2023

Requirementsanalyse

SOLIDITY DAPP

TADRAŁA,PIOTR P.P.

O-PP-CMK

CONTENTS

INLEIDING

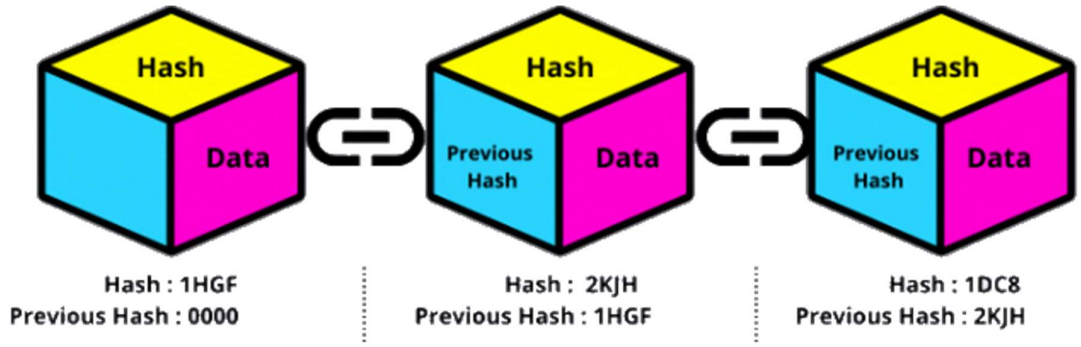
Om de onderzoeksvragen die door InforDB zijn voorgesteld in de praktijk toe te lichten, moet een functioneel prototype worden ontwikkeld dat de aspecten van Dapps in de praktijk demonstreert. Ik begin met het ontwikkelen van eenvoudige Smart Contracts om het concept gemakkelijk te visualiseren. Vervolgens ga ik verder met het ontwikkelen van een functionele applicatie met behulp van smart contracts.

In dit document beschrijf ik wat een blockchain is, wat Smart Contracts zijn en hoe Smart Contracts gebruikt worden voor het ontwikkelen van Dapps. Vervolgens maak ik een requirements analyse voor mijn functioneel prototype.

BLOCKCHAIN

BASISCONCEPT VAN BLOCKCHAIN

In zijn meest basis vorm is een blockchain een openbaar ledger van transacties en gegevens dat wordt vastgelegd op een gedecentraliseerd netwerk van computers. Transacties worden vastgelegd in 'blokken', waarbij elk blok kennis heeft van het vorige blok middels een hash. Al deze blokken vormen een chain die voortdurend groter wordt, aar komt de term 'blockchain' vandaan. De kennis van het vorige blok maakt het onmogelijk om vastgelegde gegevens te corrupten zonder de consensus van de blockchain te breken.



TRANSACTIES EN BLOKKEN

Wanneer een gebruiker een transactie wil verzenden, moet hij een fee betalen. Zodra een transactie is ondertekend, wordt deze in de mempool (que) geplaatst. Transacties met de hoogste fee worden als eerste toegevoegd aan het eerstvolgende blok. Dit betekent dat mensen de mogelijkheid hebben om meer fee te betalen voor een snellere transactie, of minder fee te betalen en mogelijk wat langer moeten wachten.

De manier waarop een transactie wordt gevalideerd, is afhankelijk van het consensusmechanisme van de blockchain, zoals bijvoorbeeld PoW (Proof of Work, Bitcoin) of PoS (Proof of Stake, Ethereum).

Miners, in het geval van PoW, 'zoeken' nieuwe blokken, zodra een blok is gevonden, mogen ze dit blok inclusief alle transacties aan de blockchain toevoegen. Als beloning ontvangt de miner een 'base block reward' en alle transactie fees van dat blok

DECENTRALISATIE

Decentralisatie vormt het basisprincipe van een blockchain. In een gedecentraliseerd systeem ontbreekt een centrale autoriteit die transacties of gegevens beheert. In plaats daarvan worden alle gegevens verdeeld over duizenden nodes die over de hele wereld verspreid zijn. Resultaat hiervan is een hoog niveau van security, omdat een malicious actor de controle zou moeten hebben over meer dan 51% van alle nodes die de transacties valideren om een ongelidge transactie in het netwerk te kunnen verwerken.

REACHABLE BITCOIN NODES
Updated: Sat Oct 14 14:29:43 2023 CEST

16720 NODES CHARTS

IPv4: -3.3% / IPv6: -1.0% / .onion: +2.4%

Top 10 countries with their respective number of reachable nodes are as follows.

RANK	COUNTRY	NODES
1	n/a	10643 (63.65%)
2	United States	1531 (9.16%)
3	Germany	1261 (7.54%)
4	France	424 (2.54%)
5	Netherlands	328 (1.96%)
6	Canada	265 (1.58%)
7	Finland	234 (1.40%)
8	United Kingdom	193 (1.15%)
9	Russian Federation	175 (1.05%)
10	Switzerland	147 (0.88%)

All (98) »

Map shows concentration of reachable Bitcoin nodes found in countries around the world. LIVE MAP

BRONNEN

- Scott Nevil 2013, What is proof of work
- DNB, Blockchain: de techniek achter crypto's

SMART CONTRACTS

WAT ZIJN SMART CONTRACTS

Smart Contracts zijn gedecentraliseerde scripts en apps die op een blockchain worden uitgevoerd. Ze werken als digitale overeenkomsten waarin de regels van transacties zijn vastgelegd. Dankzij de decentralisatie van smart contracts is er geen middleman nodig bij overeenkomsten of transacties tussen twee partijen. Aangezien alles op een blockchain, inclusief smart contracts, opensource is, zijn de voorwaarden van een smart contract voor iedereen volledig transparant.

WERKING VAN SMART CONTRACTS

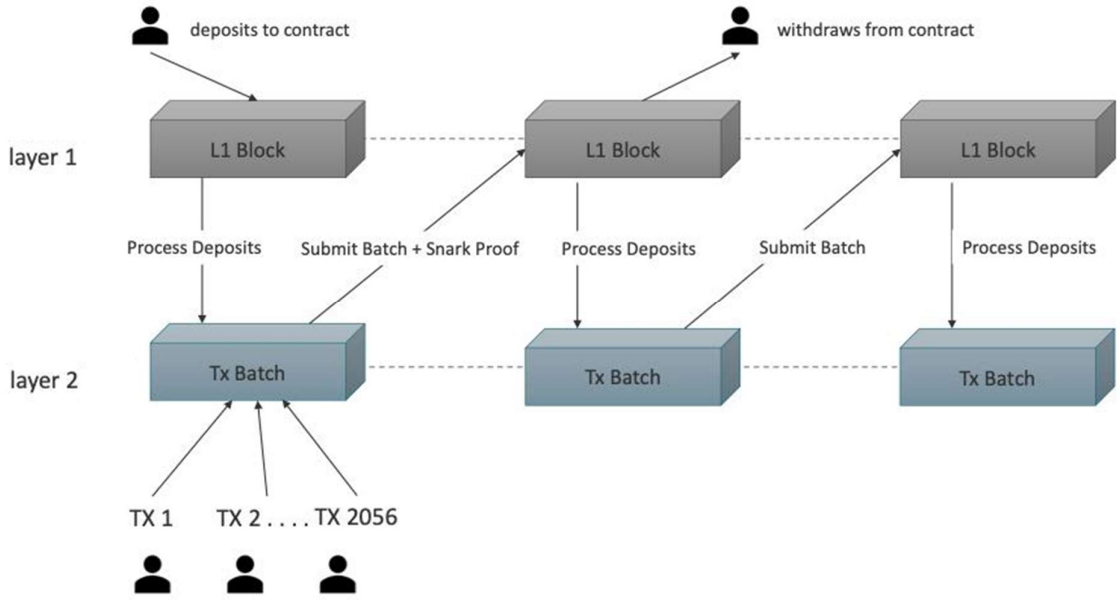
Niet alle blockchains ondersteunen smart contracts, aangezien dit afhankelijk is van de onderliggende technologie. Een van de meest bekende blockchains die smart contracts ondersteunt, is Ethereum. Ethereum is specifiek ontworpen om de functionaliteit van smart contracts mogelijk te maken. Deze functionaliteit maakt het Ethereum netwerk schaalbaarder dan bijvoorbeeld Bitcoin, doordat smart contracts het concept van layer 2's introduceren.

Een smart contract kan worden gezien als een script met ingebouwde functies. Wanneer je met een smart contract in interactie komt, stuur je in principe een transactie naar het smart contract, waarin de gegevens staan over welke functie je wilt activeren en de bijbehorende parameters. Zodra een smart contract op het netwerk wordt geplaatst is deze immutable, dit betekent dat het inhoud onmogelijk is om aan te passen.

LAYER 2

Een layer 2-netwerk functioneert als een soort side-chain bovenop Ethereum. Het combineert blokken op zijn eigen netwerk en stuurt deze samengevoegde gegevens vervolgens als een enkele transactie naar de main chain, in dit geval Ethereum. Hierdoor kan een layer 2-chain profiteren van de security van de hoofdchain, terwijl

transactie fees worden verdeeld over een groot aantal individuele transacties. Dit leidt tot een hogere efficiëntie en scalability.



SOLIDITY

Solidity is een object oriented programmeertaal die wordt gebruikt voor het ontwikkelen van smart contracts op Ethereum en enkele andere blockchains. Solidity is bedacht door Gavin Wood, een co-founder van Ethereum, in 2014. De verdere ontwikkeling is mogelijk gemaakt door een team van programmeurs, waaronder Christian Reitwiessner. Solidity is ontworpen om het mogelijk te maken om op dapps de bouwend middels smart contracts die op de EVM (Ethereum Virtual Machine) draaien. De taal is gebaseerd op de ECMAScript-syntax.

VOORBEELDEN

Het volgende stukje code komt uit de ERC-20-standaard, die wordt gebruikt bij het aanmaken van tokens. In het voorbeeld is de transfer() functie te zien, die het mogelijk maakt voor een gebruiker om tokens naar een andere persoon over te dragen.

example.sol

```sol
pragma solidity ^0.8.0;

// Definitie van een Smart Contract
contract SimpleToken {

    // Definitie van de eigenaar
    address public owner;

    // Mapping van alle balansen van de gebruikers
    mapping(address => uint256) public balances;

    // Events om transacties vast te leggen
    event Transfer(address indexed from, address indexed to, uint256 value);

    // Constructor die wordt uitgevoerd bij het maken van het contract
    constructor() {
        owner = msg.sender;
    }

    // Functie om tokens over te dragen van de ene gebruiker naar de andere
    function transfer(address to, uint256 value) public {
        // Require verifeert een conditie
        require(balances[msg.sender] >= value, "Onvoldoende saldo");

        balances[msg.sender] -= value;
        balances[to] += value;

        emit Transfer(msg.sender, to, value);
    }
}

```

BRONNEN

•Matt Timmermans 2021, Solidity – Heldere uitleg over deze programmeertaal

DAPPS

WAT ZIJN DAPPS

Dapps, gedecentraliseerde applicaties, zijn het belangrijkste concept van een blockchain met smart contracts mogelijkheden. Deze applicaties voeren al hun functionaliteiten uit via smart contracts, zoals inloggen of handelen. Het verschil tussen traditionele apps en Dapps is dat er geen centrale entiteit nodig is om de

Dapp te beheren, in plaats daarvan draait alles op de blockchain. Aangezien de source code van Dapps openbaar is, heeft iedereen inzicht in de werking van elke functionaliteit en de mogelijke gevolgen daarvan. Aangezien smart contracts immutable zijn, kun je ook zeker van zijn dat een correct geprogrammeerd smart contract consistent blijft functioneren.

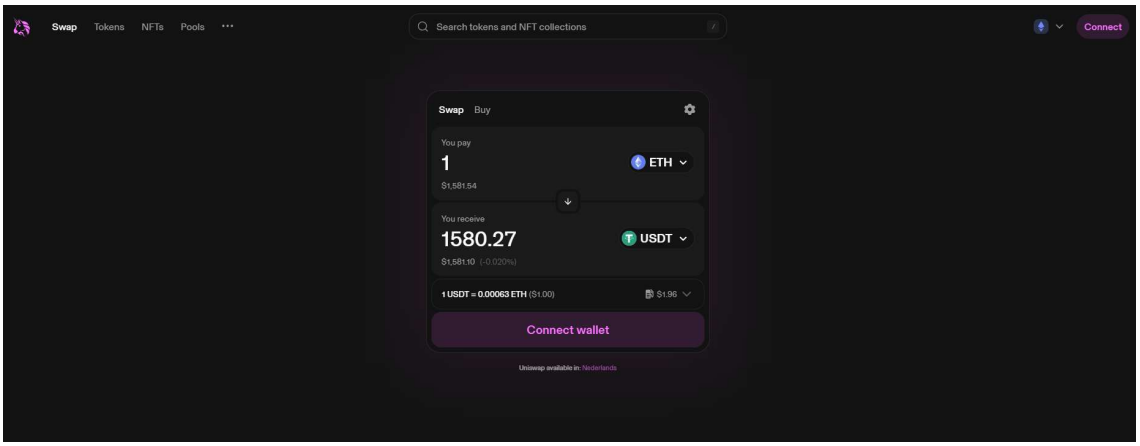
ONTWIKKELING VAN DAPPS

Ontwikkeling van Dapp begint met het schrijven van de smart contract. Deze wordt vervolgens compiled tot bytecode die op een blockchain kan worden degeployed middels en transactie. Vervolgens wordt de ABI (Application Binary Interface) gekoppeld aan een smart contract. De ABI functioneert als een interface waarmee externe applicaties of gebruikers kunnen communiceren met het smart contract. Hierdoor kunnen gebruikers interactie hebben met de Dapp en transacties initiëren en alsnog een begrijpelijk overzicht hebben over wat de code doet. Aangezien smart contracts immutable zijn, zie je vaak V2, V3 etc versies van dapps als er behoefte is aan nieuwe functionaliteiten.

VOORBEELDEN

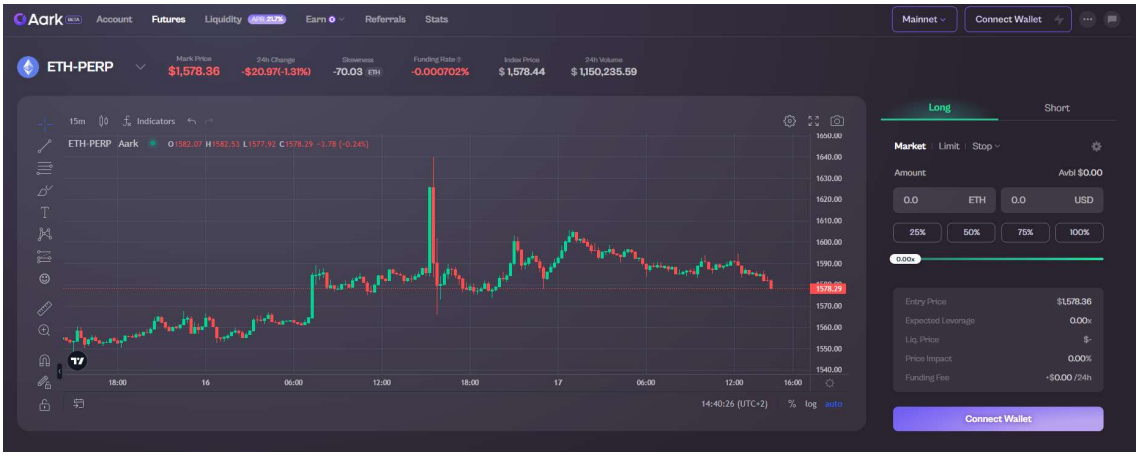
UNISWAP

Protocol voor het wisselen van tokens op de EVM blockchains.



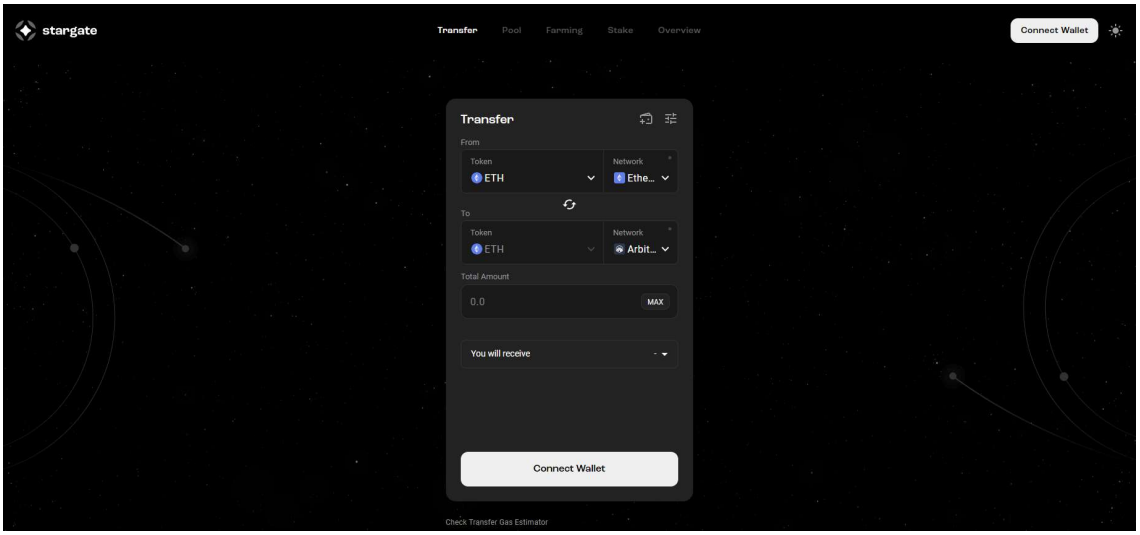
AARK

Exchange op basis van het order book model met leverage mogelijkheden.



STARGATE

Protocol voor het wisselen van tokens tussen de EVM blockchains.



BRONNEN

- Patrick Suiker 2018, Dapps – Wat zijn het en hoe gebruike je ze
- Mathijs 2023, Wat zijn Dapps? Alles over decentrale applicaties

REQUIREMENTS

CONTEXT

Het uiteindelijke resultaat zal uit twee delen bestaan. In eerste instantie ga ik aan de slag met het deployen van tokens op basis van de ERC-20 standaard. Dit zal meer of minder bedoeld zijn om mijn kennis van solidity te verbeteren. Vervolgens worden deze tokens in het tweede deel van de applicatie gebruikt. Het belangrijkste onderdeel van de applicatie zal een gedecentraliseerde exchange op basis van het AMM -model (Automated Market Maker) zijn.

Een Automated Market Maker (AMM) model is een systeem voor het ruilen van tokens waarbij algoritmen worden gebruikt in plaats van traditionele order books om de prijs beheren. De prijsverhouding tussen twee tokens in een liquidity pool is simpelweg de ratio tussen de twee tokens, waarbij de liquidity van Token A evenveel waard is als de liquidity van Token B binnen de pool.

REQUIREMENTSANALYSE

Om een volledig functionele Dapp te creëren, moeten zoals eerder genoemd twee producten worden ontwikkeld: een Smart Contract token, dat volgens de ERC-20 standaard zal worden ontwikkeld voor security en makkelijke integratie binnen bestaande systemen, en een gedecentraliseerd Exchange waar gebruikers pools van ERC-20 tokens kunnen creëren en vervolgens tokens kunnen wisselen, waarbij de prijs wordt berekend op basis van het Automated Market Maker model. De Dapp moet aan de volgende functionaliteiten voldoen.

ERC-20 TOKENS

- Solidity Script ontwikkelen op basis van de ERC-20 standaard.
- Tokens deployen op een testnet blockchain, zoals Ethereum Goerli.

- De tokens moeten een ABI bevatten van alle functies.

DEX

- Solidity script ontwikkelen die het mogelijk maakt om liquidity pools van ERC-20 token pairs te creëren.
- Een liquidity pool moet een standalone contract zijn, die de wissel tussen de twee geselecteerde tokens beheert.
- Prijs van de tokens moet automatisch berekend worden op basis van het AMM-model
- Users moeten een mogelijkheid hebben om met bestaande wallets te connecten, zoals Metamask of Coinbase Wallet.

ALGEMEEN

- Er dienen unittests uitgevoerd en gerapporteerd te worden.

INTEGRATIE BINNEN BESTAANDE SYSTEMEN EN INFRASTRUCTUUR

De Dapp moet integratiemogelijkheden bieden met bestaande systemen, de tokens moeten voldoen aan de ERC-20-standaard om interoperabiliteit met andere Dapps mogelijk te maken. De DEX moet de mogelijkheid bieden om liquidity pools te creëren van van bestaande tokens. Bovendien moeten users mogelijkheid hebben om met de meest gangbare wallets te communiceren met de Dapp.

ACCEPTATIECRITERIA

Het volledige project moet aan de volgende eisen voldoen

- Er dienen tokens comform ERC-20 standaard gedeployed te worden.
- Er dienen Liquidity pools van de ERC-20 tokens gemaakt te worden.
- Dex moet in staat te zijn om tokens te wisselen
- Users moeten mogelijkheid hebben om met de meest gangbare wallets te connecten.
- Users moeten zelf liquidity kunnen toevoegen.

STAPPENPLAN

1. Requirements verzamelen en vastleggen (Huidig document)
2. Achitectuur opstellen
3. Applicatie ontwikkelen.
4. Configuratie-, change- en releasemanagement implementeren.
5. Uitvoeren van unittests

SLOT

In dit document is de context van dit project toegelicht en is aangegeven wat het eindproduct inhoudt. De volgende stap is het opstellen van een architectuur voor een applicatie die voldoet aan de eerder genoemde functionaliteiten en concepten.