Toekomst van ons geld, bitcoin en de blockchain

Kees van Hee

12-10-2018

Agenda

- Wat is geld eigenlijk?
- Cryptografie in een notendop
- Bitcoin
- Blockchain
- Voor en nadelen
- Ons systeem*

**Een nieuw Monetair Systeem en een nieuw Monetair Beleid*

Kees van Hee en Jacob Wijngaard

white paper, zie: www.robuustgeld.nl

Wat is geld eigenlijk?

- Geld is ruilmiddel en spaarmiddel
- Sparen is ruilmiddel voor de toekomst
- Ruilmiddel is altijd een schaars goed, dat:
  - Makkelijk is op te slaan
  - Makkelijk is te vervoeren
  - Niet bederfelijk is
- Voorbeelden:
  - Vee (pecunia komt van pecus)
  - Zout
  - Zilver
  - Goud

Geld is ‘schaars goed’

- Schaars = de *moeite* die het kost om het verkrijgen
- Vroeger: *moeilijk* te verkrijgen materialen
- De waarde van een munt was de goud- of zilverwaarde*
- Later: het was de *moeite* die het kost om de munt (na) te maken **
- Een cryptogeld (Bitcoin) is in wezen een *getal*: een getal dat moeilijk te vinden is
- Fiduciair geld: heeft zelf geen materiele waarde, maar ze hebben wel het vertrouwen(wettig betaalmiddel)

*Wet van Gresham: Bad money drives out good money
(muntwaarde > materiaal waarde is bad money)

** 1 op $10.000 is vals en 1 op € 20.000

Chartaal, Giraal en Digitaal geld

- Chartaal geld:
  - Munten
  - Bankbiljetten
- Basisgeld (ook wel positive money of sovereign money)
  - Chartaalgeld
  - Reserves van de banken bij DNB
- Giraalgeld:
  - Geld tegoeden bij banken (claims op de bank)
  - Ook wel ‘schuldgeld’
  - Kan ook in de vorm van wissels (papier)
- Digitaal geld: geld in digitale vorm
  - Dus *informatie* als geld
  - Meestal zijn het ‘bewijzen’ van *giraal* geld
  - Cryptogeld: Bitcoin en Ethereum zijn digitaal basisgeld (en nog 50 andere)

Monetair beleid *toen*

- Jaren lang een gouden standaard: munten en bankbiljetten waren inwisselbaar voor goud
- Bretton Woods (1944, start van IMF):
  - Vaste wisselkoersen t.o.v. dollar (alleen door overheden beperkt aanpasbaar)
  - Alleen dollar inwisselbaar voor goud
  - Enigszins gebaseerd op Keynes idee van een uitwisselingsmunt: de ‘bancor’
- USA heeft het contract verbroken:
  - Tussen 1971 en 1973, in ‘73 wisselde UK 3miljard dollar in goud om
  - I.v.m. Vietnam oorlog
  - “Nixon shock”: munten gingen zweven t.o.v dollar

Monetair beleid *nu*

- Centale Bank (ECB) koopt (dubieuze) leningen op met geld dat zij daarvoor creeert. Zo schept ECB *basisgeld*.
- Banken kunnen ook lenen bij CB: dat is deels ook creatie van basisgeld.
- De banken creeren *giraal* geld, zij kunnen zelf beslissen een klant een lening te geven (bv € 10.000)
- De bank past zijn balans als volgt aan:
  - Activa: + 10.000 debiteuren
  - Passiva: +10.000 rekening courant tegoed
- Kapitaaleisen Basel III: solvabiliteit
  - eigenvermogen/geleend vermogen> 7% (wordt verhoogd)

Bezwaren tegen huidige systeem

- Geldbewaring en transport zijn in handen van particuliere bedrijven (banken) die niet helemaal te vertrouwen zijn
- Banken creeren zelf geld en beinvloeden zo onze economie
- Wij kunnen maar €100.000 safe op de bank bewaren
  ("Staatsgarantie" banken moeten dit gezamenlijk betalen)
- We bezitten bijna geen basisgeld, alleen *claims*: tegoeden bij banken
- Betalingsverkeer is een essentiele functie in de maatschappij en de infrastructuur is handen van de banken!!!
- Voor banken is betalingsverkeer eigenlijk een bijproduct en daardoor ook niet efficient!!!

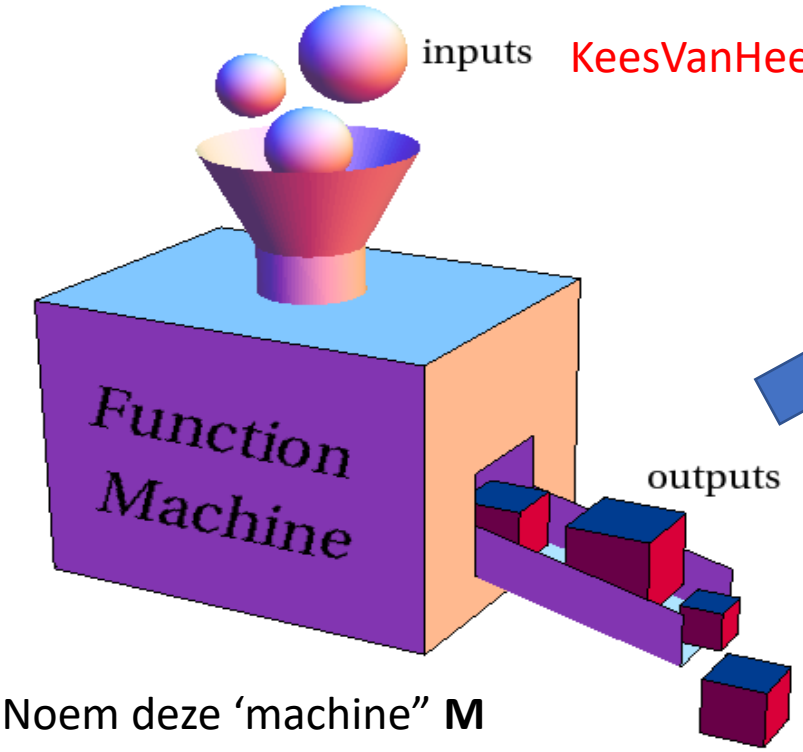
Tegenbewegingen

- Positive Money in UK, gestart 2010
- Stichting Ons Geld, gestart 2012
- Burger Initiatief Stichting Ons Geld:
  - 2015 (113.000 handtekeningen)
  - George van Hout (toneelstuk: De Verleiders)
  - Ontvankelijk verklaard door Tweede Kamer
  - WRR doet NU onderzoek
- Bezwaren van deze clubs:
  - Men wil geen schuldgeld meer (positive money only)
  - Men wil niet dat banken geld kunnen scheppen
  - Men wil dat het geld dat CB schept niet alleen naar de banken gaat maar direct naar de burgers en bedrijven
  - Men wil dat ons geld veilig is opgeborgen en geen risico loopt als bank failliet gaat

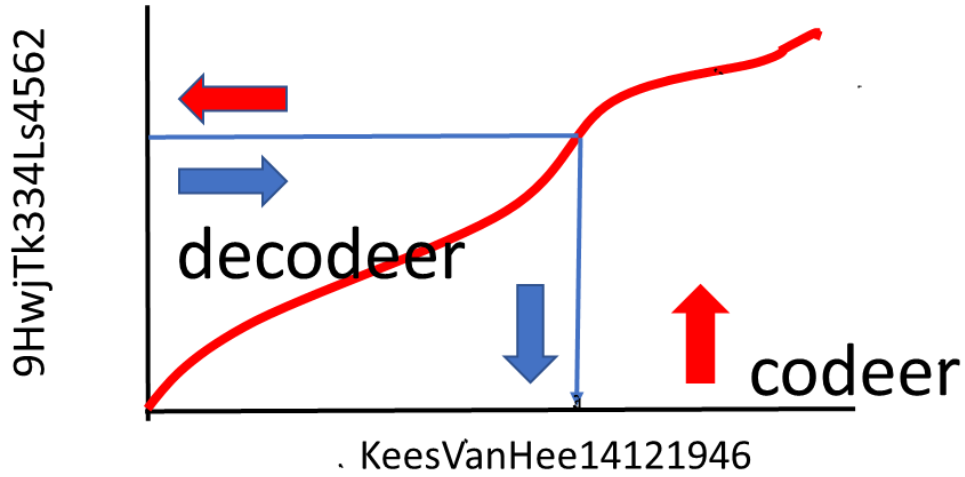
Cryptografie in een notendop

- Twee geheimschrift principes:
  - *Sleutel functies*
  - *Hash functies*

M (KeesVanHee14121946) = 9HwjTk334Ls4562



We noteren dit als:



Noem deze 'machine" **M**

Versleuteling: *public key infrastructure* (PKI)

- Deelnemers hebben 2 sleutels een *publieke* **P** en *geheime* **S** (secret)
- Publieke sleutel staat in een ‘telefoonboek’ (certificaat van TTP)
- Sleutels werken als *functies*: de ene is de inverse van de andere:

S(KeesVanHee14121946)=9HwjTk334Ls4562

P(9HwjTk334Ls4562)=KeesVanHee14121946

- en ook

P(KeesVanHee14121946)=76TTRiKls34OpQ

S(76TTRiKls34OpQ)=KeesVanHee14121946

- We noteren dit als: **P**(**S**(bericht))=bericht

Codes zijn gebaseerd op de *discrete logaritme:* zoek X zodat g^X=Y mod(p)

Digitale handtekening

Persoon X wil een bericht B naar persoon Y sturen zodat:

- alleen Y het kan lezen
- Y zeker weet dat het van X komt

1. X verstuurt naar Y: $\mathbf{P_Y}(\mathbf{S_X}(B,X))$
2. Y past eigen geheime sleutel toe: $\mathbf{S_Y}\ (\mathbf{P_Y}(\mathbf{S_X}(B,X)))= \mathbf{S_X}(B,X)$
3. Y zoekt publieke sleutel P_X van X op en past toe**:** $\mathbf{P_X}(\mathbf{S_X}(B,X))=[B,X]$

Y heeft nu het bericht B en weet dat het van X komt

Hash functie

- Een *hash* functie **H** lijkt op een sleutelfunctie, maar er is geen inverse functie
- Makkelijk **H**(X) te berekenen, maar practisch onmogelijk om voor gegeven Y een X te vinden zodat **H**(X)=Y
- Hash functies maken van een lange rij een korte (*vingerafdruk*)
- Voorbeeld van hashing:
  - Pas **H** toe op: 391846378289290945672289
  - Knip in blokken: 39184 |63782 | 89290 | 94567 |2289
  - *Tel de cijfers per blok op en blijf dit doen tot er maar één overblijft*
    3+9+1+8+4=25; 2+5= 7
  - De blokken worden: 7, 8, 1, 4, 3
  - Dus **H**(391846378289290945672289)= 78143
  - En **H**(391864387289290945672289)= 79643

- Er is een standaard hash functie uitgegeven door NSA is: SHA-256 (Secure Hash Algorithm)

SHA256(KeesVanHee14121946)=F7E35E5A5EA4531796279E5C07CCD4A87231CE6CD15E46A44D48ACA88C7AE91F

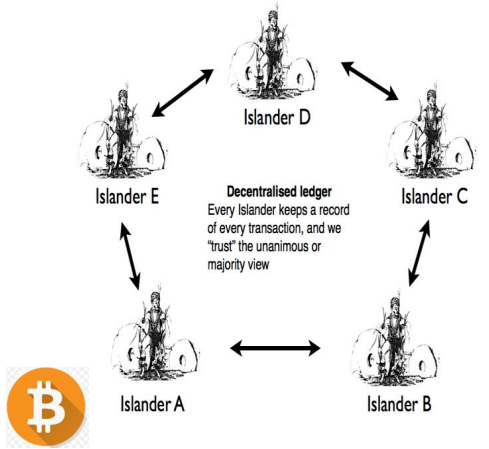
http://passwordsgenerator.net/sha256-hash-generator/

Basisbegrippen van de Bitcoin

- Je verdient nieuwe bitcoins als je een *zeldzaam getal* vindt
- Het kost veel rekenwerk zo'n *getal* te vinden
  Dat gebeurt door een *pool* van samenwerkende computers
- Het vinden van zo'n getal heet *mining*
- Het rekenwerk heet een *proof-of-work*
- Je bezit bitcoins in de vorm van een *transactie-record* van de transactie waarmee je ze verkregen hebt.
- Een *transactie-record* is een gestructureerd stuk tekst (*record*), b.v.:

  [*van*: X; *bedrag*: B; *naar* :Y; *datum*: D]

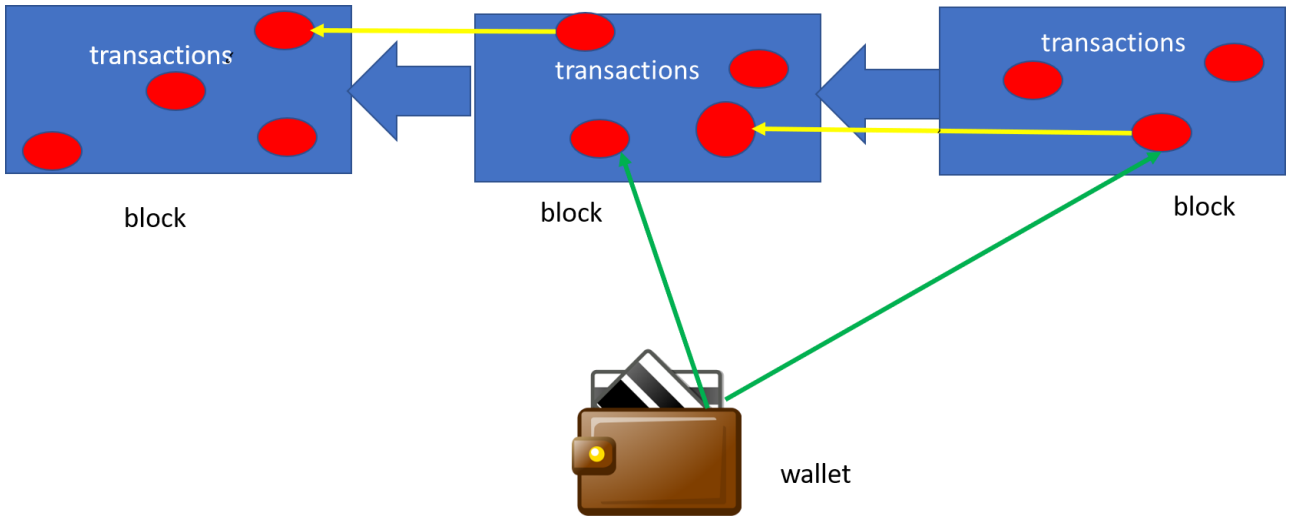
Wat is een eigenlijk?



- Een bitcoin is een *eenheid* van digitaal geld
- Er is geen *individuele* bitcoin: je kunt alleen een *aantal* bitcoins hebben
- Er zijn ook fracties van bitcoins: millibitcoin en de satoshi:0,00000001
- Je krijgt bitcoins in een *transactierecord.*
- De transacties staan de *blockchain*
  (een grote database die op veel computers staat en waarin niets veranderd kan worden, maar wel toegevoegd)

Je hebt een *wallet* waarin je bewaart:

- adres van je 'bitcoin', d.w.z. de lokatie van het transactierecord
- het aantal bitcoins in het transactierecord
- geheime sleutel die er bij hoort, om te bewijzen dat hij van jou is

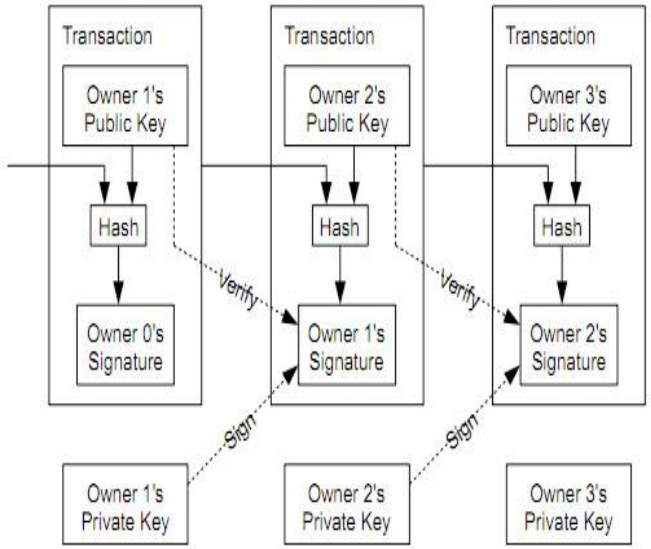


Bitcoin transactie

- Persoon X wil B bitcoins overdragen aan persoon Y
- Dan moet X een transactie T hebben waarin hij B verkregen heeft
- Dan creëert hij de volgende transactie-record T':

$$T' = [T, P_Y, B, \mathbf{S_X}(\mathbf{H}(T, P_Y, B))]$$

  - T is de vorige transactie, waarmee X de bitcoin heeft gekregen
  - P_Y de publieke sleutel die Y heeft gekozen voor de transactie; S_Y bewaart Y in zijn *wallet*
  - B is het aantal bitcoins van de transactie
  - $\mathbf{S_X}$ is geheime sleutel uit de wallet van X van de vorige transactie (P_X staat in T)
  - **H**(…) is een *vingerafdruk*
- Een transactie is natuurlijk alleen geldig als X minstens B eenheden heeft
- Transacties kunnen bestaan uit meerdere input records en meerdere output records (als een record niet genoeg saldo heeft dan meer inputs en als er wisselgeld is dan meer outputs)
- Y kan verder de 'bitcoin' weer uitgeven als hij S_Y maar bewaart in zijn *wallet*



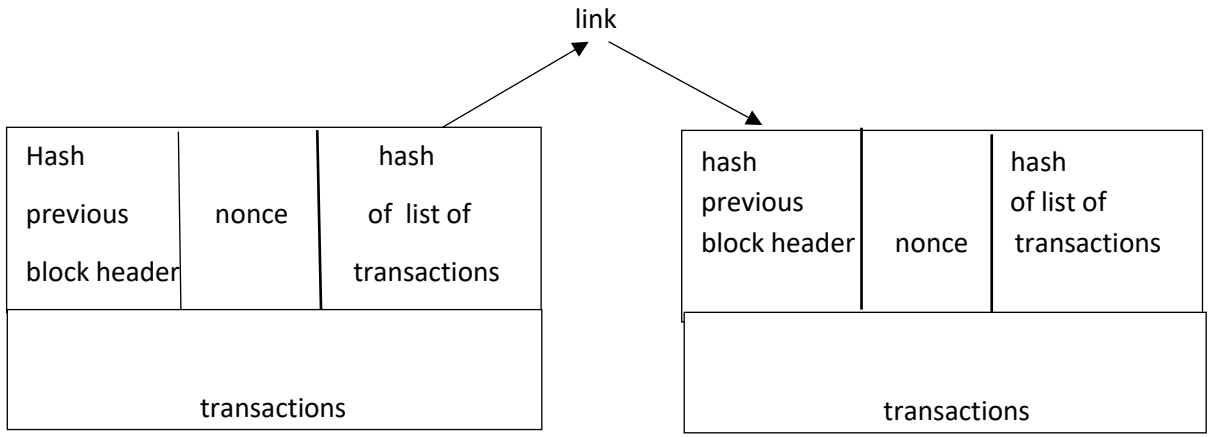
Schema uit het originele artikel van Satoshi Nakamoto 2008

Blockchain van transacties

- Probleem: Hoe voorkomen we dat je een bitcoin *twee* keer uitgeeft?
- Hiervoor is *blockchain* ‘uitgevonden’:
- Verzameling *datablocks* die naar elkaar verwijzen: nieuwe verwijst naar oude met hash van vorige block
- Elk block bevat transacties (ca 4000, die weer naar elkaar verwijzen)
- Elke transctie komt maar 1 keer voor en kan *nooit* meer gewijzigd worden
- Een transactierecord is geldig als er geen opvolger is waar de bitcoins in zijn gebruikt
- Om te zien of transactierecords geldig moeten dus alle nieuwe blocks gecontroleerd worden of er *geen volgende* transactierecord is

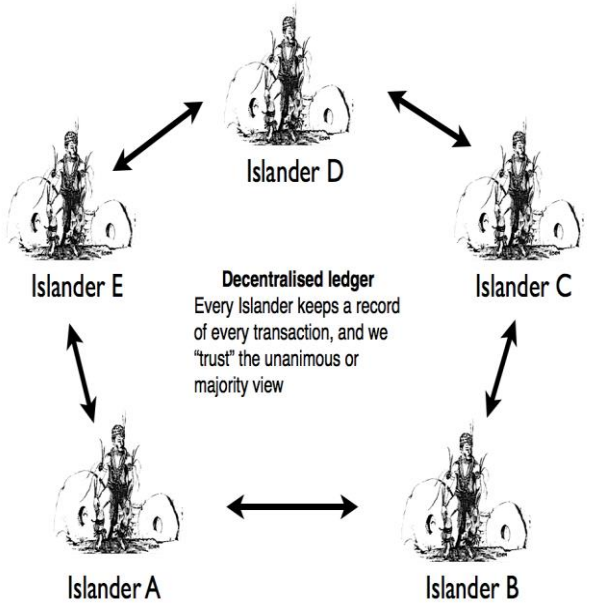
Blockchain:

- Blocks hebben
  - Kopregel
  - Transactie verzameling
- Kopregel:
  - Hash van vorig block
  - *Nonce:* getal dat berekend moet worden
  - Hash van dit block:
    hash former, transactions en nonce



Blockchain: database gedistribueerd over network

- Blockchain staat op alle computers van het network (P2P)
- Nieuwe transactie wordt aan iedereen gestuurd
- Alle computers in het network gaan de transactie *controleren*
- Alle goedgekeurde transacties verzamelen ze in een nieuw *block*
- De eerste die klaar is (*nonce* gevonden!) maakt het block officieel: stuurt naar allen en allen checken en bevestigen (eventueel stemmen)
- Blocks die niet af waren worden weer uitelkaar gehaald. Iedereen begint opnieuw met de nog niet verwerkte transacties
- Een nieuw block begint met een *start-transactie* van een miner die de goede nonce van het vorige block gevonden heeft.
- Elke 10 min een nieuw block.



Je kunt dit spel natuurlijk ook spelen met andere soorten transacties: handelstransacties, actes van de notaris of patienten dossiers

Het rekenwerk voor de miners

- Gebruik de standard hash functie SHA-256
- Laat B de inhoud van het block zijn
- Zoek een *nonce* X zodat **H**(B,X) begint met N nullen (nu 72)
- Kans op getal met 1 nul: 0.5 , met 72 nullen: $1/(2^{72}) \approx 2.5.10^{-22}$
- Dat kan alleen maar door alle getallen X vanaf 1 te proberen. Hoe groter N hoe moeilijker dat is!
- Per 2016 blocks wordt de moeilijkheidgraad (N) aangepast om zo te bereiken dat er elke 10 min een nieuw block komt
- Als je dat lukt krijg je een aantal bitcoins als beloning en een vaste fee
- Elke 210.000 blocks wordt de beloning gehalveerd (1 x per 4 jaar)
- In ca 2040 krijgt de miner alleen nog een fee!!
- Dan zullen er ca 21miljoen bitcoins zijn.
- De ‘gedolven’ bitcoins vormen de eerste transactie van een nieuw block
- Dat maakt bitcoins schaars!

Voor- en nadelen

Voordelen

- Bitcoins zijn anoniem
- Geen centrale organisatie
- Geen banken met grote overhead
- Digitaal geld, basisgeld dat niet door de overheid bijgedrukt kan worden

Nadelen

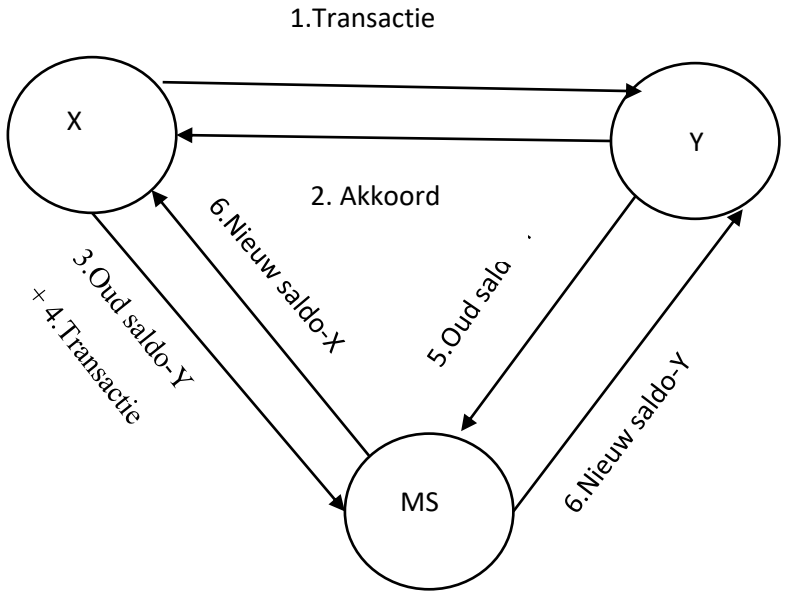
- Onduidelijk wat de munt eigenlijk is
- Systeem kan uitsterven als bitcoin zijn waarde verliest en mining te duur wordt
- Wisselgeld
- Veel rekenwerk, kost heel veel energie en veel tijd:* 2400MW=3 kerncentrales
- Daardoor ongeschikt voor kleine betalingen
  (10 min voor je zekerheid hebt!)
- Er zit een ‘duistere’ organisatie achter die de spelregels bepaalt

*vlg Prof Bart Preneel,
“Hoe warmen de bitcoins de aarde”
YouTube: Universiteit van Vlaanderen

Ons voorstel in een notendop (eind 2014 bedacht)

- Met dezelfde cryptografie
- Iedereen heeft één of meer *digitale kluisjes*
- We noemen ze *rekeningen* met elk een *saldo*
- Iedereen bewaart *zelf* zijn *saldo-records*.
- Maar mag dit ook uitbesteden aan een financiele dienstverlener
- Bij betaling van X naar Y wordt een *transactie-record* gemaakt
- Er is een *centrale organisatie* (MS) maar die bewaart *alleen*:
  - Rekeningnummers
  - Volgnummer van het actuele saldo (niet het saldo)
- Nodig om tweemaal uitgeven te voorkomen
- MS weet dus bijna niets van de klanten!

Transactieverwerking door MS



- **Transactie-record:**
  [van: X; bedrag: B; naar :Y; $S_X(X,B,Y);S_Y(X,B,Y)$]
- **Saldo-record:**
  [acnr: X; saldo: A; vlgnr: M; $S_{MS}(X,A,M)$]
- **MS-record:**
  [acnr:X; vlgnr: M; $S_{MS}(X,M)$]
- **Voorbeeld saldo:**
  [acnr:NL27INGB0001614778; saldo: €15.637,76;
  vlgnr:2017234; **********]
- **Voorbeeld transactie:**
  [van:NL27INGB0001614778; bedrag: €10,59;
  naar: NL43VLB1231834229; **********]

De *** is een versleuteling van het voorgaande. Daardoor te controleren met de P sleutel van MS. Maar alleen te veranderen door MS met zijn S sleutel**

Vergelijking met andere systemen

- Bitcoin houdt *transacties* bij in een blockchain (wisselgeld)
- Ethereum houdt *saldi* bij in een block chain (geen wisselgeld)
- Wij houden alleen *volgnummers* van rekeningen bij in een database
- Onze database mag gedistribueerd zijn: redundantie voor efficiency en veiligheid. Onze database mag een block chain zijn maar hoeft niet.
- Onze transacties zijn veel efficienter. Onze opslag is evenredig met aantal rekeningen , bij anderen met aantal transacties!

Wees voorzichtig met 

Dank voor uw aandacht!

Zie ook: *Een nieuw Monetair Systeem en een nieuw Monetair Beleid*
white paper: Kees van Hee en Jacob Wijngaard
www.robuustgeld.nl