. Na het opbouwen van consensus gebruikmakend van een federatief zoekalgoritme worden de

transacties die zijn toegevoegd aan het block doorgegeven aan de kern van valideerders om geschreven te worden naar het grootboek.

Transacties niet Ingesloten in het Register

Transacties die niet zijn ingesloten in de lijst van voltooide transacties worden gemarkeerd als afgewezen. Informatie hierover wordt onmiddellijk vertoond aan de verzender (initiator) van de transactie.

Transacties die niet zijn ingesloten in het grootboek verblijven in de set van kandidaten en worden opgeslagen in de netwerknodes. Alle nieuwe transacties ontvangen door de server op het moment van consensus arriveren daar ook, en dan beging het zoekproces opnieuw. Zulk een continue cyclische operatie van het netwerk maakt het mogelijk transacties te doen in een redelijke korte tijdsperiode terwijl een hoge graad van betrouwbaarheid en relevantie van informatie wordt behouden.

3. Transactieverwerking

Transacties

Een transactie is de minimum eenheid van het systeem dat het platform informeert over de uitvoering van contractmethoden of directe overschrijvingen tussen accounts zonder het creëren van een smart contract, gevolgd door plaatsing van het resultaat in het peer-to-peer netwerk.

Consensusopbouw

Het systeem gebruikt een federatief model om een consensus op te bouwen – het stemmen van vertrouwde valideernodes, en ook het consensusopbouwende algoritme – een algoritme voor het doorgeven van een eindige-staat-automaat. Consensus werkt met cycli (tijdsstappen), per tijdsstap worden transacties afgeleid en geplaatst in een pool (one-dimensional array [ééndimensionale serie]). Na geplaatst te zijn in de pool worden alle transacties verzonden naar de vertrouwde nodes om een respons te ontvangen. Als de respons ontvangen is kan de transactie waarvoor het verzoek tot toevoeging was verzonden toegevoegd worden aan het grootboek van deze valideerder. Daarna wordt deze verzonden naar de volgende valideerder in het netwerk. Wanneer de consensus is opgebouwd – aan het einde van de keten waar de overschrijvingslegaliteit volledig is geconfirmeerd, wordt de transactie verzonden naar validering met een markering voor het schrijven en opslaan naar het grootboek.

Transactieverwerking

Om de gedecentraliseerde aard van het system te bereiken moet iedere server zowel grootboekopslagruimte hebben alsook een volledige behandelaar van alle transacties zijn.

Het system gebruikt het concept van systeemkernen. Met kernen bedoelen we een databehandelaar die een specifieke productietaak uitvoert, ongeacht de beschikbaarheid en operationaliteit van de overgebleven systeemcomponenten. Iedere kern ontvangt, bij de invoer, op het moment dat de taak wordt uitgevoerd, een lijst van variabelen voor verwerking. En krijgt altijd een resultaat bij de uitvoer – positief, ieder ander of een fout. Als resultaat bevat de systeemkern altijd de responsieve code, als toevoeging aan de hoofddataset. Deze structuur is nodig voor de hoogst mogelijke snelheid van ieder proces, dat onafhankelijk van elkaar moet werken.

Grootboekinvoerstructuur

Om significante grootboekprestaties te bereiken, maar tegelijkertijd zonder te comprimeren op veiligheid, stellen we voor een grootboekdatabase te gebruiken zonder het opbouwen van de Merkle tree [Merkle-boom] van de hashcode van het vorige block en het transactieresultaat.

De Merkle tree (TTH – Tiger Tree Hashing) is een type van hashfunctie gebruikt om de integriteit van data te checken om een unieke identificeerder van de keten te verkrijgen en de volgorde te herstellen. De data

wordt verdeeld in kleine delen – blocks die individueel gehashd worden gebruikmakend van de Leaf Tiger Hash, daarna wordt de Internal Tiger Hash een-voor-een gecalculeerd vanuit ieder paar van hashes. Als de hash geen paar heeft wordt deze onveranderd overgestuurd naar de nieuwe keten. Vervolgens wordt de Internal Tiger Hash opnieuw gecalculeerd in de keten voor ieder paar. Deze procedure wordt herhaald totdat er één hash over is.

Wanneer het grootboek gebruikt wordt met de Merkle trees is de transactieverwerkingssnelheid erg laag, en de belasting op berekeningsbronnen is erg hoog. Naar onze mening is geen rationeel gebruik van dataopslag.

CREDITS Grootboekstructuur

We bieden aan om Merkle trees te verlaten en het transactielogboek te gebruiken in het CREDITS system; met iedere invoer bestaande uit een hashcode van het transactieblock om toe te voegen aan de lijst van kandidaten als toevoeging aan het grootboek. Ook heeft de invoer de node-identificeerder en de tijdsmarkering wanneer deze gegenereerd werd. De grootboekinvoer bevat de transactierichting, zijn initiële en uiteindelijke accounts, het type afschrijving, het aantal afschrijvingseenheden, het type storting en het aantal stortingseenheden. Dit principe verhoogt de snelheid van transactieverwerking, verhoogt de complexiteit van niet legitieme grootboekverandering en sluit mogelijke veranderingen achteraf in de grootboekinvoer uit.

Blockgrootte

De tijdseenheid is de cyclus van zoeken naar de hoofd- en vertrouwde nodes, en de tijdscyclus wordt gecalculeerd afhankelijk van de netwerkcomplexiteit. Per tijdseenheid bevat het netwerk N transacties, gegenereerd en overgeschreven voor verwerking naar het netwerk, vanaf het einde van de vorige cyclus tot de start van de volgende cyclus, om de status van "Kandidaat om toegevoegd te worden aan het grootboek." De transacties geselecteerd van netwerk N worden geplaatst op het block. De blockgrootte is afhankelijk van het aantal transacties daarin.

Zoeken naar Transactieparticipanten

CREDITS peer-to-peer netwerk kan gepresenteerd worden als een grafiek met gebruikersaccounts in de vorm van toppunten en een veelheid van mogelijke transacties in de vorm van gestuurde lijnen die twee toppunten (accounts) met elkaar verbinden. Aangezien alle lijnen een initieel en eindigend toppunt hebben kun je altijd een georiënteerde grafiek (orgraph) construeren.

Als we de volgende voorwaarden nemen voor identificatie:

- Iedere transactie heeft altijd een verzender en een ontvanger;
- Ieder toppunt (account) kan altijd verbonden worden met een ander toppunt met een directe lijn (transactie);
- Ieder toppunt van de grafiek (account) heeft een eindig aantal van directe lijnen (inkomende en uitgaande transacties).

In relatie met voorgaande kunnen we zeggen dat de georiënteerde grafiek de benodigde route bevat om aan de noodzakelijke transactievoorwaarden te voldoen en een eenvoudige keten te construeren. Aangezien het een eindige reeks van toppunten is waar ieder toppunt (behalve het laatste) verbonden is met het volgende toppunt in de reeks door een lijn.

Datatransmissiekanaal

Ieder kanaal van communicatie tussen de hoofdnetwerknode en de algemene node van het CREDITS network is een aparte thread [draad] (multithreading) waarin data wordt verzonden in versleutelde vorm wanneer de transactie wordt uitgevoerd.

Om netwerkveiligheid te verzekeren wordt alle data tussen de valideerdernodes verzonden in een versleutelde vorm, en iedere verbinding tussen nodes is laag-niveau gebaseerd op de netwerkbibliotheek. Als de data-overschrijving zich voordoet met een fout moet de thread automatisch geïnterrumpeerd worden, de corresponderende invoer wordt voor schrijven geplaatst in het logsysteem, en dan naar het logbestand. Data wordt

verstuurd via gespecificeerde variabelen. Verzonden data wordt versleuteld gebruikmakend van het symmetrisch RC4 algoritme. Aangezien dit algoritme onder een gemeenschappelijke geheime sleutel werkt wordt deze sleutel verzonden wanneer een verbinding wordt gemaakt tussen nodes en wordt verzonden in een versleutelde vorm in overeenstemming met het Diffie-Hellman algoritme.

Het Diffie-Hellman algoritme staat twee partijen toe om een gemeenschappelijke geheime sleutel te ontvangen gebruikmakend van een kanaal dat niet beschermd is tegen doorplaatsing maar beschermd tegen communicatiekanaalverandering. De ontvangen sleutel kan gebruikt worden om boodschappen uit te wisselen gebruikmakend van symmetrische versleuteling. Het algoritme is gebaseerd op de complexiteit van het berekenen van discrete logaritmes. Daarin, zoals in veel andere algoritmes met een publieke sleutel, worden de calculaties modulo uitgevoerd met een zeker groot primair getal P.

Eerst wordt een zeker natuurlijk getal A, kleiner dan P, geselecteerd op een speciale manier. Als we de waarde X willen versleutelen dan calculeren we

$$Y = AX \bmod P.$$

En het is gemakkelijk om Y te berekenen wanner we X hebben. Het omgekeerde probleem van X te berekenen vanuit Y is nogal gecompliceerd. Exponent X wordt precies het discrete logaritme Y genoemd. Aldus, de complexiteit kennend van het berekenen van het discrete logaritme, kan nummer Y publiek verzonden worden op ieder communicatiekanaal, aangezien met een grote modulus P de initiële waarde X bijna onmogelijk zal zijn om te kiezen. Het Diffie-Hellman algoritme om een sleutel te genereren is gebaseerd op dit mathematische feit.

Iedere actie in het systeem is verbonden aan de tijdsmarkering, het nummer van het vorige block, de login van de gebruiker en de ID van het smart contract. Dit maakt het vinden van duplicaten bij uitvoering mogelijk. Als een duplicaat gevonden is nemen we de eerste transactie uit de pool, de rest wordt als niet legitiem beschouwd.

Actie in het Systeem

Een actie in het system is een transactie die de eenvoudigste overschrijving van de waarde van account naar account karakteriseert of de overschrijving van het resultaat van de contractmethode naar de valideerder, voor het daaropvolgende zoeken naar een oplossing in het consensus-zoek-subsysteem.

Om de duplicatie van de transactie in hetzelfde blok met dezelfde identificeerder te voorkomen accepteert het systeem een overeenstemming dat de enige ware en correcte transactie die is die eerst aankwam bij het validatiesubsysteem voor verwerking. Aangezien al alreeds is opgenomen in het validatiesysteem dat een transactie alreeds is gemaakt vanuit de huidige account en er geen waardes over zijn in de account om de transactie uit te voeren kan een consensus niet gevonden worden. Aldus is het probleem van dubbele verspilling opgelost.

Wanneer de transactie wordt uitgevoerd wordt informatie tonvangen van de valideerder en bevestigd, de informatie over de grootboekstatusverandering wordt automatisch verdeeld onder alle nodes van de vertrouwde lijst, waarna het grootboek gesynchroniseerd wordt.

Om altijd een up-to-date transactiegrootboek te hebben onder alle vertrouwde nodes voor de huidige valididatienode is het nodig om de nieuw arriverende transactie in het grootboek van alle nodes iedere keer te synchroniseren. Om dit probleem op te lossen moet een aparte poort voor synchronisatie gebruikt worden (als er zulk een gelegenheid is). Deze gelegenheid zal de snelheid versnellen van het verwerken van de informatie die inkomt in de validatiekern vanwege de verdeling van de belasting op de poort. De gesynchroniseerde thread wordt altijd uitgevoerd, dit is cyclisch. Prioriteit voor de toewijzing van RAM en CPU belasting (gebruikmakend van CPU cycli) is lager dan gemiddeld. Het geheugen slaat de laatste 1.000 operaties en hun staat op (in een versleutelde vorm gebruikmakend van een synchroniserend algoritme), en dit vergroot de snelheid van het reageren op verzoeken van andere validatienodes.

Een Transactie Toevoegen voor Validatie

Een transactie toevoegen aan het grootboek wordt alleen opgeroepen vanuit het validatiesubsysteem onmiddellijk na consensusopbouw en het compileren van een witte lijst met de resultaten van transacties die wordt opgeslagen in het grootboek. Oproepen van derde-partijsystemen is onmogelijk om de veiligheid te verbeteren.

Inkomende parameters – het object dat de transactie karakteriseert. De resulterende waarde ResultValue<0 – uitvoering wordt verlaten met een fout, de resulterende waarde is een mogelijke foutcode / 0 < ResultValue – de functie werd uitgevoerd zonder fouten, het resultaat is het nummer van de invoer in het grootboek.

Inkomende parameter – het object dat het unieke label bevat van de transactie, de verzender, de ontvanger, de overgeschreven waarde, de waarde-overeenkomst, de gewenste waarde, de hoeveelheid van de overgeschreven waarde, de hoeveelheid van de gewenste waarde en andere systeeminformatie die veranderd kan worden wanneer nodig.

Kosten van Transacties

Het system gebruikt op dit moment CREDITS, welke dienen:

- Als een intern middel van betaling voor het systeemgebruik;
- Om verschillende munteenheden te wisselen binnen het systeem;
- Om diverse waarden binnen het systeem te wisselen;
- Om operaties onder smart contracts te creëren en te verwerken;
- Om informatie te kopen van derde-partijbronnen voor diensten binnenin het systeem.

De kosten van een transactie kunnen variëren afhankelijk van de netwerkbelasting, op een particuliere gebruiker van het systeem, die theoretisch een enorme vloed van transacties op een bepaalde piektijd kan sturen. We stellen voor de materiaalmethode te gebruiken en de impact op de systeemgebruikers om de netwerkbelasting te controleren.

de kosten van het uitvoeren van transacties zullen in de eerste drie jaar van de systeemoperatie individueel gesteld worden voor verschillende typen van transacties en operaties. In de toekomst zal een algoritme voor het automatisch genereren van de transactiekosten ontwikkeld worden.

4. Smart Contracts

Introductie

Een smart contract (slim contract) in het CREDITS systeem is een elektronisch algoritme dat een set van voorwaarden beschrijft waarmee acties en gebeurtenissen in de echte wereld of digitale systemen geassocieerd kunnen worden.

Om zelfcontrolerende smart contracts te implementeren is een gedecentraliseerde omgeving nodig die volledig de menselijke factor uitsluit, en om de overschrijving van de kosten van een smart contract te gebruiken is een cryptogeldeenheid onafhankelijk van de centrale autoriteit nodig.

Entiteiten

Een slim contract in CREDITS bestaat uit de volgende entiteiten:

1. Eigenschappen (publieke variabelen) – de systeementiteit die de publieke data opslaat die nodig is voor werking van het contract in het CREDITS systeem.
2. Method is de entiteit van het CREDITS systeem die verantwoordelijk is voor het observeren van de logica en volgorde van acties bij het uitvoeren van de transactie (acties onder het contract).

Participanten in het CREDITS systeem tekenen de smart contracts gebruikmakend van de methode-oproep die de contracteigenschappen modificeert, door het lanceren van het proces voor verificatievolgzaamheid van condities en coördinatie.

Een smart contract wordt bekrachtigd na het tekenen door de partijen. Om automatische vervulling van verplichtingen te verzekeren is een omgeving van bestaan nodig die volledig automatisch de contractvoorwaarden uitvoert. Dit betekent dat smart contracts alleen bestaan binnen een omgeving die onverhinderde toegang heeft tot de uitvoerbare code naar het smart contractvoorwerp.

Alle contractvoorwaarden moeten een mathematische omschrijving hebben en duidelijke logica van uitvoering. Aldus is het hoofdprincipe van een smart contract volledige automatisering en betrouwbaarheid van contractuele relaties tussen de partijen.

Smart Contractmethode

De CREDITS smart contract methode is de systeementiteit verantwoordelijk voor het volgen van de logica en opeenvolgende acties gedurende de transactie (acties onder het contract).

De logica en opeenvolging van acties worden beschreven door een programmacode (module) die opdrachten bevat; hun opeenvolgende uitvoering staat toe het gewenste resultaat te bereiken. Deze code kan systeemcommando's behandelen (bijvoorbeeld de taakcommando), gebruikersopdrachten (apart geschreven functies), contracteigenschappen (statisch of dynamisch geïnitieerde variabelen beschikbaar van iedere contractmethode), en methoden van ieder ander derde-partijcontract beschikbaar voor alleen de eigenaar van het verbonden (derde-partij)contract. Voor meer popularisatie wordt de ontwikkeling beschikbaar gesteld in scripttaal (bijvoorbeeld JavaScript).

De methode (programmacode) maakt mogelijk; het gebruik van alle wijds gebruikte scripttaalfuncties (commando) (taken, conditionele en niet-conditionele sprongen), het creëren van functies en procedures (subroutines), verbinding van derde-partijbibliotheken.

Virtuele Uitvoerbare Machine

The contract methode van het CREDITS systeem wordt uitgevoerd in de virtuele omgeving van het systeem (Virtuele Machine, hierna gerefereerd aan als VM). Wanneer een methode wordt opgeroepen voor een particulier contract wijst VM een geheugengebied aan en laadt de contractbytecode die in zich de geïnitieerde (of opnieuw gedefinieerde, wanneer andere contractmethoden worden opgeroepen) methoden en de variabelen bevat. VM start met verwerking van de methode bytecode, bij loopperiode, variabelen en code worden geladen in zijn geheugengebied en commando's worden succesvol uitgevoerd, hun resultaat wordt overgeschreven naar het peer-to-peer netwerk voor daaropvolgende plaatsing in het grootboek.

De initiator van de uitvoeringsmethode is de gebruiker van het systeem, voor wie deze methode wordt gelanceerd.

Waardevoorwaarde

CREDITS cryptogeldeenheid is ook een indicator van de waarde-voorwaarde van een contracteenheid om twee complete verschillende eenheden te vergelijken en een consensus op te bouwen wanneer het contract door de partijen uitvoerend of accepterend. In plaats van iedere aparte waarde-/gatewaycombinate te registreren dienst CREDITS cryptogeldeenheid als een samendromming om waarde-overschrijvingen te effectueren. Dit is mogelijk omdat iedere waarde liquide is met betrekking tot CREDITS geldeenheid, wat betekent dat iedere waarde liquide kan zijn in relatie tot iedere andere waarde.

Uitvoering van de Smart Contractvoorwaarden

De contractvoorwaarde in het CREDITS systeem is de waarde van de trigger (gecheckte) velden die nodig zijn om het (complete) contract te sluiten.

Voltooiing van de smart contractvoorwaarden is een procedure wanneer de trigger (gewenste) velden gecontroleerd zijn voor een equivalente gewenste waarde. Er zijn drie mogelijke manieren om een oplossing te vinden om de contractvoorwaarden te vervullen:

1. Het contract wordt afgesloten tussen twee of meer partijen voor de overdracht van waarde. In dit geval betreft de contractvervulling de provisie van het kostenequivalent van de waarde van de overdragende partij vanuit de ontvangende partij.
2. Het contract wordt afgesloten tussen de partijen voor de overdracht van waarde,

maar betaling moet gemaakt worden bij het vervullen van een bepaald aantal condities (bijvoorbeeld levering van waarde aan de ontvangende partij).

3. Een contract voor omzetting van de ene waarde naar de andere met een kostenequivalent in de vorm van CREDITS wordt geplaatst in het systeem. In dit geval start het platform met het zoeken naar het kortst mogelijke pad van het wisselen van de ene waarde voor de andere via omzetting in andere contracten. Iedere vervulling van het contract kan geleverd worden per één transactie, of per diverse transacties, wat de gelegenheid zal opleveren om de benodigde kwantiteit van waardeenheden te verzamelen om het contract te voltooien.

Databronnen

Voor correcte en volledig uitgerust werk, checken en voorzien in additionele informatie om een meer gebalanceerde en optimale oplossing te leveren gebruikt CREDITS derde-partij dataproviders. De nood om additionele databronnen in het systeem te introduceren is vanwege de ontoereikendheid van publieke informatie over één of diverse contractpartijen (bijvoorbeeld verkrijgen van de kredietstatus van de lener voor het maken van een beslissing om een krediet uit te geven).

Om te werken met derde-partij informatiesystemen kan het platform een integratiebus aanroepen die op afstand een verzoek kan genereren aan een derde-partij systeem (site) in een format voor datapresentatie op betaalde basis voor de systeemparticipanten met betaling in CREDITS.

Het verzoek wordt verzonden in een versleutelde vorm naar poorten en adressen geleverd door informatiesystemen anders dan de standaardsystemen. Het resultaat van het verzoek kan iedere respons zijn op de dienst die de noodzakelijke informatie bevat om een beslissing te maken, of een foutcode die de onmogelijkheid karakteriseert van het ontvangen van de benodigde respons en mogelijke stappen om de fout te elimineren.

The request is sent in an encrypted form to ports and addresses provided by information systems other than the standard ones. The result of the request can be any response to the service containing the necessary information to make a decision, or an error code characterizing impossibility of receiving the required response and possible steps to eliminate the error.

5. Implementatieplan

Technisch Plan van Projectimplementatie

	S1	S2	S3	S4	S5
	Pre-Alpha	Alpha	Beta	Release kandidaat	Release
Opslag, Consensus mFA Consensus	FA : Sleutel Design Implementatie	mFA : Sleutel Design Implementatie PoW (Proof-of-Work) and PoC (Proof-of-Capacity)	mFA Optimalisatie	–	–

Data Opslag	Decentralisatie Grootboek, NoSQL Opslag implementatie	MessagePack History	–	Blockchain backup	–
CVM (Credits virtual machine)	Design en Implementatie	Integratie met ecosysteem	Optimalisatie	Check fouten	–
Derde-partij systeem	–	Design en Implementatie	Integratie in volledig systeem	–	Optimalisatie
Inferentie Engine	Formele Specificatie en Sleutel Design Elementen	Redeneerder Integratie met Blockchain	Redeneerder Optimalisatie	–	–
Gebruikersinterface	Implementatie	Web UX design	–	–	–
Wallet	Wallet Formele specificatie		UX design Applicatie Test	–	Android, iOS, Desktop Wallets
RPC & REST API	Formele specificatie	Blockchain Explorer	–	Derde-partij Explorer	–

CREDITS Cryptogeldeenheid

Na het lanceren van de releaseversie van het systeem zal een vaste hoeveelheid van 1.000.000.000 CREDITS uitgegeven worden. Deze zullen ingewisseld worden voor ERC20 standaardtokens, uitgebracht tijdens de initiële tokenverkoop. Ze zullen gewisseld worden met een vaste wisselprijs: 1 ERC20 standaard token = 1 CREDITS monetaire eenheid.