dApp en Blockchain Development

Een workshop waarin we de wereld van Smart Contracts op de Ethereum blockchain induiken.

Bèr `berkes` Kessels

Over

Over Bèr Kessels

- Bèr Kessels
- @berkes, github.com/berkes
- Ruby, Blockchain en Open Source developer
- Founder van PlaceBazaar

Over de presentatie

- Staat op: github.com/berkes/ethpres
- URL komt aan het einde nog eens voorbij.

Notes en prikbord

https://beta.etherpad.org/p/fundfissa (ook op Moodle)

Over mij en Blockchain

- April 2011 begonnen met Bitcoin
- In bouw aan een startup, placebazaar.org. Bouw dit op Ethereum

Inhoud

- Welk probleem lost Blockchain op?
- Wat is een Smart Contract?
- Wat is een dApp (Web 3.0)?
- Wat is Ethereum?

Wat is Blockchain?

Een onveranderlijk, gedistribueerd grootboek

An Immutable, Distributed Ledger

Wat bedoelen we met een grootboek?

Een database met daarin opeenvolgende transacties:

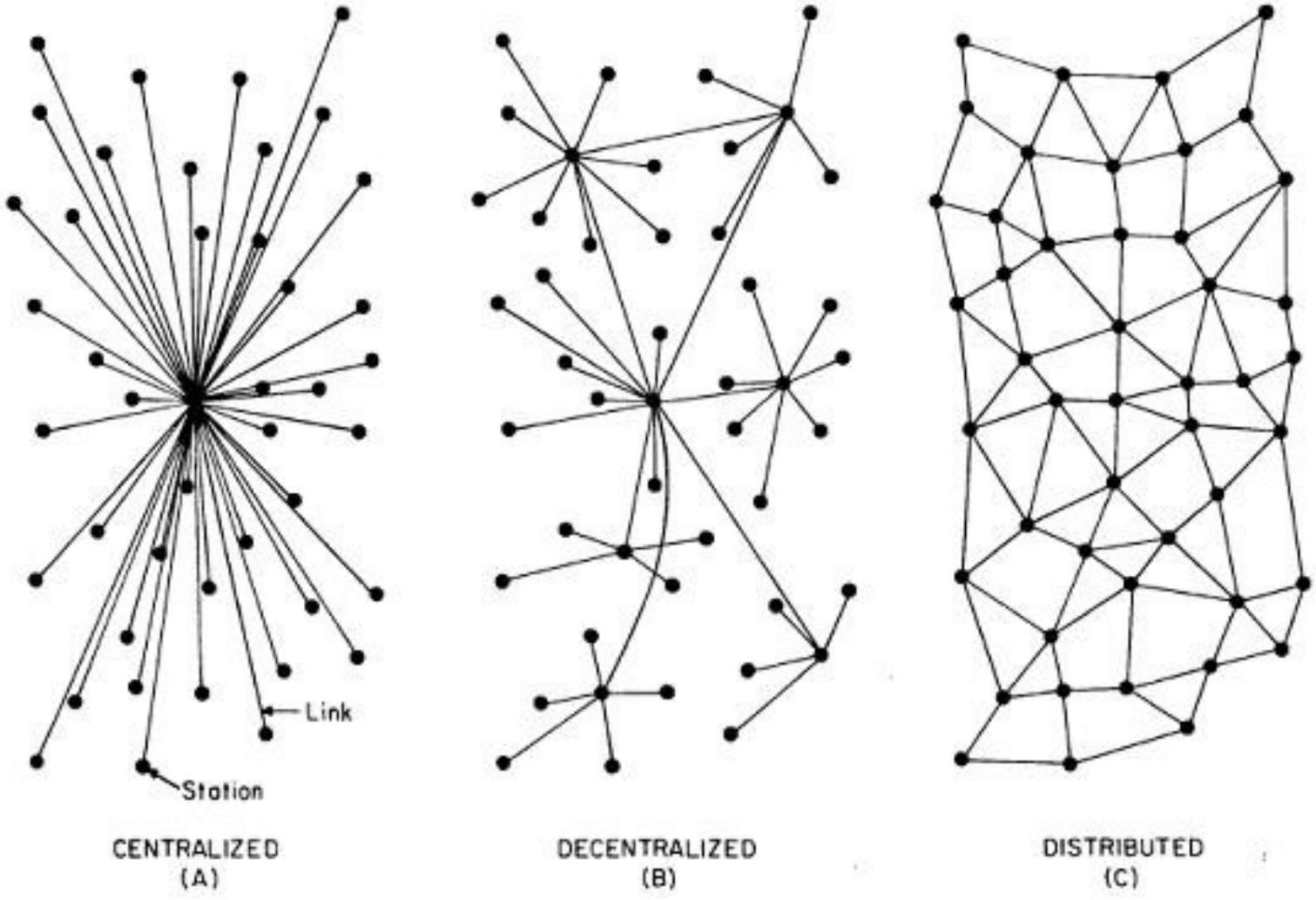
Beginstand: Anne: 6, Bob 0, Carol 1

1. **Anne** geeft 5 aan **Bob**
2. **Bob** geeft 2 aan **Carol**
3. **Carol** geeft 1 aan **Anne**

Eindstand: Anne 2, Bob 3, Carol 2

Gedistribueerd:

Centraal, Decentraal, Gedistribueerd



Figuur 1: Centraal vs Decentraal vs Gedistribueerd

Bijvoorbeeld contant geld:

Gedistribueerd (cash) is:

- Trustless (vertrouwenloos).
- Permissionless (vergunningsvrij)
- Uncensorable (oncensureerbaar)
- Verifiable (controleerbaar)

Dat komt voort uit “Distributed”

- Geen centraal “point of failure”
- Kan wereldwijd opschalen
- Heet ook wel “Peer to Peer, P2P”

Waarom werkt muntgeld?

Ideeën?

En hoe moet dat digitaal?

- Tot 2008 werd dit onmogelijk geacht
- Altijd via centrale autoriteiten

Centraal:

Een blockchain!

Een onveranderlijk, gedistribueerd grootboek!

Onveranderlijk? (Immutable)

- Data kan niet aangepast worden nadat het in het grootboek is weggeschreven
- Data is controleerbaar
- Er is consensus nodig om het grootboek bij te werken

Wat is consensus?

- “Iedereen” komt overeen wat de huidige status is
  - Bijvoorbeeld: wat zijn de saldi van iedereens rekeningen

Distributed

- Complex
- Maar zonder centrale partijen!

Ethereum

Wat is een Smart Contract?

Code

```
pragma solidity ^0.4.0;

contract HelloMyNameIs {
  string name;
  address issuer;

  function HelloMyNameIs() public {
    issuer = msg.sender;
  }

  function getName() public constant returns(string) {
    return name;
  }

  function setName(string newName) public returns(string) {
    require(issuer == msg.sender) ;
    name = newName;
    return name;
  }
}
```

Wat is een dApp (Web 3.0)?

De voordelen?

- **Geen centraal controlerende instantie**. Je appartement verhuren is tussen jou en de huurder. AirBnB of andere "Siren Servers" hebben daarin geen plek.
- **Data eigendom**. Je blijft eigenaar van jou data. Jij kunt deze verkopen. Of niet.
- **Minder heftige hacks**. Decentraal betekent dat een hacker het hele netwerk moet overnemen om toegang te krijgen, ipv een centrale server.
- **Permissionless**. Of je nu een onderdrukte minderheid, of gezochte terrorist bent, niemand kan je toegang onthouden tot diensten.

Smart Contract proberen

- Ga naar HelloMyNameIsOpen (link op eterpad)
- Onder “Read” bekijk de huidige naam.
- Onder “Write” connect met metamask.
- Verander de naam. Kies “write”. Bevestig in MetaMask.
- Bekijk transactions. Bekijk de naam.

Zelf een Smart Contract releasen

Remix IDE

Interact met ander contract

- Laad het contract van één van je collega's in.
- gebruik hiervoor `At Address` formulier.
- Probeer `setName`, merk de foutmelding op.

What we just did

Compile en deploy

- Met de “Web3.js” API compileren en deployen.
- Dit wordt een “account”: een entiteit met een wallet, op een adres.
- Twee soorten accounts: contracts (zonder private key) en users (met private key)

Interactie via formulieren

- Contract heeft een Interface (ABI).
- Een client kan dan functies aanroepen op het contract.
- Iedere interactie met state-change is een *transactie*.

Clients

- **CLI:** geth, web3/node.js, solc
- **Officiëel:** mist wallet
- **Web:** remix IDE
- **dApp:** JS op jou site
- En vele andere wallets

Transacties

- Transactie is een verandering, write, op blockchain uitvoeren.
- Succesvol aanroepen van `setName` is een transactie.
- Leesacties zijn `gratis` en instant.
- Transacties duren even.

Gas

- Draaien van code kost wat:

- Note gas
- Alle `OPCODES` hebben vooraf bepaalde gas-cost

Gas, Ethers, Gwei

- Ether is betaalmiddel
- Uitgedrukt in “wei”, kleinst deelbare eenheid
- 1e18 wei = 1 ether
- gas is dynamisch, 1 gas kost X wei ethgasstation.info

Operaties kosten Gas

- Iedere operatie kost gas
- Sommige operaties zijn duur, andere goedkoop
- Wanneer er te weinig gas is, wordt een OutOfGas exceptie geraised

Waarom Gas?

- Gebruiker/aanvaller betaalt
- Endless loops niet mogelijk (Halting problem)
- Miners ontvangen gas voor het draaien van de code

Solidity

- Defacto standaardtaal voor het schrijven van Ethereum smart contracts
- Alternatieven: Serpent (Python), LLL (Lisp), Viper (Python), Bamboo (OCaml)

FundFissa

Wat uitleg en details

```
Contract: Fissa
  deploy
    ✓ sets an eventName
    ✓ sets StartsAt
    ✓ sets a ticketPrice in wei
    ✓ has a threshold
    ✓ has a participants map
    ✓ has a balances map
    ✓ has an organizer
  purchase
    ✓ adds buyer to list of participants (55ms)
    ✓ increments buyer amount in participants (48ms)
    ✓ increments buyer balance (54ms)
    ✓ transfers ether from buyer into the contract (55ms)
    ✓ does not allow payments lower than ticketPrice (39ms)
    ✓ does not allow payments higher than ticketPrice
    ✓ sends a Purchase event
  isExpired
    ✓ is expired
    ✓ reports to be expired
    ✓ no longer allows purchase()ing
  isFunded
    ✓ is false when threshold is not met (43ms)
    ✓ is true when threshold is met (44ms)
  withdraw when not isExpired()
    ✓ is not allowed for buyer (42ms)
  withdraw when isExpired() but not isFunded()
    ✓ allows buyer to withdraw from their balance (54ms)
    ✓ does not allow withdrawing more than our balance (46ms)
  withdraw when isExpired() and isFunded()
    ✓ refutes buyer to withdraw from their balance
```

Security

- It runs forever. Immutable.
- Kill-switch.
- Logical errors.
- Programming errors.

Presentatie

- github.com/berkes/ethpres