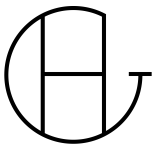


HoGent

Faculteit Bedrijf en Organisatie

Hoe kan de blockchain technologie de problemen oplossen in verband met supply chain in de agricultuur?

Wouter Van Hecke

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Angeline Van Achter
Co-promotor:
Hadrien Van Durme

Instelling: Hogeschool Gent

Academiejaar: 2017-2018

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Hoe kan de blockchain technologie de problemen oplossen in verband met supply chain in de agricultuur?

Wouter Van Hecke

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Angeline Van Achter
Co-promotor:
Hadrien Van Durme

Instelling: Hogeschool Gent

Academiejaar: 2017-2018

Tweede examenperiode

Woord vooraf

Dit is de bachelorproef met als onderwerp: 'Hoe kan de blockchain technologie de problemen oplossen in verband met supply chain in de agricultuur?'. Deze bachelorproef is geschreven in het kader van mijn laatste jaar Toegepaste informatica aan de Hoge School.

Het voorbije jaar was ik al geïnteresseert in blockchain technologie, daarom zag ik deze bachelorproef ook als een perfecte kans om dieper in te gaan in bepaalde aspecten. Samen met mijn stage, dat ook over blockchain ging, heb ik al een redelijke basis dat mij verder kan helpen naar de toekomst.

Graag zou ik volgende personen bedanken.

Ik zou graag als eerste mijn co-promotor, Hadrien Van Durme, bedanken voor het opvolgen en advies geven bij deze bachelorproef. Het is dankzij mijn co-promotor dat ik hyper ledger composer heb leren kennen en hier al een klein netwerk heb kunnen opstellen. Ook heeft hij mij geholpen deze bachelorproef zo correct mogelijk te houden.

Tevens wil ik zeker ook mijn stage mentor Jens Mortier bedanken voor mij deze geweldige kans te geven om stage te lopen bij het bedrijf Trase en mij daardoor ook in de juiste richting te leiden.

Ten slotte wil ik ook nog mijn promotor Angeline Van Achter bedanken als promotor en begeleider.

Samenvatting

Technologie blijft maar evolueren.

Bedrijven moeten mee met de tijd, anders worden ze achtergelaten en voorbij gestoken door andere bedrijven. Maar bedrijven hebben niet altijd de tijd / het vermogen om deze nieuwe technologieën te onderzoeken. Dit soort transformatie neemt veel tijd en geld in beslag. In de wereld van supply chain is dit niet anders, en met de opkomst van Blockchain zijn er heel veel mogelijkheden. Daarom is het belangrijk voor bedrijven dat gebruik van maken een supply chain, om deze mogelijkheden goed te onderzoeken om met een zo goed mogelijke uitkomst te komen. In deze bachelorproef is er gefocust op het bouwen van supply chain netwerk op een blockchain en hier bepaalde testen op uit te voeren.

Natuurlijk is er eerst een literatuurstudie gedaan om kennis op te doen over aspecten zoals blockchain en supply chain. Alle stappen van literatuurstudie tot het bespreken van resultaten zijn te vinden in deze bachelorproef, zie het einde van de inleiding voor een duidelijke indeling hiervan. De conclusie van het onderzoek is dat het goed mogelijk is om blockchain technologie te combineren met supply chain, ookal is blockchain zelf nog in ontwikkeling en zal nog niet alles lukken. Deze technologie is nog lang niet perfect, dus het is belangrijk om hier aan te blijven werken en blijven onderzoeken. Een uitbreiding van het supply chain netwerk zou een goed begin zijn, het realiseren van een inlog systeem met het dynamisch aanmaken van producten bijvoorbeeld.

Inhoudsopgave

1 Inleiding 11

1.1 Probleemstelling 11

1.2 Onderzoeksvraag 12

1.3 Onderzoeksdoelstelling 12

1.4 Opzet van deze bachelorproef 13

2 Stand van zaken 15

2.1 Blockchain + Supply chain 15

2.2 De agricultuur in China 16

2.3 Huidige problemen van een supply chain 16

3 Methodologie 19

3.1 Literatuurstudie 19

3.2 Prototype 19

3.3 Testen 20

4 Bitcoin 21

4.1 De historiek van Bitcoin 21

4.2 Voor- en nadelen van bitcoin 22

4.3 Wallets 23

4.4 Anoniemiteit 24

4.5 De derde partij 24

5 De blockchain technologie 25

5.1 Wat is blockchain? 25

5.2 Eigendom / Identicatie / Authenticatie / Authorisatie 27

5.3 'Double spend' probleem 28

5.4 Hashing 29

5.5 Cryptografie 30

5.6 Het opslaan van data 30

6 Supply Chain 33

6.1 Wat is een supply chain? 33

6.2 Supply Chain Management 34

7 Problemen met supply chain 35

8 HyperLedger composer prototype 37

8.1 Wat is hyperledger composer? 37

8.2 Netwerk voorstelling 37
8.3 Gebruikte technologieën 39
8.4 Demonstratie 45
8.5 Mogelijke uitbreidingen 46
8.6 Conclusie Hyperledger composer 46

9 Uitvoeren testen 47
9.1 Schetsen van de situatie 47
9.2 Uitgevoerde testen 47
9.3 resultaten 49
9.4 conslusie resultaten 54

10 Toekomstig werk 55

11 Conclusie 57

A Onderzoeksvoorstel 59

B Bijlage 63
B.1 Demonstratie applicatie 63
B.2 Resultaten testen 68
B.3 Netwerk data 71

Bibliografie 73

1. Inleiding

Technologie blijft evolueren, dit geldt voor alle sectoren. In deze bachelorproef wordt er dieper ingegaan in de sector van supply chain, meer specifiek voor de agricultuur. Deze technologie begint te verouderen, maar is zeer moeilijk om zomaar te gaan vervangen. De huidige supply chain management systemen zoals ERP en SAP brengen een paar problemen met zich mee, problemen waar een blockchain technologie wel hulp kan bieden. Deze problemen worden onderzocht en er wordt op een praktische manier naar een oplossing gezocht.

Als praktische oplossing wordt een een fictief netwerk opgesteld dat een supply chain representeerd. Op dit netwerk zijn er bedrijven zoals winkels en fabrieken te vinden, net zoals de producten die ze kopen/verkopen. Natuurlijk kan dit netwerk zo uitgebreid mogelijk gemaakt worden, maar voor dit prototype hebben we het gehouden rond het proces van tomaten.

Voor een bedrijf dat dit netwerk zou gebruiken, is de snelheid ervan ook een belangrijk onderdeel. Daarom worden er testen gehouden op de snelheid van het netwerk. Deze testen gaan van het aanmaken van producten tot het ophalen van 1 enkel product.

1.1 Probleemstelling

De bedoeling van deze bachelorproef is om bedrijven dat gebruik maken van supply chain, een prototype te tonen van hoe een supply chain eruit kan zien met een blockchain technologie. Dit kan gaan van bedrijven zoals Colruyt, waar het heel belangrijk is, ook naar de eindgebruiker toe, van waar het product precies komt, onder welke omstandigheden dat

het getransporteert werd en natuurlijk hoe lang het product onderweg was met als laatste een van de belangrijkste vragen: Hoe lang blijven deze producten nog goed?

Aan de andere kant is deze bachelorproef ook handig voor IT-developers dat actief zijn in supply chain technologie. Dankzij dit prototype hebben ze een duidelijk beginpunt, het is dus bedoeling dat er verder kan gewerkt worden bovenop dit protoype.

1.2 Onderzoeksvraag

'Hoe kan de blockchain technologie de problemen oplossen in verband met supply chain in de agricultuur?' Dit is de hoofdonderzoeksvraag. Er wordt gekeken naar de huidige problemen van een supply chain en hoe blockchain een rol kan spelen in het oplossen van deze problemen of hoe blockchain een meerwaarde kan zijn.

Deze hoofdonderzoeksvraag wordt nog eens onderverdeeld in drie kleinere deelvragen:

1. **Hoe kan er met zekerheid gezegd worden dat de producten authentiek zijn?**
   Bij deze vraag wordt er dieper op ingegeaan hoe men met zekerheid kan zeggen dat er niet gebeurt is met de producten dat niet mag en vooral aantonen van fabrieken deze producten deze producten komen. Wat tegenwoordig een heel belangrijk details is voor mensen.
2. **Heeft het aantal gegevens op het netwerk een invloed op het ophalen en wegsturen van informatie?**
   Zoals al eerder aangehaald werd, is het voor een bedrijf zeer belangrijk dat het netwerk optimaal is. Daarom is het handig als er enkele testen worden uitgevoerd op het netwerk.
3. **Welke problemen kunnen niet opgelost worden met blockchain?**
   Blockchain is op zich ook nog in ontwikkeling en zeker nog ver van perfectie. Daarom is het ook belangrijk om aan te halen welke aspecten niet / nog niet mogelijk zijn om met blockchain te realiseren.

1.3 Onderzoeksdoelstelling

Er worden een paar zaken verwacht van de resultaten:

Voor de bevolking is het tegengaan van vervalsing en zekerheid hebben over hun producten topprioriteit. Dit gebeurt de dag van vandaag zeker niet altijd. Maar blockchain kan hier zeker een verandering in brengen. Omdat eens aangebrachte veranderingen voor altijd op de blockchain staat, kan dit door de eindpartij ook zelf bekeken worden of het inderdaad klopt wat er precies op het label staat.

Er wordt verwacht van de resultaten van de testen dat deze snel zal zijn en dat er relatief

gezien niet veel verschil zal zitten tussen testen met een groter aantal gegevens.

Omdat blockchain zelf nog in zijn beginfase zit, zullen er waarschijnlijk ook wel aspecten zijn dat het nog niet kan oplossen. Supply chain zit ook heel complex in elkaar. Er zal naar de toekomst toe zeker nog heel veel werk in kruipen naar gelang implementatie toe.

1.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 4 wordt een introductie gegeven over Bitcoin, dit is een belangrijke basis om verder te gaan met blockchain.

In Hoofdstuk 5 wordt een introductie gegeven over blockchain, hier worden belangrijke concepten aangehaald wat belangrijk zal zijn voor bepaalde vragen.

In Hoofdstuk 6 wordt er gekeken naar wat een supply chain precies inhoud en wat de belangrijke aspecten hier in zijn.

In Hoofdstuk 7 worden de problemen van een supply chain onderzocht en gekeken hoe de blockchain op die vlakken kunnen helpen.

In Hoofdstuk 8 wordt uitleg gegeven over het gemaakte prototype met hyperledger composer. Hierin wordt er vertelt hoe het netwerk is opgemaakt met een demonstratie van de gemaakte mobile applicatie.

In Hoofdstuk 9 worden de testen besproken dat los gelaten worden op het netwerk met de uiteindelijke conclusie van de resultaten.

In Hoofdstuk 10 wordt besproken wat er verder onderzocht kan worden in de richting van supply chain gecombineerd met blockchain / welke andere testen er eventueel nog kunnen uitgevoerd worden.

In Hoofdstuk 11, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2. Stand van zaken

2.1 Blockchain + Supply chain

In het eerste artikel (Goodnight, 2017) wordt er informatie gegeven over wat de supply chain is en precies het doel is van supply chain management. Het geeft ook uitleg over wat de belemmeringen zijn van een supply chain en hoe de blockchain technologie in zijn tijd deze problemen kan helpen wegwerken. Een supply chain wordt omgeschreven als een sequentie van processen dat een product ondergaat. Een supply chain is alles behalve lineair, het is een netwerk van netwerken. De ene uitgang kan de ingang zijn van het volgende product. Maar het gaat niet alleen maar om de productstroom, ook over geldstroom, informatiestroom...

Het belangrijkste van een supply chain is het beheren van de staat, dit kan gezien worden als de laatst geweten status van een proces of transactie. Deze dat wordt meestal ingegeven door meerdere bedrijven, zo kan het ontstaan het ene bedrijf informatie niet invult dat ergens anders wel ingevuld wordt. Zo onstaat er inconsistentie. Nog andere nadelen van de staat op dit moment is dat deze informatie niet real time zijn, er bestaat een vertraging tussen het uitwisselen van informatie tussen het digitale systeem en de fysieke supply chain. Deze informatie dat bijhouden wordt is heel bedrijfs centrisch gelegen. Er kan niet gekeken worden naar informatie buiten het eigen ERP systeem. Dit is namelijk het hoofdprobleem van ERP-systemen.

Dit is alles behalve een netwerk van waarde, maar met de opkomst van de Blockchain technologie, samen met IoT en zelfs AI, kan dit allemaal veranderen in de toekomst. De huidige ERP systemen lijden aan een paar problemen dat kan worden opgelost met de blockchain technologie. Deze problemen zijn het manueel fouten maken bij het ingeven

van gegevens en papierwerkfouten. De informatie ligt heel centrisch, maar het moet wel meerdere keren ingegeven worden door verschillende bedrijven. De aard van deze problemen ligt bij het feit dat bedrijven elkaar elkaar niet vertrouwen met hun informatie en daarom alles gecentreerd houden.

Dankzij het opkomen van IoT, kan de informatie beheert en gecontroleerd worden in real time. Het automatiseren van processen kan vergemakkelijkt worden, terwijl dit wel voor meer transacties per seconden zorgt. De Blockchain technologie helpt daarnaast om één gedecentraliseerde database te verkrijgen, waar er transparantie ontstaat van de producten en processen. Dankzij de blockchain technologie kan er echt een netwerk van waarde worden opgesteld omdat de bedrijven elkaar niet hoeven te vertrouwen, het gedecentraliseerde netwerk zal het vertrouwen afdwingen.

2.2 De agricultuur in China

In het tweede artikel vertelt Hays (2008) over problemen in de agricultuur ter plaatse van China. Hij vertelt over de problemen van de boeren en de algemene problemen van de agricultuur. Het inkomen van de boeren is aan het zakken terwijl hun kosten alleen maar stijgen, terwijl de prijzen van het voedsel gedaald is en er minder vraag is van de middel klasse in China. De boeren hebben geen zekerheid tegen een grote ramp, zoals droogte en dat zo hun oogst niet zal slagen. Het land waarop ze werken is ook niet hun eigendom, ze huren dit van hun overheid. Hierdoor hebben ze weinig tot geen zeg over wat er moet gebeuren met het land in verband met verbeteringen. Ze hebben wel zelf de keuze wat ze planten, daar is er ook een moeilijkheid. Het is niet makkelijk om te voorspellen wat ze precies nodig hebben en hoeveel ze er van nodig hebben.

Nu als het gaat over de gehele agricultuur in China, kan het volgende gezegd worden. Omdat de wegen en infrastructuur er zo slecht zijn, gebeurt het vaak dat de boer zijn producten niet op tijd bij de markten krijgt en ze daardoor rot aankomen. De kwaliteit van de producten zijn soms heel slecht of vervormd, door het gebruik van pesticiden en kunstmest. Vele landschappen zijn aan het uitdrogen of het landschap wordt onbruikbaar.

2.3 Huidige problemen van een supply chain

Het derde artikel (Uhlenberg, 2017) geeft een visie van welke huidige problemen er bestaan voor supply chain management. Maar eerst wordt er besproken wat een goede supply chain inhoud.

Een supply chain moet de rauwe materialen kunnen bekijken, hierbij kan de supply chain manager het inplannen van nieuwe producten maken, effectieve inventaris opmaken om zoveel mogelijk kosten te verminderen en zich kunnen aanpassen aan de vraag van de consument. Alle informatie moet makkelijk verkrijgbaar zijn zodat er zich geen situaties voordoen als het moeten stopzetten van de productie. Een van de grootste problemen is

dat de data real time moet zijn om een veel beter overzicht te krijgen van wat er allemaal gebeurt.

Het eerste punt dat aangehaald wordt is de globalizering. Bedrijven verhuizen hun productiefabrieken meer naar het buitenland voor lagere kosten en minder belastingen, lagere transport kosten enzovoort. Maar dit brengt natuurlijk meer complexiteit binnen hun supply chain. De bedrijven moeten hun materialen meer organiseren en samen werken met andere bedrijven voor hun maken van producten, opslaan van het resultaat en het transport ervan.

De markt is constant in beweging. Aankoop prijzen en verkoop prijzen veranderen constant, net zoals de vraag en het aanbod van de producten. De producten hebben een kortere levenscyclus, dit forceert de bedrijven om hun supply chain aan te passen. Het is moeilijk om te voorspellen hoeveel producten er precies moeten worden gemaakt, dat geldt dus ook voor hoeveel materiaal er moet worden aangekocht.

Het laatste punt dat aangehaald wordt is de kwaliteit van de producten. Door de druk van sociale media moet er veel meer gekeken worden op consistente kwaliteit van de producten. Dit geldt voor het maken van het eindproduct, maar dit is ook belangrijk bij het aankopen van de rauwe materialen. De bedrijven moeten letten dat hun product tijdens de verwerking kwaliteitsvolle processen ondergaat, bv. tijdens het inpakken ervan, tijdens transportatie, zeker tijdens het kweken van eten zoals tomaten of granen.

Tijdens deze processen moet er ook stevig wat papierwerk worden bijhouden. Dit is op zich al een heel makkelijk punt om fouten te maken. Een fout in het ingeven van gegevens kan al grote gevolgen hebben in het proces van het product. Er wordt ook rekening gehouden met documenten zoals vergunningen, licenties, certificaten.

3. Methodologie

3.1 Literatuurstudie

Voor deze bachelorproef is er een basis nodig aan kennis van bepaalde termen, waar onder blockchain en supply chain. Om verder te gaan met het onderzoek moesten deze termen eerst onderzocht worden aan de hand van een literatuurstudie. Bitcoin was de eerste term dat onderzocht werd, omdat het daar eigenlijk allemaal begonnen in. Bij het onderzoeken van blockchain hoort dus ook wat onderzoek naar het ontstaan van Bitcoin. Hierbij werden verschillende artikels bekeken om kennis te vergaren over Bitcoin. Vervolgens was blockchain aan de beurt, om hier informatie over te verkrijgen was onder andere het boek Blockchain Basics benut. In dit boek stonden er aspecten wat de term blockchain precies inhoud en wat het wil bereiken. Ook was er informatie te vinden hoe het op een abstract level inhoudelijk in elkaar zit, wat een goed zicht gaf om te begrijpen wat blockchain precies is en hoe het best gebruikt kan worden.

Daarna was het de beurt aan supply chain. Hierbij werd er in het begin onderzocht wat supply chain en supply chain management precies inhouden. Dan kwamen de problemen van een supply chain. Omdat blockchain al onderzocht was, kon er op het einde van de literatuurstudie al geantwoord worden op de vraag wat welke problemen een blockchain (nog) niet kan oplossen.

3.2 Prototype

Als tweede deel werd een prototype gemaakt van een supply chain op een blockchain. Hiervoor waren er 2 opties: Ethereum en Hyperledger composer. Met Ethereum zou er

een smart contract gemaakt worden om zo het netwerk te simuleren. Maar Hyperledger composer is specifiek bedoeld om netwerken op te bouwen. Daarom lag de keuze al snel bij Hyperledger composer. Als eerste werd er een netwerk bedacht dat bestaat uit bedrijven en producten. Omdat de focus hier gelegd werd bij agricultuur, zullen de producten gesimuleerd worden als tomaten. Er werden verschillende soorten bedrijven gekozen dat elk met hun eigen producten werkten, bijvoorbeeld lasagna of spaghetti saus, dat dus ook gemaakt wordt van tomaten. Om deze gegevens op het netwerk te zetten, moet er wel eerst een netwerk zijn. Dit netwerk opstellen was ook helemaal nieuw, dus moest er onderzocht worden hoe dit precies in elkaar zat. In de loop van dit prototype werd er een Hyperledger composer netwerk opgesteld dat het netwerk simuleerde, een NodeJS applicatie dat de informatie regelde van het netwerk en een mobiele applicatie dat met dit netwerk kon communiceren. Hoe dit netwerk met de applicaties werden opgesteld, komt verder aan bod in Hoofdstuk 8.

3.3 Testen

Ten slotte, werden er enkele testen geschreven dat werken uitgevoerd op het netwerk. De snelheid van een netwerk is heel belangrijk voor een bedrijf dat dit netwerk gebruikt. Omdat deze testen lijken op de NodeJS applicatie, zijn er stukken code uit de applicatie gehaald en hergebruikt. Deze testen kunnen dus ook gezien worden als NodeJS scripts. Omdat het prototype gemaakt was in een virtuele machine, had het netwerk ook zo zijn limitaties. Daarom konden sommige testen spijtig genoeg niet uitgevoerd worden omdat ze te lang duurde en het netwerk dit niet aankon. Hierbij horen de testen van het verwijderen van veel gegevens. Alle (andere) testen werd uitgevoerd met een verschillend aantal gegevens, met de bedoeling om te kijken of het aantal gegevens dat zich op het netwerk bevond, een groot verschil uitmaakte voor de snelheid van de transacties. Hieronden bevinden zich alle testen dat zijn uitgevoerd:

1. Creatie van informatie
2. Verwijderen van informatie
3. Ophalen van alle producten / bedrijven
4. Opzoeken van één bepaalde ID

4. Bitcoin

Voordat we beginnen met de hoofdzaken, is het belangrijk dat we eerst de terminologie rond blockchain goed snappen. Vandaar deze intro over Bitcoin en blockchain.

4.1 De historiek van Bitcoin

Het onstaan van Bitcoin gaat helemaal terug tot in 2008, toen een mysterieuze persoon genaamd Satoshi Nakamoto een paper (Nakamoto, 2008) uitbracht onder de titel: "Bitcoin, a peer to peer electronic cash system".Wie deze persoon precies is, is tot op deze dag nog altijd niet gekend. Satoshi Nakamoto is in 2011 van het internet verdwenen.

In het jaar 2009 werd voor het eerst het gedecentraliseerde Bitcoin system publiek gemaakt. Dit hield in dat mensen bitcoins konden minen - een wiskundig proces waarbij nieuwe bitcoins worden gecreeërd - als ook transacties versturen en verifiëren op de Bitcoin blockchain. Meer over blockchain in het volgende hoofdstuk.

In het jaar 2010 kreeg een bitcoin voor de eerste keer een waarde, er werden maar liefst 10 000 bitcoins ingewisseld voor 2 pizza's. Op het moment van het schrijven van deze bachelorproef, zou dat een waarde hebben van 60 miljoen euro. Dit lijkt nu een belachelijk groot bedrag, maar dit moment was ook heel memorabel in de geschiedenis van bitcoin. Het was de allereerste keer dat er een echt product werd gekocht en dus dat bitcoin voor de eerste keer als een echt betaalmiddel gebruikt werkt.

Er werd meer en meer interesse getoond in deze gedecentraliseerde technologie, er kwamen nieuwe projecten boven water met het doel om de Bitcoin blockchain te verbeteren. Deze

projecten hebben ook hun eigen coin of token, een paar bekende voorbeelden zijn: de Ethereum blockchain met hun coin genaamd Ether, de Ripple blockchain met hun coin genaamd XRP. Er zijn in totaal meer dan 1000 van deze coins, meestal worden ze onder een overkoepelende term gezet, namelijk altcoins.

Een blockchain draait vooral rond het decentraliseren van centrale systemen en het oplossen van het vertrouwensprobleem tussen verschillende partijen. Op dit punt in tijd stond de beveiliging nog niet helemaal op peil en bestaan er nog altijd kwetsbaarheden in het netwerk. Omdat bitcoin op dat moment redelijk nieuw was en een digitale waarde voorstelde, was dit ook een doelwit voor hackers. In het jaar 2014 gaat de bekende site Mt. Gox offline nadat ze gehacked waren en er maar liefst 850,000 bitcoins gestolen werden, wat nu een waarde zou hebben van maar liefst 5.1 miljard euro.

Maar zelfs dat hield Bitcoin niet tegen, over de jaren heeft het als maar meer en meer aandacht gekregen en is de waarde blijven stijgen. In het jaar 2016 kwam er een nieuw concept genaamd ICO's of wel "Initial coin offering", ook vergelijkbaar met crowdfuncing of "IPO". Start up bedrijven konden op deze manier hun initieel bedrag verkrijgen en de mensen die hun geld gaven coins in de plaats geven. Die coins hebben op zich ook een waarde dat kan zakken of stijgen.

Tot op de dag van vandaag is bitcoin blijven stijgen in waarde tot op een hoogtepunt van 18 000 euro. (Marr, 2017)

4.2 Voor- en nadelen van bitcoin

Sommige aspecten van Bitcoin kunnen zowel als positief als negatief aanzien worden. voorbeelden hiervan zijn: transactiesnelheid, privacy, controle...

Voordelen

Omdat er geen centrale autoriteit is op een blockchain ben je onafhankelijk van banken of de overheid. Bitcoin is een globale munt waar je dus overal met kan betalen.
Het verwerkingsproces van het verzenden op de Bitcoin blockchain is in vergelijking met het huidige banken systeem veel sneller. Als je een transactie wil uitvoeren bij een bank, kan dit gemakkelijk 3 dagen duren. Bij Bitcoin zal dit ongeveer gebeuren in 1 uur. Er zijn al projecten zoals Ripple of NEO waar de transactieduur verminderd werd tot een paar seconden/minuten.
Omdat je altijd in het bezit bent van je eigen coins, heb je zelf de complete controle, indien je de coins veilig houdt in een wallet. Niemand kan aan deze coins behalve de persoon die in het bezit is van de privé sleutel. Meer informatie over wallets in het volgende hoofdstuk.
Als er transacties gebeuren op een blockchain weet niemand wie je bent, wat ze wel kunnen zien is bijvoorbeeld van welk adres de bitcoins komen en naar welk ander adres ze verstuurd wordt. Er werd dus een pseudo-anonimiteit op de blockchain ontwikkeld waar

niemand uw identiteit precies kent.

Nadelen

Aan bijna elk voordeel zit er ook een nadeel.
Omdat er altijd maar meer en meer mensen het Bitcoin netwerk toetreden, wordt het alsmaar duurder en langer voor een transactie uit te voeren, meer uitleg over blockchain transacties in het hoofdstuk van Blockchain.
Omdats sommige blockchains volledig anoniem zijn, is het veel makkelijker om deze coins te gebruiken voor het aankopen van drugs en wapens op de zwarte markt, omdat je veel moeilijker op te sporen bent.
Er is geen centraal controlepunt door een overheid bijvoorbeeld, maar dat wil niet zeggen dat ze zich er niet met bemoeien. In vele landen wordt er al gesproken over het invoeren van belastingen op de winst dat je krijgt als je belegd in coins. Ze zullen dit ook proberen reguleren, wat dus eigenlijk het overgestelde is van wat een blockchain wil zijn.
Volledige controle over je eigen geld is natuurlijk heel goed, maar wat gebeurt er als er iets fout gaat? Je staat er helemaal alleen voor want er is niemand waar je naar toe kunt voor hulp. Je moet dus altijd heel voorzichtig zijn met wat je doet met je geld. (Verdanov, 2017)

4.3 Wallets

Als je in het bezit wil zijn van je eigen coins, zal je deze moeten opslaan in je eigen wallet. Er zijn er paar verschillende wallets zoals een "cold storage wallet", een ’paper wallet’ of een "web wallet". Bij het aanmaken van een nieuwe wallet krijg je 3 belangrijke gegevens: een adres, een publieke sleutel en een private sleutel. Met je private sleutel krijg je toegang tot je geld terwijl je publieke sleutel openlijk bekend staat op de blockchain. Meer informatie over beide sleutels in de blockchain hoofdstuk.
Elke wallet heeft zijn eigen adres, dat gebruikt wordt en geld naar overzetten. Stel dat persoon A 5 bitcoins naar persoon B wil sturen, dan zal persoon A dit naar het adres van persoon B sturen.

Er bestaan een paar verschillende wallets die hier kort besproken zullen worden:

Online web wallet

Een voorbeeld van een ’web wallet’ is MetaMask. Dit is een wallet dat je kan openen in een webbrowser.

Software wallet

Een software wallet is een wallet dat je zal draaien op je eigen computer en zo je coins kan

beheren. Een voorbeeld van een software wallet is Exodus.

Hardware wallet

Een hardware wallet kan je vergelijken met een usb stick, speciaal geprogrammeerd als wallet. Het voordeel van een hardware wallet is dat je private sleutel nooit het apparaat verlaat en dus ook niet met het internet verbonden is tot zo lang dat je er niet zelf met connecteert. Een hardware wallet is de aangeraden optie voor mensen dat al een groter bedrag hebben.

Paper wallet

Met een 'paper wallet' zal je de publieke en private sleutel afdrukken om een papier en zo bijhouden. Dit stukje papier is dan de enige toegang dat je hebt tot je geld. (Rosic, 2017)

4.4 Anoniemiteit

Op een blockchain blijft een persoon normaal pseudo-anoniem tot dat deze persoon onderheven wordt aan het proces van "Know your customer". Er zijn wel blockchains waar een persoon helemaal anoniem kan zijn, bv op de blockchain van Monero. Op sommige exchange site of uitwisselingsite moeten mensen hun gegevens invullen voordat ze cryptocurrency kunnen kopen. Dit is een website waar je verschillende coins kan kopen en verkopen. Maar voordat je dat kan doen, moet je eerst geregistreerd staan op hun database. De coins die op de exchange staan, heb je dus zelf niet in eigen bezit, maar ze zijn wel in het bezit van de exchange, omdat deze exchange in het bezit is van de privé sleutel van de persoon in kwestie.

4.5 De derde partij

Een van de zaken dat een blockchain kan doen is de man in het midden wegwerken. Het grootste voorbeeld is nu een bank. De mogelijk om geld snel en goedkoop naar iemand te versturen zonder dat je moet rekenen op een bank is gigantisch. Omdat een blockchain een peer to peer systeem is, bestaat er een connectie van persoon A naar persoon B zodat er geen tussen persoon nodig is.

5. De blockchain technologie

(Drescher, 2017)

5.1 Wat is blockchain?

Verschil gecentraliseerd / gedecentraliseerd netwerk

Bij het maken van een software systeem, kom je voor de keuze te staan over de architectuur van het netwerk, dit kan zowel centraal als decentraal. Deze twee zijn de belangrijkste architecturen. Bij een centraal netwerk zal er altijd een centraal component aanwezig zijn waar al het verkeer naartoe lijdt. Dit maakt het ook een makkelijk doelwit voor hackers. Als het lukt om het centrale punt neer te halen, ligt heel het netwerk plat. In vergelijking met een gedecentraliseerd netwerk waar er geen centreel punt is, het netwerk blijft runnen ookal wordt er 1 component (of node) uit het systeem wordt gehaald. In een gedecentraliseerd systeem is elke node indirect met elkaar verbonden, dat wil niet zeggen dat elke node met elk andere node verbonden zal zijn, maar via via er wel een connectie met heeft.

Dit zijn de voordelen van een blockchain:

Het grootste voordeel van een blockchain is het oplossen van het vertrouwensprobleem. Partijen vertrouwen elkaar meestal niet, er komt nog een derde partij tussen dat de twee andere partijen wel vertrouwen dat alles vertrouwbaar verloopt. Deze derde partij vraag

hier ook geld voor. Een blockchain zorgt ervoor dat deze derde partij niet meer nodig is door de manier hoe een blockchain systeem wordt opgesteld.
In vergelijking met een centraal systeem, waar de kracht meestal uit een (super) computer wordt gehaald, wordt de kracht in een gedecentraliseerd systeem gehaald uit de combinatie van alle nodes in het netwerk. Hierdoor is de rekenkracht meestal veel groter.

Alhoewel de initiële kost voor het opzetten van de gedecentraliseerd systeem hoger ligt dan bij een gecentraliseerd systeem, op langer termijn zal je nog altijd veel kosten bespraren omdat het veel duurder is om een (super / centrale) computer te beheren en onderhouden. Het is ook veel makkelijker om in een gedecentraliseerd systeem een computer te vervangen.

Als er in een gedecentraliseerd netwerk één van de computers crasht, heeft dit weinig gevolgen op het volledige netwerk.

Door het toevoegen van meer computers in het netwerk, verhoog je de rekenkracht van het hele netwerk. Daarom is er een natuurlijk vermogen om het netwerk uit te breiden. Hierdoor zijn er wel meer ingangen tot inbraak in het netwerk.

Natuurlijk heeft een gedecentraliseerd systeem ook zijn nadelen:

In een gedecentraliseerd netwerk is er geen coördinatie door een centraal punt, daarom moet er rekenkracht opzij gehouden worden zodat elke node dit zelf kan doen. Niet zomaar elke node is goed genoeg voor het netwerk, zo moet een node voldoen aan een minium specificatie om zich te kunnen voorstellen als een node van het netwerk.

Coördinatie vereist communicatie, dit moeten de nodes ook zelf doen.

Een geschreven software voor op een gedecentraliseerd systeem moet nog een paar extra taken vervullen. Namelijk de nadelen dat hierboven beschreven staan. Hierdoor worden de programma's veel complexer omdat natuurlijk alle nodes over een en dezelfde informatie moet beschikken.

In een publieke blockchain kan iedereen toetreden, hierdoor is er niet geweten of elke node te betrouwen in. Deze onbetrouwbare node zouden verkregen informatie kunnen vervalsen en terug doorsturen. Hierdoor zal er dus een stevige beveiliging aanwezig moeten zijn in het programma. In vergelijking met een private blockchain waar de node moet toegelaten worden door fundament dat de private blockchain heeft opgericht.

Integriteit, een heel belangrijk aspect van een blockchain

Integriteit. Voor een blockchain staat integriteit op nummer 1. Een blockchain kan aanzien worden als een instrument voor het bekomen en onderhouden van de integriteit van een decentraliseerd systeem. Maar wat bedoeld men nu precies?
Het is een heel belangrijk niet-functioneel aspect van een software programma, de gebruikte

data moet correct en volledig zijn, het programma moet op een correcte manier werken / niet vast lopen en de gegevens mogen altijd maar toegankelijk zijn voor de mensen dat daadwerkelijk toegang hebben. Als deze waarden geschonden worden, kan dit grote gevolgen hebben voor de gebruikers. Een voorbeeld is bij het vast lopen van een excel bestand terwijl er niet opgeslagen werd. De programmeurs doen er alles aan om dit zoveel mogelijk te verkomen en moest het wel gebeuren, ten minste de data veilig te stellen. Zodra er data wordt weggeschreven naar ee blockchain, staat dit er voor altijd op. Stel dat er een programma wordt opgezet en hier zitten fouten in. Dan staat dit ook fout op de blockchain en kan dit niet meer angepast worden.

Hoe ziet een blockchain er nu eigenlijk uit?

Je kan een blockchain letterlijk zien als een ketting van blokken. In elke blok zitten de gegevens van alle transacties die in het verleden tot op het heden al gebeurd zijn. Deze gegevens zijn niet te vervalsen dankzij het gebruik van hash-functies, meer daarover in één van de volgende hoofdstukken. Elke node in het netwerk probeert de eerste te zijn dat een nieuwe block kan maken, vanaf er iemand een blok gemaakt, moeten al de andere nodes stoppen met hun blok en de gemaakte blok verifiëren op correctheid. Als deze blok inderdaad correct is, nemen ze deze op in hun geschiedenis en proberen ze terug de volgende block te creëren of met andere woorden "minen".

Terminologie van een blockchain?

Een blockchain kan gezien worden als een data structuur. Het is een manier van data te organiseren. De data wordt opgeslaan in een blok en elke blok wordt met elkaar geconnecteerd. De ordering waarin dit gebeurt is van groot belang. Stel dat er bitcoin uitgewisseld worden over 4 adressen, dan wil je de juiste volgorde weten van welk adres het kwam en naar waar het gaat, en zeer belangrijk, wie dus de eigenaar is.
Een blockchain kan ook gezien worden als een algoritme, het gaat hem hier dan over de sequentie van instructies dat er op het netwerk gebeurt in verband met creëren van een blok, verifiëren van een blok, de juiste historiek kiezen...
De blockchain technologie wordt anderzijds ook aanzien als een soort van overkoepelende mantel van verschillende zaken, zoals de combinatie van de data structuur, het algoritme, maar zowel van cryptografie dat instaat voor eigendom op de blockchain en de beveiliging voor op het gedecentraliseerde netwerk.

5.2 Eigendom / Identicatie / Authenticatie / Authorisatie

Een kernbegrip van een blockchain gaat over de eigendom. Weten wie de eigenaar is en van wat de persoon de eigenaar is. Maar hoe wordt er nu bepaald wie een eigenaar is. Stel dat deze informatie zich bevindt op één plaats, dan hebt je één punt dat je moet vertrouwen. Dit doet een blockchain anders. Op een blockchain wordt deze informatie op meerdere plaatsen opgeslaan en wordt er dus gekeken naar de wet van meederheid. Stel dat er acht van de tien nodes zeggen dat transactie A correct is en de andere twee nodes zeggen dat transactie A niet correct is, dan is er meer kans de transactie A inderdaad klopt.

Dit concept is te vergelijken met een rechtbank, waar er meerdere mensen een beslissing moeten maken in plaats van één.

Het kunnen bewijzen van de eigendom

Het bewijzen van activieten van een persoon gebeurt via een signature. Elke persoon heeft een privé sleutel en een publieke sleutel. Als deze persoon een bericht of transactie encrypteert met zijn privé sleuten, is dit bericht alleen maar decrypteerbaar met de publieke sleutel van die persoon. Zo weet men met zekerheid dat de persoon in kwestie daadwerkelijk het bericht heeft gestuurd. Hierover meer bij het gedeelte van cryptografie.

Identicatie / Authenticatie / Authorisatie

Identificatie betekent het beweren van wie je bent. Bijvoorbeeld zeggen dat Jan noemt. Hier komt er nog niets te pas van hoe je dit gaat bewijzen.
Authenticatie betekent het bewijzen van wie je bent. Bijvoorbeeld je rijbewijs voorleggen. Authorisatie betekent het toegang krijgen tot iets waar je voor moest bewijzen dat je daadwerkelijk toegang had. Bijvoorbeeld een persoon dat 18+ moet zijn om te mogen rijden. Dit kunnen we ook reflecteren naar de blockchain technologie. Op een blockchain is iedereen uniek, ze worden gescheiden van elkaar, niet door een naam, maar wel door een privé sleutel. Meer over private sleutels later. Het authenticatie gedeelte op de blockchain wordt afgehandeld met behulp van "two-way hashing". Dit is een combinatie van je privé sleutel, je publieke sleutel en hash-functies. Meer hierover in het deel over hashing.

Integriteit op een blockchain heeft dus onder andere te maken over echte uitspraken over eigendommen.

5.3 'Double spend' probleem

Een van de meest schendende voorbeelden van de integriteit is het 'Spending money twice' probleem.

Wat is het probleem?

De informatie over de eigendom staat op elke node in het netwerk opgeslaan. Als er één verandering gebeurt van deze informatie, moeten alle andere nodes dit ook weten. Maar dit neemt natuurlijk tijd in, dus zal de blockchain in een status komen waarvoor er nodes zijn die de nieuwe informatie al kennen wanneer dat nog niet zo is bij andere nodes. Dit zou gemanipuleerd kunnen worden door de eigenaar van bijvoorbeeld een auto. Stel nu dat de eigenaar van een auto 'persoon A', zijn auto wil verkopen aan 'persoon B' en dus de eigendom doorgeven. Vanaf deze auto doorgegeven wordt en de nodes bezig zijn met het aanpassen van hun informatie, kan 'persoon A' naar een andere node gaan waar de informatie nog niet is toegekomen en daar de auto verkopen aan 'persoon C'. Dit is het

probleem dat wordt gezien als het 'spending money twice' probleem.

Hoe wordt dat probleem opgelost?

Om dit probleem op te lossen wordt dus de technologie van de blockchain gebruikt. Dit zal duidelijker worden aan de hand van volgende deelhoofdstukken.

5.4 Hashing

Op een blockchain krijg je te maken met heel veel data, deze data moet je op een snelle en efficiënte manier uniek maken en kunnen vergelijken. Vandaar dat er hier hashing wordt toegepast. Hash functies zijn kleine programma's dat eender welke data als input krijgt en die data verandert in een combinatie van hexadecimale getallen.

Aspecten van cryptografische hash functies

Bij het gebruik van cryptografische hash functies, zijn de hash values altijd uniek. Identieke input zal altijd een identieke hash value geven.
De output is pseudorandom, dat betekent dat de output van een bijna identieke input, veel verschil kan hebben.
Een hashing functie is een 'one-way functie', dat betekent dat je bijna onmogelijk van de output terug naar de input kan gaan.
Er zijn miljoenen verschillende combinatie van hash values, dus de kans dat er twee keer dezelfde hash value wordt gevonden voor een identieke input is immens klein.

Een voorbeeld van hashing

Verschillende hash patronen

Het eerste en simpelste voorbeeld is onafhankelijk hashen. In dit partoon wordt van elke data individueel de hash value berekend.

In het volgende voorbeeld bespreken we het herhaald hashen. In dit partoon wordt de input eerst gehashed, en daarna wordt de hash value nog eens gehashed. Dit is mogelijk omdat een hash value ook een vorm van data is en daarom kan dienen als input.

Het volgende partoon gaat over combinerend hashen. Stel dat we 2 inputs willen hashen met het combinerend hashing partoon, dan zal als eerste de input aan elkaar geplakt worden. Daarna wordt er één hash value gecreëerd van de gecombineerde input. Een nadeel aan dit patroon is dat de hash value van de individuele inputs niet gekent is.

Het sequentieel hashing patroon, Dit is een combinatie van de het herhaalde hashing patroon en het combinerende hashing patroon.De eerste input zal gehashed worden. Deze

hash value zal gecombineerd worden met de volgende input, wat op zijn beurt weer een nieuwe hash value geeft. Zo zal dit proces blijven door gaan tot dit compleet is.

Als laatste patroon bespreken we het hiërarchische hashing patroon ofwel een Merkle Tree genoemd. Dit patroon heeft een omgekeerde boom structuur. Stel dat je werkt met twee inputs, dan worden deze twee eerst apart gehashed en daar zullen deze twee hash values samen gehashed worden dankzij het combinerende patroon.

5.5 Cryptografie

Symmetrische cryptografie

Bij symmetrische cryptografie bestaat er maar één sleutel voor het encrypteren en het decrypteren. Degene dat het bericht encrypteerde, dus het dus ook terug decrypteren.

Asymmetrische cryptografie

Asymmetrische cryptografie zit een beetje ingewikkelder in elkaar. Hierbij heb je twee complementaire sleutels. Als een bericht met sleutel A wordt geëncrypteerd, kan dat alleen maar met sleutel B. Vandaar de privé sleutel en de publieke sleutel. De privé sleutel is alleen voor de eigenaar die het bericht verstuurd en de publieke sleutel wordt aan iedereen gegeven, zonder te weten of ze betrouwbaar zijn of niet. Merk op dat er dus een bericht kan geëncrypteerd worden met ofwel de privé sleutel of de publieke sleutel, beiden met een verschillende functie. Als een bericht met de privé sleutel geëncrypteerd wordt, kan je dit vergelijken met een broadcast bericht waar iedereen dit kan lezen, want iedereen heeft de publieke sleutel. Als eens bericht met de publieke sleutel geëncrypteerd wordt, dan kan je dit vergelijken met een soort brievenbus. Waar iedereen berichten kan sturen maar alleen de eigenaar ze kan decrypteren.

5.6 Het opslaan van data

De blockchain structuur

Er werd al eerder uitgelegd dat de structuur van een blockchain vergeleken kan worden met een ketting van blokken. Nu zullen we een beetje dieper ingaan op die structuur. Als er een blok wordt gemaakt met data in, dan wordt deze block gehashed, en deze hash wordt opgeslaan in de header van de volgende blok. Maar hoe ziet de data er precies uit in een blok? De data kan vergeleken worden met een boom structuur. In de kruik staan de transacties, deze worden allemaal gehashed. Daarna worden 2 hash values terug samen gehashed tot op het punt dat er maar 1 hash value aan de wortel van de boom staat. Deze laatste hash value wordt opgeslaan in de header, samen met de hash value van de vorige

block.

Oude transacties proberen wijzigen

Daarom is de blockchain dus zo fraudebestendig, mdat alles sequentieel gehashed wordt en zelfs een kleine verandering aan de input een grote verandering kan brengen aan de hash value. Als er iemand de data van transactie 2 wil aanpassen, dan zal de hash value van R2 niet meer kloppen, met het gevolg dat de hash values van R12, B1 en B2 ook niet meer zullen kloppen. Hierdoor kan de fout heel snel opgespoort worden. Als iemand de informatie in een transactie wil veranderen, dan zal deze persoon elk hash value terug moeten veranderen en dit kost heel veel tijd en computerkracht.

6. Supply Chain

Nu dat we waar bitcoin en blockchain voor staan, is het tijd om te kijken naar supply chain. Wat is een supply chain? Hoe werkt het? Wat loopt er zoal mis?

6.1 Wat is een supply chain?

Een supply chain stelt het fysieke netwerk voor tussen de bedrijven, de leveranciers en de retailers. Het stelt alle stappen voor dat een product ondergaat van begin tot einde. Dit gaat over hoe/waar het product begon als rauw material, het transport naar de fabrieken, het proces van de rauwe materialen in de fabriek, distributie over de wereld, transport naar de winkels en zo uiteindelijk naar de klant. Het stelt het hele proces voor.

In een supply chain zijn er een paar flows: information flow, product flow, money flow. Information flow vloeit door heen het hele netwerk, van een naar andere peers. Product flow vloeit van de maker naar de eindklant. Money flow vloeit van de eindklant terug naar de maker.

Het aspect van betrouwbaarheid binnen een supply chain is heel belangrijk. Er is noodzaak aan vertrouwen van de leveranciers, zodat er geweten is dat je kan rekenen op kwaliteitsvolle producten dat ook optijd geleverd zullen worden. Natuurlijk hebben leveranciers soms ook hun eigen leveranciers, zo worden er al snel meerdere netwerken betrokken bij het op tijd zijn van het proces. Éen fout kan al snel meerdere gevolgen hebben.

Het is dus voordelen om de beste oplossing te vinden voor de gekozen supply chain, betere optimalisatie betekent op zijn beurt ook weer mindere kosten en dat betekent minder kosten

voor de eindklant. Optimalisatie is key.

6.2 Supply Chain Management

Bij een supply chain, hoort naturlijk ook een supply chain management systeem. Dit is meer te vergelijken met het digitale deel van de supply chain. Supply chain management kan gezien worden als een mantel van activiteiten dat bestaat uit het plannen, controleren en uitvoeren de processen van de supply chain, om dit zo vlot en kost efficiënt mogelijk te maken. Andere taken zijn onder andere controleren en beheren van de inventaris, het regelen van transportatie, het documenteren van alles...

Een supply chain kan heel duur en complex zijn, daarom is het belangrijk om zo efficiënt mogelijk te werk te gaan. Er zijn en paar software paketten op de markt dat verspreid gebruikt worden: ERP, Oracle en SAP. Je kan deze software de ruggengraat noemen van een bedrijfscentrische supply chain. Supply chain management zorgt voor de planningen van alle activiteiten, risico beheer, opvolgen en optimaliseren van de inventaris, het regelen van transport, alles documenteren, dag tot dag productie... (Sme, 2017)

7. Problemen met supply chain

8. HyperLedger composer prototype

8.1 Wat is hyperledger composer?

Hyperledger composer is een open ontwikkelings tool om blochchain applicaties te maken. Een business netwerk kan makkelijk worden omgezet in een blockchain netwerk. Het maakt gebruik van middelen, zoals tomaten, certificaten, vervoer. Het maakt gebruik van transacties dat inwerkt in de middelen. Daarnaast heb je ook nog toegang tot toegans-controle, identiteiten en andere deelnemers in het netwerk, zoals de koper en verkoper, leveranciers en fabrieken.

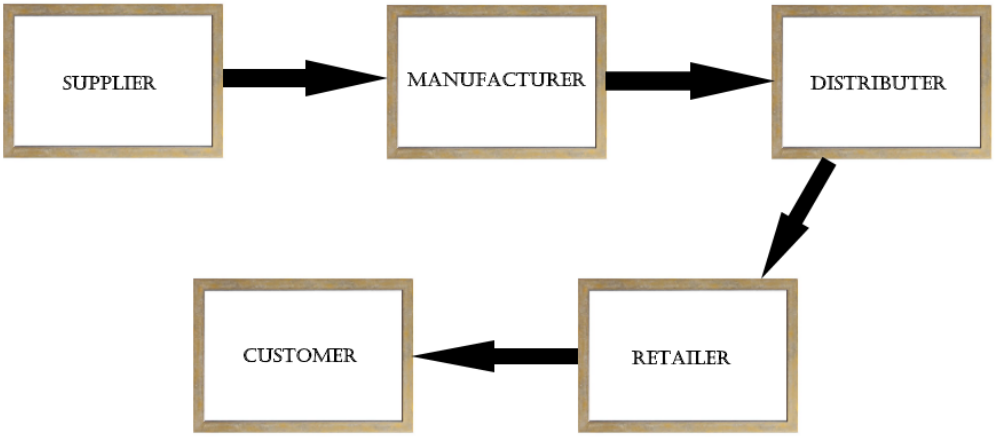
Met hyperledger composer kan een protoype opgezet worden om de supply chain te simuleren in de agricultuur. (Hyperledger, 2015)

8.2 Netwerk voorstelling

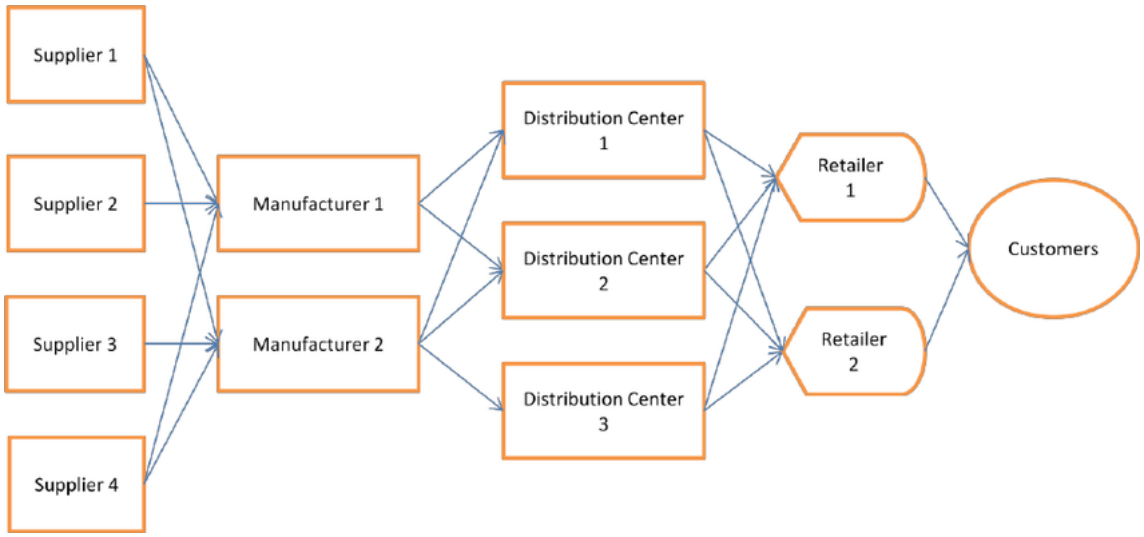
Om deze simulatie zo goed mogelijk te laten verlopen, werd er een fictief netwerk opgesteld van bedrijven. Deze bedrijven bevatten de volgende soorten: "supplier", "manufacturer", "contributer", "retailer", "customer". Elk soort bedrijf heeft een paar voorbeelden gekregen. Buiten de bedrijven mogen we de producten ook zeker niet vergeten. In dit netwerk draait alles om tomaten. Wat er nodig is om tomaten te maken, van deze tomaten producten maken en deze te verkopen. Naast de bedrijven en producten, hebben alle bedrijven ook een eigen Wallet, voor de transacties zo goed mogelijk te simuleren. Hier onder wordt er een tabel gegeven van alle informatie dat beschikbaar is:

Zie Bijlage C voor alle gegevens te bekijken.

Een netwerk in supply chain is nooit zomaar een ketting van bedrijven. Het is een heel netwerk op zich. De ene output van een bedrijf kan de input betekenen van een ander bedrijf. Een bedrijf heeft veel verschillende zaken nog voor het maken van een product, maar in deze simulatie wordt er gefocust op tomaten. Zie hier een foto van 1 ketting in een heel netwerk:



Maar meestal is een supply chain netwerk niet gewoon 1 ketting, maar dan bestaat dit uit een netwerk van kettingen, zoals dit te zien is op onderstaande foto (Celikbilek, 2015).



8.3 Gebruikte technologieën

HyperLedger Composer

Met behulp van de HyperLedger composer, is er een netwerk opgesteld van bedrijven en producten. Hiervoor zijn een paar stappen nodig:

1. Hyperledger fabric starten starten

```
cd ~/fabric-tools
./startFabric.sh
./createPeerAdminCard.sh
```

Copy

Om het netwerk te gebruiken en er moeten we het eerste starten met bovenstaande starten. In dit voorbeeld is hyperledger fabric al geïnstalleerd.

2. Netwerk starten

```
composer network install --card PeerAdmin@hlfv1 --archiveFile tutorial-
network@0.0.1.bna
```

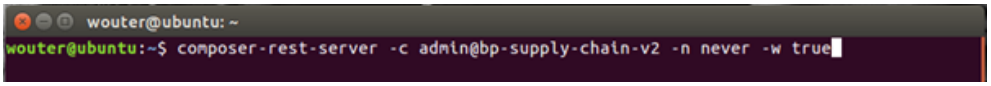
Copy

```
composer network start --networkName tutorial-network --networkVersion 0.0.1 --
networkAdmin admin --networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1 --
file networkadmin.card
```

Copy

Bij het installeren en het starten van het netwerk worden gegevens meegegeven zoals de admin kaart, de bna file waar alles gecompesseerd instaat, de versie van het netwerk dat ingesteld wordt in de package.json...

3. Composer rest server starten



Via HyperLedger wordt er gezorgd voor een rest-api server. Met behulp van deze lokale server kunnen er verzoeken verzonden worden vanaf een applicatie naar de rest-api server dat op zich de nodige data uit het netwerk zal halen. Hier wordt meer op in gegaan in het deel van NodeJS.

Het netwerk bestaat uit een aantal bestanden: een model bestand, waar alle bedrijven en producten worden gedefiniëerd, maar ook de verschillende transacties. Een logica bestand waar de logica instaat tijdens het uitvoeren van een transactie. Er wordt ook gebruik

gemaakt van een acces file, hierin staan alle beveiligingsregels van wie wat mag doen op het netwerk. Er wordt niet verder ingegaan op dit bestand in deze simulatie.

Dit is de gebruikte structuur van het model bestand. Op de foto is onder andere het product en het bedrijf te zien, net zoals een transactie en een event declaratie.

In de volgende foto is dan weer de logica te zien van een transactie. Hier in wordt het register van alle producten opgehaald en wordt het product dat aangepast wordt geüpdatet. Daarna wordt er een nieuwe event aangemaakt dat zichtbaar zal zijn in de NodeJS applicatie.

```
19  namespace org.obliviate.supchain
20
21  asset Product identified by productID {
22    o String productID
23    o UsedProduct productName
24    o Integer quantity
25    o Double price
26    --> CompanyParticipant owner
27  }
28
29  participant CompanyParticipant identified by companyID {
30      o String companyID
31      o String companyName
32      o String country
33      o ParticipantTypes companyType
34      o UsedProduct[] productTypes
35      --> Wallet networkWallet
36  }
37
38  transaction TradeProduct {
39      --> Product productOldOwner
40      --> Product productNewOwner
41      o Integer quantity
42  }
43
44  event TradeProductSuccesEvent {
45    o String productName
46    o Double price
47    o Integer quantity
48    o String oldOwnerName
49    o String newOwnerName
50  }
51
```

```
/**
 * Sample transaction
 * @param {org.obliviate.supchain.CreateMoreSupply} createSupply
 * @transaction
 */
async function createSupply(tx) {
    if(tx.product.owner.companyType == 'SUPPLIER'){
        //access
        const oldQuantity = tx.product.quantity;
        tx.product.quantity = tx.product.quantity + tx.quantity;

        const assetRegistry = await getAssetRegistry('org.obliviate.supchain.Product');
        await assetRegistry.update(tx.product);

        // Emit an event for the modified asset.
        let event = getFactory().newEvent('org.obliviate.supchain', 'CreateMoreSupplySuccesEvent');

        event.companyName = tx.product.owner.companyName;
        event.productName = tx.product.productName;
        event.oldQuantity = oldQuantity;
        event.newQuantity = tx.product.quantity;

        emit(event);
    }else{
        //no access
        let event = getFactory().newEvent('org.obliviate.supchain', 'TransactionFailedEvent');

        event.transactionFunction = "CreateMoreSupply";
        event.message = "This product doesn't belong to a supplier company";

        emit(event);

        throw new Error('See event for information');
    }
}
```

NodeJS

Omdat het netwerk elk keer weer leeg is als het opnieuw wordt opgestart, moeten we een manier vinden het netwerk terug te vullen met informatie. Dit kan gedaan worden door de "composer-client"package te gebruiken in NodeJS.

Eerst zal er een connectie gemaakt moeten worden met het netwerk via de "composer-client":

```
'use strict';

const BusinessNetworkConnection = require('composer-client').BusinessNetworkConnection;

let cardname = 'admin@bp-supply-chain-v2';
```

```
static async start() {
    let cf = new ComposerFiller();
    await cf.init().then(() => {cf.listen()});
}
```

```
class ComposerFiller {

    constructor() {
        this.SupChainConnection = new BusinessNetworkConnection();
    }

    async init() {
        return this.businessNetworkDefinition = await this.SupChainConnection.connect(cardname);
    }
```

Er zijn 3 verschillende scripts geschreven: een script om het netwerk te vullen met gegevens, een script om alle informatie in het netwerk te verwijderen en een script dat luistert naar event uit het netwerk.

```
let participant = factory.newResource('org.obliviate.supchain', 'CompanyParticipant', this.info[i][j][0]);
let wallet = factory.newResource('org.obliviate.supchain', 'Wallet', "W-" + this.info[i][j][0]);

participant.companyName = this.info[i][j][1];
participant.country = this.info[i][j][2];
participant.companyType = this.info[i][j][3];

if(i == 3){
    participant.productTypes = this.info[i][j][4];
}else{
    participant.productTypes = [this.info[i][j][4]];
}

wallet.privateKey = "E9873D79C6D87DC0FB6A57786333" + this.info[i][j][0] + "89F4453213303DA61F20BD67FC233AA33262";
wallet.publicKey = "6FPZBT79E9G1VPR0FB6A5778633389F4453846303DA61F2R9FKS5C233AA33262";
wallet.address = "E9873D79C6D87F2R9FKS5C233AA3";
wallet.balance = 100000;

participant.networkWallet = factory.newRelationship('org.obliviate.supchain', 'Wallet', "W-" + this.info[i][j][0]);

participants.push(participant);
wallets.push(wallet);
```

```
await participantRegistry.addAll(participants).then(() => {console.log('All participants added')});
await assetsRegistry.addAll(products).then(() => {console.log('All assets added')});
await walletRegistry.addAll(wallets).then(() => {console.log('All wallets added')});
```

In het script om het netwerk met gegevens te vullen, wordt er van elk gegeven het bijhorende object gemaakt, dit wordt gepushed in een overkoepelende array en dan in één keer naar het netwerk verstuurd.

```
async removeAllParticipants() {
    try {
        let participantRegistry = await this.SupChainConnection.getParticipantRegistry('org.obliviate.supchain.CompanyParticipant');
        let assetsRegistry = await this.SupChainConnection.getAssetRegistry('org.obliviate.supchain.Product');
        let walletRegistry = await this.SupChainConnection.getAssetRegistry('org.obliviate.supchain.Wallet');
        await participantRegistry.getAll().then((res) => {participantRegistry.removeAll(res)});
        await assetsRegistry.getAll().then((res) => {assetsRegistry.removeAll(res)});
        await walletRegistry.getAll().then((res) => {assetsRegistry.removeAll(res).then(() => {console.log('All info removed')})});
    }catch(error) {
        console.log(error);
        throw error;
    }
}
```

In het script om alle informatie te verwijderen, wordt eerst alle informatie opgehaald met de registers en daarna allemaal verwijderd.

```
listen() {
    console.log('Started to listen to events');
    this.SupChainConnection.on('event', (evt) => {
        var output = "";
        if(evt.$type == 'TransactionFailedEvent'){
            output = `The transaction "${evt.transactionFunction}" failed. ${evt.message}.\n`;
            console.log(output);
        }else if(evt.$type == 'CreateMoreSupplySuccesEvent'){
            output = `The quantity of the product ${evt.productName} for company ${evt.companyName} has been raised from ${evt.oldQuantity} to ${evt.newQuant
            console.log(output);
        }else if(evt.$type == 'TradeProductSuccesEvent'){
            output = `A trade happened between the companies: ${evt.oldOwnerName} and ${evt.newOwnerName}.\nThe trade consisted of ${evt.quantity} units of $
            console.log(output);
        }
        else if(evt.$type == 'CreateNewProductSuccesEvent'){
            output = `The company ${evt.companyName} turned ${evt.quantityOld} ${evt.productNameOld} into ${evt.quantityNew} new ${evt.productNameNew}.\n`
            console.log(output);
        }
    });
}
```

In het script dat luistert naar de verschillende events, wordt er elke keer dat er een event ontvangen wordt naar de naam gekeken en gebaseerd daarop wordt de juiste melding terug gegeven.

```
wouter@ubuntu:~/HLC/NodeJS applications/NodeJS HLC Portal$ node main.js
Started to listen to events
The quantity of the product SEEDS for company Zhengyuan has been raised from 300000 to 303000.

A trade happened between the companies: Bata Food and Lovekitchen.
The trade consisted of 300 units of TOMATOES at a price of 0.52 euro per unit.

The company Heinz turned 15 TOMATOES into 5 new SPAGHETTI SAUCE.
```

React Native

Als voorbeeld werd er een mobiele applicatie geschreven.Later in de demonstratie wordt hier grondiger op in gegaan. In deze applicatie heeft de gebruiker de keuze tussen de soorten bedrijven en daarna een specifiek bedrijf binnen die branche. De gebruiker krijgt de keuze tussen 3 acties dat een bedrijf kan verrichten, van zodra de correcte gegevens worden ingevuld, krijgt de gebruiker een overzicht te zien van alle specifieke stappen dat ondergaan werden tijdens de transactie. Hieronder volgen nog een paar voorbeelden van gebruikte code:

```
34      getDistinctTypes = () => {
35          var types = [];
36
37          for(let i = 0; i < this.state.participants.length; i++){
38              if(types.length == 0){
39                  types.push(this.state.participants[i].companyType);
40              }else{
41                  if(types.indexOf(this.state.participants[i].companyType) == -1){
42                      types.push(this.state.participants[i].companyType);
43                  }
44              }
45          }
46
47          this.setState({
48              types: types,
49          })
50      }
51
```

In de bovenstaande foto wordt er naar elk type gekeken van elk bedrijf. Alleen als dat type nog niet in de array staat, wordt deze toegevoegd. Dit is belangrijk om dynamisch alle soorten bedrijven op te halen.

```
case "Create more supply":
    let fun = "org.obliviate.supchain.CreateMoreSupply";
    let product = "resource:org.obliviate.supchain.Product#" + this.state.selectedProductID;
    let quantity = this.state.amount;
    api.createMoreSupply(fun, product, quantity).then((res) => {
        if(res.transactionId){
            this.setState({sending: false, amount: 0, selectedProductID: null, selectedParticipantID: null});
            this.props.navigation.navigate('Events', {transactionID: res.transactionId, from: this.state.action});
        }else{
            //problems
            Alert.alert(res.error.message);
        }
    });

    break;
```

Dit is een voorbeeld van een transactie verzonden wordt vanuit de front-end. Het is nodig om de juiste transactienaam mee te geven, samen met de bijhorende variabelen zoals het product en het aantal.

```
getInfo1 = (res) => {
    //CreateMoreSupply

    //product
    //participant
    let product;
    let participant;
    let productID = res.product.split("#")[1];
    api.getProductByID(productID).then((res) => {
        product = res;
        let participantID = res.owner.split("#")[1];
        api.getParticipantByID(participantID).then((res2) => {
            participant = res2;

            this.setState({
                myProduct: product,
                myParticipant: participant,
                isLoading : false,
            });
        });
    });
```

Elke transactie op het netwerk heeft zijn eigen transactieID. Dit kregen we terug in de vorige foto. Met dit transactionID kunnen we de volledige transactie terug ophalen en daar de gegevens ook terug uithalen.

```
urlNgRok : 'http://170a9b69.ngrok.io',

getAllParticipants() {
    var url = `${this.urlNgRok}/api/CompanyParticipant`;
    return fetch(url).then((res) => res.json());
},

getAllProducts() {
    var url = `${this.urlNgRok}/api/Product`;
    return fetch(url).then((res) => res.json());
},

createMoreSupply(_fun, _product, _amount) {
    return fetch(`${this.urlNgRok}/api/CreateMoreSupply`, {
        method: 'POST',
        headers: {
            'Accept': 'application/json',
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({$class : _fun, product : _product, quantity : _amount})
    }).then((res) => res.json());
},
```

Om dit alles te kunnen doen, moet er natuurlijk een api worden aangesproken. Omdat de API normaal op een localhost draait, gebruiken we NgRok om dit open te stellen op het internet. Daarna kan je de correcte link aanroepen en ofwel een get, post, delete, put verzoek sturen naar de server.

8.4 Demonstratie

Demonstratie 1

Zie Bijlage A waar de screenshots staan van de applicatie. Deze screenshots zijn per demonstratie in de juiste volgorde gezet.

Bij het openen van de applicatie wordt de keuze gegeven tussen de verschillende soorten bedrijven, in het eerste voorbeeld wordt de "SUPPLIER"genomen. Daarna komt er een nieuw scherm waar de keuze gemaakt wordt tussen de verschillende bedrijven.

Elk soort bedrijf heeft zijn eigen acties dat ze kunnen uitvoeren op het netwerk. Een supplier kan nu alleen maar zijn eigen voorraad vergroten. Bij het uitvoeren van de actie wordt er gevraagd naar het bepaalde product en hoeveelheid. De getoonde producten in de lijst zijn alleen maar producten dat dat ene bedrijf kan maken.

In de laatste stap van de applicatie wordt er stap voor stap getoond wat er precies allemaal gebeurt tijdens de transactie. Er wordt informatie getoond zoals de naam van het bedrijf en de voorraad van het bepaalde product.

Demonstratie 2

De uitleg bij demo 2 en 3 zijn grotendeels hetzelfde. De flow van de applicatie is makkelijk af te leiden ui de screenshots.

Zie Bijlage A demo 2.

Demonstratie 3

Zie Bijlage A demo 3.

8.5 Mogelijke uitbreidingen

Natuurlijk is dit nog maar een minimaal prototype, er zijn zeker nog veel mogelijke uitbreidingen.

Een van de belangrijkste uitbreidingen dat hier nog gemaakt kunnen worden is het implementeren van een zoekfunctie dat voor de gebruiker een perfecte route gaat zoeken. Er zou dan gefilterd kunnen worden op snelste route, goedkoopste route, de route met de beste kwaliteit of combinatie van meerdere factoren. Omdat de prijzen van de bedrijven openlijk verkrijgbaar zijn op het netwerk, is deze optie zeker doenbaar.

Een logische uitbreiding is het toevoegen van meerdere producten in de supply chain. Zoals voor de bedrijven dat lasagne of soep maken, zullen er meer producten nodig zijn dan alleen maar tomaten. Dit geldt natuurlijk ook voor meerdere bedrijven.

In dit prototype is niet verder ingegaan op de access file, dat kan ook een verdere uitbreiding worden. Dankzij de access file wordt er beslist wie precies wat mag doen op het netwerk.

8.6 Conclusie Hyperledger composer

De vraag 'Hoe kan er met zekerheid gezegd worden dat de producten authentiek zijn?' kan nu al beantwoord worden met de met de informatie dat er tot nu toe al verzameld is. Op de manier dat informatie wordt opgeslagen op een blockchain kan er gezegd worden dat deze informatie niet aangepast kan worden, de data kan niet vervalst worden. Als een persoon deze informatie wil veranderen, zal de hash waarde niet meer overeenkomen met de historiek van andere nodes. Deze persoon zou daarom alle hash waarden moeten gaan aanpassen, en dat is nu onmogelijk. Op het netwerken heeft een bedrijf ook bepaalde rechten, dus een bedrijf kan niet zo maar de voorraad van een ander bedrijf gaan aanpassen. Dankzij deze historiek dat bijgehouden wordt kan er perfect aangetoond worden van waar een product precies afkomstig is en kan deze informatie dus ook als betrouwbaar aanzien worden.

9. Uitvoeren testen

9.1 Schetsen van de situatie

Voor een bedrijf is de snelheid van het netwerk ook zeker een belangrijk aspect. Een bedrijf kan niet zitten wachten op transacties dat in queue staan om uitgevoerd te worden. Daarom worden er ook een paar snelheidstesten losgelaten op het netwerk. Dit netwerk werd opgesteld op een virtuele machine, de testen duren daarom dus ook langer dan normaal. Op volgende transacties wordt de snelheid getest:

1. Creatie van informatie
2. Verwijderen van informatie
3. Ophalen van alle producten / bedrijven
4. Opzoeken van één bepaalde ID

9.2 Uitgevoerde testen

Elke test wordt 30 keer uitgevoerd...
Creatie van informatie

Er worden twee verschillende testen uitgevoerd voor de creatie van de informatie. De gegevens kunnen in één lijst verstuurd worden om aan te maken op het netwerk of dit kan gegeven per gegeven verstuurd worden. Hier wordt getest of er een verschil bestaat tussen deze twee manieren. Er werd getest met 5, 50 en 100 producten. Er worden geen testen

uitgevoerd met meer producten omdat er vanaf dan timeout errors worden gegeven.

```
async createProductsOnce(i) {
    let begin = new Date().getTime();
    let assetsRegistry = await this.SupChainConnection.getAssetRegistry('org.obliviate.supchain.Product');
    let factory = this.businessNetworkDefinition.getFactory();

    var products = []

    for(let k = 0; k < i; k++){
        let product = factory.newResource('org.obliviate.supchain', 'Product', (i + new Date().getTime()).toString());

        product.productName = "SEEDS";
        product.quantity = 10;
        product.price = 1;

        product.owner = factory.newRelationship('org.obliviate.supchain', 'CompanyParticipant', 111);

        products.push(product);
    }

    await assetsRegistry.addAll(products).then(() => {console.log((new Date().getTime() - begin) / 1000)});
}
```

```
async createProductsOnePerOne(i) {

    let assetsRegistry = await this.SupChainConnection.getAssetRegistry('org.obliviate.supchain.Product');
    let factory = this.businessNetworkDefinition.getFactory();

    for(let k = 0; k < i; k++){
        let product = factory.newResource('org.obliviate.supchain', 'Product', (i + new Date().getTime()).toString());

        product.productName = "SEEDS";
        product.quantity = 10;
        product.price = 1;

        product.owner = factory.newRelationship('org.obliviate.supchain', 'CompanyParticipant', 111);

        await assetsRegistry.add(product);
    }
}
```

Verwijderen van informatie

De volgende transactie dat getest zal worden is het verwijderen van alle informatie. Dit kan ook gebeuren één per één of alles in één keer. Hierbij wordt er ook getest met 5, 50 en 100 producten.

```
async deleteProductsOnce() {
    let begin = new Date().getTime();
    let assetsRegistry = await this.SupChainConnection.getAssetRegistry('org.obliviate.supchain.Product');
    await assetsRegistry.getAll().then((res) => {assetsRegistry.removeAll(res).then(() => {console.log((new Date().getTime() - begin) / 1000)})});
}
```

```
async deleteProductsOnePerOne() {
    let assetsRegistry = await this.SupChainConnection.getAssetRegistry('org.obliviate.supchain.Product');
    let products = await assetsRegistry.getAll();
    for(let j = 0; j < products.length; j++){
        assetsRegistry.remove(products[j]);
    }
}
```

Ophalen van alle producten / bedrijven

Bij deze test is het de bedoeling om te kijken of er een groot verschil bestaat tussen het opvragen van alle informatie. Hier wordt er nogmaals gebruik gemaakt van 5, 50 en 100

producten.

```
console.log("getting all 50 products");
await this.createProducts(50);
for(let j = 0; j < 30; j++){
    let begin = new Date().getTime();
    let products = await assetsRegistry.getAll().then(() => {console.log((new Date().getTime() - begin) / 1000)});
}
```

Opzoeken van één bepaalde ID

Als laatste wordt er onderzocht of er een verschil bestaat tussen het ophalen van één bepaald product in een lijst van meerdere producten. Nogmaals worden er hier 5, 50 en 100 producten gebruikt.

```
console.log("getting id 2 out of 50 products");
await this.createProducts(50);
for(let j = 0; j < 30; j++){
    let begin = new Date().getTime();
    let products = await assetsRegistry.get("2").then(() => {console.log((new Date().getTime() - begin) / 1000)});
}

console.log("getting id 45 out of 50 products");
await this.createProducts(50);
for(let j = 0; j < 30; j++){
    let begin = new Date().getTime();
    let products = await assetsRegistry.get("45").then(() => {console.log((new Date().getTime() - begin) / 1000)});
}
```

9.3 resultaten

Alle resultaten zijn uitgedrukt in seconden...

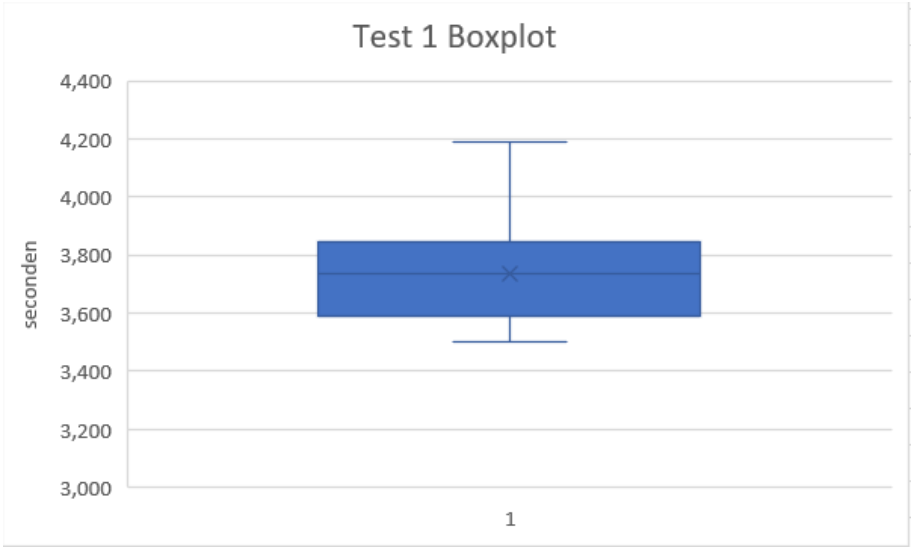
1. Creatie van informatie

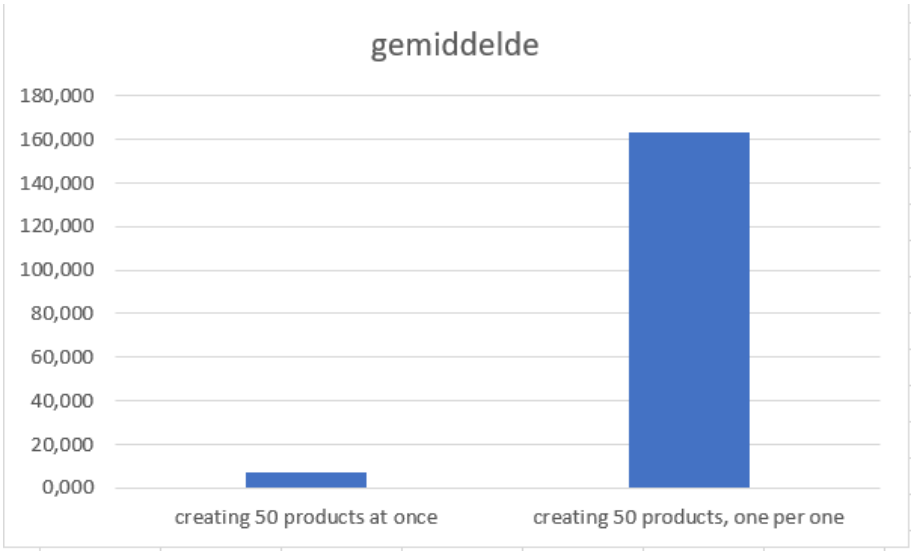
   Alle producten in 1 keer

   Voor de ruwe resultaten, zie Bijlage B.

   Alle producten apart

   Voor de ruwe resultaten, zie Bijlage B.

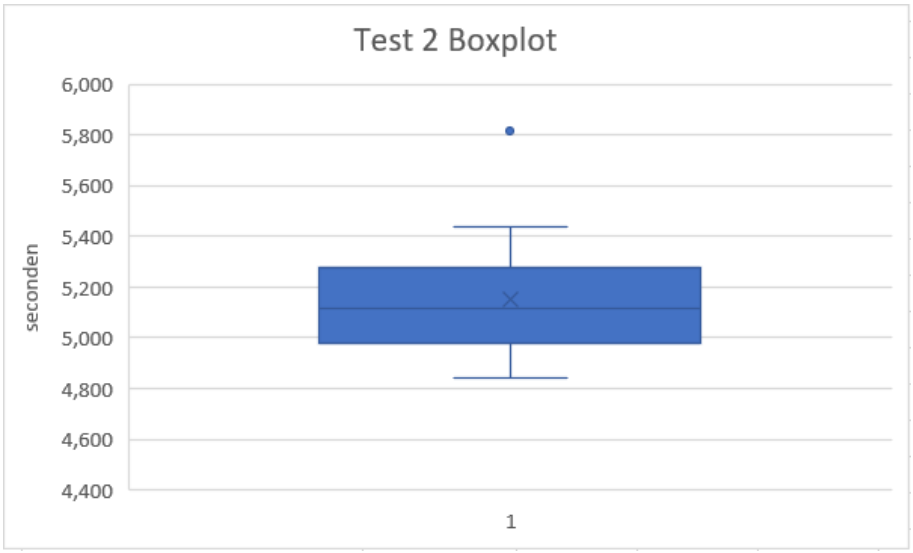


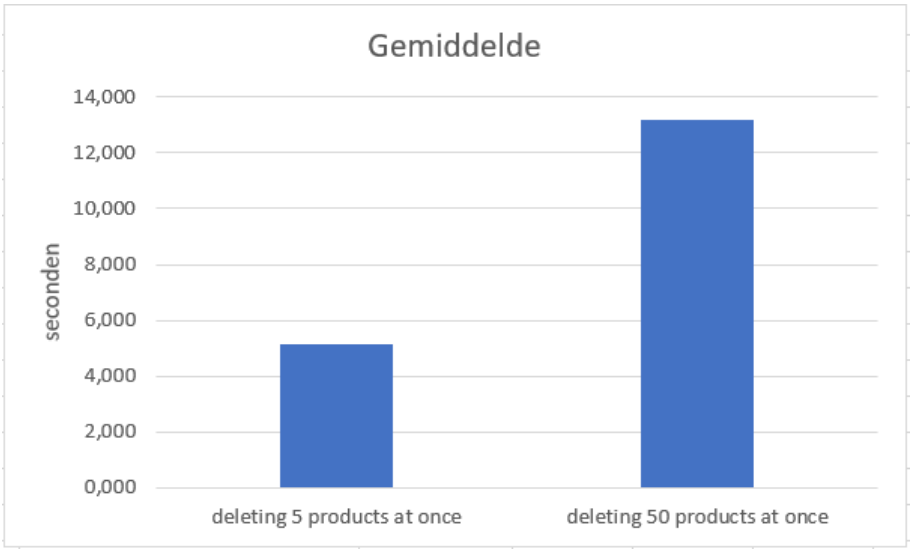


Als er gekeken wordt naar het gemiddelde van 50 producten in één keer toevoegen of 50 producten één voor één toevoegen, kan er geconcludeerd worden dat er toch een redelijk groot verschil bestaat tussen deze twee manieren. Er wordt elke keer een extra call gedaan naar het netwerk en daar wordt er tijd verloren. Voor het apart toevoegen van 50 prodcuten kan het wel tot bijna 3 minuten duren.

2. Verwijderen van informatie

   Voor de ruwe resultaten, zie Bijlage B.

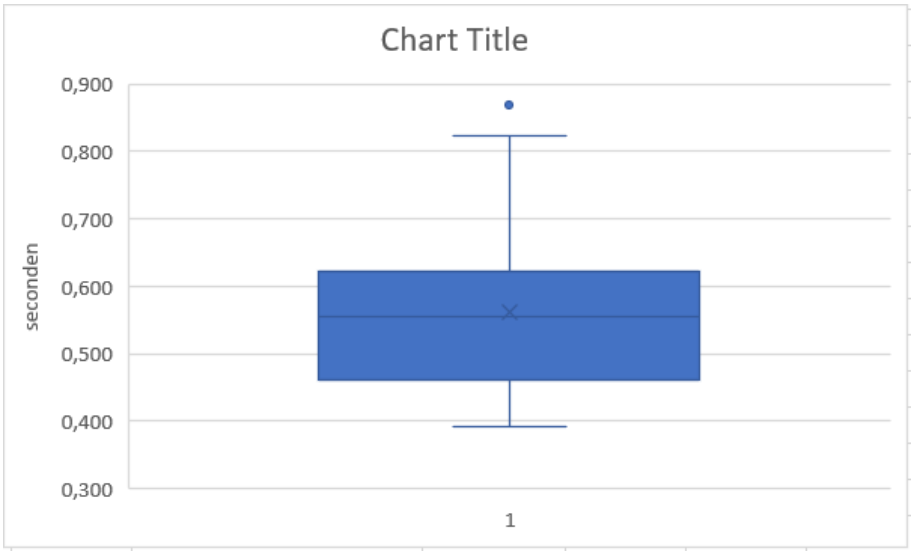


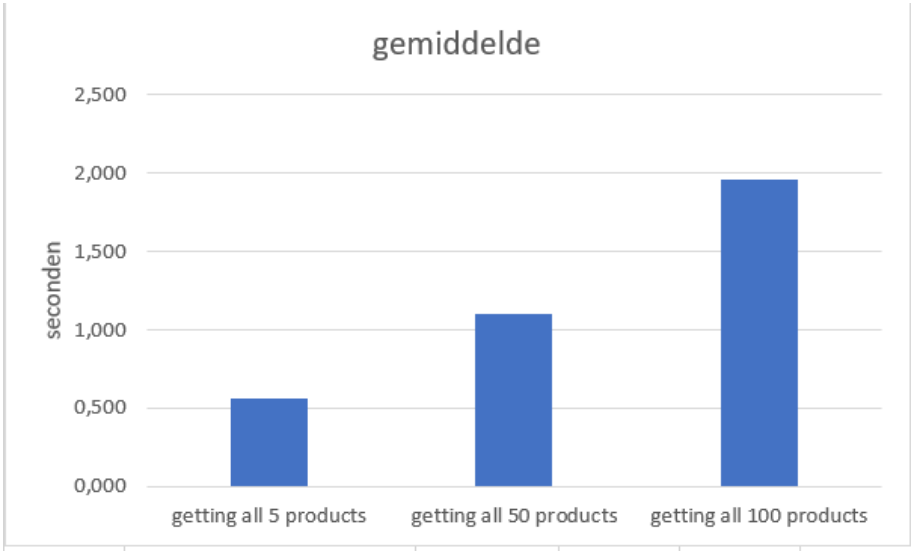


Bij het verwijderen van gegevens konden de testen van het product een voor een te verwijderen niet uitgevoerd worden. Omdat deze veel te lang duurde gaf het netwerk een time out. De tijd bij deze testen is langer dan verwacht, het verwijderen duurt langer dan het aanmaken. Opnieuw liggen de resultaten dicht bij elkaar.

3. Ophalen van alle producten / bedrijven

   Voor de ruwe resultaten, zie Bijlage B.

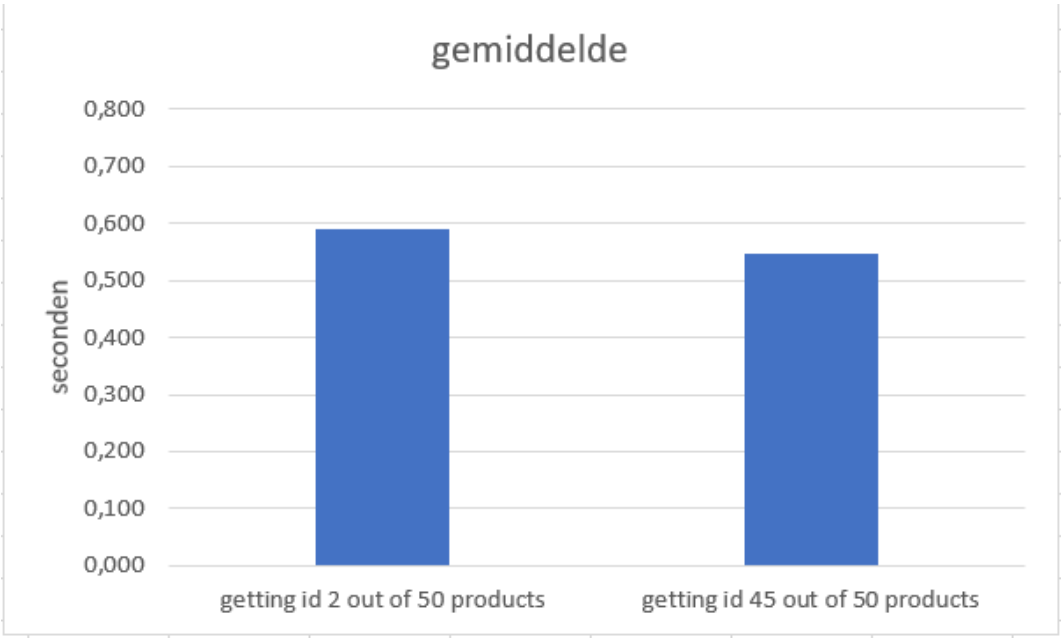


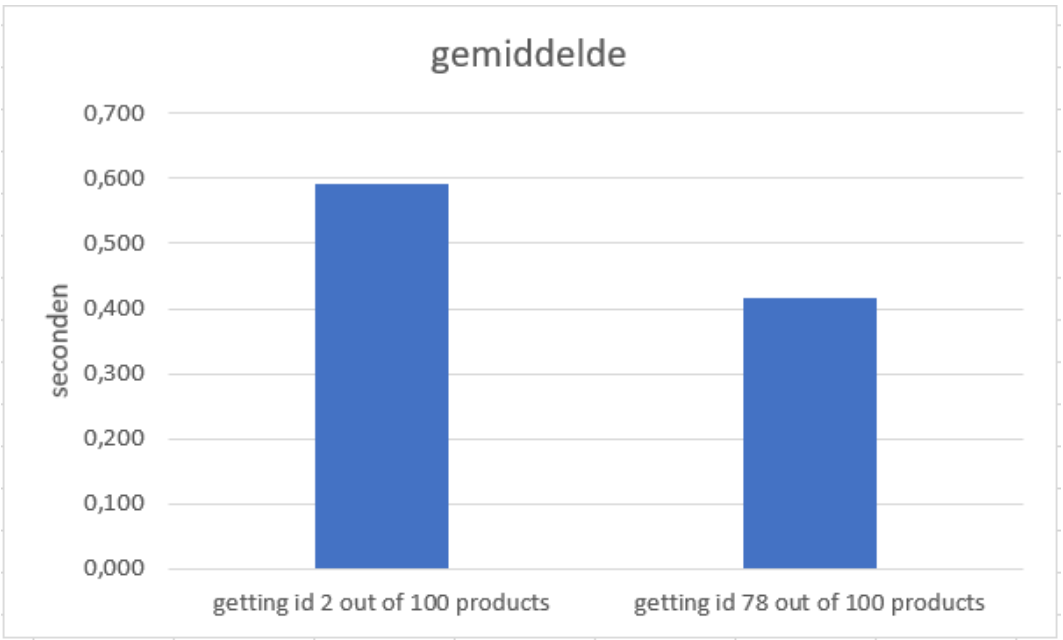


In deze liggen de resultaten een beetje meer verspreid van elkaar als er gekeken wordt naar de boxplot. Het ophalen van alle gegevens gaat nog relatief gezien snel in vergelijking met het toevoegen en verwijderen.

4. Opzoeken van één bepaalde ID

   Voor de ruwe resultaten, zie Bijlage B.





Wat heel merkwaardig is bij deze testen is dat het blijkt dat de producten met de hogere id's minder lang duren dan de producten met een klein id. Dit kan zijn omdat de lijst omgekeerd onderzocht wordt, maar hier werd niet meer informatie over gevonden.

9.4 conslusie resultaten

Spijtig genoeg konden sommige testen niet worden uitgevoerd omdat het netwerk te traag was op de virtuele machine. De conclusie uit deze testen is dat het altijd beter is om een hele lijst mee te geven aan een functie dan elk object apart te versturen.De resultaten zijn redelijk consistent in de uitvoeringstijd als er gekeken wordt naar de boxplotten. Soms waren er verassende resultaten zoals het verwijderen dat langen duurt dan het toevoegen en het ophalen van een hoger id dat minder lang duurde dan het ophalen van een lager id.

10. Toekomstig werk

Zoals eerder al aangehaald in deze bachelorproef, blijft technologie maar evolueren. Het is belangijk dat er mee wordt gegaan met de tijd. Blockchain technologie staat nog in zijn peuterschoentjes en er is nog veel dat onderzocht moet worden in verband met decentralisatie, beveiliging en uitbreidbaarheid. Het hele gebeuren rond supply chain is ook heel complex, er zal diep onderzoek gedaan moet worden hoe deze twee technologieën perfect in harmonie kunnen samenwerken. Bijvoorbeeld kan er een meer dynamisch netwerk opgesteld worden waardat de gebruikers zich werkelijk inloggen als een bedrijf en zelf ook producten kunnen toevoegen en bij andere bedrijven verkopen en kopen.

Wat nog verder getest kan worden is de transparantie van het netwerk. Wat mag een bedrijf bekijken op het netwerk en wat niet. Kiest een bedrijf er zelf voor of het zichtbaar is naar de buitenwereld enzovoort.

Deze bachelorproef creëert ook nog een paar andere vragen wat verder onderzocht moet worden:

1. Voor welke bedrijven is het nuttig om met blockchain technologie te integreren?
2. Wat is nuttiger voor een supply chain netwerk? Publiek of privaat?
3. Welke blockchain technologie kan het beste gebruikt worden voor een supply chain?

11. Conclusie

De onderzoeksvragen worden snel nog eens hieronder gezet:

1. **Hoe kan er met zekerheid gezegd worden dat de producten authentiek zijn?**
2. **Heeft het aantal gegevens op het netwerk een invloed op het ophalen en wegsturen van informatie?**
3. **Welke problemen kunnen niet opgelost worden met blockchain?**

In de conclusie worden de onderzoeksvragen nog eens overlopen en wordt er hier een antwoord op gegeven.

Dankzij de blockchain technologie kunnen we met zekerheid zeggen dat de authentiteit van producten behouden wordt. Eens de informatie op de blockchain staat, kan die transactie niet meer aangepast worden. Op het netwerk heeft alleen de eigenaar rechten tot zijn producten en kiest dus zelf wat er met gebeurt. Hierdoor kunnen andere bedrijven geen producten gaan aanpassen van andere bedrijven. Dankzij de historiek van de blockchain kan er gezien worden wat er exact met elk product gebeurt, en zo kunnen de eindgebruikers dus ook met zekerheid zien waar hun prouct precies vandaag komt.

Het aantal gegevens heeft dus duidelijk wel een invloed op de snelheid van het ophalen en wegschrijven van informatie. Uit de resultaten kan er afgeleid worden dat het effeciënter wordt als er een grotere lijst wordt ingevoerd. Spijtig genoeg zijn sommige testen niet kunnen doorgaan door het te kort komen van de virtuele machine waarop het netwerk

draaide.

Ten slotte werd er gekeken naar wat een blockchain (nog) niet kan oplossen als blockchain gecombineerd wordt met supply chain. §§§Next up§§§

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage. Door veranderingen na het voorstel is dit voostel niet helemaal up to date. Dit voorstel moest dus ook niet aangepast worden voor deze bachelorproef.

Veiligheid van data op de blockchain technologie en het gebruik van smart contracts

Onderzoeksvoorstel Bachelorproef

Wouter Van Hecke[1]

Samenvatting
Dit onderzoek gaat over de blockchain technologie en een concept dat de blockchain technologie met zich meebrengt: ”smart contracts”. Dit is een concept dat onze manier van omgaan met huidige contracten helemaal kan veranderen. Er zit veel potentieel in (de) blockchain technologie/smart contracts en is daarom ook belangrijk dat dit verder onderzocht wordt, zowel voor jonge programmeurs die zich verder willen verdiepen in smart contracts en voor geavanceerde programmeurs die de huidige problemen verder willen behandelen. In deze bachelorproef wordt onderzocht hoe de blockchain technologie zijn data beveiligd en hoe deze technologie kan toegepast worden in supply chain en voting systems. Het doel van deze bachelorproef is aan te tonen waarom een blockchain heel veilig is en hoe het in de praktijk gebruikt kan worden.

Sleutelwoorden
Onderzoeksdomein. Blockchain — Smart Contracts — Ethereum

Contact: [1] wouter.vanhecke.w9895@student.hogent.be

Inhoudsopgave

1 Introductie **1**
2 State-of-the-art **2**
2.1 A gentle introduction 2
2.2 Impossible use cases 2
2.3 Smart contracts in de pharmacy wereld 2
3 Methodologie **2**
4 Verwachte resultaten **2**
5 Verwachte conclusies **2**
Referenties **2**

1. Introductie

Technologie blijft steeds evolueren. De blockchain technologie is daar een belangrijk voorbeeld van. Deze nieuwe technologie begint meer en meer bekend te worden, kijk maar naar de stijging in waarde van bitcoin het afgelopen jaar. Maar de blockchain technologie is zoveel meer dan alleen maar bitcoin en andere cryptocurrencies. Omdat een blockchain een gedecentraliseerd systeem is, is het van hoog belang om ervoor te zorgen dat data zo beveiligd mogelijk is om de integriteit van het systeem te behouden. Een ander concept dat blockchain met zich mee brengt zijn de ”smart contracts”. Een smart contract is een zelf uitvoerbaar script dat opgeslagen wordt op de blockchain. Hiermee kan je bijvoorbeeld programmeren dat persoon A op een bepaald tijdstip persoon B zal betalen, dit zal op dat exacte tijdstip dan automatisch uitgevoerd worden dankzij het smart contract. Omdat dit nog een relatief nieuwe technologie is, bestaat er nog veel verwarring rond dit concept. Daarom is het belangrijk dat deze technologie verder onderzocht wordt. Het doel van deze bachelorproef is aan te tonen waarom een blockchain heel veilig is en hoe het in de praktijk gebruikt kan worden. De hoofdvraag van deze bachelorproef wordt dus:

Hoe wordt data op de blockchain beveiligd?

Omdat de blockchain technologie een gedecentraliseerde technologie is, kan iedereen het netwerk binnentreden, ongeacht de intentie van de persoon. Daarom is het heel belangrijk dat alle data op de blockchain goed beveiligd wordt. Bij deze vraag wordt er onderzocht hoe de blockchain zijn data precies beveiligd en alleen maar de correcte informatie vrijgeeft aan de huidige eigenaar.

Andere onderzoeksvragen bestaan uit:

Hoe wordt de blockchain technologie gebruikt voor supply chain? De blockchain technologie vindt bij meer en meer industrieën een thuis. Bij deze vraagt wordt er onderzocht hoe de blockchain technologie een thuis heeft gekregen bij supply chain.

Hoe kan de blockchain technologie gebruikt worden als een voting system? Omdat elke gebruiker op de blockchain anoniem is, maar er tegelijkertijd ook geen kennis is over de betrouwbaarheid van de gebruiker, wordt er onderzocht hoe de blockchain toch gebruikt kan worden als een voting system en waarom dit toch een velige keuze kan zijn.

NEO en Ethereum zijn allebei platformen dat smart contracts ondersteunen, maar hoe verschillen ze van elkaar?

Stel nu dat je een applicatie wil schrijven dat gebruik maakt van smart contracts en nu moet je kiezen op welke blockchain technologie je deze applicatie wil maken. In dit deel wordt er onderzocht wat de verschillen zijn tussen de blockchain platformen 'NEO' en 'Ethereum'.

2. State-of-the-art

Er zijn al meerdere artikels en thesissen geschreven over smart contracts. Artikels hebben het vaak over wat een smart conctract nu eigenlijk is, terwijl andere bachelorproeven of thesissen al snel meer gedetailleerd ingaan op een specifieke toepassing met smart contracts.

2.1 A gentle introduction

Het eerste artikel (antonylewis2015, 2016) geeft een kort, maar krachtige introductie over smart contracts. Het geeft een paar basis definities, samen met een paar praktische voorbeelden in verband met banken. Het geeft uitleg over wat een smart contract doet en hoe je het kan gebruiken. Ook vertelt de auteur over zijn visie van de toekomst voor smart contracts.

2.2 Impossible use cases

Het tweede artikel, geschreven door Greenspan (2016), vertelt meer over wat niet mogelijk is met smart contracts op dit moment. Het haalt drie use cases naar boven waar mensen veel over spreken en die op dit moment nog niet mogelijk zijn. Hij geeft een duidelijk, praktisch voorbeeld en bespreekt hoe je dit probleem kan omzeilen.

2.3 Smart contracts in de pharmacy wereld

In deze thesis, geschreven door Bergquist (2017), wordt het probleem met apothekers aangehaald. De auteur spreekt over hoe patiënten soms verkeerde medicatie voorgeschreven krijgen en daardoor nog zieker kunnen worden. Dit ligt in de handen van de dokter die de examinatie heeft afgelegd en hoe al het vertrouwen bij de dokter ligt. De dokter schrijft de medicatie op een prescriptie en zet hier een stempel op als validatie. De auteur heeft geprobeerd om dit om te zetten naar een smart contract dat werk met privacy en validatie.

3. Methodologie

De vragen zullen beantwoord worden aan de hand van literatuurstudies. Er zal onderzocht worden hoe de blockchain technologie zijn data beveiligd en hoe smart contract aspecten kunnen helpen zoals supply chain, voting, kostefficiëntie...

4. Verwachte resultaten

Hoe wordt data op de blockchain beveiligd? Voor deze onderzoeksvraag wordt verwacht dat de data op een blockchain zeer goed beveiligd wordt dankzij encryptie. Het vervalsen van gegevens zal onmogelijk zijn voor mensen met valse intenties omdat de blockchain technologie werkt met een gedecentraliseerd systeem waar je alleen maar gegevens aan kan toevoegen en niet verwijderen/aanpassen.

Hoe wordt de blockchain technologie gebruikt voor supply chain? Voor deze onderzoeksvraag wordt verwacht dat producten succesvol op de blockchain kunnen worden opgeslaan en zorgt voor authenticiteit van producten.

Hoe kan de blockchain technologie gebruikt worden als een voting system? Omdat de blockchain technologie zorgt voor een systeem waar annonimiteit en vertrouwloos vertrouwen topprioriteiten zijn, kan een blockchain technologie als een voting system gebruikt worden. Omdat een blockchain werkt met hash values en een gelinkte list waar constant de hash value van de vorige block wordt opgeslagen, kan vervalsing niet voorkomen, omdat de hash values dan niet meer overeenkomen.

NEO en Ethereum zijn allebei platformen dat smart contracts ondersteunen, maar hoe verschillen ze van elkaar? Om deze vraag te beantwoorden zullen de twee platformen naast elkaar gelegd worden en vergeleken worden op aspecten zoals kosten, transactievermogen, snelheid, gebruikte programmeertaal, toegankelijkheid.

5. Verwachte conclusies

Uit de onderzoeken zal blijken dat er nog veel groeipotentieel is voor blockchain technologieën. Smart contracts zullen in de toekomst als standaard gebruikt worden voor contracten omdat er veel minder logische fouten gemaakt kunnen worden tijdens het uitvoeren van het contract. Blockchain technologie kan opgenomen worden binnen zowel kleine als grote bedrijven. Door het gebruik van blockchain te standaardiseren zal er veel geld bespaard worden op vlak van onkosten / het betalen van derde partijen, omdat hun werk nu geautomatiseerd kan worden.

Referenties

antonylewis2015. (2016). A gentle introduction to smart contracts. Verkregen van https://bitsonblocks.net/2016/02/01/a-gentle-introduction-to-smart-contracts/

Bergquist, J. H. (2017). *Blockchain Technology and Smart Contracts* (masterscriptie, Uppsala University).

BlockGeeks. (2016). Smart Contracts: The Blockchain Technology That Will Replace Lawyers. Verkregen van https://blockgeeks.com/guides/smart-contracts/

Bokhorst, J. (2016). Smart contracts binnen de blockchain: contracteren 2.0? Verkregen van https://www.ordina.nl/nl-nl/blogs/2016/oktober/smart-contracts-binnen-de-blockchain-contracteren-20/

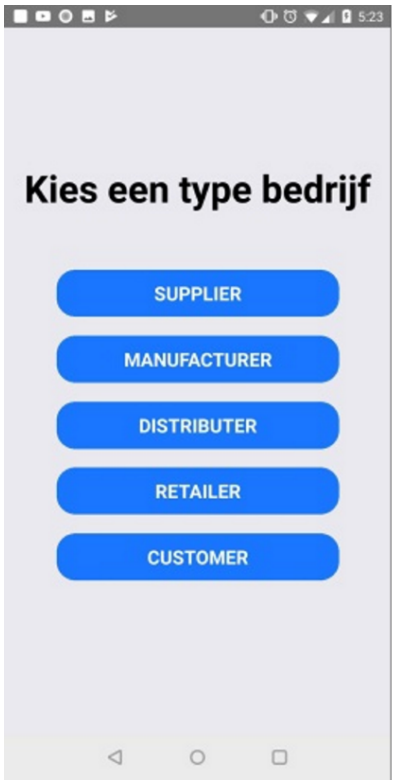
Greenspan, G. (2016). Why Many Smart Contract Use Cases Are Simply Impossible. Verkregen van https://www.coindesk.com/three-smart-contract-misconceptions/

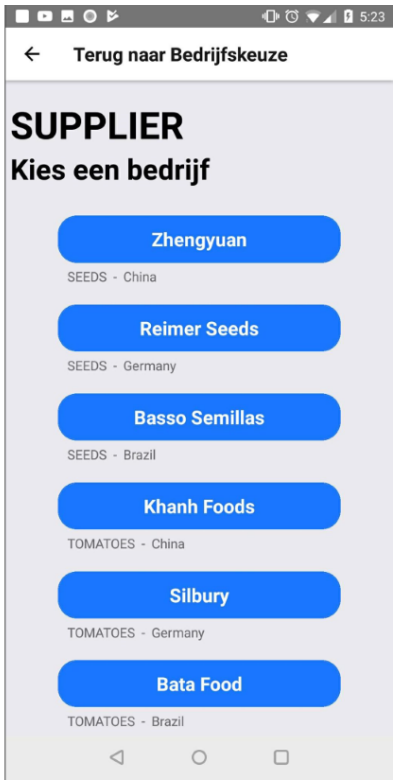
Hertig, A. (Unknown). How Do Ethereum Smart Contracts Work? Verkregen van https://www.coindesk.com/information/ethereum-smart-contracts-work/

B. Bijlage

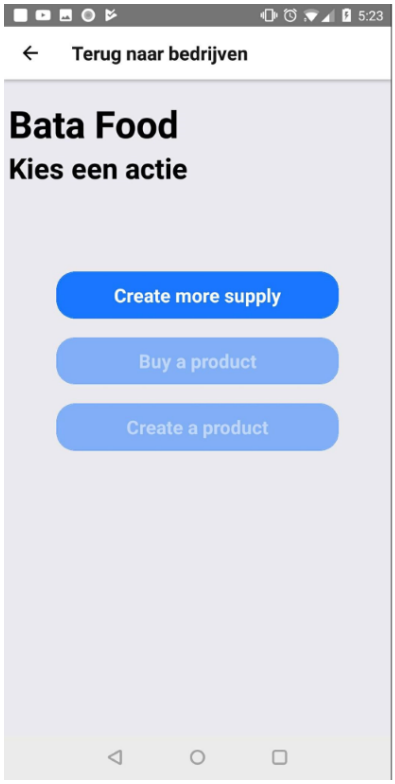
B.1 Demonstratie applicatie

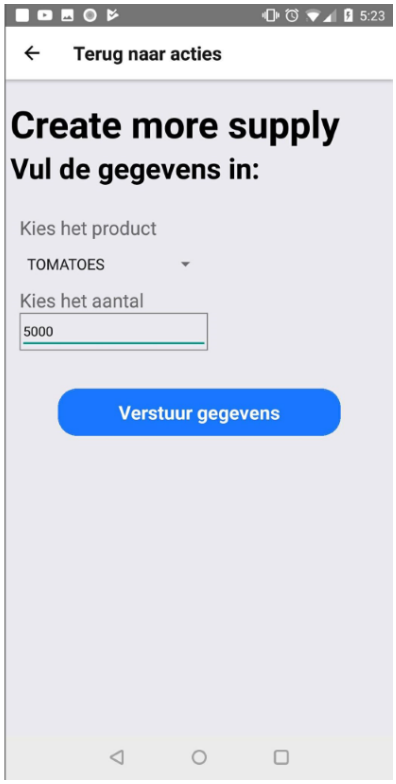
Demonstratie 1



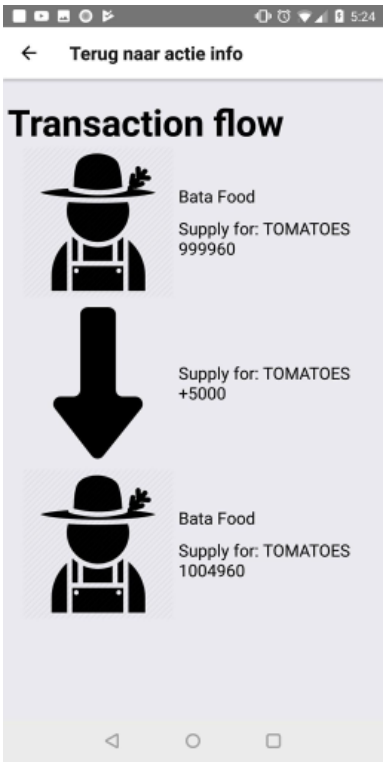


...

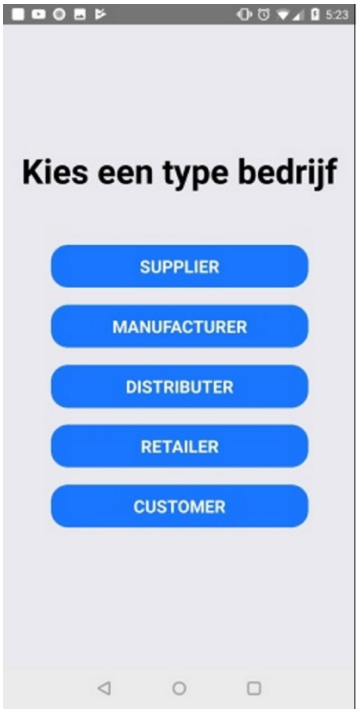

5:23
Terug naar bedrijven
Bata Food
Kies een actie
Create more supply
Buy a product
Create a product

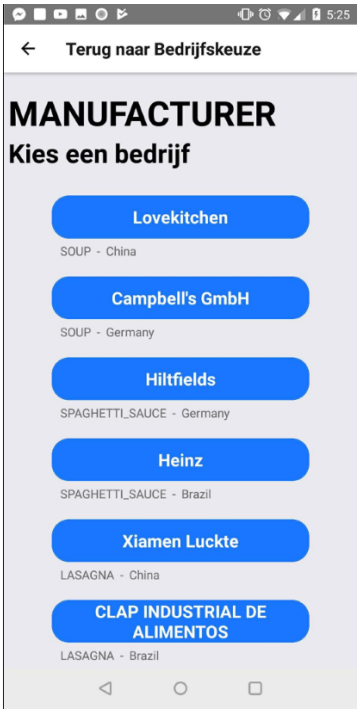

5:23
Terug naar acties
Create more supply
Vul de gegevens in:
Kies het product
TOMATOES
Kies het aantal
5000
Verstuur gegevens

...

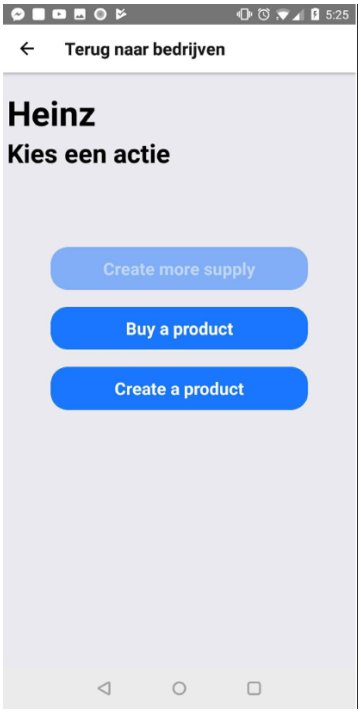

5:24
Terug naar actie info
Transaction flow
Bata Food
Supply for: TOMATOES
999960
Supply for: TOMATOES
+5000
Bata Food
Supply for: TOMATOES
1004960

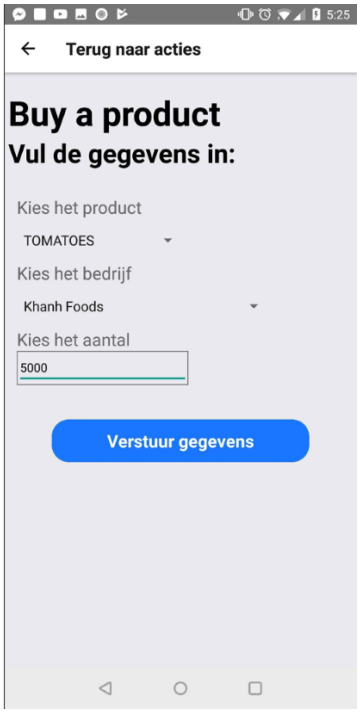
Demonstratie 2



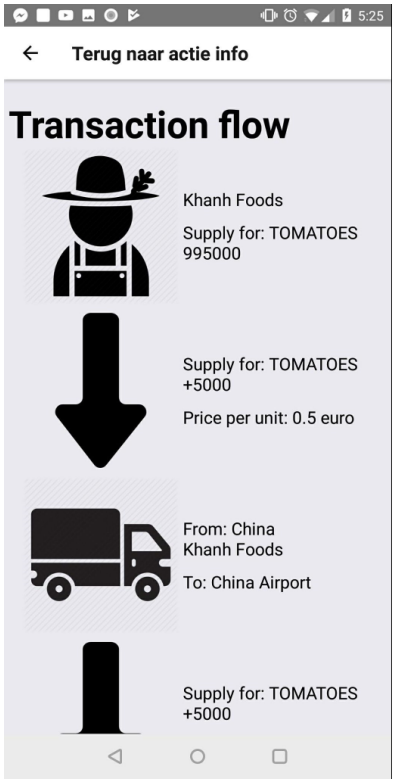


...





...

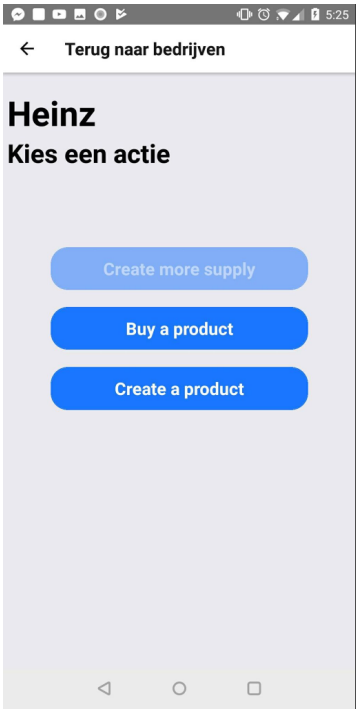

5:25
Terug naar actie info
Transaction flow
Khanh Foods
Supply for: TOMATOES
995000
Supply for: TOMATOES
+5000
Price per unit: 0.5 euro
From: China
Khanh Foods
To: China Airport
Supply for: TOMATOES
+5000


5:26
Terug naar actie info
Transaction flow
Price per unit: 0.5 euro
From: China
To: Brazil
Supply for: TOMATOES
+5000
Price per unit: 0.5 euro
From: Brazil Airport
To: Brazil
Heinz

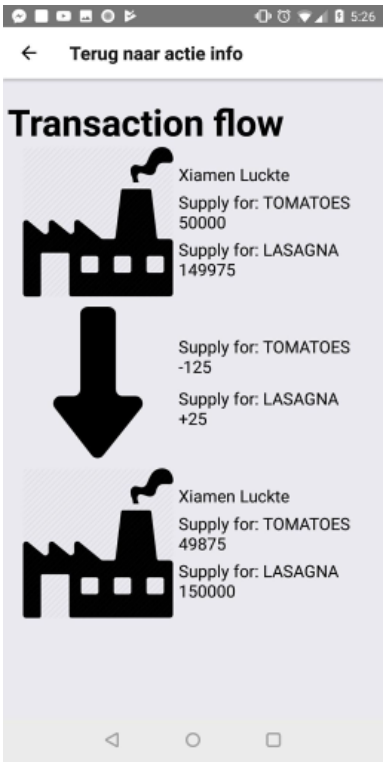
...


5:26
Terug naar actie info
Transaction flow
Price per unit: 0.5 euro
From: Brazil Airport
To: Brazil
Heinz
Supply for: TOMATOES
+5000
Price per unit: 0.5 euro
Heinz
Supply for: TOMATOES
55040

Demonstratie 3



...



B.2 Resultaten testen

creatie van producten in één lijst

```
creating 5 products at once
3.885
3.858
3.946
3.777
3.738
3.555
3.596
3.569
3.736
3.51
3.6
3.714
3.839
3.782
3.504
3.503
3.706
3.733
4.19
3.586
3.821
4.089
3.671
3.819
3.603
3.765
4.003
3.59
3.524
3.896
Done
```

```
creating 50 products at once
7.166
7.195
7.157
7.014
6.975
7.366
6.986
6.901
6.784
6.527
6.92
6.95
7.324
7.273
7.572
7.136
6.72
6.939
8.034
7.487
7.263
7.313
7.235
6.811
6.811
7.748
8.108
7.367
7.784
7.805
Done
```

```
creating 100 products at once
16.779
14.631
15.484
14.975
15.151
24.089
20.632
19.708
19.514
14.864
16.634
14.372
15.288
14.769
15.082
14.487
14.717
15.075
15.001
15.054
15.022
14.079
15.558
16.145
15.817
15.838
14.925
15.562
14.865
14.171
Done
```

creatie van producten elk apart

```
creating 5 products, one per one
17.144
16.907
17.606
20.05
18.005
17.512
16.095
17.029
16.404
17.793
18.059
18.212
17.185
17.462
17.952
18.841
17.494
17.241
17.149
16.761
18.048
15.74
18.67
17.229
17.984
17.054
16.562
16.051
17.196
17.464
Done
```

```
creating 50 products, one per one
156.376
157.344
159.944
156.073
158.803
159.716
158.646
161.548
155.565
163.905
169.879
168.7
163.75
162.292
163.92
165.653
168.432
158.105
157.899
160.339
160.292
181.697
196.959
169.04
183.015
159.193
158.8
154.998
155.525
155.161
Done
```

verwijderen van producten in één lijst

```
deleting 5 products at once
5.311
4.972
5.07
4.932
4.98
4.985
5.167
4.978
5.44
5.268
5.42
5.364
5.122
4.917
5.362
4.936
5.058
5.064
5.114
5.053
4.953
5.257
5.138
5.396
5.247
5.03
4.843
5.816
5.233
```

```
deleting 50 products at once
10.273
12.951
13.085
12.866
13.092
12.309
14.489
15.357
14.752
14.459
13.479
13.345
13.497
13.441
12.585
13.21
13.788
13.943
13.817
14.76
12.114
12.622
12.171
12.647
13.352
12.206
12.292
12.618
12.881
12.963
```

Ophalen van alle producten

```
getting all 5 products
0.629
0.602
0.637
0.603
0.822
0.869
0.572
0.621
0.506
0.448
0.583
0.647
0.518
0.559
0.766
0.725
0.49
0.518
0.465
0.606
0.621
0.475
0.392
0.417
0.413
0.41
0.436
0.526
0.401
0.551
Done
```

```
getting all 50 products
1.254
0.864
1.135
0.938
1.392
1.447
1.318
1.007
1.8
1.293
0.861
0.96
0.961
1.189
1.313
1.037
1.163
1.351
0.901
0.972
0.996
1.174
0.857
0.726
0.878
0.897
0.95
1.014
1.19
1.201
Done
```

```
getting all 100 products
2.299
1.68
1.615
2.34
2.079
2.146
1.429
2.295
1.929
1.665
2.32
1.945
2.039
2.461
2.147
1.54
2.879
1.577
1.37
2.262
2.069
1.538
2.126
1.55
1.55
1.894
2.26
2.032
1.847
1.801
Done
```

Ophalen van één bepaalde ID

```
getting id 2 out of 5 products
0.607
0.45
0.447
0.558
0.555
0.587
0.533
0.727
0.641
0.481
0.46
0.496
0.532
0.55
0.549
0.491
0.572
0.339
0.604
0.501
0.535
0.482
0.422
0.445
0.609
0.537
0.519
0.538
0.566
0.46
Done
```

```
getting id 2 out of 50 products
0.705
0.785
1.01
0.863
0.65
0.823
0.536
0.539
0.525
0.678
0.542
0.499
0.477
0.567
0.544
0.532
0.536
0.567
0.483
0.522
0.596
0.587
0.442
0.608
0.524
0.578
0.55
0.462
0.493
0.495
Done
```

```
getting id 45 out of 50 products
0.572
0.588
0.526
0.557
0.494
0.682
0.592
0.515
0.502
0.467
0.503
0.572
0.545
0.533
0.519
0.61
0.593
0.456
0.647
0.538
0.525
0.451
0.523
0.349
0.895
0.552
0.574
0.507
0.545
0.423
Done
```

Ophalen van één bepaalde ID

```
getting id 2 out of 100 products
0.429
0.584
0.559
0.53
0.548
0.552
0.509
0.62
0.742
0.691
0.568
0.513
0.622
0.571
0.564
0.618
0.643
0.623
0.744
0.604
0.643
0.575
0.691
0.596
0.585
0.498
0.575
0.712
0.565
0.448
Done
```

```
getting id 78 out of 100 products
0.535
0.44
0.4
0.41
0.421
0.413
0.415
0.358
0.432
0.319
0.438
0.465
0.388
0.368
0.385
0.397
0.317
0.395
0.464
0.412
0.39
0.544
0.415
0.4
0.432
0.506
0.455
0.4
0.315
0.445
Done
```

B.3 Netwerk data

Soort bedrijf	Naam bedrijf	Land	Product
SUPPLIER	Zhengyuan	China	SEEDS
SUPPLIER	Reimer Seeds	Duitsland	SEEDS
SUPPLIER	Basso Semillas	Brazilië	SEEDS
SUPPLIER	Khanh Foods	China	TOMATOES
SUPPLIER	Silbury	Duitsland	TOMATOES
SUPPLIER	Bata Food	Brazilië	TOMATOES
SUPPLIER	Seawinner	China	FERTILIZER
SUPPLIER	Argus Fertilizer	Brazilië	FERTILIZER
SUPPLIER	Soil supplier 1	China	SOIL
SUPPLIER	Soil supplier 2	Brazilië	SOIL
SUPPLIER	Amprion	Duitsland	ELEKTRICITY
SUPPLIER	Berlin Water Works	Duitsland	WATER
SUPPLIER	Weidemann GmbH	Duitsland	MACHINES
SUPPLIER	Schoeller Allibert	Duitsland	CONTAINERS

Soort bedrijf	Naam bedrijf	Land	Product
MANUFACTURER	Lovekitchen	China	SOUP
MANUFACTURER	Campbell's GmbH	Duitsland	SOUP
MANUFACTURER	Hiltfields	Duitsland	SPAGHETTI SAUCE
MANUFACTURER	Heinz	Brazilië	SPAGHETTI SAUCE
MANUFACTURER	ALIMENTOS	Brazilië	LASAGNA
MANUFACTURER	Xiamen Luckte	China	LASAGNA
MANUFACTURER	Tianz	China	KETCHUP
MANUFACTURER	Hiltfields	Duitsland	KETCHUP

Soort bedrijf	Naam bedrijf	Land	Product
DISTRIBUTER	DISTRIBUTER SAMPLE 1	Brazilië	PLANE
DISTRIBUTER	DISTRIBUTER SAMPLE 2	China	PLANE
DISTRIBUTER	DISTRIBUTER SAMPLE 3	Duitsland	PLANE
DISTRIBUTER	DISTRIBUTER SAMPLE 4	China	PLANE
DISTRIBUTER	DISTRIBUTER SAMPLE 5	Duitsland	PLANE
DISTRIBUTER	DISTRIBUTER SAMPLE 6	Brazilië	PLANE
DISTRIBUTER	DISTRIBUTER SAMPLE 7	China	CAR
DISTRIBUTER	DISTRIBUTER SAMPLE 8	China	CAR
DISTRIBUTER	DISTRIBUTER SAMPLE 9	Duitsland	CAR
DISTRIBUTER	DISTRIBUTER SAMPLE 10	Duitsland	CAR
DISTRIBUTER	DISTRIBUTER SAMPLE 11	Brazilië	CAR
DISTRIBUTER	DISTRIBUTER SAMPLE 12	Brazilië	CAR

Soort bedrijf	Naam bedrijf	Land	Product
RETAILER	COLRUYT	Duitsland	SPAGHETTI SAUCE
RETAILER	WALMARKT	Brazilië	SOUP
RETAILER	PRISMA	China	LASAGNA
RETAILER	PANOS	Duitsland	TOMATOES
RETAILER	FRITUUR T'HOEKSKE	Duitsland	KETCHUP

Soort bedrijf	Naam bedrijf	Land	Product
CUSTOMER	Wouter Van Hecke	China	SPAGHETTI SAUCE
CUSTOMER	Jens Mortier	China	SOUP
CUSTOMER	Hadrien Van Durme	Duitsland	LASAGNA
CUSTOMER	Sam Maesschalck	Duitsland	TOMATOES
CUSTOMER	Diego Carboni	Brazilië	KETCHUP

Bibliografie

Celikbilek, C. (2015, mei 1). A Fuzzy Approach for a Supply Chain Network Design Problem - Scientific Figure on ResearchGate. Verkregen van https://www.researchgate.net/figure/A-generic-supply-chain-network_fig1_280158117

Drescher, D. (2017). *Blockchain Basics*.

Goodnight, H. (2017, juli 10). Managing Supply Chains on the Blockchain: A Primer. Verkregen van https://blog.sweetbridge.com/managing-supply-chains-on-the-blockchain-a-primer-1f7dc293e3d9

Hays, J. (2008). AGRICULTURE IN CHINA. Verkregen van http://factsanddetails.com/china/cat9/sub63/item348.html

Hyperledger. (2015). HyperLedger Composer. Verkregen van https://hyperledger.github.io/composer/latest/

Marr, B. (2017, december 6). A Short History Of Bitcoin And Crypto Currency Everyone Should Read. Verkregen van https://www.forbes.com/sites/bernardmarr/2017/12/06/a-short-history-of-bitcoin-and-crypto-currency-everyone-should-read/#5eeca9563f27

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Verkregen van https://bitcoin.org/bitcoin.pdf

Rosic, A. (2017, juli 16). Paper Wallet Guide: How to Protect Your Cryptocurrency. Verkregen van https://blockgeeks.com/guides/paper-wallet-guide/

Sme, S. (2017, april 2). What is Supply Chain Management (SCM)? Verkregen van https://scm.ncsu.edu/scm-articles/article/what-is-supply-chain-management-scm

Uhlenberg, A. (2017, september 18). Key Issues in Supply Chain Management and How to Overcome Them. Verkregen van https://www.liaison.com/blog/2017/09/18/key-issues-supply-chain-management-overcome/

Verdanov. (2017, november 29). Bitcoin: wat zijn de voordelen en nadelen van bitcoins? Verkregen van https://financieel.infonu.nl/geld/121774-bitcoin-wat-zijn-de-voordelen-en-nadelen-van-bitcoins.html