Blockchain voor het hbo

Lectoraat Data Intelligence en
Lectoraat Optimaliseren Kennisintensieve
Bedrijfsprocessen – Zuyd Hogeschool

mr. Jehan Edriouch
dr. Roger Bemelmans
dr. ing. Martijn Zoet
drs. Jan Beumers

Zuyd Hogeschool

Zuyd Hogeschool

Blockchain voor het hbo

Lectoraat Data Intelligence en Lectoraat Optimaliseren Kennisintensieve Bedrijfsprocessen – Zuyd Hogeschool

mr. Jehan Edriouch
dr. Roger Bemelmans
dr. ing. Martijn Zoet
drs. Jan Beumers

Introductie

Stel jezelf eens de vraag hoe vaak je het afgelopen jaar de term Bitcoin hebt horen vallen. De kans is groot dat dit een hot topic was tijdens een verjaardagsfeest, borrel of zelfs tijdens het sporten. Kun je je nog herinneren hoeveel personen aanwezig waren die al over Bitcoin beschikten, er net uit waren gestapt, of op het punt stonden deze of een andere cryptomunt aan te schaffen? Misschien beschik je er inmiddels zelf wel over. Bitcoin is geen onomstreden munt en wordt vaak geassocieerd met "*the dark web*", criminaliteit, piramidespellen en de tulpenbollenmanie. Waar tijdens deze gesprekken niet altijd (uitgebreid) over wordt gesproken is de technologie die het fundament onder deze cryptocurrency vormt en echt als de belofte van deze tijd wordt geduid (zelfs als het nieuwe internet), "*Blockchain*" of "*de blockchain*".

Dit kan worden verklaard door het feit dat er weinig mensen zijn die je in begrijpelijke taal echt kunnen uitleggen wat deze technologie exact inhoudt en welke invloed deze zal hebben op bijvoorbeeld de financiële sector. Ook kan het zijn dat je juist de mensen hebt getroffen die je heel enthousiast meedelen hoe fantastisch blockchain is en dat hiermee alle wereldproblemen zullen worden opgelost, maar het daar verder bij laten.

Als je openstaat voor een kennismaking met blockchaintechnologie is het lezen van dit boekje een goede eerste stap. Het is geschreven voor de eerste 40 studenten die vanaf mei 2018 zullen deelnemen aan de Blockchain-minor van Zuyd Hogeschool. Deze studenten zullen op een praktische wijze kennismaken met blockchain en zelf blockchain-applicaties ontwikkelen. Dit boekje vormt een deel van de theoretische basis. Ben je geen student, maar wil je je toch in blockchain verdiepen, dan kan dat natuurlijk ook met dit boekje. We hebben bewust gekozen voor het beperken van de scope van dit eerste boekje en hebben ons als doel gesteld dat iedereen die dit boekje heeft gelezen in ieder geval de volgende vragen kan beantwoorden:

1. Wat is de geschiedenis van blockchain?
2. Wat is blockchain eigenlijk?
3. Hoe werkt blockchain?
4. Wat zijn de kenmerkende eigenschappen?
5. Wat zijn verschillende varianten en toepassingen van Blockchain?
6. Wat zijn smart contracts?
7. Wat is een DAO?
8. Wat is een ICO?
9. Hoe ziet het ecosysteem eruit?

In dit boekje beantwoorden we de belangrijkste 9 vragen die je helpen de basis van blockchain te begrijpen. Ons streven is dit boekje ieder studiejaar aan de hand van de belangrijkste ontwikkelingen en onze eigen praktijkervaringen te actualiseren. We pretenderen niet dat dit boekje allesomvattend is, maar helpen je voor nu op weg. Als je na het lezen niet (langer) bang bent om een inhoudelijk gesprek over blockchaintechnologie te voeren, én we erin zijn geslaagd je te motiveren nog meer over dit onderwerp te weten te komen, is ons doel bereikt.

Inhoudsopgave

1. Geschiedenis

Op 31 oktober 2008, rond het hoogtepunt van de financiële crisis, inmiddels alweer bijna 10 jaar geleden, publiceerde een persoon of groep personen, handelend onder de naam Satoshi Nakamoto (hierna: Satoshi), een whitepaper genaamd "*Bitcoin: A peer to peer electronic cash system*" [1].

Er wordt zowel online als offline druk gespeculeerd over wie deze Satoshi is, maar tot heden is zijn/haar/hun officiële identiteit nog niet formeel bevestigd. Voor het gemak verwijzen wij naar Satoshi als "*hij*".

In de whitepaper wordt gepleit voor een elektronisch betaalsysteem waarin (voor elkaar onbekende) partijen direct, zonder tussenkomst van een derde (zoals bijvoorbeeld een bank), transacties kunnen realiseren. Het betaalmiddel van dit systeem is de inmiddels befaamde Bitcoin. De allereerste Bitcoin-transactie dateert van 3 januari 2009.

Volgens Satoshi verbeterde zijn betaalsysteem het bestaande systeem op een aantal punten, zowel praktisch als principieel. De verbeterpunten die hij in het bestaande systeem constateerde waren de volgende:

- Te hoge transactiekosten vanwege de bemiddelingskosten die de bemiddelende/vertrouwde derde partijen in rekening brengen;
- Kosten die worden veroorzaakt door het niet kunnen terugdraaien van betalingen van diensten die al geleverd zijn en niet ongedaan gemaakt kunnen worden;
- De noodzaak van vertrouwen in de ander in situaties waarin prestaties wel kunnen worden teruggedraaid/ongedaan gemaakt kunnen worden;
- Het voor lief nemen van fraude.

Nu was het idee van een elektronisch betaalsysteem waarin partijen peer to peer, direct met elkaar, kunnen handelen niet revolutionair, althans in ieder geval niet nieuw.

De verbeteringen die door Satoshi zijn ingevoerd zijn het tegengaan van het zogenaamde double spending, het twee keer uitgeven van dezelfde munt, en het invoeren van een timestamp, een chronologisch bewijs dat de volgorde van transacties vaststelt. Deze verbeteringen zijn cryptografisch van aard en het is Satoshi gelukt om een aloud probleem dat wordt geduid aan de hand van het Byzantine Generals Problem op te lossen.

Hierdoor is het Satoshi gelukt een systeem te creëren waarin het vertrouwen is ingebed in de technologie, de blockchain. De veiligheid zit namelijk in de nodeverhouding, waarover later meer.

Deze twee aspecten bestaan natuurlijk ook in het bestaande elektronische betaalsysteem, maar worden hierin juist ondervangen door een Trusted Third Party (ook wel aangeduid als TTP), die iedere transactie op deze punten controleert. Juist de noodzaak van het bestaan van deze partij was een principieel bezwaar van Satoshi, omdat dit inhoudt dat de volledige macht volledig bij deze autoriteit (lees: de bank) ligt.

De oplossing van Satoshi is een systeem waarin bij een transactie de ontvanger er zeker van is dat de eerdere eigenaar geen eerdere transactie heeft uitgevoerd en dat de eerste transactie de enige geldige transactie is. Het komt erop neer dat voorkomen moet worden dat een eerder uitgegeven munt een tweede keer wordt uitgegeven.

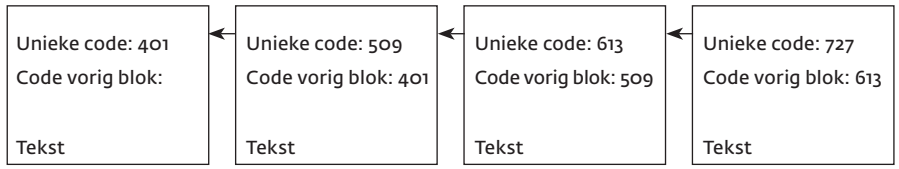
Om dit vast te kunnen stellen moet er sprake zijn van volledige transparantie, met andere woorden moeten alle transacties kunnen worden gecontroleerd.

In het Bitcoin-betaalsysteem zijn dan ook alle transacties zichtbaar. Deze transparantie is een van de onderscheidende kenmerken van de Blockchain.

2. Wat is blockchain?

Hoewel het fenomeen blockchain revolutionair is, zijn de technieken (lees: ingrediënten) waarmee een blockchain wordt gebouwd niet nieuw. Een blockchain is, precies zoals het woord aangeeft, een keten van blokken. De vraag is dan wel, wat zijn dat voor een soort blokken en hoe vormen die dan een keten? De blokken zijn in feite niets anders dan stukjes data (tekst) en ze vormen een keten doordat ieder stukje data een verwijzing heeft naar het vorige stukje data. We kunnen ons de keten voorstellen als een verzameling A4'tjes (pagina's) met tekst (iedere pagina is een blok), waarbij iedere pagina een eigen unieke code heeft en ook een verwijzing heeft naar de unieke code van een andere (de vorige) pagina (deze verwijzing zorgt er dus voor dat er een keten van blokken ontstaat).

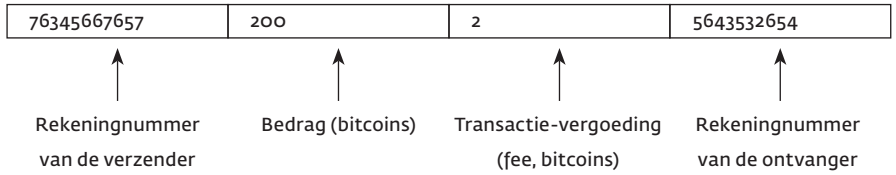
Een block in de Bitcoin-blockchain is maximaal 1 MB (1 Mega Byte = 1 miljoen bytes) groot. 1 byte bestaat uit 8 bits (een bit is een 1 of een 0). In 1 byte kunnen dus 2^8=256 verschillende waarden worden opgeslagen. Als ieder karakter (letter, cijfer, leesteken etc.) in een byte wordt opgeslagen, dan passen in 1 MB ongeveer 200 A4'tjes (5000 karakters per pagina).



Blockchain van pagina's. Het eerste blokje in de keten (het zogenaamde Genesis blok) is door Satoshi begin 2009 zelf aangemaakt.

Bitcoin is zoals gezegd een digitale munt en in de blokken worden transacties bijgehouden. Een transactie beschrijft wie hoeveel geld (lees: Bitcoins) aan wie overdraagt (zoals journaalposten in een grootboek). Een transactie is een record met een aantal velden.

Transactie



De rekeningnummers zijn unieke publieke sleutels, dit wordt verder toegelicht bij de paragraaf public-private key encryptie.

De transactievergoeding (fee) is een vergoeding voor de miners, dit wordt in hoofdstuk 3 "Werking Blockchain" verder toegelicht.

De uitdaging

Voor zover nog weinig spannends of revolutionairs aan de blockchain. Wat de blockchain van Satoshi zo bijzonder maakt is dat er geen centrale autoriteit (TTP= Trusted Third Party, bijvoorbeeld een bank) nodig is om de data (de pagina's) in de blockchain te bewaren, te bewaken of te controleren. Verder kan iedereen (betrouwbaar of onbetrouwbaar) meedoen en data (nieuwe pagina's met nieuwe transacties) aan de blockchain toevoegen (of dat in ieder geval proberen). We hebben dus een keten van blokken die nergens centraal wordt opgeslagen (iedereen die dat wil heeft een eigen lokale versie) en niemand controleert of iemand die een transactie wil toevoegen wel betrouwbaar is. De uitdaging is dus om het systeem zodanig in te richten dat alleen geldige transacties worden toegevoegd, zonder dat de persoon achter de transacties bekend is en zonder een centrale partij die de controles uitvoert.

Bitcoin weetjes (eind 2017)

- 1 satoshi (kleinste eenheid): 0,000.000.01 BTC
- Omvang van de blockchain: 90 GB
- Aantal blokken: 470.000
- Laatste blok met nieuwe Bitcoins: jaar 2140
- Geschat wordt dat 10-25% van alle Bitcoins verloren zijn (bijv. mensen die hun private key zijn vergeten).

Het probleem

Met een centrale partij (bijv. een bank) is het relatief eenvoudig een betrouwbaar en consistent grootboek bij te houden. De bank weet wie iedereen is en weet ook hoeveel geld iedereen heeft. Een transactie (geld overmaken van Alice naar Bob) gaat altijd via de bank. Alice geeft aan de bank door dat zij een bepaald bedrag, bijvoorbeeld, €100, wil overmaken naar Bob, de bank verlaagt dan het saldo van Alice (indien zij minstens €100 in bezit heeft) met €100 en verhoogt het saldo van Bob met €100. Makkelijk, betrouwbaar en de bank zorgt dat alles klopt. Dit werkt prima, zolang de bank betrouwbaar is (de bank is de TTP).

Wat nu als we een dergelijk systeem willen, maar dan zonder een TTP. Er zijn uiteraard meerdere redenen waarom een systeem zonder TTP gewenst is. Bijvoorbeeld omdat er simpelweg geen betrouwbare partij aanwezig is, of omdat er weinig tot geen vertrouwen is in een autoriteit (bijvoorbeeld in landen met een onbetrouwbare, corrupte, overheid).

Of omdat een TTP ook een single-point-of-failure kan betekenen, als bijvoorbeeld een bank failliet gaat of er wordt digitaal ingebroken (gehackt) dan zijn de gegevens (het geld) niet meer veilig.

Zonder centrale autoriteit is er ook geen centrale betrouwbare opslag. De uitdaging is dus om gezamenlijk een grootboek bij te houden, zodanig dat er consensus is over de geldige actuele inhoud. Waarbij het probleem is dat de deelnemers (personen die een transactie aan het grootboek willen toevoegen) mogelijk onbetrouwbaar zijn. Dit probleem staat ook bekend als het Byzantine Generals Problem [2].

Byzantine Generals problem

Het Byzantine Generals problem is het probleem van een aantal generaals die gezamenlijk een gecoördineerde aanval op de vijand willen uitvoeren. De generaals zijn allemaal verspreid (en moeten dus via boodschappers met elkaar communiceren), ook bestaat de mogelijkheid dat
een of meer generaals onbetrouwbaar zijn. De uitdaging voor de betrouwbare generaals is om overeenstemming te bereiken over een gezamenlijk aanvalsplan (in termen van blockchain, een gezamenlijk grootboek).

2 Generals problem:
Generaal A stuurt een boodschap naar generaal B met de mededeling dat hij om 15:00 uur aanvalt, maar verwacht wel een bericht terug dat generaal B meedoet. Generaal B stuurt bij ontvangst direct een boodschap terug dat hij ook om 15:00 uur aanvalt. Maar generaal B wil wel zekerheid dat generaal A zijn bericht ontvangen heeft. De boodschapper die van B terug naar A gaat kan immers onderweg overvallen worden, zodat het bericht van generaal B niet bij generaal A aankomt. Generaal A moet dus weer een bevestiging van ontvangst terugsturen, maar verwacht op zijn beurt ook weer een bevestiging van ontvangst. Het is inmiddels duidelijk dat dit nog een tijdje zo door kan gaan, maar dat het probleem niet oplosbaar is, de generaals blijven onderling op nieuwe bevestigingen wachten.

Het probleem bestaat eigenlijk uit 4 deelproblemen:

1. ***Volledigheid***. Een bericht van een generaal aan de anderen kan mogelijk niet aankomen (bijv. omdat de boodschapper wordt overvallen). In termen van blockchain: een geldige transactie bereikt niet de anderen en wordt niet opgenomen in het grootboek.
2. ***Authenticatie***. Een onbetrouwbare generaal kan zich voordoen als een andere generaal. In termen van blockchain: Chuck wil het geld van Alice aan Bob geven.
3. ***Integriteit***. De boodschap van een betrouwbare generaal kan vervalst worden (bijv. door de boodschapper). In termen van blockchain: Alice doet een geldige transactie maar de inhoud van de transactie wordt gewijzigd.
4. ***Consistentie***. Een onbetrouwbare generaal kan verschillende tegenstrijdige boodschappen versturen naar verschillende generaals. In termen van blockchain: Alice heeft weliswaar €10, maar geeft die zowel aan Bob als aan Charlie (double spending).

Wat blockchain in de kern doet is het Byzantine Generals Problem oplossen (in ieder geval in de praktijk, zie kader), in termen van blockchain: dat alle deelnemers overeenstemming hebben over de inhoud van het grootboek. Hiervoor worden een aantal cryptografische technieken gebruikt. Het eerste deelprobleem (niet arriveren van een bericht) wordt verder niet expliciet opgelost. In het internet-tijdperk kunnen berichten heel vaak, heel snel achter elkaar en naar heel veel ontvangers tegelijkertijd verstuurd worden, dan komt een bericht in de praktijk uiteindelijk toch wel aan.

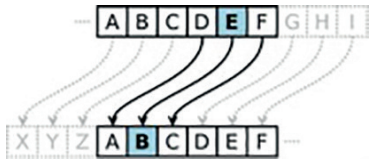
Cryptografie (geheimschrift) bestaat al meer dan 2000 jaar. Zo zouden de Spartanen omstreeks 500 v.Chr. al een stok (Scytale[3]) met een doek gebruiken om geheime boodschappen naar elkaar te kunnen versturen. Een kenmerk van cryptografie is dat er altijd een sleutel (bijv. wachtwoord) moet zijn om het versleutelde bericht weer te kunnen ontcijferen (decrypten). In het geval van de Spartanen was de stok de sleutel. Om een bericht te versleutelen (encryptie) werd een lint om een stok gewikkeld en vervolgens werd op dat lint de boodschap geschreven.

Eenmaal van de stok afgewikkeld is alleen een lint met onleesbare tekens zichtbaar. De ontvanger heeft uiteraard exact dezelfde stok nodig om het bericht weer te ontcijferen (door het lint om de stok te wikkelen en de tekst te lezen zoals die geschreven is).

Figuur 1

Julius Caesar maakte ook gebruik van encryptie om geheime boodschappen te versturen. Alle letters werden vervangen door de letters 3 plekken verder in het alfabet (een A werd een D, een M een P en een Y een B). Het woord “encryptie” wordt zo “hqfubswlh”. In het geval van Caesar is de sleutel het getal 3. Alle vertrouwelingen wisten dat de onleesbare tekst weer leesbaar werd door alle letters 3 plaatsen terug te zetten.

Figuur 2



Figuur 1: *https://commons.wikimedia.org/wiki/File:Skytale.png*
Figuur 2: *https://commons.wikimedia.org/wiki/File:Caesar_cipher_left_shift_of_3.svg*

De oplossing

Om het Byzantine Generals Problem op te lossen maakt blockchain o.a. gebruik van een aantal reeds bestaande cryptografische technieken [4]: public-private key encryptie, cryptografische hashfuncties en hashcashen [5].

Public-private key encryptie

Dit is een vorm van versleuteling waarbij er twee sleutels nodig zijn, een publieke sleutel en een privésleutel. Deze twee sleutels horen bij elkaar (vormen een paar) en zijn elkaars tegenhanger (ook wel asymmetrische encryptie genaamd). Er zijn verschillende algoritmen (recepten), zoals het RSA algoritme [6], waarmee deze twee sleutels aangemaakt kunnen worden. Moderne cryptografische recepten zijn meestal gebaseerd op priemgetallen (zie kader).

Het bijzondere aan de public-private key paren is dat een bericht dat versleuteld is (encrypted) met de publieke sleutel alleen ontcijferd (decrypted) kan worden met de bijbehorende privésleutel, en vice versa (dus encrypted met de privésleutel, dan alleen decrypten met de publieke sleutel).

Priemgetallen

Een priemgetal is een getal dat enkel door zichzelf en door 1 gedeeld kan worden. Getal 1 wordt zelf, vanwege praktische redenen, geen priemgetal genoemd.

Ieder niet-priemgetal kan geschreven worden als de vermenigvuldiging van een aantal priemgetallen en dat kan maar op 1 manier. Bijvoorbeeld: 84=2*2*3*7 (en dat kan alleen op deze manier).

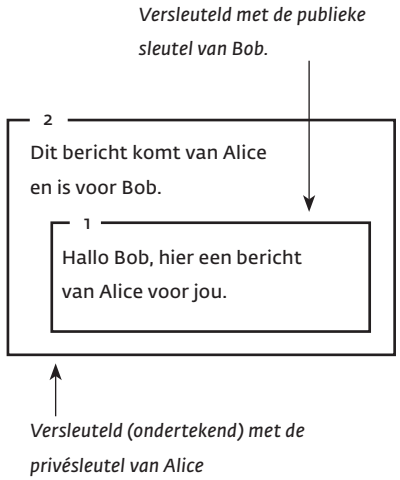
Moderne cryptografie is gebaseerd op het feit dat een groot getal ontbinden in priemgetallen zeer veel tijd kost (zelfs voor een computer). Een bericht versleuteld met priemgetallen is daardoor erg lastig te kraken.

Een bericht proberen te ontcijferen met iedere willekeurige andere sleutel leidt enkel tot een nog steeds onleesbaar bericht (maar nooit, in de praktijk, tot het oorspronkelijke bericht). Dit geeft (naast het versleutelen van berichten) nog een extra mogelijkheid, namelijk authenticatie (is iemand wel wie hij zegt dat hij is, een digitale handtekening zetten). Stel Alice heeft een privésleutel (die zij dus netjes voor zichzelf houdt) en een publieke sleutel die zij aan iedereen vertelt. Alice wil nu een bericht sturen naar Bob, maar wel zodanig dat:

- Duidelijk is dat het bericht ook echt van Alice komt.
- Alleen Bob het kan lezen.

Dan kan op de volgende manier:

1. Eerst versleutelt Alice het bericht met haar privésleutel. Alice zet dus haar digitale handtekening, zodat duidelijk is dat dit bericht alleen van Alice kan komen (alleen met de publieke sleutel van Alice is dit bericht immers te lezen).
2. Vervolgens versleutelt Alice dat bericht, samen met de tekst "Dit bericht komt van Alice en is voor Bob", met de publieke sleutel van Bob.



Bericht 2 is voor iedereen onleesbaar en kan alleen ontcijferd worden met de privésleutel van Bob. Dan wordt de tekst "Dit bericht komt van Alice en is voor Bob" leesbaar, terwijl het getekende bericht ("Hallo Bob, hier een bericht van Alice voor jou") onleesbaar blijft.
Nu gebruikt Bob de publieke sleutel van Alice om bericht 1 te ontcijferen. Als dat leidt tot leesbare tekst is duidelijk dat het bericht inderdaad van Alice afkomstig is (niemand anders is immers in staat een bericht te versleutelen dat met de publieke sleutel van Alice ontcijferd kan worden).
Met public-private key encryptie is

deelprobleem 2 (authenticatie) van het Byzantine Generals Problem opgelost. Met public-private key kan een digitale handtekening gezet worden, bijvoorbeeld door de publieke sleutel zelf te versleutelen met de privésleutel.

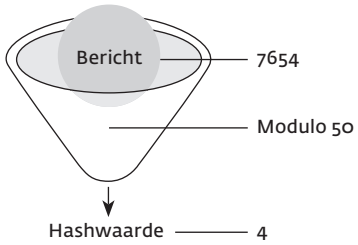
In het voorbeeld op de vorige pagina is de 2-de versleuteling (het onleesbaar maken van het bericht voor iedereen behalve Bob, met de publieke sleutel van Bob) niet nodig in de context van het Byzantine Generals Problem, maar is weergegeven om de mogelijkheden te laten zien.

Cryptografisch hashen

Hashen is het omzetten van een stuk tekst van willekeurige (grote) omvang naar een (kleiner) stuk tekst met een vaste lengte. Een eenvoudig voorbeeld is een groot getal hashen door modulo 50 toe te passen (modulo 50 geeft de rest na deling door 50), dus 7654 modulo 50 geeft 4 en 48 modulo 50 geeft 48. De hashwaarde is de uitkomst van het hashen. Cryptografisch hashen is hashen, maar dan met 2 bijzondere eigenschappen:

1. Het oorspronkelijke bericht is niet terug te herleiden uit de hashwaarde (dus eenrichtingsverkeer).
2. Verschillende berichten leiden niet tot dezelfde hashwaarde (in de praktijk, in theorie kan dat wel).

Cryptografisch hashen wordt toegepast om de integriteit van data te garanderen (dus voorkomen dat data ongemerkt gewijzigd kan worden).



Stel, Alice wil een groot bestand (bijvoorbeeld een film) verzenden naar Bob. Omdat het bestand te groot is besluit Alice het in meerdere kleine stukjes te verzenden, Bob ontvangt de film dus in meerdere kleinere porties. De vraag is hoe Bob weet of hij alle stukjes correct heeft ontvangen. Alice geeft hiervoor aan Bob de cryptografische hashwaarde van het oorspronkelijke bestand (dus van de hele film) door. Als Bob denkt alle kleinere porties binnen te hebben kan hij makkelijk controleren of hij inderdaad de hele film (zonder fouten) binnen heeft gekregen. Hij voegt de stukjes samen en neemt dan de hashwaarde van de samengevoegde stukjes (de film) en vergelijkt die met de hashwaarde die Alice hem eerder heeft gegeven. Komen die hashwaarden overeen dan heeft Bob de film in zijn geheel correct ontvangen, alleen de film van Alice leidt immers tot precies die hashwaarde. Er zijn verschillende algoritmen (recepten), zoals het SHA algoritme [7], waarmee van een bericht de hashwaarde berekend kan worden.

Hiermee is deelprobleem 3 (integriteit) van het Byzantine Generals Problem opgelost, in hoofdstuk 3 (werking blockchain) zien we dat het ook gebruikt wordt, samen met hashcashen (zie beneden), om deelprobleem 4 (double spending) op te lossen.

Hashcashen

Hashcashen is een arbeidsintensief proces (Proof of Work, veel rekenwerk voor een computer), waarmee een bepaalde cryptografische hashwaarde wordt gezocht. Zoals eerder gesteld is cryptografisch hashen eenrichtingsverkeer: van een (groot) bericht kan makkelijk de hashwaarde bepaald worden, maar op basis van een hashwaarde is het praktisch onmogelijk een bijbehorend bericht te vinden. In theorie zijn er uiteraard meerdere grote berichten die allemaal leiden tot dezelfde kleinere hashwaarde, in de praktijk is het echter nagenoeg onmogelijk een van die berichten te vinden. Hashcashen is eigenlijk het omgekeerde van cryptografisch hashen. De hashwaarde is bekend en nu moet een bijbehorend bericht bepaald worden. In de praktijk is dit echter onmogelijk. Bij hashcashen wordt dan ook geen specifieke hashwaarde gebruikt (zoals 130), maar eisen waaraan de hashwaarde moet voldoen (bijv. kleiner dan 130). In het volgende hoofdstuk zien we hoe hashcashen gebruikt wordt om te voorkomen dat iemand zijn geld 2 keer uitgeeft.

3. Werking Blockchain

Voordat we de werking van blockchain gaan bespreken passen we eerst de schematische weergave van de blockchain aan met de besproken cryptografische technieken. In hoofdstuk 2 hebben we een blok als volgt beschreven:

Unieke code: 727 Code vorig blok: 613 Tekst

De unieke code wordt nu de cryptografische hashwaarde van het gehele blok (en kan dus niet in het blok zelf worden opgeslagen). De 'code vorig blok' is de cryptografische hashwaarde van het gehele vorige blok. De inhoud van ieder blok, de data waar het om gaat, is een lijst met transacties. Een blok ziet er dan als volgt uit:

Hashwaarde vorig blok <lijst met cryptografisch ondertekende geldige transacties> Nonce

Nonce: willekeurig nummer. De nonce wordt gebruikt om een blok te maken waarvan de cryptografische hashwaarde voldoet aan bepaalde eigenschappen (in Bitcoin bepaald door de Difficulty).

Een blockchain bestaat uit een aantal blokken, waarbij het aantal blokken in de tijd toeneemt (de keten wordt langer). Er is geen centrale blockchain (geen TTP), maar iedereen heeft een eigen lokale versie van de blockchain. De uitdaging is ervoor zorgen dat al die lokale versies wel dezelfde gegevens bevatten (dat de Byzantine Generals over hetzelfde aanvalsplan beschikken; dat iedereen hetzelfde grootboek hanteert).

Iedereen kan een nieuw blok aan de keten toevoegen (in ieder geval aan zijn of haar eigen lokale keten). De uitdaging is te zorgen dat alle anderen ook dat blok toevoegen aan hun lokale versie, zodat iedereen weer dezelfde (nu 1 blok langere) keten heeft.

Bij een blockchain zijn er meerdere partijen die een rol spelen (zie kader), in de praktijk kunnen deze verschillende rollen gecombineerd zijn, of nog verder uitgesplitst. In de uitleg hieronder gaan we uit van 3 rollen: beheerders, miners en gebruikers.

Blockchain stakeholders

Bij het gebruik van een blockchain zijn in de praktijk meestal meerdere soorten gebruikers betrokken.

- ***Ontwerpers***. Dit zijn de mensen die het systeem bedenken en de regels afspreken en vastleggen.
- ***Ontwikkelaars***. Dit zijn de mensen die het ontwerp realiseren, die de software bouwen.
- ***Database beheerders***. Dit zijn de mensen die een volledige blockchain lokaal hebben opgeslagen en onderhouden.
- ***Werkers (miners)***. Dit zijn de mensen (computers) die nieuwe blokken aanmaken en aan de beheerders vragen deze blokken toe te voegen aan hun keten.
- ***Gebruikers (klanten)***. Dit zijn de mensen die die transacties willen doen, elkaar geld willen overmaken.

In de basis gelden, voor iedere blockchain, 3 hoofdregels:

1. Alle eerlijke deelnemers houden zich aan de hoofdregels.
2. Alleen geldige blokken worden aan de blockchain toegevoegd.
3. Als er verschillende geldige ketens zijn, dan geldt de langste keten (met de meeste blokken) als de geldige.

Een nieuw blok toevoegen aan de blockchain

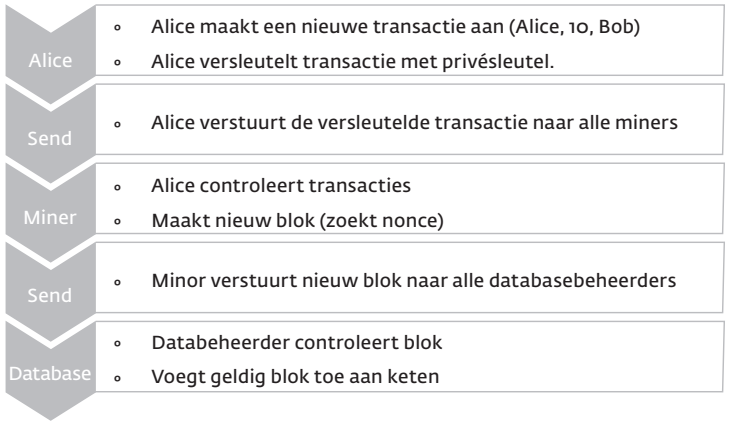
Om een nieuw geldig blok toe te voegen aan de keten moeten 3 gegevens ingevuld worden (hashwaarde vorig blok, lijst met transacties en de nonce). Verder geldt dat de hashwaarde van een blok aan bepaalde voorwaarden moet voldoen (bijv. de hashwaarde moet kleiner zijn dan 1000).

1. Hashwaarde vorig blok. Dit is eenvoudig, de hashwaarde kan snel berekend worden met bekende algoritmen (bijv. SHA).
2. Lijst met nieuwe geldige transacties. Een transactie is geldig als de persoon die het bedrag wil overmaken ook minstens dat bedrag in bezit heeft. Iedere miner bepaalt zelf welke transacties hij wil opnemen in zijn blok.
3. Nonce. De waarde van de nonce

(een willekeurig getal) moet zodanig zijn dat de hashwaarde van het gehele blok (hashwaarde vorig blok + lijst met transacties + nonce) aan de voorwaarden voldoet. De nonce bepalen (hashcashen dus) is een kwestie van uitproberen. Dit is wat de miners doen, een geldige nonce zoeken.

Stel een gebruiker wil een transactie doen, Alice wil €10 overboeken naar Bob.

- Alice maakt eerst zelf een transactie aan (een bericht dat zij €10 wil overmaken aan Bob) en versleutelt deze transactie met haar privésleutel.
- Vervolgens verstuurt Alice de versleutelde transactie naar alle miners.
- De (iedere) miner controleert de ontvangen transacties op geldigheid: Met de publieke sleutel van Alice kan gecontroleerd worden of de transactie echt van Alice komt en met een geldige keten kan het actuele saldo van Alice bepaald worden.
- De miner plaatst de geldige transacties in een nieuw blok en zoekt naar een geldige nonce.
- Zodra de miner een geldige nonce heeft gevonden (en daarmee een geldig nieuw blok) stuurt de miner het blok naar de database beheerders.
- De databasebeheerders controleren het blok zelf ook op geldigheid en voegen het daarna toe aan hun eigen lokale keten.



Drijfveer voor miners

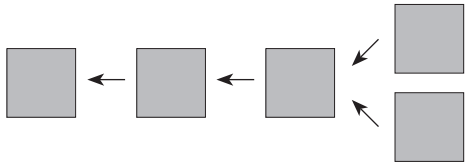
De miners verrichten het zware werk, zij stellen hun computers ter beschikking om transacties te controleren en nieuwe blokken aan te maken. Iedereen kan minen, dat kan met eigen software maar daarvoor zijn ook veel (en meestal gratis) softwareprogramma's beschikbaar [8]. Miners krijgen een vergoeding voor het werk dat zij verrichten. Iedereen die een transactie doet geeft ook aan hoeveel van die transactie voor de miner bedoeld is (een fee als transactievergoeding). Daarnaast ontvangt de miner die een nieuw blok aan de blockchain toevoegt nog een extra fee van (op dit moment) 12,5 nieuwe bitcoins. Er zijn in totaal 21 miljoen Bitcoins en alleen miners kunnen nieuwe Bitcoins krijgen. Als alle Bitcoins op zijn krijgen de miners alleen nog de transactie-fees. Om de fees te krijgen voegt de miner zelf een laatste transactie toe aan het blok (uiteraard voordat hij de juiste nonce gaat zoeken) waarin hij zichzelf de fees toekent.

Betrouwbaarheid van de miners

Een onbetrouwbare miner kan de nonce van een eerlijke miner niet gebruiken voor zijn eigen blok, de nonce werkt immers alleen voor de transacties in de lijst van de eerlijke miner. In die lijst staat ook de transactie dat de eerlijke miner alle fees krijgt.

Stel dat een onbetrouwbare Chuck van Alice €10 heeft gekregen. Om te voorkomen dat Chuck de transactie (die door Alice met haar privésleutel ondertekend is) zelf nogmaals naar de miners stuurt (en zo nog eens €10 probeert te ontvangen) wordt aan een transactie een soort volgnummer toegevoegd (2 keer hetzelfde volgnummer mag niet). In de bitcoin blockchain is het volgnummer (UTXO, Unspent transaction Output) een verwijzing naar eerdere transacties, die opgeteld groter of gelijk zijn aan het transactiebedrag.

Het kan voorkomen dat 2 verschillende miners op hetzelfde moment allebei een geldig blok naar de database beheerders sturen. De database beheerders maken dan tijdelijk een splitsing aan in de keten, waardoor eigenlijk 2 ketens ontstaan (waarvan alleen het laatste blok anders is).



De miners zullen echter per nieuw blok dat ze minen moeten kiezen voor een van de 2 takken, omdat in een blok een verwijzing staat naar het vorige blok (namelijk de cryptografische hashwaarde van het vorige blok). Omdat verder de afspraak is dat de langste keten de geldige is, hebben alle miners er belang bij om aan de dezelfde (langste) keten te werken. Al het werk dat verricht wordt aan een tak die uiteindelijk niet in de langste keten blijkt te zitten, is immers voor niets geweest. Alleen de langste keten is de geldige en dus zijn ook alleen alle transacties en alle fees in die keten geldig (alle andere takken bestaan in feite niet).

Betrouwbaarheid van de blockchain

Chuck heeft een bedrag van €100 overgemaakt aan Bob en heeft inmiddels van Bob een tegenprestatie (bijv. een telefoon) ontvangen. Stel dat de transactie hieronder in blok 3 is opgeslagen. Als Chuck nu probeert die transactie ongedaan te maken (dus de telefoon houden, maar het geld ook), dan moet hij daarvoor de inhoud van blok 3 aanpassen. Als hij dat doet dan verandert ook de cryptografische hashwaarde. Dan is (zeer waarschijnlijk) ook de nonce niet meer geldig. Dan wordt het lastiger voor Chuck, hij moet nu immers een nieuwe geldige nonce gaan zoeken voor blok 3 en de nieuwe hashwaarde die daaruit weer volgt komt in blok 4 (daar staat immers de hashwaarde van het vorige blok). Maar als hij in blok 4 iets verandert dan klopt ook daar de nonce niet meer en moet hij ook voor blok 4 een nieuwe nonce gaan zoeken. Het is inmiddels duidelijk dat dit ook voor blok 5 geldt. Aangezien de langste keten de geldige keten is, kan Chuck dit alleen winnen als hij sneller is dan alle andere miners samen (en de huidige langste keten inhaalt).

Dus zolang de rekenkracht van alle eerlijke miners groter is dan de rekenkracht van oneerlijke miners, zullen de oneerlijke miners niet in staat zijn de langste keten in te halen en blijft de langste keten de eerlijke geldige keten, waarmee ook deelprobleem 4 (consistentie) van het Byzantine Generals Problem opgelost is.

Hoe eerder in de keten hoe kleiner de kans dat een blok aangepast kan worden (daarvoor moeten immers alle blokken die daarop volgen ook aangepast worden).

Betrouwbaarheid van de database beheerders

Voor de databasebeheerders geldt hetzelfde als voor de miners. Zolang minstens de helft van alle beheerders eerlijk is, blijft de langste keten de geldige. Eerlijke miners gaan dan uit van de database (blockchain) die de meeste database beheerders hebben. In tegenstelling tot de miners krijgen de database beheerders geen vergoeding voor het bijhouden en opslaan van de blockchain.

4. Kenmerkende eigenschappen

Wanneer is er sprake van een blockchain of van dé Blockchain? Hierover bestaat discussie en deze vraag laten we voor nu in het midden. Wij vinden het wel van belang om in ieder geval de kenmerken van blockchain in de context van Bitcoin op een rijtje te zetten:

- ***Geen centrale opslag***. De blockchain wordt door iedereen lokaal opgeslagen, er is geen centrale database en geen centrale controle (TTP).
- ***Niet wijzigbaar.*** Transacties die in de blockchain zijn opgenomen zijn niet meer aanpasbaar. Er kunnen alleen nieuwe transacties aan het einde van de keten worden toegevoegd (met een nieuw blok).
- ***Onmogelijkheid op foutcorrectie.*** Het niet kunnen aanpassen van de bestaande blokken betekent overigens ook dat er geen foutcorrecties kunnen plaatsvinden. Een transactie ongedaan maken kan alleen als de ontvanger het bedrag terugstort.
- ***Energieverbruik (Proof of Work)***. De miners zorgen ervoor dat de database consistent blijft. De miners controleren of de transacties geldig zijn, maar het echte werk zit in het zoeken naar een geldige nonce. De prijs die betaald wordt voor de afwezigheid van een TTP is in feite het energieverbruik van de miners bij het zoeken naar een geldige nonce.
- ***Omloopsnelheid***. Het vinden van een geldige nonce duurt ongeveer 10 minuten, gegeven de totale rekenkracht van alle miners samen. Hoe meer miners hoe sneller de nonce gevonden zou worden. Om gemiddeld toch op 10 minuten uit te komen wordt de voorwaarde waaraan de hashwaarde moet voldoen aangepast (de voorwaarde heet bij Bitcoin de Difficulty). Consequentie hiervan is dat een transactie pas na 10 minuten in de blockchain wordt opgenomen, dan weet de ontvanger van de transactie ook pas (met enige zekerheid) dat het geld op zijn rekening is bijgeschreven.
- ***Schaalbaarheid***. De blokken in de blockchain van Bitcoin hebben nu een maximale grootte van 1 MB. Dat betekent dat mogelijk niet alle transacties in het volgende nieuwe blok kunnen worden meegenomen. Bij veel gebruikers (veel transacties) is de consequentie dat het mogelijk veel langer dan 10 minuten kan duren voordat een transactie in de blockchain wordt opgenomen. Het vergroten van de blokken is een mogelijkheid, nadeel daarvan is echter dat dan meer opslagcapaciteit nodig is om de blockchain te bewaren.
- ***Privacy***. Omdat alle transacties op geldigheid gecontroleerd moeten worden is volledige transparantie nodig. Van iedereen (in ieder geval van ieder rekeningnummer) is bekend hoe groot het totale saldo is, maar ook wie wanneer van wie welk bedrag heeft gekregen en dus ook wie wanneer aan wie welk bedrag heeft gegeven.
- ***Semi-democratisch gehalte***. Blockchain is een systeem zonder centrale autoriteit, er is geen centrale partij die de spelregels bepaalt en toeziet op naleving van de regels. Maar toch zijn er spelregels en soms worden de regels aangepast (bijv. dat de maximale blokgrootte naar 2MB gaat). Dit gebeurt door allerlei discussies op internetfora. Zodra een grote groep het met elkaar eens is worden de regels aangepast (door de ontwerpers), vervolgens passen de ontwikkelaars de software aan en zorgen de databasebeheerders en miners dat zij de regels volgen. Het komt ook voor dat er geen consensus bereikt wordt en dan splitsen de groepen zich (hard fork).

Dan ontstaat de situatie dat er 2 (of meer) aparte blockchains zijn, met hun eigen regels en munteenheid. Op 1 augustus 2017 is dit met Bitcoin gebeurd, waardoor een nieuwe blockchain is ontstaan met de naam Bitcoin Cash.

- ***Kwetsbaarheid***. Het systeem is betrouwbaar zolang meer dan 50% van de miners betrouwbaar is. Als een groep onbetrouwbare miners 50% of meer van de totale rekenkracht kan opbrengen dan kunnen zij de blockchain aanpassen (een nieuwe langste keten produceren).
- ***Sleutel kwijt is geld kwijt***. Transacties moeten met de privésleutel van de verzender worden versleuteld, anders is een transactie niet geldig. Als iemand zijn privésleutel kwijtraakt is het totale saldo van dat rekeningnummer voor altijd verloren.

Het grote voordeel van een blockchain is dat er geen centrale autoriteit (TTP) nodig is, met de afwezigheid van een TTP is er ook geen single-point-of-failure (spof). Een centrale partij (spof, single point of failure) kan aangevallen worden waardoor deze onbereikbaar is, er kan worden ingebroken of er kunnen andere problemen optreden bij de centrale partij waardoor het hele systeem platligt.

Blockchain bevat in de kern eigenlijk geen nieuwe technieken, maar realiseert door op een slimme manier bestaande cryptografische technieken te combineren, een openbare gedistribueerde database die nagenoeg niet te kraken is.

5. Verschillende toepassingen en varianten

De beschrijving van het concept blockchain is in de eerdere hoofdstukken vooral gebaseerd op de wijze waarop een blockchain in de cryptocurrency Bitcoin wordt gebruikt. Er zijn echter meerdere manieren waarop een blockchain kan worden gerealiseerd, maar er zijn ook meerdere toepassingsmogelijkheden (naast cryptocurrencies). Deze toepassingen en varianten komen in dit hoofdstuk aan bod, maar voordat we deze behandelen wordt eerst een blik geworpen op een belangrijk onderscheidend kenmerk van blockchaintechnologie, namelijk het consensusmechanisme.

Proof of Work

De betrouwbaarheid van de Bitcoinblockchain is voornamelijk gebaseerd op het werk van de miners (het vinden van de juiste nonce), die ervoor zorgen dat er overeenstemming is over de juiste inhoud van de blockchain. Het model om overeenstemming te realiseren wordt een consensusmodel genoemd en in het geval van Bitcoin heet dat consensusmodel Proof of Work. Het nadeel van Proof of Work is dat het veel rekenkracht (energieverbruik) kost, dat is dus feitelijk de prijs die betaald wordt om overeenstemming te bereiken over de juiste inhoud van de database (de blockchain).

Proof of Stake

Het op dit moment meest veelbelovende alternatief voor Proof of work is Proof of Stake [9]. In Proof of Stake mogen alleen miners die aan een bepaalde voorwaarde voldoen een nieuw blok aanmaken. Deze voorwaarde is gebaseerd op het belang (stake) dat de miners hebben in de blockchain. Het idee is: hoe groter het belang (hoe hoger het saldo) hoe groter de kans om een nieuw block te mogen aanmaken. Deze miners hebben immers een groot belang bij een geldige keten. In de meest eenvoudige versie mag een miner een block aanmaken indien hij voldoet aan de volgende voorwaarde: *hashwaarde vorig blok + adres miner < saldo miner * factor.* Hoe hoger het saldo van de miner hoe groter de kans om een blok te kunnen aanmaken. De factor wordt gebruikt om de kans omhoog of omlaag bij te stellen. Deze methode vergt van de miner niet veel werk om een nieuw blok aan te maken (anders dan bij

proof-of-work). Nadeel hiervan is wel dat in het geval van een splitsing een miner sneller een nieuw blok aan beide ketens kan toevoegen, de drijfveer om aan dezelfde langste keten te werken is niet aanwezig (dit wordt het nothing-at-stake probleem genoemd). Dit nadeel zorgt ervoor dat de grootste blockchains (Bitcoin en Ethereum) op dit moment nog werken met proof-of-work. Ethereum heeft plannen om op korte termijn over te stappen op Proof of Stake.

Proof of Activity

Dit is een systeem dat Proof of Work combineert met Proof of Stake. Het toevoegen van een nieuw blok, gebeurt evenals bij het Proof of Work systeem door de inzet van computerrekenkracht om een ingewikkelde som op te lossen. In het nieuw te vormen blok zijn echter geen transacties opgenomen. Het blok wordt geaccordeerd door validerende nodes die worden geselecteerd op basis van voldoende saldo. Hierin is het Proof of Stake principe gelegen. Zowel de miners als de validator ontvangen hiervoor een vergoeding.

Verschillende varianten

Een technologie die een dusdanige disruptieve aard heeft creëert onder meer zogenaamde blockchainevangelisten, blockchainvijanden en natuurlijk ook blockchainopportunisten.

De belangrijkste onderscheidende kenmerken van blockchaintechnologie en de mogelijke invloed op bestaande sectoren en industrieën, zoals bijvoorbeeld de aanzienlijke kostenreductie, het wegvallen van de noodzaak van TTP's (denk aan de bank, de notaris, de Kamer van Koophandel, het Kadaster, de accountant), maar ook radicale transparantie en het principe van niet wijzigen/terugdraaien van transacties roepen verschillende (tegen)reacties op.

Waar sommigen alle mogelijkheden die deze technologie met zich meebrengt toejuichen, proberen anderen toepassingen te laten verbieden en/of te reguleren, maar is er ook sprake van "*cherry picking*" (het enkel kiezen van de eigenschappen die opportuun zijn). Dit komt tot uiting in de verschillende bewegingen en toepassingen op het gebied van deze technologie. Zo kunnen de toepassingen in ieder geval worden ingedeeld in een public en een private blockchain, maar zijn er ook tussenvormen mogelijk.

Public & permissionless blokchain:
Deze variant biedt toegang en inzage aan iedereen. Iedereen kan een node creëren, kan de transacties inzien en kan transacties (en blocks) creëren. Er is dan ook geen sprake van censuur. De bekendste voorbeelden van deze van varianten zijn vanzelfsprekend Bitcoin en Ethereum (waarover meer in het volgende hoofdstuk). Deze variant wordt ook wel vergeleken met het internet. Het huidige uitganspunt is dat deze variant meer vragen oproept op het gebied van wet- en regelgeving.

Private & Permissioned blockchain:
Deze variant geeft geen inzage in de transacties en geeft enkel toegang en inzage aan deelnemers en partijen die hiervoor toestemming hebben gekregen. Nieuwe deelnemers kunnen op basis van toestemming deelnemen of aan de hand van een vooraf bepaald regulerend kader. Deze variant wordt ook wel eens vergeleken met het intranet en kan worden gekozen als men zorgen heeft over privacy, niet voldoende vertrouwen heeft in het systeem, de tussenpersoon/autoriteiten niet volledig wenst uit te sluiten, en/of enkel gebruikt wenst te maken van de kostenreductie. Het uitgangspunt is dat deze variant eenvoudiger te reguleren is, omdat alle partijen in kaart kunnen worden gebracht en bijvoorbeeld eenvoudiger is vast te stellen wie waarvoor verantwoordelijk is. Voorbeelden hiervan zijn MONAX en Multichain (deze voorbeelden worden niet nader toegelicht).

Hybride blockchain/ Federated Blockchain/Consortium Blockchain:
Bij deze variant is er sprake van het combineren van de public/permissionless en private/permissioned elementen. Zo kan er onderscheid worden gemaakt tussen twee levels waarbij niet alle transacties voor alle deelnemers zichtbaar zijn, maar wel kunnen worden gecontroleerd. Ook kan het zijn dat bijvoorbeeld wel alle transacties voor alle deelnemers zichtbaar zijn, maar slechts door een vooraf bepaald aantal deelnemers kan worden gecontroleerd. Deze variant komt onder meer voor in de bancaire wereld en in de verzekeringsbranche. Een voorbeeld hiervan is Corda. Deze variant van het R3-consortium wordt hieronder kort toegelicht.

Ethereum

Blockchaintechnologie wordt vaak beschreven als het nieuwe internet. In deze context wordt ook vaak verwezen naar Ethereum. Ethereum is een platform dat het mogelijk maakt door middel van blockchaintechnologie decentrale applicaties, DApps, te ontwikkelen, dus een blockchain app platform.

Ook aan Ethereum liggen de beginselen transparantie, vertrouwen en algoritmische interpretatie ten grondslag, maar vanwege het feit dat Ethereum als doel heeft de algemene, ondersteunende technologie te worden waarmee alle toepassingen, door wie dan ook, waar dan ook gebruikt, ontwikkeld kunnen worden met gebruik van blockchaintechnologie, wordt over Ethereum gezegd dat dit wel eens het internet 3.0 of een wereldcomputer zou kunnen worden. Overigens moet een en ander nog worden waargemaakt en worden er op dit moment bijvoorbeeld vraagtekens geplaatst bij de veiligheid van Ethereum's smart contracts [10].

Waar Bitcoin zich beperkt tot financiële transacties en in deze zin meer beperkt is heeft Ethereum zich als doel gesteld alle transacties mogelijk te maken door middel van het gebruik van smart contracts. Door het gebruik van deze smart contracts kunnen meer condities aan de transacties worden toegevoegd en leent dit systeem zich bijvoorbeeld voor de overdracht van verschillende goederen (bijv. huizen), maar ook voor het vrijgeven van/de toegang tot goederen als aan bepaalde voorwaarden is voldaan. De werking van deze smart contracts wordt aan de hand van enkele voorbeelden nader toegelicht in hoofdstuk 6.

Wonderkind Vitalik Buterin heeft de basis van Ethereum in 2013 gelegd op 19 jarige leeftijd. Zijn grootste bezwaar tegen Bitcoin was de beperkte transactie-scope. Daarnaast was hij voorstander van een meer universele/algemene programmeertaal. In 2014 is het hem gelukt om samen met zijn medeoprichters meer dan 18 miljoen dollar op te halen voor de oprichting van dit platform. De eerste versie van Ethereum, Ethereum Frontier werd gelanceerd in 2015.

Ethereum is continue in ontwikkeling en is bijna toe aan de derde upgrade. Ook Ethereum hanteert op dit moment nog het Proof of Work-principe, maar is wel aanzienlijk sneller dan Bitcoin. Toch is het doel om over te stappen op het Proof of Stake- principe en de verwachting is dat dit in 2018 zal gebeuren.

Daarnaast heeft Ethereum de DAO (Decentralized Autonomous Organizations) geïntroduceerd (waarover meer in hoofdstuk), welk fenomeen ook wel bekend is geworden door de zogenaamde "*blockchain hack*", waarbij 80 miljoen dollar aan ether werd "*gestolen*". Hierdoor liep het imago van Ethereum tijdelijk een deukje op en werd verondersteld dat de blockchaintechnologie toch niet onfeilbaar was. Dit kwam mede, omdat ten onrechte naar buiten werd gebracht dat "*de blockchain was gehackt*." Het betrof echter geen hack, maar er was sprake van een fout in de code [14]. Deze fout en het verloren gaan van 80 miljoen dollar aan ether heeft geleid tot een splitsing en het bestaan van 2 versies van Ethereum, Ethereum en Ethereum Classic. Deze "*crisis*" is een goed voorbeeld van het belang van governance in een blockchain-context en zal nader worden toegelicht in hoofdstuk 9.

Waar in het Bitcoin-systeem sprake is van fees en de miners worden betaald met Bitcoin is in het Ethereum systeem Ether de currency. Om de kosten van een Ethereumtransactie uit te drukken wordt gebruik gemaakt van de eenheid gas. Iedere transactie in Ethereum is onderworpen aan een gasLimit en aankopen vinden plaats op basis van de geldende gasPrice (de prijs van een eenheid).

Het vertrouwen in Ethereum lijkt echter te zijn hersteld, want in februari 2017 is de Enterprise Ethereum Alliance (EEA) opgericht en inmiddels telt deze alliantie meer dan 400 leden. EEA brengt gevestigde en startende bedrijven uit verschillende sectoren, experts en academici bijeen met als doel het Ethereum protocol te verbeteren, best practices en standaarden te ontwikkelen en deze naar de markt te brengen. Naar verwachting verschijnt er in de loop van 2018 een uniform business standard voor blockchain, uitgebracht door EEA.

Hyperledger Fabric

Een ander bekend blockchainplatform is Hyperledger Fabric. Hyperledger Fabric is onderdeel van het Hyperledger Project [11] dat in 2015 op initiatief van de Linux Foundation is gelanceerd. Dit platform brengt

evenals de EEA verschillende partijen (in dit geval meer dan 180 bedrijven) bij elkaar.

Hyperledger Fabric is weliswaar een open source netwerk, maar is eveneens een permissioned netwerk dat niet is gestoeld op cryptocurrency en mining. Hyperledger is ontwikkeld met het oog op zakelijk gebruik en is volgens IBM modulair en schaalbaar opgebouwd.

Een groot verschil met Bitcoin en Ethereum is dat Hyperledger Fabric niet op basis van cryptocurrency en mining functioneert. Deze incentives worden niet als uitgangspunt genomen, omdat door de deelnemers wordt gewerkt op basis van een gemeenschappelijk doel.
Het uitganspunt van Hyperledger Fabric is dan ook dat juist het vertrouwen in de partij met wie je zaken doet van belang is, in tegenstelling tot het vertrouwen in de onfeilbaarheid van de technologie zoals bij Bitcoin het vertrekpunt is. Alle deelnemers aan het netwerk dienen elkaar te kennen.

Hierdoor is dan ook de keuze gemaakt af te wijken van een systeem waaraan wie dan ook, waar dan ook kan deelnemen en is deelname enkel mogelijk na toelating. Deze afwijkende benadering brengt ook een andere filosofie op het gebied van consensus met zich mee.

Waar Bitcoin en Ethereum (nu nog) kiezen voor een Proof of Work consensusmechanisme is dit bij Hyperledger Fabric niet het geval. Het modulaire karakter staat de deelnemers immers toe zelf een te integreren consensusmechanisme te kiezen. Dit kan zowel een Fault-Tolerance mechanisme als een Byzantine Fault Tolerance mechanisme zijn. Het eerstgenoemde mechanisme biedt weliswaar minder bescherming, maar het uitganspunt is dat bij bekende en betrouwbare partijen hiertoe geen/een beperkte noodzaak is. Daarnaast vergt een Byzantine Fault Tolerance mechanisme meer computerrekenkracht en is dit vaak niet de meest efficiënte optie.

Corda (R3)

Dat de banken zich niet uit het veld hebben laten slaan door de opkomst van deze disruptieve technologie die ervan wordt beschuldigd juist deze TTP te kunnen vervangen blijkt uit de vorming van het R3-consortium [12], dat inmiddels (volgens eigen zeggen) meer dan 200 banken, financiële instellingen, wetgevers, handelsassociaties, dienstverleners en technologiebedrijven wereldwijd tot haar leden mag rekenen. Dit consortium heeft (niet geheel verrassend) in 2017, 107 miljoen dollar aan kapitaal verzameld.

Het antwoord op de blockchaintechnologie van dit consortium is genaamd Corda, een open source blockchainplatform dat zich specifiek richt op de financiële sector, in het bijzonder op de (financiële) overeenkomsten die worden gesloten tussen deze instellingen. Corda geeft niet alle deelnemers inzage in alle transactiedata.
Alleen partijen die hiervoor een reden hebben krijgen toegang. In Corda kunnen verschillende consensusmechanismes worden geïntegreerd, maar in beginsel wordt consensus op overeenkomstniveau bereikt. Dat de TTP niet volledig worden uitgeschakeld komt bij Corda tot uiting in de mogelijkheid om toezichthoudende autoriteiten toegang te geven tot transacties door middel van het genereren van regulerende en handhavende nodes in het systeem.

6. Smart Contracts

De term smart contracts is kort aangestipt bij de bespreking van Ethereum en wordt in dit hoofdstuk meer gedetailleerd behandeld.

Smart contracts, in het Nederlands *'slimme contracten' of autonome contracten*, is een onderwerp dat vaak in één zin met blockchain wordt genoemd. Voorbeelden die in deze context vaak worden aangehaald zijn de snoepautomaat, huurcontracten en notariële aktes. Ondanks het feit dat smart contracts een redelijk nieuw fenomeen lijkt komt het concept al uit 1995. In 1995 definieerde Nick Szabo de term en een visie omtrent het onderwerp. Deze visie kon op dat moment niet gerealiseerd worden vanwege technologische beperkingen. Beperkingen die met de komst van de Blockchain voor een deel zijn verdwenen.

Maar wat is nu precies een smart contract? Wat zijn de type smart contracts? Wat is de opbouw van een smart contract? En hoe specificeer en controleer je een smart contract? Deze vragen worden in de komende paragraven één voor één beantwoord.

Wat is een smart contract?

Een smart contract is "*een overeenkomst die door een machine wordt uitgevoerd bestaande uit een zeer specifieke set van voorwaarden en daarbij behorende uitkomsten, mogelijk digitaal getekend, die (digitale) valuta of activa tussen twee of meer partijen beheert en kan worden geadministreerd, afgedwongen en uitgevoerd door een derde partij, waarbij het contract kan aantonen dat er aan de gestelde voorwaarde is voldaan ten tijde van executie.*" Dit is een uitgebreide definitie welke aan de hand van twee casus, een snoepautomaat en een huurovereenkomst, zal worden toegelicht.

Casus 1: De snoepautomaat

De transactie met een snoepautomaat bestaat uit vier activiteiten. Als eerst typt een persoon op de snoepautomaat in welk artikel hij/zij wil aanschaffen. De snoepautomaat geeft op zijn beurt de prijs van het artikel weer waarna de persoon dit bedrag in euro's in de machine stopt. De persoon geeft nu zijn akkoord voor het leveren van het product en de machine levert het desbetreffende artikel. Dit proces is grafisch weergegeven in figuur 3.

figuur 3: proces snoepautomaat

Casus 2: De huurovereenkomst

In het geval van de huurovereenkomst werkt een transactie als volgt. De verhuurder en huurder sluiten een huurovereenkomst. In deze huurovereenkomst staat dat de huurder elke maand op de 22e zijn huur dient over te maken en dat hij/zij in ruil daarvoor toegang krijgt tot het appartement. In de situatie zoals beschreven heeft het huis geen sloten met een fysieke sleutel maar een digitale sleutel die geregistreerd staat op de mobiele telefoon van de huurder.

Waar bevindt het smart contract zich?

Als eerst dient te worden vastgesteld waar het smart contract zich in beide casus bevindt. In de casus met betrekking tot de snoepautomaat is een smart contract aanwezig om de transactie tussen de persoon en de automaat vorm te geven. In dit contract staat vormgegeven dat wanneer de persoon een keuze heeft gemaakt en genoeg geld heeft ingeworpen hij het gekozen product krijgt.

In de casus met betrekking tot de huurovereenkomst is de huurovereenkomst zelf vastgelegd in een smart contract.

In dit specifieke geval is één specifieke voorwaarde van het smart contract als volgt: wanneer het huurbedrag op de 23e niet is gestort op de bankrekening van de verhuurder wordt de toegang tot het appartement ingetrokken tot het moment dat er wel betaald is.

Belangrijk om te constateren is dat in beide casus het smart contract dient te worden geactiveerd voordat de voorwaarden gecontroleerd kunnen worden. In het geval van de snoepautomaat wordt het smart contract geactiveerd nadat de betaling door de klant is uitgevoerd. Bij het huurcontract wordt het smart contract automatisch elke 23e van de maand geactiveerd. Een smart contract is dus een reactief object dat geactiveerd dient te worden. Het is daarmee dus geen proactief

object dat continue de omgeving scant om te kijken of er aan de voorwaarde is voldaan en zichzelf daarna executeert.

De elementen van een smart contract

Het eerste criterium zoals gesteld door de hiervoor genoemde definitie is dat de overeenkomst wordt uitgevoerd door een machine. In het geval van de snoepautomaat is de individuele snoepautomaat de machine die het contract uitvoert. In het tweede geval wordt de huurovereenkomst uitgevoerd door een server en daarmee is ook voldaan aan het eerste criterium. Daarnaast stelt de definitie dat een smart contract een zeer specifieke set van voorwaarden en uitkomsten heeft. De voorwaarden van het contract in de snoepautomaat staan weergegeven in tabel X en Y. Hierbij worden er twee voorwaarden gesteld om te bepalen of de transactie mag worden uitgevoerd: *"de waarde van de ingevoerde euro's"* en *"de voorraad van het gekozen artikel"*. Een aanvullende clausule beschrijft daarna de voorwaarde die wordt gesteld om het product daadwerkelijk uit te leveren. Waarom deze beide zijn gesplitst wordt toegelicht in de paragraaf *"opbouw van een smart contract"*. In beide gevallen wordt maar een deel van de voorwaarden getoond.

	De transactie moet gesteld worden op akkoord indien: 1. de waarde van de ingevoerde euro's is groter of gelijk aan de waarde van het artikel; 2. de voorraad van het gekozen artikel is groter dan nul.	

tabel X: voorwaarden smartcontract deel 1 snoepautomaat

	De activiteit *"leveren product"* moet worden uitgevoerd indien: 1. de waarde van de transactie is gelijk aan akkoord.	

tabel Y: voorwaarden smartcontract deel 2 snoepautomaat

Ook voor de huurovereenkomst dienen voorwaarden te worden opgesteld, een voorbeeld staat in tabel Z. Hierbij dient meteen gesteld te worden dat de voorwaarden met betrekking tot de huurovereenkomst in de praktijk uitgebreider zijn.

	De toegang tot het huis moet gesteld worden op actief indien: 1. het gestorte bedrag is gelijk aan huursom.	

tabel Z: voorwaarden smartcontract huurovereenkomst

Het derde criterium is dat het contract mogelijk digitaal is getekend. Let hierbij op het woord mogelijk waarmee een advies wordt uitgedrukt en geen verplichting. Aan deze voorwaarde wordt bij de snoepautomaat niet voldaan. In deze situatie is dit ook niet wenselijk, want in dat geval zou de snoepautomaat van iedere klant eerst een digitale handtekening moeten ontvangen voordat het product mag worden geleverd. Bij het huurcontract is het wel aan te bevelen een ondertekend smart contract toe te passen. Het gaat hier namelijk om een transactie met een hogere mate van economisch en sociale waarde waarbij zowel de huurder als de verhuurder in een zekere mate dienen te worden beschermd. Deze ondertekening kan in een blockchain met een digitale sleutel plaats vinden.

Het vierde criterium is dat het contract een (digitale) valuta of activa tussen twee partijen dient te beheren. In beide casus is dit het geval. In de eerste situatie beheert het contract de relatie tussen het te kopen product en de betaling van de koper. In het tweede geval beheert het contract de relatie tussen toegang tot het huis en de betaling van de huurder. Het een na laatste criterium is dat het contract kan worden beheerd, geadministreerd en afgedwongen door een derde partij.

Ook hierbij geldt weer dat dit een advies is en geen verplichting.

Bij de snoepautomaat is hier namelijk geen directe aanleiding toe. Het contract zou via het netwerk beheerd kunnen worden op een blockchain. De vraag is alleen of dit echte toegevoegde waarde oplevert. Hiermee is ook meteen duidelijk dat een smart contract niet altijd gekoppeld hoeft te zijn aan een blockchain. Het afdwingende karakter is misschien meer van toepassing op deze casus. Stel, de euro's worden geaccepteerd maar de machine keert niet uit omdat het product 'blijft hangen'. In dat geval zou een derde (bijvoorbeeld een monteur) het contract alsnog kunnen executeren.

In het geval van de huurovereenkomst is opslag op de blockchain wel een voor de hand liggende keuze. De reden hiervoor is dat zowel de verhuurder als de huurder het contact in willen zien of willen kunnen aantonen dat het contract daadwerkelijk is gesloten tegen de voorwaarden die daarin genoemd staan. Hiermee is er dus een administrerende en beherende rol weggelegd voor het blockchainprotocol. In deze specifieke huurcasus wordt er gewerkt met een digitale sleutel. Door gebruik te maken van een digitale sleutel kan het contract ook zelf de toegang tot het appartement regelen waarmee er ook een uitvoerende rol voor het blockchainprotocol is gerealiseerd. Maar niet elk appartement en huis gebruikt een digital sleutel om toegang ertoe te verkrijgen. In veel gevallen is de toegang tot het appartement of huis nog altijd een fysieke sleutel. In dit geval kan het smart contract de huurder geen toegang ontzeggen en heeft het daarbij 'hulp' nodig van een derde partij. Daarmee is er naast het digitale netwerk ook een rol weggelegd voor het offline netwerk.

Als laatste wordt gesteld dat het contract kan aantonen dat er aan de gestelde voorwaarde is voldaan tijdens de executie van het contract. In het geval van de snoepautomaat betekent dit dat de automaat de volgende elementen dient te registeren: A) de hoeveelheid euro's die in de machine zijn ingevoerd B), de hoeveelheid euro's die als wisselgeld dient te worden uitgekeerd, C) het product dat is uitgegeven en D) de 'code'/voorwaarden van het contract op het moment dat het werd uitgevoerd. Elk van deze elementen kan de snoepautomaat (of het netwerk waarop de automaat is aangesloten) registeren en bewaren. Deze informatie kan in de toekomst worden gebruikt om de legitimiteit van de executie van het contract aan te tonen. Ditzelfde principe geldt ook voor de huurovereenkomst. Het blockchain protocol dient te registreren: A) welk bedrag er voor de huursom is binnengekomen, B) wat de huursom op dat specifieke moment is, C) het afsluiten of niet afsluiten van de deur en D) de 'code'/voorwaarden van het contract op het moment dat het werd uitgevoerd.

Om te bepalen of iets een smart contract is kunnen de hierboven genoemde criteria worden toegepast. Omgedraaid kunnen de criteria ook worden gebruikt om te bepalen of in een specifieke situatie een smart contract uitkomst kan bieden.

Hiervoor dienen dan de volgende vragen gesteld te worden:

- Kan of dient de overeenkomst door een machine te worden uitgevoerd?
- Zijn de voorwaarden en uitkomsten tot in detail beschreven?
- Kan of dient de overeenkomst digitaal te worden ondertekend?
- Beheert het contract de (digitale) valuta of activa tussen twee of meer partijen?
- Dient het contract op een later tijdstip te kunnen aantonen dat er tijdens de executie van het contract aan de in het contract gestelde voorwaarden werd voldaan?

Type smart contracts

Op het hoogste abstractieniveau kan er onderscheid gemaakt worden tussen twee typen smart contracts: implementatieonafhankelijk of implementatieafhankelijk. Een implementatieafhankelijk smart contract is een smart contract dat is opgesteld in een notatievorm die is afgestemd om leesbaar te zijn voor één specifiek blockchainprotocol. Een voorbeeld van een implementatieafhankelijke smart contract is een smart contract geschreven in Solidity. Zie tabel X voor voorbeeld code in Solidity.

```
function KoopProduct(uint coin)
{
    if (coinAccount[msg.sender] > coin)
      {
          coinAccount[msg.sender] -= coin;
          ProductAccount[msg.sender] += (coin / kWh_rate);
          }
}
```

tabel X: voorwaarden smartcontract deel 1 snoepautomaat

Solidity is een niche programmeertaal voor smart contracts die geschikt zijn voor het Ethereum protocol. Een voorbeeld van een ander blockchain protocol is het Neo protocol.
Een smart contract geschreven in Solidity werkt niet op Neo. Neo op haar beurt heeft haar eigen talen waarin smart contracts worden geschreven. In tegenstelling tot Ethereum hebben de ontwikkelaars van Neo gekozen voor meerdere talen waarin een smart contract kan worden gespecificeerd, namelijk Java, C# en Python. Dit maakt het eenvoudiger om de smart contracts over te dragen naar een ander blockchain protocol mocht dit nodig zijn. Ondanks de verhoogde overdraagbaarheid zijn de smart contracts vanwege specifieke functies die alleen binnen het Neo protocol functioneren nog steeds beperkt overdraagbaar. Om smart contracts te specificeren die geschikt zijn voor meerdere blockchain protocollen dient er een implementatieonafhankelijk smart contract te worden ontwikkeld.

Een implementatieonafhankelijk smart contract is een smart contract dat is opgesteld in een notatievorm die niet is afgestemd om geïmplementeerd te worden op één specifiek blockchain protocol. Dit smart contract wordt opgesteld in gestructureerd Engels of Nederlands en wordt daarna via een lexer/parser vertaald naar een specifiek blockchain protocol. Hiermee kan een contract worden getransformeerd naar zowel Solidity als Neo. Een voordeel hiervan is dat het contract één keer hoeft te worden opgesteld en daarna naar verschillende protocollen kan worden getransformeerd. Een bijkomend voordeel hiervan in het huidige tijdsbeeld is dat wanneer een specifiek protocol nog verdwijnt daarmee de kennis in de smart contract niet automatisch verdwenen is.

De opbouw van een smart contract

Een smart contract is onder te verdelen in vier niveaus en zes onderdelen. De vier niveaus zijn de clausule (niveau 1), de individuele voorwaarden (niveau 2), de feittypen (niveau 3) en de feitwaarden (niveau 4), zie figuur 4. Hierbij bestaat niveau 2 uit de afleiding van de voorwaarden voor één specifieke clausule.
Niveau 3 bestaat uit de conditie(-feiten) en het conclusie(feit) en niveau 4 beschrijft voor zowel de conditie(feiten) en het conclusie(-feit) de feitwaarden. Elk onderdeel wordt hier verder kort toegelicht.

Een conclusie(feit) is het resultaat dat dient te worden bepaald, bijvoorbeeld: "toegang tot het huis". De daadwerkelijke waarden die conclusie kan aannemen worden conclusiefeitwaarden genoemd. In dit geval zijn de conclusiefeitwaarden: "verleend" en "niet verleend". Om de conclusie te kunnen bepalen dienen condities(feiten) te worden geëvalueerd. In dit specifieke geval wordt de hoogte van het binnengekomen bedrag geëvalueerd. Ook voor de conditiefeiten dienen feitwaarden bepaald te worden. In dit geval de hoogte van de huursom.
De laatste component van het smart contract is de afleiding van de specifieke clausule. Een voorbeeld van een afleiding is: "*De toegang tot het huis moet gesteld worden op actief indien: het gestorte bedrag is gelijk aan huursom.*"

figuur 4: niveau's smart contract

Een belangrijke eigenschap met betrekking tot opbouw van een smart contract is de structurering van de clausules. Een clausule vanuit het perspectief van een daadwerkelijk contract wordt gedefinieerd als een paragraaf in een overeenkomst die een specifiek onderwerp behandelt. Vanuit het perspectief van een smart contract dient deze definitie te worden aangescherpt. Hierbij geldt namelijk niet alleen dat de clausule één onderwerp mag behandelen maar tevens dat de clausule dient te voldoen aan het single responsibility principe. Wat wordt er verstaan onder het single responsibility principe? Hiermee wordt bedoeld dat elke clausule in het contract maximaal één conclusiefeit mag bevatten of één specifieke taak mag vervullen. Aanvullend is een belangrijke eigenschap dat een clausule in het contract altijd één van de volgende hoofddoelen heeft: A) een communicatietaak, B) een registratietaak of C) een kennistaak. Een communicatietaak bestaat uit het communiceren van een specifiek element uit het contract. Het registreren (of wijzigen) van een getal wordt onder een registratietaak verstaan en het bepalen van conclusie op basis van condities valt onder een kennistaak.

Het specificeren van een smart contract

Het gaat voor dit boekje te ver om in detail te beschrijven hoe een gedegen en voldoende gespecificeerd smart contract gemaakt dient te worden. Wel beschrijven we in deze paragraaf aan de hand van een simpel ezelsbruggetje de gevaren van slecht of onduidelijk gespecificeerd contract.

Wat is de uitkomst van de volgende som: 6 / 2 * 3 = ? . Is het door jou gegeven antwoord: 1? Dan moeten we je teleurstellen, 1 is niet het juiste antwoord. Is het door jou gegeven antwoord: 9? Dan heb je het juiste antwoord gegeven. Één som waar twee verschillende groepen mensen een ander antwoord opgeven. Waar komt dit verschil in antwoord vandaan? Bij het eerste antwoord moeten we je teleurstellen, het antwoord zegt waarschijnlijk iets over je leeftijd. Als het door jou gegeven antwoord 1 is, dan heb je waarschijnlijk gebruik gemaakt van het ezelsbruggetje: Meneer Van Dale Wacht Op Antwoord. Een ezelbruggetje waarmee veel van ons (iets oudere mensen) hebben leren rekenen. Het probleem? Dit ezelbruggetje klopt al sinds 1992 niet meer. Dit heeft te

maken met de algemene introductie van de computer/computerchips. Een computer maakt namelijk geen verschil tussen “delen door” en “vermenigvuldigen met”. Anders geformuleerd, voor een computer is “gedeeld door 2” en “vermenigvuldigden met 0,5” hetzelfde. Daarom is er vanaf 1992 een nieuw ezelsbruggetje: Hoe Moeten Wij Van De Onvoldoendes Afkomen. Hierbij is de volgorde waarin je berekening uitvoert als volgt:

1. (haakjes)
2. machtsverheffen en worteltrekken, in de volgorde van de opgave
3. vermenigvuldigen en delen, in de volgorde van de opgave
4. optellen en aftrekken, in de volgorde van de opgave

Het voorgaande voorbeeld geeft aan hoe een simpele rekensom al verkeerd kan worden uitgelegd. Hoe zal dat zijn met smart contracts als deze niet eenduidig zijn gespecificeerd?

De controle van een smart contract

Ondanks het feit dat met de grootste voorzichtigheid smart contracts dienen te worden geformuleerd kunnen er altijd fouten in het contract zitten. Daarom dient een smart contract altijd, net als broncode, gecontroleerd te worden op twee elementen: verificatiefouten en validatiefouten.
Verificatiefouten zijn fouten met betrekking tot semantiek (betekenis) en structuur. Aan de andere kant zijn validatiefouten fouten met betrekking tot de toepassing van het bedrijfsregelmodel.

De verificatie van smart contracts kan op meerdere manieren worden ingericht, waarbij altijd rekening gehouden dient te worden met twee variabelen: betrouwbaarheid en automatiseringsgraad.
De variable betrouwbaarheid bestaat uit twee mogelijkheden: preventieve controle of detectieve controle.

Een preventieve controle is een controle die ongewenste evenementen/fouten voorkomt. Voorbeelden van evenementen zijn het invoeren van een conflicterende voorwaarden of het invoeren van niet bestaande conditiefeiten. Een detectieve controle is een controle die evenementen /fouten identificeert nadat deze hebben plaatsgevonden. Om de fout te herstellen dient een aanvullend bedrijfsproces te worden gerealiseerd.

De automatiseringsgraad kent twee variabelen: geautomatiseerd en niet geautomatiseerd. Bij een automatische detectieve controle wordt op het moment van invoer een controle uitgevoerd. Het systeem zal gelijk een melding maken van een gedetecteerde fout. Dit kan bijvoorbeeld voorkomen bij het invoeren van een niet bestaand conditiefeit. Het systeem zal controleren of deze term voorkomt in de vocabulaire. Als dit niet het geval is geeft het systeem een fout terug.
Bij automatische preventieve controle zorgt het systeem er van tevoren al voor dat er geen fouten gemaakt kunnen worden. Dit gebeurt bijvoorbeeld aan de hand van vooraf ingestelde keuzemogelijkheden. Er kan dan niets anders worden ingevuld dan vooraf is vastgesteld. Hierdoor kunnen er geen fouten gemaakt worden.

De validatie van een smart contract kan op meerdere manieren worden ingericht, op basis van 1) collegiale toetsing en/of op basis van scenario's. Collegiale toetsing is een proces waarbij collega's, aan de hand van vooraf vastgestelde toetscriteria feedback geven aan collega's. Collegiale toetsing kan plaatsvinden door middel van workshops of een gedefinieerd werkproces. De tweede vorm van validatie is validatie op basis van scenario's. Bij validatie op basis van scenario's gelden concrete gevallen als uitgangspunt. Een validatiemedewerker/computersysteem doorloopt de geschreven voorwaarden met vooraf gedefinieerde scenario's en controleert of vooraf gedefinieerde uitkomsten behaalt zijn.
Elk scenario bevat de initiële situatie, de feiten (en bijbehorende gegevens) die van toepassing zijn en het vooraf gedefinieerde resultaat.

7. Decentralized Autonomous Organization (DAO)

In het voorgaande hoofdstuk is uitvoerig aandacht besteed aan smart contracts en de werking hiervan. Een interessant verschijnsel in deze context is de DAO, met name als men kijkt naar de inrichting van het digitale landschap en het vormen van zelfstandig functionerende entiteiten in de toekomst.

De DAO, die bestaat uit een aantal met elkaar verbonden smart contracts, vormt namelijk een decentrale, autonome, zelfstandig functionerende organisatie waarbij op basis van de code wordt gehandeld. De regels die dienen te worden nageleefd kunnen door personen en organisaties worden bepaald en door middel van code in de smart contracts worden opgenomen, maar daarna functioneert de DAO, in beginsel, zonder tussenkomst van deze personen of organisaties.

De meest beruchte DAO is de DAO die heeft geleid tot de opsplitsing van Ethereum in Ethereum en Ethereum Classic (waarover later meer).

Deze DAO was ontworpen als investeringsfonds dat als doel had gedecentraliseerde apps die in Ethereum waren ontwikkeld te financieren. Deze DAO was op een punt onbetwist succesvol. Immers, binnen een zeer korte termijn slaagde deze DAO erin meer dan 150 miljoen dollar aan ether in te zamelen. De constructie met betrekking tot dit fonds was de volgende [13].

Oprichters van de DAO hadden zeggenschap over de keuze van de te financieren applicaties en investeerders konden met gebruik van ether DAO Tokens aanschaffen in ruil voor stemrecht. In deze zin is het vergelijkbaar met een aandeel. Het besluitvormingsproces ten aanzien van de te financieren applicaties was tamelijk eenvoudig. Vooraanstaande leden van de Ethereum-gemeenschap dienden hun zegen te geven aan de toepassingen die zij geschikt achtten voor ontwikkeling, voordat door de houders van de DAO Tokens kon worden gestemd over de te financieren apps. Indien in ieder geval 20% ten gunste van een app had gestemd werd een deel van het budget toegekend aan de ontwikkeling van deze app.

Het was voor investeerders mogelijk zich terug te trekken uit de DAO door middel van een zogenaamde "*split function*". Deze mogelijkheid tot exit hield het recht tot teruggave van de geïnvesteerde ether in en bood daarnaast de mogelijkheid een beperkte uitvoering van de DAO te creëren, ook wel de "*child DAO*" genoemd. In deze "*split function*'' die ook een wachtperiode inhield van 28 dagen voordat de ether kon worden uitgegeven schuilde echter een enorm gevaar dat heeft geleid tot het verdwijnen van ether ter waarde van (destijds) 80 miljoen dollar.

Waar aanvankelijk werd gesproken van een hack van het systeem, bleek het uiteindelijk een fout in de code te betreffen. De code stelde de "hacker" namelijk in staat meerdere malen een verzoek tot teruggave van ether te doen voordat de registratie hiervan in het systeem kon worden opgenomen. Dat de vermeende hacker zeer vasthoudend en geduldig was bleek uit het feit dat hij zoveel pogingen had gedaan dat hij hiermee 80 miljoen dollar aan ether had vergaard.

Na de ontdekking van deze "*verduistering*" brak voor de Ethereumgemeenschap een doorslaggevende periode van 28 dagen aan. De vermeende hacker had namelijk gekozen voor de "*child DAO*" wat inhield dat gedurende 28 dagen geen toegang tot de vergaarde Ether bestond.

Deze wachtperiode bood de Ethereum-community de mogelijkheid om met elkaar in gesprek te gaan over wat de geschikte reactie was op deze situatie. Fundamenteel in dezen was dat het dilemma uiteindelijk neerkwam op de keuze tussen mens en technologie/code.

Niet alleen heeft deze programmeerfout zeer duidelijk heeft gemaakt hoe zeer het van belang is dat de code correct is en dat ook op dit niveau fouten kunnen worden gemaakt met vergaande gevolgen.

Ook heeft het tot een aantal belangrijke vragen geleid op het gebied van governance en wet- en regelgeving. Zo kan men zich immers afvragen of het systeem wel dient te worden aangepast als het uitgangspunt is dat de code doorslaggevend is en conform code is/wordt gehandeld. Daarnaast kan men zich afvragen of er wel van een hacker/ diefstal mag worden gesproken indien op een (weliswaar in dit geval zeer) handige wijze gebruik is gemaakt van een slordigheid in de code. In het hoofdstuk 9 wordt aan de hand van deze case kort een aantal punten met betrekking tot het ecosysteem van de blockchain besproken en worden de verschillende rollen en belangen aangestipt .

8. Initial Coin Offering (ICO)

Eind 2017 bereikten de Bitcoin-gekte en koers verschillende hoogtepunten. Hiervan werd handig gebruik gemaakt door verschillende partijen die ook een nieuwe (dienst rondom een) cryptocurrency in het leven riepen.

De financiering van het realiseren van verschillende (diensten rondom) cryptocurrencies vindt vaak plaats door middel van een ICO, een Initial Coin Offering, oftewel het aanbieden van een eerste token. Tokens kunnen vaak worden gekocht in ruil voor een recht op een aandeel in het project, een dienst of een deel van de winst. Om niet onder het regulerend kader te vallen wordt regelmatig gekozen voor het verschaffen van een recht op een dienst. Het is voor een leek echter niet altijd duidelijk welke investering een verstandige is. Immers worden vaak geweldige rendementen beloofd en de meest fantastische business modellen geëtaleerd, maar blijft het met beperkte kennis van de technologie en markt risicovol om hierin te investeren [15]. Dit betekent overigens niet dat alle ICO's gevaarlijk zijn, maar de Autoriteit Financiële Markten (AFM) raadt het, vanwege de risico's en het feit dat ICO's vaak buiten de invloed van de AFM liggen, af om onder de huidige omstandigheden te investeren in ICO's [16].

9. Het ecosysteem

Je vraagt je op dit moment mogelijk af hoe de verhouding mens technologie is bij blockchaintechnologie, hoe de technologie zich ontwikkelt, of er sprake is van politiek(e beslissingen), welke regels er gelden en wie ervoor zorgt dat deze worden nageleefd. In dit hoofdstuk wordt kort beschreven hoe het ecosysteem er in dit kader uit ziet en wat de belangen van de verschillende spelers kunnen zijn aan de hand van Bitcoin en Ethereum. Dit is een zeer beperkte beschrijving die als doel heeft je een idee te geven van de verhouding tussen de verschillende spelers en de verschillende belangen.

Bitcoin

Satoshi, je kent hem vast nog wel, heeft zich uit Bitcoin teruggetrokken uit 2011 (althans is hij vanaf dat moment niet meer zichtbaar actief geweest).

Satoshi heeft de broncode van Bitcoin overgedragen aan een van de belangrijkste developers [17]. De broncode werd vervolgens weer gedeeld met een aantal andere developers waardoor er een kernteam bestond van zogenaamde "*core developers*". Niet iedereen is fan van deze developers. Als het gaat om de rol van deze ontwikkelaars wordt ook wel gezegd dat zij zich onder andere bezig houden met de bescherming van de oorspronkelijke ideologie en de verdere ontwikkeling van de technologie met deze ideologie in het achterhoofd. Vanwege het tamelijk open karakter kunnen ontwikkelaars door middel van zogenaamde Bitcoin Improvement Proposals (BIPs) [18] voorstellen doen met betrekking tot de verbetering van het protocol en hierover overeenstemming bereiken.

Een punt van kritiek op de groep van kerndevelopers is dat men te voorzichtig is en zich bijvoorbeeld onvoldoende bezig houdt met de schaalbaarheid van Bitcoin en een praktische toepassing hiervan in de toekomst. Een belangrijk punt van discussie is onder andere de grootte van de blokken i.v.m. de transactiesnelheid geweest [19]. De kerndevelopers willen namelijk voorkomen dat kleinere miners buiten spel worden gezet vanwege het feit dat voor het genereren van grotere blokken met meer transacties ook meer computerrekenkracht nodig is. Dit zou in strijd kunnen met het decentrale karakter.

Ook binnen het team van kerndevelopers heeft dit geleid tot verschillende meningen en als gevolg gehad dat dat sommige developers zich hebben afgesplitst en hun eigen Bitcoin-variant (Bitcoin is namelijk geen beschermde naam) in het leven hebben geroepen. Een aantal voorbeelden hiervan zijn Bitcoin XT, Bitcoin Classic, Bitcoin Cash en Bitcoin Unlimited. Als we het over de oorspronkelijke variant hebben wordt ook wel gesproken van Bitcoin Core.[20]

Waar ligt de kracht van de gebruikers in dit speelveld? Aangezien het in dit geval om software gaat is de gebruiker in die zin koning. De gebruiker bepaalt namelijk welke variant hij graag wil gebruiken. De overwegingen op basis waarvan verschillende gebruikers de keuze maken liggen uiteen, maar logischerwijs is de marktwaarde van de currency na een wijziging van het protocol een belangrijke.

Een punt van zorg als het gaat om decentralisatie is onder meer de positie van de miners en de invloed die zij hebben in het ecosysteem. Over de miners wordt namelijk gezegd dat zij met behulp van de computerrekenkracht “het zware werk” doen en de blockchain draaiende houden. Deze miners wil men daarom niet kwijtraken.

De incentive van de miners is natuurlijk het terugverdienen van de investering en de kans te vergroten het raadsel op te lossen, hiermee een nieuw blok te creëren en de bijbehorende beloning te incasseren. Zij beoordelen een voorstel met betrekking tot de wijziging van het protocol met deze doelen als uitgangspunt.

Het is met deze incentive dan ook niet verrassend dat de miners voorstanders zijn van het mogelijk maken van het creëren van grotere blokken, omdat dit leidt tot meer transacties waarvoor transactiekosten in rekening kunnen worden gebracht. Let wel, de verwachting is dat 75% [21] van de miners in het netwerk deel uitmaakt van mining pools die voornamelijk in China gevestigd zijn. Ook op dit punt wordt wel eens de vraag gesteld in hoeverre er nog sprake is van een decentraal systeem.

Ethereum

Zoals in hoofdstuk 7 is beschreven werd de Ethereum-gemeenschap geconfronteerd met een lastig vraagstuk op het moment van de zogenaamde DAO-hack. Er diende overeenstemming te worden bereikt over de vraag of de code diende te worden gerespecteerd en daarmee de fout in de code voor lief diende te worden genomen, of dat er door middel van een aanpassing van het protocol de betreffende transacties moesten worden teruggedraaid. Juist deze ongedaanmaking zou in strijd zijn met een aantal van de grondbeginselen van blockchaintechnologie, namelijk het niet kunnen wijzigen/ongedaan maken van transacties.

Door middel van een zogenaamde “hard fork”, een zeer ingrijpende wijziging van het protocol dat door alle gebruikers in het netwerk moet worden geaccepteerd om te kunnen worden ingevoerd, werden deze transacties dan ook teruggedraaid, althans werd het mogelijk gemaakt om door middel van een nieuw smart contract de onttrokken ether te retourneren [22]. Dit was geen onomstreden beslissing. Aan de ene kant was namelijk een aanzienlijk deel van het totaal aantal ether verdwenen en wilde men dit graag terug zien, aan de andere kant wilde men niet afwijken van de code is law-benadering en hiermee het gedachtegoed van blockchain ondermijnen.

Het Ethereum-kernteam, bestaande uit de Ethereum Foundation en een aantal kernontwikkelaars beschikte niet over een handboek dat uitsluitsel gaf over hoe om te gaan met een dergelijke situatie. De oplossing met betrekking tot een uitspraak over het al dan niet toepassen van een hard fork was een ouderwetse stemming. Binnen een korte tijdspanne kon men voor of tegen stemmen.

Uiteindelijk heeft maar een klein deel van de gemeenschap gestemd en stemde de meerderheid voor een hard fork. Een beslissing die dus niet heel veel draagvlak binnen de gemeenschap leek te hebben en die uiteindelijk heeft geleid tot de opsplitsing van het netwerk. Dit betekent ook dat niet alle gebruikers zijn overgestapt op het nieuwe protocol. Hierdoor is de oude, oorspronkelijke, keten blijven bestaan. De achterblijvers, onder andere bestaande uit ontwikkelaars en miners noemen deze keten Ethereum Classic.

Bovenstaande voorbeelden geven weer dat als het gaat om de besluitvorming op protocolniveau, de inrichting en de toepassing van de technologie en de verhouding tussen ideologie en marktwerking, er verschil-

lende spelers met verschillende belangen en uitgangspunten zijn. Weliswaar zijn de algoritmes en consensusmechanismen eenduidig en doorslaggevend als het gaat om het controleren en valideren van transacties, over het besturen van de blockchain in zijn geheel is op dit moment nog geen consensus bereikt. De verwachting is dan ook dat als het gaat om de toekomst van blockchaintechnologie de uitdagingen niet zijn gelegen in de technologische kant, maar juist in governance en wet- en regelgeving [23]. In ons volgende boekje zal hieraan daarom meer aandacht worden geschonken.

Bibliografie

1. Nakamoto, Satishi. Bitcoin: A Peer-To-Peer Electronic Sash System. *https://bitcoin.org/bitcoin.pdf. 2008*.
2. The Byzantine Generals Problem. Leslie Lamport, Robert Shostak, and Marshall Pease. 1982, ACM Trans. Program. Lang. Syst. 4, 3, pp. 382-401.
3. Russel, Frank. Information Gathering in Classical Greece. sl : The University of Michigan Press, 1999.
4. Menezes, Alfred J., van Oorschot, Paul C. en Vanstone, Scott A. Handbook of Applied Cryptography. sl : CRC Press, 1996.
5. Back, Adam. Hashcash - A Denial of Service Counter-Measure. 2002.
6. Boneh, Dan. Twenty Years of Attacks on the RSA Cryptosystem. sl : AMS, Vol. 46, No. 2, 1999, AMS, Vol. 46, No. 2, pp. 203-213.
7. Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, Yarik Markov. The first collision for full SHA-1. *https://shattered.io/* : CWI Amsterdam, Google Research, 2017.
8. Valkenburgh, Peter van. What is "open source" and why is it important for cryptocurrency and open blockchain projects? coincenter.org. [Online] 17 oktober 2017. *https://coincenter.org/entry/what-is-open-source-and-why-is-it-important-for-cryptocurrency-and-open-blockchain-projects*.
9. Buterin, Vitalik. What Proof of Stake Is And Why It Matters. Bitcoin Magazine. [Online] 20 11 2013. *https://bitcoinmagazine.com/articles/what-proof-of-stake-is-and-why-it-matters-1377531463/*.
10. Nikolic, Ivika, Kollura, Aashish en Sergey, Ilya, Finding The Greedy, Prodigal and Suicidal Contracts at Scale, *https://arxiv.org/pdf/1802.06038.pdf*
11. Hyperledger, *https://www.hyperledger.org/projects/fabric*
13. Siegel, David. Understanding The Dao Attack. *https://www.coindesk.com/understanding-dao-hack-journalists/*
12. Brown, Richard Gendal. Introducing R3 Coda. A distributed Ledger Designed for Financial Services. *http://www.r3cev.com/blog/2016/4/4/introducing-r3-corda-a-distributed-ledger-designed-for-financial-services*
14. Simonite, Tom. $ 80 Million Hack Shows the Danger of Programmable Money. MIT Technology Review. *https://www.technologyreview.com/s/601724/80-million-hack-shows-the-dangers-of-programmable-money/*
15. Orcutt, Mike. What the Hell is an Initial Coin Offering? MIT Technology

Review. *https://www.technologyreview.com/s/608799/what-the-hell-is-an-initial-coin-offering/*

16. AFM. AFM waarschuwt bij grote risico's bij Initial Coin Offerings. *https://www.afm.nl/nl-nl/nieuws/2017/nov/risico-ico*
17. De Filippi, Primavera, Loveluck Benjamin. The invisible politics of Bitcoin: governance crisis of a decentralised infrastructure. Internet Policy Review, Volume 5, Issue 3. *https://policyreview.info/articles/analysis/invisible-politics-bitcoin-governance-crisis-decentralised-infrastructure*
18. BIP. *https://github.com/bitcoin/bips.*
19. Williams-Grut, Oscar, Price, Rob. A Bitcoin civil war is threatening to tear the digital currency in 2-here's what you need to know. *http://www.businessinsider.com/bitcoins-hard-fork-bitcoin-unlimited-segregated witness-explained-2017-3?international=true&r=US&IR=T*
20. Bitcoin Core. *https://bitcoincore.org/en/team/*
21. Murtaugh, Dan. Bitcoin can drop 50% and China's miners will still make money. *https://www.bloomberg.com/news/articles/2018-01-10/bitcoin-can-drop-50-and-china-s-miners-will-still-make-money*
22. Del Castillo, Michael. The Hard Fork: What's about to happen to Ethereum and The Dao. *https://www.coindesk.com/hard-fork-ethereum-dao/*
23. Harper, Jim. Understanding Bitcoin's Scaling Debate: Politics Comes First. *https://www.coindesk.com/understanding-bitcoins-scaling-debate-politics-comes-first/*