

# Afstudeerverslag

*Allianz Group - HeadFWD*

Door: Calum Iain Munro



Software Development, ICA, VT

HBO afstudeerverslag:

**Versie: 1 (definitief)**

**Datum: 15 juni 2018**

**Gegevens opdrachtgever:**

Bedrijf: HeadForward B.V.

Contactpersonen: Daniël Siahaya

**Gegevens opleiding:**

Opleiding: HBO bachelor Informatica

School: Hogeschool van Arnhem en Nijmegen

Begeleider: Misja Nabben

Assessor: Rein Harle

**Gegevens opdrachtnemer:**

Teamlid: Calum Iain Munro (549288)

## INHOUDSOPGAVE

---

<b>1 Samenvatting</b>	<b>4</b>
1.1 Project omgeving . . . . .	4
1.2 Projectopdracht . . . . .	4
1.3 Belangrijkste Resultaten . . . . .	4
1.4 Conclusies . . . . .	5
<b>2 Inleiding</b>	<b>6</b>
<b>3 Context</b>	<b>7</b>
3.1 Begeleiders . . . . .	7
<b>4 Opdrachtomschrijving</b>	<b>8</b>
4.1 Doelstelling . . . . .	8
<b>5 Methode</b>	<b>9</b>
5.1 Technieken . . . . .	11
5.1.1 C4 framework . . . . .	12
<b>6 Process en resultaten</b>	<b>13</b>
6.1 Verloop van het project . . . . .	13
6.1.1 Fase: Initiatieffase (iteratie 0) . . . . .	13
6.1.2 Fase 2: Definitiefase (iteratie 0 - 1) . . . . .	14
6.1.3 Fase 3: Analyse / Realisatie Fase (Iteraties 1 - 5) . . . . .	14
6.1.3.1 Iteratie 1 - Onderzoek naar Blockchain en huidige proces . . . . .	15
6.1.3.2 Iteratie 2 - Onderzoek naar de architectuur van de blockchain en smart contracts . . . . .	16
6.1.3.3 Iteratie 3 - Analyse + Realisatie fase . . . . .	17
6.1.3.4 Iteratie 4 - Analyse + Realisatie fase . . . . .	18
6.1.3.5 Iteratie 5 - Realisatie fase . . . . .	18
6.1.4 Fase 4: Afrondingsfase (Iteraties 6 - 7) . . . . .	19
6.2 Resultaten en Producten . . . . .	19
<b>7 Conclusie &amp; Aanbevelingen</b>	<b>21</b>
7.0.1 Aanbeveling . . . . .	21
<b>8 Reflectie</b>	<b>22</b>
8.1 Situatiebeschrijvingen . . . . .	23
8.1.1 Projectaanpak keuze . . . . .	23
8.1.2 Communicatie met de opdrachtgever . . . . .	24
8.1.3 Oplossingsrichtingen planmatig vergelijken . . . . .	24
8.1.4 Analyse software design documenteren . . . . .	25

<b>Bibliografie</b>	<b>25</b>
<b>9 Bijlagen</b>	<b>27</b>
<b>A Smart contract: Claims</b>	<b>28</b>

# HOOFDSTUK 1

## SAMENVATTING

---

Deze samenvatting beschrijft in hoofdpunten het afstudeertraject. Deze hoofdpunten zijn:

- De project omgeving
- De projectopdracht
- De belangrijkste behaalde resultaten
- De getrokken conclusies.

Het doel van deze samenvatting is om de belangrijkste onderdelen van dit afstudeerproject te tonen.

### 1.1 Project omgeving

Dit project heeft plaatsgevonden in het kantoor van HeadFWD, gevestigd aan het John M. Keynesplein 12-46 in Amsterdam. Het project is uitgevoerd door Iain Munro. Hij is hierbij begeleid door Daniël Siahaya, die met zijn rol als Product Owner in direct contact stond met de opdrachtgever, Allianz Group.

### 1.2 Projectopdracht

Tijdens de afstudeerstage bij HeadFWD is er onderzoek uitgevoerd naar de mogelijkheden om Blockchain technologie toe te passen voor het claimproces van verzekeringen bij Allianz Group.

### 1.3 Belangrijkste Resultaten

Het resultaat van dit afstudeeropdracht is vooral het Smart Contract geschreven in Solidity en het onderzoek naar de Blockchain technologie. Dit resultaat is terug te vinden in bijlagen A. Hiermee is de hoofdvraag van dit afstudeerproject: “Hoe is de blockchain technologie in te zetten om het claimproces van Allianz te automatiseren?” beantwoord.

## **1.4 Conclusies**

Na afloop van het project kunnen alle hoofdfunctionaliteiten van het claim systeem uitgevoerd worden. Gedurende het gehele project naar voren is gekomen dat de technieken die gebruikt worden met de Blockchain technologie daadwerkelijk functioneel zijn, maar niet altijd even goed meewerken.

Gesteld kan worden dat meer tijd nodig is om een goed werkend product neer te kunnen zetten. Verder is de conclusie die betrekking heeft tot mijn proces dat ik positieve feedback heb ontvangen van mijn bedrijfsbegeleider Daniël. Ik heb van een lastig project met veel onduidelijkheden en onstabiele techniek een goed resultaat geboekt en mijn proces juist en inzichtelijk uitgevoerd. Hieruit is te concluderen dat ik geslaagd ben in het behalen van het gewenste projectresultaat en de competenties vanuit school.

## HOOFDSTUK 2

### INLEIDING

---

Voor u ligt het afstudeerverslag van Iain Munro, dat is geschreven naar aanleiding van het afstudeerproject dat loopt van 5 februari 2018 tot en met 29 juni 2018 (2017/2018 periode 3 & 4). De Afstudeeropdracht is uitgevoerd in opdracht van Allianz Group via het stagebedrijf HeadFWD. Deze opdracht bestond uit het ontwikkelen van een 'proof of concept' voor de opdrachtgever, waarbij vooraf analyse en onderzoek werd uitgevoerd.

Het doel van dit verslag is om inzicht te geven over de inhoud, aanpak, verloop en resultaten van dit project. Het eerste gedeelte van dit verslag heeft betrekking op de context van het project. Hierna volgt de opdrachtomschrijving, waarin kort de opdracht en het uitgevoerde onderzoek worden beschreven. Aangezien dit al eerder uitgebreid is beschreven wordt hier alleen kort samengevat met een verwijzing naar het Plan van Aanpak.

Vervolgens worden de gebruikte methoden en technieken besproken. Hierbij wordt voornamelijk gekeken naar de reden waarom deze technieken zijn gebruikt, en in hoeverre deze nuttig waren voor het project. Verder staat er een korte beschrijving van alle (tussen)resultaten van het project.

In de procesbeschrijving worden de belangrijkste activiteiten en besluiten verantwoord. Hierbij wordt verwezen naar de verschillende competenties die de bekwaamheid aantonen van de afstudeerde als beginnend beroepsprofessional. Verder worden de opgeleverde resultaten en producten toegelicht. Hierdoor zou er een duidelijk beeld moeten ontstaan en zou het helder moeten zijn in hoeverre deze voldoen aan de gestelde eisen. Het volledige eindproduct is meegeleverd met dit document en een verwijzing is te vinden in de bijlagen.

Dit verslag wordt afgerond met de conclusies en aanbevelingen. Hierin wordt teruggekoppeld naar de opdracht- en doelstelling van het project. De vraag "In hoeverre ben ik geslaagd in het gewenste projectresultaat en daarmee de doelstelling(en) van het project te behalen" wordt hier beantwoord, en de belangrijkste conclusies worden weergegeven. Op de momenten waar het gaat over mijn proces maak ik gebruik van de ik vorm, omdat dit gaat over mijn proces. Naar aanleiding van deze conclusies wordt er een aanbeveling gedaan voor eventuele vervolgacties of aandachtspunten voor het opgeleverde resultaat.

Het einde van dit verslag bestaat uit een bronnenlijst waarbij de opgeleverde producten zoals het Plan van Aanpak, Onderzoeksverslag en Analyse documenten zijn meegeleverd met dit verslag.

# HOOFDSTUK 3

## CONTEXT

---

Het afstudeerproject wordt uitgevoerd bij HeadFWD voor een van hun klanten, Allianz Group. HeadFWD is een consultancy bedrijf en ontwikkelt, beheert en host daarnaast maatwerk software en data- en integratie oplossingen [2].

De Allianz Group is een Duitse verzekерingsmaatschappij[2] en heeft HeadFWD gevraagd om bepaalde bedrijfsprocessen te automatiseren, waaruit dit afstudeerproject is gevormd. Verdere details over de opdracht zijn beschreven in het volgende hoofdstuk.

De werkzaamheden voor afstudeeropdracht zijn bij HeadFWD op kantoor in Amsterdam uitgevoerd. Het project wordt full-time alleen uitgevoerd door Iain Munro onder begeleiding van Daniël Siahaya die de rol als productowner heeft vervuld.

### 3.1 Begeleiders

Voor dit afstudeerproject zijn er vanuit de Hogeschool Arnhem Nijmegen Misja Nabben als stagebegeleider aangesteld en Rein Harle als Assessor. Zie het Plan van Aanpak[2] voor de contactgegevens.

# HOOFDSTUK 4

## OPDRACHTOMSCHRIJVING

---

Zoals behandeld in het plan van aanpak verzekert de opdrachtgever, Allianz Group panden voor miljoenen. Zogenoemde co-insurance verzekeringen worden in de praktijk gedeeld met meerdere verzekeraars, om zo het risico te verspreiden.

Het probleem en ook gelijk de aanleiding voor dit project is dat het claimproces te veel tijd kost voordat deze wordt uitgekeerd naar de klant. Dit komt doordat het door verschillende bedrijven en hun handmatige bedrijfsprocessen gevalideerd moet worden. Hierdoor kan een claim vaak 3 maanden of langer duren voordat deze werkelijk wordt uitbetaald.

### 4.1 Doelstelling

De doelstelling van dit project is het versnellen van het claimproces. Zodat claims sneller uitbetaald worden. Om dit doel te bereiken wordt er een proof of concept (POC) ontwikkeld, waarin de nieuwe situatie wordt gedemonstreerd.

Om dit doel te bereiken is er eerst onderzoek uitgevoerd naar de blockchain technologie en de verschillende mogelijke oplossingen. Hoe ik tot deze vragen ben gekomen staat in mijn onderzoeksverslag[1]. Verder is er analyse uitgevoerd om de requirements en use cases in kaart te brengen voor het proof of concept.

De subdoelstelling van dit afstudeerproject is kennis opbouwen over de blockchain en smart contract technologieën. Hiermee kunnen in vervolgprojecten of voor andere use cases hier gebruik van gemaakt kan worden.

# HOOFDSTUK 5

## METHODE

---

Om het project kwalitatief goed uit te kunnen voeren is er gebruik gemaakt van verschillende methoden en technieken. Er is gezien de HBO vaardigheden competenties en kenmerken van het project gekozen voor een agile iteratieve projectmethodiek. Als projectmethodiek is er gekozen voor Kanban, die verder beschreven is in het plan van aanpak [PVA].

De reden voor deze aanpak is omdat hiermee werk gestructureerd opgesplitst, ingeschat, geprioriteerd en ingepland wordt in iteraties. Dit geeft de vereiste flexibiliteit om snel te kunnen schakelen bij aanpassingen of nieuwe requirements. Dit vereist het gebruik van experimentele software, materie en technologieën die nog niet bekend waren voordat dit project begon.

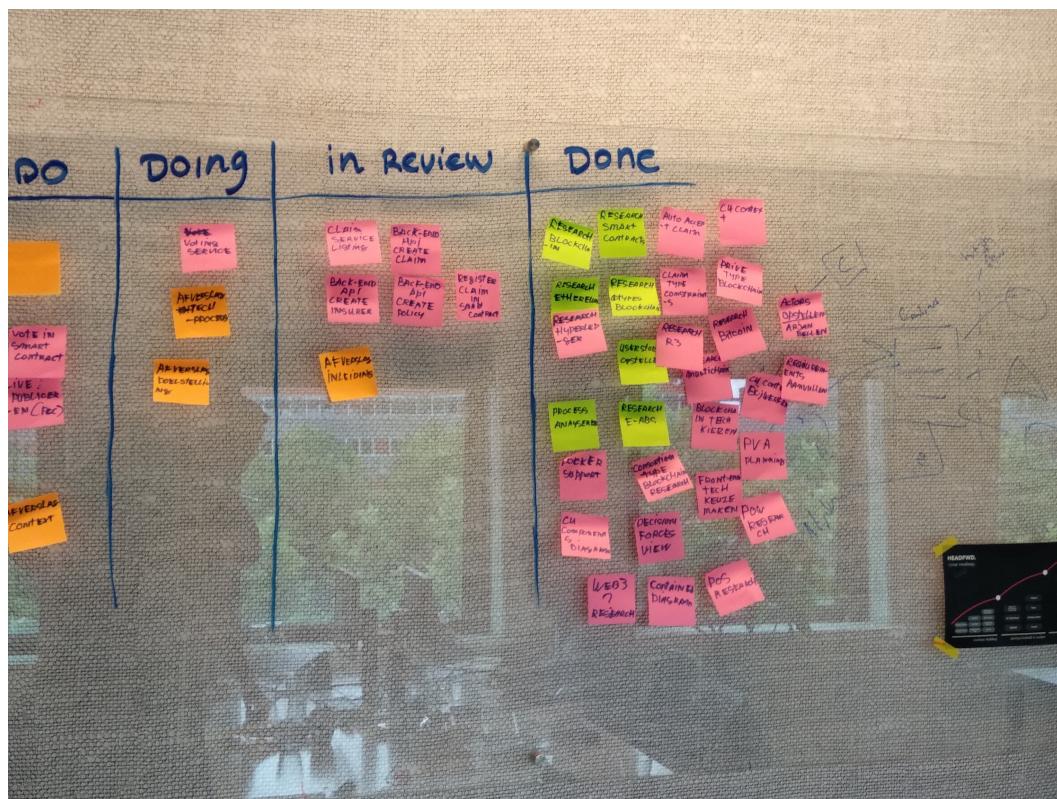
Deze praktische aanpak maakt het mogelijk dat bevindingen tijdens het onderzoeken naar een onderwerp gelijk gebruikt kunnen worden in een stuk ontwikkeling van het prototype (proof of concept). Verder is er voor Kanban gekozen omdat het project niet in teamverband wordt uitgevoerd. Kanban, in tegenstelling tot Scrum, vereist niet de teamvergaderingen die normaal gesproken nodig zijn als het project in teamverband wordt uitgevoerd (daily standups, retrospectives etc..).

Met de uitvoering van Kanban is er gekozen om gebruik te maken van het whiteboard op kantoor, zie figuur 5.1. Het bord bestaat uit vier kolommen: To Do, In Progress, Review en Done. Net als de bekende Scrum methodiek staat hier al het opgesplitste werk op, voor zowel de school als voor het stagebedrijf. Zo wordt alles eerst onder het kopje To Do gezet. Het werk wat een hoge prioriteit heeft, wordt bovenaan geplaatst. Wanneer ik begin met een nieuwe ticket dan wordt deze verplaatst naar de In Progress kolom. Op het moment dat ik het werk heb afgerond wordt deze eerst opgestuurd voor feedback en acceptatie. Dit gebeurt door de ticket in de review kolom te plaatsen.

Zodra een ticket goedgekeurd is door de stakeholder, dan wordt het werk als klaar beschouwd. In het geval van het proof of concept onderzoek of analyse documenten, wordt deze review uitgevoerd door de productowner (Daniël Siahaya) of in sommige gevallen door de opdrachtgever (Allianz Group, Arjan Zaal). Het werk voor school wordt vanuit de review kolom naar To Do gezet nadat er feedback ontvangen is van de stagebegeleider en assessor.

Door deze aanpak is er één oogopslag duidelijk wat de huidige stand van zaken is. Dit geeft ook inzicht in het proces, want hierdoor wordt al snel duidelijk tegen het einde van een iteratie of er niet te veel nog staat bij de To Do of Doing kolom.

Hierdoor kan ik iedere iteratie nieuwe doelstellingen maken en controleren of het project nog op schema ligt. Op momenten dat dit niet zo is, kan er een juiste afstemming gemaakt worden. Deze retrospective<sup>1</sup> wordt uitgevoerd na iedere iteratie met de bedrijfsbegeleider, waar ik reflecteer op mijn eigen werkprocessen van die iteratie. Deze retrospective momenten hebben in het begin wat meer tijd gekost, maar met het verloop van het project zijn deze momenten steeds minder nodig. Na een iteratie wordt dan ook de resultaten of bevindingen van onderzoek behandeld en gekeken naar wat er te doen staat in de volgende iteratie. Om deze feedback loop zo kort mogelijk te houden is er gekozen voor 2 weken iteraties, zodat er voldoende tijd is om resultaten te boeken en voldoende feedback momenten zijn.



Figuur 5.1: Foto genomen van het Kanban bord op 11-06-2018

---

<sup>1</sup> <https://www.atlassian.com/team-playbook/plays/retrospective>

## 5.1 Technieken

Er zijn meerdere technieken gebruikt tijdens het project, waarvan een aantal die bekend zijn vanuit de studie. Om te beginnen is er een work breakdown structure gebruikt om werk op te splitsen en vervolgens op te schrijven op post-its op het kanban bord. Om te achterhalen wat de requirements, user stories en het minimum viable product zijn, is er analyse uitgevoerd en verbaal overleg uitgevoerd met de opdrachtgever en de product owner. Deze analyse is vastgelegd in een requirements document en specificeert op een gestructureerde en gestandaardiseerde manier door UML en C4 diagrammen te maken. Deze opgestelde eisen zijn gemaakt gedurende het softwareontwikkeltraject en vervolgens evalueert op basis van gebruikersbehoeften per user story deze door te nemen met de klant. (SD-1)

Voor het vermelden van de gebruikte bronnen is de APA richtlijnen gebruikt en tevens verplicht vanuit de HAN. Voor het indelen van de te maken producten en requirements is de MoSCoW-methode toegepast. Er wordt tijdens de ontwikkeling van het proof of concept gebruik gemaakt van GIT, versiebeheer systeem. Waarbij de GitFlow best practice is gebruikt om met iedere nieuwe feature een nieuwe git branch te maken en deze te mergen met een pull-request.

Verder is op basis van ontwerp meerdere werkende en betekenisvolle data-intensieve en gedistribueerde software systemen gerealiseerd. Zo is bij de ontwikkeling van het Smart Contract rekening gehouden met hergebruik, het verminderen van complexiteit en onderhoudbaarheid. Deze code bevat ook de core business rules en is daarom ge-unit test.

Er is geen gebruik gemaakt van continuous integration of deployment gezien de tijdsdruk en het feit dat het een prototype is dat na dit afstudeerproject niet verder gebruikt zou worden.

Tijdens het onderzoek zijn er prototyping en experimenten uitgevoerd naast literatuuronderzoek. Zo is voor de Blockchain technologie kleinschalig, systematisch en methodisch onderzoek uitgevoerd. De bevindingen en conclusies hieruit hebben geholpen om bepaalde besluiten in het project te onderbouwen.

Er is geen digitaal Kanban bord gebruikt zoals Trello of Jira, omdat het fysieke bord voldoende is geweest. Dit is mede mogelijk omdat ik iedere dag op kantoor ben en niemand anders op afstand hier naar hoeft te kijken.

### **5.1.1 C4 framework**

Voor het modelleren van de architectuur is er gekozen voor de Simon Brown's C4 Framework. Deze techniek is mij geleerd tijdens het Advanced Software Development semester. De reden hiervoor was dat door de gekozen agile projectaanpak, software design niet met een heavy upfront design wordt uitgevoerd. Met een agile aanpak wordt bijvoorbeeld UML vooral gebruikt om informele aantekening te maken die bestaan uit vierkanten en lijnen om een verbaal verhaal duidelijk te krijgen. Het probleem hiermee alleen is dat dit niet heel inzichtelijk is of het waard is om in een document mee te nemen.

Door het gebruik van C4 framework heb ik een formele manier van communicatie waarmee ik aan de verschillende stakeholders de software architectuur kan uitleggen. Hiermee kan ik agile ontwikkelen en tegelijkertijd de competenties SD-2 en SD-3 vanuit school aantonen en de architectuur duidelijk krijgen. Dit doet C4 door 4 verschillende diagrammen te ontwikkelen met variërende niveaus van abstractie.

Deze argumentatie is hetgeen wat mij is geleerd tijdens de SWA lessen en is terug te vinden in slide 11 van de Software Architecture Course: Simon Brown's C4 Framework [3].

# HOOFDSTUK 6

## PROCESS EN RESULTATEN

---

In dit hoofdstuk beschrijf ik het proces en de resultaten van het afstudeerproject. Bij afwijkingen van het originele plan dat beschreven in het Plan Van Aanpak is dit beargumenteerd. Verder wordt er in dit hoofdstuk alleen het proces en de resultaten geconstateerd. Reflectie hierop en mijn keuzes worden in een volgend hoofdstuk beschreven en in dit hoofdstuk constateer ik alleen wat er is gebeurd.

### 6.1 Verloop van het project

Om het verloop van het project gestructureerd te verwoorden, is het project te verdelen in 4 fases: Initiatie fase, definitie fase, analyse en realisatie fase en afronding fase. Dit zijn ook de fases die opgenomen zijn in de planning in het plan van aanpak en zijn in de onderstaande subparagrafen op chronologische volgorde beschreven.

Door de agile aanpak is in derde fase er beide analyse en realisatie tegelijkertijd uitgevoerd. De gekozen Kanban agile methode laat dit ook toe. De reden hiervoor is om risico's van het project te verminderen door de software, het onderzoek en enige analyse tegelijkertijd stukje voor stukje uit te voeren in iteraties <sup>1</sup>.

#### 6.1.1 Fase: InitiatiEFFASE (iteratie 0)

In deze eerste fase is het doel om een gelijk beeld van het project te verkrijgen van alle betrokkenen aan het project en hiermee goedkeuring te krijgen om het project te starten. Dit is gedaan door de stand van zaken vast te stellen en onderzoek te doen naar de aanleiding en doelstelling van het project.

Deze fase is samen met de definitiefase uitgevoerd waar verdere analyse is gedaan naar het te ontwikkelen product. Dit is gedocumenteerd in het Plan Van Aanpak waarin het beoogde projectresultaat in termen van: randvoorwaarden, functionele en operationele prestaties, eisen, wensen en projectgrenzen duidelijk wordt. Een planning in de vorm van een GANTT chart met daarin de deadlines die vanuit school bekend zijn, geeft een beeld voor de komende fases.

---

<sup>1</sup><https://www.twynstraguddekennisbank.nl/projectmanagement/faseer-het-project>

### **6.1.2 Fase 2: Definitiefase (iteratie 0 - 1)**

In de tweede fase is er verder gekeken naar het te ontwikkelen product. In deze fase is het Plan van Aanpak verder afgerond en vervolgens goedgekeurd, door zowel de stagebegeleider en assessor als de bedrijfsbegeleider. De ontwikkelmethode, Kanban is gekozen en de planning voor de aankomende realisatie- en analysefase zijn opgesplitst in 7 iteraties van 2 weken. Daarbij wordt per iteratie gekeken naar wat er nodig is aan werk voor de komende 2 weken.

Om te beginnen is het duidelijk dat er een onderzoek gestart moet worden. De aanleiding hiervoor is dat het doel van het proof of concept moet aantonen dat het elektronische claim systeem use case van Allianz met de Blockchain Technologie geautomatiseerd kan worden.

Omdat ik zelf nog niet bekend ben met de Blockchain Technologie en het duidelijk is dat er in een later stadium een geïnformeerd besluit gemaakt over moet worden, is het duidelijk dat de eerste iteratie zich vooral bezighoudt met een algemeen onderzoek naar de Blockchain Technologie. Hiermee is de definitiefase afgerond en begint de analyse / realisatie fase.

### **6.1.3 Fase 3: Analyse / Realisatie Fase (Iteraties 1 - 5)**

Er is tijdens deze fase meerdere malen analyse en realisatie uitgevoerd. In de onderstaande subparagraphen beschrijf ik per iteratie hoe het proces is gelopen en waar aan gewerkt is. De grootste afwijking met het originele plan, in het plan van aanpak is het opsplitsen van de resultaten in het onderzoeksverslag dat bestaat uit een literatuuronderzoek, analyse en het proof of concept experiment. Dit wordt gedaan na het ontvangen van de feedback op de 80% versie en om verwarring te vermijden wordt er in het proces hieronder verwezen naar de namen van de losse documenten die aan het einde van het project met dit afstudeerverslag zijn meegeleverd.

### **6.1.3.1 Iteratie 1 - Onderzoek naar Blockchain en huidige proces**

In de eerste iteratie is er gekeken naar hoe het huidige claim proces werkt. Deze analyse vult de analyse die is gestart voor het Plan van Aanpak aan met gesprekken met de opdrachtgever Allianz Group en de Product Owner. Uit deze analyse zijn requirements opgesteld en user stories en actoren van het te ontwikkelen proof of concept gedocumenteerd in het Software Architecture Document. (SD-1, SD-2)

Daarnaast is er een literatuuronderzoek uitgevoerd naar een technische basis termen rondom de blockchain-technologie en gerelateerde onderwerpen. Dit is gedaan om een aantal basis technische termen in cryptografie, blockchain-technologie en gerelateerde onderwerpen duidelijk te krijgen. Hiermee weet ik waar ik het over heb wanneer het proof of concept in vervolg iteraties behandeld wordt.

De focus is met name de manier waarop de blockchain gebruikt zou kunnen worden voor het doeleinde van de klant. Tijdens de eerste week is al snel duidelijk dat er verschillende types Blockchains zijn: Consortium, Privé en openbaar, en die zijn vervolgens onderzocht. (SD-7)

### **6.1.3.2 Iteratie 2 - Onderzoek naar de architectuur van de blockchain en smart contracts**

Tijdens de tweede iteratie is er verder onderzoek uitgevoerd naar de blockchain technologie en is het uitgebreid naar Smart Contracts en de blockchain zijn architectuur. Dit allemaal om duidelijk te krijgen hoe hiermee een gedecentraliseerde blockchain applicatie (Dapps<sup>2</sup>) ontwikkeld zou kunnen worden.

Verder is het onderzoek aangevuld door bij het onderzoek naar de verschillende types blockchain te kijken wat handig is voor het proof of concept en welke populaire blockchain projecten een gedecentraliseerde applicatie op gebouwd zou kunnen worden. (SD-7)

Tijdens deze iteratie is de analyse naar welke factoren in het eindproduct verwerkt moet worden afgerond en vervolgens goedgekeurd door zowel de klant als de product owner. Dit is gedaan door de scope van het proof of concept met de MoSCoW analyse te definiëren.

Hieruit is gekozen dat de Must Haves betrekking hebben op het registeren van Claim aanvraag. Deze ziet op het moment dat iedere verzekeraar hierop zijn stem heeft ingevoerd of de claim geaccepteerd of geweigerd wordt. Verder was een Must Have dat dit allemaal door Blockchain transacties inzichtelijk en controleerbaar moet zijn voor iedere gebruiker van het systeem. Op het moment dat dit niet mogelijk is dan heeft het proof of concept een negatiever resultaat voor Allianz, want het doel was zonder een centraal punt alle transacties gevalideerd kunnen worden.

Should haves waren bijvoorbeeld om het werk te verminderen dat claim aanvragen met een bedrag onder 1000 euro en het type diefstal automatisch geaccepteerd zouden worden, mits hier voldoende tijd voor is. Daarnaast zijn functionaliteiten zoals het kunnen inloggen met een gebruikersnaam en wachtwoord opgenomen. De Could Haves, de factoren die een leuke toevoeging zouden zijn wanneer er voldoende tijd over is maar dat helaas niet het geval is, is dat polissen en nieuwe makelaars geregistreerd kunnen worden. (SD-1)

Daarnaast is in eerste instantie het plan dat het proof of concept gebruik zou maken van het bestaande E-ABS systeem. Na overleg met de stakeholders is duidelijk geworden in deze iteratie dat dit niet het doel is en dat het proof of concept alleen moet aantonen dat de use cases mogelijk zijn met de Blockchain technologie. Dit is ook gelijk een constraint waardoor er tijdens dit project niet gekeken hoeft te worden naar andere methodes dan de blockchain. Dit zou eventueel in een vervolgactie opgenomen kunnen worden.

---

<sup>2</sup>[https://en.wikipedia.org/wiki/Decentralized\\_application](https://en.wikipedia.org/wiki/Decentralized_application)

### **6.1.3.3 Iteratie 3 - Analyse + Realisatie fase**

Het onderzoek naar de blockchain is in iteratie 3 afgerond. Er is voldoende informatie om een geïnformeerd besluit te maken op de juiste oplossingsrichting voor de blockchain technologie.

Tijdens deze iteratie en komende iteraties zijn Software Architecturele besluiten gemaakt met een techniek die mij is geleerd tijdens de Software Architectuur lessen in het ASD semester. De techniek houdt in dat de requirements die invloed hebben op de architectuur en andere invloeden, zogenaamde decision forces, opgeschreven en meegenomen worden met een architecturaal besluit. Deze decision forces zijn samen met de product owner opgesteld en met ieder oplossingsrichting meegenomen. (SD-2, SD-3)

In deze iteratie gaat het vooral om het besluit van Blockchain framework waarop het proof of concept opgebouwd zou worden. Door het onderzoek naar dit onderwerp in vorige iteraties kan hier vrij snel een besluit opgenomen worden. Het type blockchain dat handig is voor het proof of concept is al bekend: consortium. Hierin is er maar een blockchain technologie die voldoende functionaliteit en documentatie biedt om het proof of concept te bouwen.

Voor het modelleren van de architectuur is er gekozen voor de Simon Brown's C4 Framework. Hier is in deze iteratie een start mee gemaakt waarmee ik 4 verschillende diagrammen kon ontwikkelen met variërende niveaus van abstractie.

Naast alle analyse is er ook een stuk software ontwikkeld in deze iteratie. Dit is namelijk het smart contract geschreven in Solidity waarmee de core functionaliteiten en business rules in zijn opgenomen. Zie bijlage A die in de eindfase alle belangrijke requirements R2, R3, R4, R6, R7, R8, R9, R10, R11, R13 en R14 opneemt en wordt uitgevoerd in de blockchain. Deze zijn vervolgens ge-unittest (SD-4, SD-5, SD-6, SD-8).

Tijdens de retrospective van deze iteratie is er besloten om de laatste twee iteraties meer focus te zetten op het bereiken van resultaten van het proof of concept.

#### **6.1.3.4 Iteratie 4 - Analyse + Realisatie fase**

Tijdens de vierde iteratie is het smart contract verder ontwikkeld. Hierbij zijn functionaliteiten zoals het automatisch accepteren van een claim geïmplementeerd. Het probleem hiermee is alleen dat dit nog niet toegankelijk is voor een eindgebruiker.

Daarnaast wordt duidelijk vanuit Allianz dat een nieuwe requirement (R1) gebruikers de mogelijkheid moet geven om in te kunnen loggen op het systeem via een webapplicatie. Daarom is er in deze iteratie besloten om te werken aan een simpele web api met gebruiker niveaus, zodat het smart contract in een later stadium gebruikt kan worden door een Grafische gebruikersinterface. (SD-1, SD-4, SD-6, SD-8)

Zodoende is er tijdens deze fase gekeken naar een geschikte back-end framework. Hierbij is vooral rekening gehouden met het feit dat er een Blockchain API beschikbaar moet zijn waarmee functies binnen het smart contract aangeroepen kunnen worden. Een ander belangrijk decision force voor dit besluit, is de bekendheid met het framework en taal, om zo de ontwikkelingskosten lager te houden. Hieruit is het Sails JS framework gekozen in combinatie met Angular doordat het geschreven is in Javascript die mogelijkheid biedt om gebruik te maken van de officiële Ethereum JavaScript API, Web3.

In deze iteratie is de software architectuur bijgewerkt met de nieuwe besluiten (SD-3). Een ander besluit wordt gemaakt op het moment dat ik erachter kom dat het itereren door een lijst van objecten, zoals een lijst van claims in een smart contract niet mogelijk is. De oplossing hiervoor is het gebruik van een database die op basis van events de aggregatie van data in transacties toegankelijk maakt. Dit besluit is ook met decision forces besloten en doordat er een link naar de transactie (etherscan.io) blijft binnen de blockchain kan de verificatie en validatie uitgevoerd worden.

#### **6.1.3.5 Iteratie 5 - Realisatie fase**

In de vijfde iteratie is de api verder ontwikkeld. Om requirement R15 te vervullen is er deze iteratie ook een webapplicatie gemaakt in Angular. Deze applicatie biedt een simpele gebruikersinterface, waarop verschillende gebruikers kunnen inloggen en via web interface van de web-api gebruik maken en daarmee functies aanroepen in het smart contract. Deze iteratie is er vooral vol ontwikkeld aan het de backend en de front-end applicatie om zoveel mogelijk functionaliteiten te realiseren.

Gezien de beperkte tijd en de hoeveelheid werk zijn er een aantal besluiten gemaakt om het werk te verminderen. Zo is er het besluit genomen om het authenticatie gedeelte van de applicatie door Firebase te laten gebeuren. Aangezien dit niet het primaire doel is van het proof of concept en er al gebruik wordt gemaakt van Firebase.

#### 6.1.4 Fase 4: Afrondingsfase (Iteraties 6 - 7)

De nazorg van dit project betreft eigenlijk alleen het afronden van documenten. Deze fase bevat ook een tussentijdse beoordeling van school, waaruit feedback is ontvangen om het grote onderzoeksverslag op te splitsen in losse documenten.

Verder is er tijdens deze fase gewerkt aan het afstudeerverslag wat u nu leest en zijn de bevindingen en conclusies uit het proof of concept op zijn geschreven in het proof of concept document.

## 6.2 Resultaten en Producten

Uit het proof of concept is gebleken dat het mogelijk is om het claimproces met de blockchain te automatiseren.

Het resultaat van dit afstudeeropdracht is vooral het Smart Contract geschreven in Solidity en het onderzoek naar de Blockchain technologie. Hiermee is de hoofdvraag van dit afstudeerproject: "Hoe is de blockchain technologie in te zetten om het claimproces van Allianz te automatiseren?" beantwoord.

De back-end en front-end code die er naast is ontwikkeld om het smart contract toegankelijk te maken, toont dat de volledige use cases van Allianz opgelost kunnen worden. De twee belangrijkste elementen van het proof of concept is dat nieuwe claims veilig geregistreerd worden in het smart contract op de blockchain. Met een wijziging van deze data is dit traceerbaar met een transactiecode. Dit is te zien in de onderstaande screenshot van de web applicatie, waar op het moment dat je op de transactiecode of verwerkingscode klikt je naar een externe derde website gaat die de transactie van data valideert in de blockchain. Alle musts en zelf een aantal

The screenshot shows a web application interface for managing claims. At the top, there is a dark blue header with the text 'HEADFWD + Allianz' and several navigation links: 'Claims', 'Polissen', 'Verzekeraars', and a yellow button labeled 'Uitloggen'. Below the header is a sub-header 'Claims' with a sub-sub-header 'Ingediende claims'. To the right of this is a yellow button labeled 'Claim indienen'. The main content area contains a table with the following columns: 'Transactiecode', 'Verwerkingscode', 'Polisnr.', 'Bedrag', 'Status', 'Claimdatum', and 'Acties'. There are seven rows of data in the table:

Transactiecode	Verwerkingscode	Polisnr.	Bedrag	Status	Claimdatum	Acties
0xfc0c2baed94...	0xfc0c2baed94...	daniel-01	€500,00	Automatisch goedkeurd	27-05-2018 12:21	Claim inzien
0x8fc5b6e1d1...	0x3415e0b980...	daniel-01	€200,00	Goedgekeurd	27-05-2018 12:23	Claim inzien
0x94be177449...	0x535336c2d7...	arjan-01	€200,00	Goedgekeurd	27-05-2018 13:59	Claim inzien
0xb19chdfdf3d...	0xbcdff179e3...	daniel-01	€1,000,00	Goedgekeurd	28-05-2018 14:37	Claim inzien
0x29244a0fcd...	0x29244a0fcd...	daniel-01	€999,00	Automatisch goedkeurd	28-05-2018 14:39	Claim inzien
0x387a9c49b4...	0x2bdb2890de...	daniel-01	€1,000,00	Afgewezen	28-05-2018 14:46	Claim inzien

Below the table, there is a grey sidebar with links to 'Zzp', 'Mkb', 'Verzekeringen', and 'Allianz'. The 'Zzp' section includes links to 'Eigen bedrijf starten', 'Arbeidsongeschikt raken', and 'Personnel aanmenen'. The 'Mkb' section includes 'Bedrijfspand kopen' and 'Bedrijf overnemen'. The 'Verzekeringen' section includes 'Pakketten', 'Arbeidsongeschikheidverzekering', 'Autoverzekering', 'Aansprakelijkheidverzekering', and 'Alle verzekeringen'. The 'Allianz' section includes 'Over ons', 'Nieuws', 'Blog', 'Vacatures', and 'Wijzer in geldzaken'. At the bottom of the sidebar, it says 'Powered by HEADFWD'.

Figuur 6.1: Foto van de user interface waar een lijst aan claims te zien is

coulds uit de MoScow analyse zijn gerealiseerd. Zo kan bijvoorbeeld ingesteld worden dat om een bepaald claimbedrag een claim automatisch geaccepteerd wordt. Maar deze berekening wordt door iedere lid van het netwerk uitgevoerd.

# HOOFDSTUK 7

## CONCLUSIE & AANBEVELINGEN

---

In dit hoofdstuk kan aan de hand van dit gehele project de vraag “In hoeverre ben ik geslaagd in het gewenste projectresultaat en daarmee de doelstelling(en) van het project te behalen” worden beantwoord en de belangrijkste conclusies getrokken worden.

Het claimproces van co-insured verzekeringen duurt te lang. Het doel is om aan te tonen dat met de recent opkomende blockchain technologie bestaande bedrijfsprocessen geautomatiseerd kunnen worden. Het resultaat is de responstijd van een claim te verminderen en dit aan te tonen met een proof of concept.

Na afloop van het project kunnen alle hoofdfunctionaliteiten van het claim systeem uitgevoerd worden. Hieruit kan geconcludeerd worden dat de doelstelling van het project is behaald. Het proof of concept is in week 23 gepresenteerd aan de opdrachtgever, Allianz waarop positieve feedback is ontvangen.

Gedurende het gehele project is naar voren gekomen dat de technieken die gebruikt worden met de Blockchain technologie daadwerkelijk functioneel zijn, maar niet altijd even goed mee werken. Het kost veel tijd en uitzoekwerk en het grootste knelpunt zit voornamelijk in de documentatie en support om een gedecentraliseerde blockchain applicatie te realiseren.

Gesteld kan worden dat meer tijd nodig is om een goed werkend product neer te kunnen zetten. De API van de Blockchain verandert ook nog steeds en de applicatie die 2 weken geleden was ontwikkeld, is nu al niet meer compatible met de laatste versie van Ethereum. Hieruit kan ook geconcludeerd worden dat de techniek nog te nieuw is voor functioneel gebruik.

Verder is de conclusie die betrekking heeft tot mijn proces dat ik positieve feedback heb ontvangen van mijn bedrijfsbegeleider Daniël. Ik heb van een lastig project met veel onduidelijkheden en onstabiele techniek een goed resultaat geboekt en mijn proces juist en inzichtelijk uitgevoerd. Hieruit is te concluderen dat ik geslaagd ben in het behalen van het gewenste projectresultaat en de competenties vanuit school.

### 7.0.1 Aanbeveling

Aan de hand van de bovenstaande conclusies kan er een aanbeveling gedaan worden. Gezien het feit dat na bijna zes maanden nog geen stabiel prototype gepresenteerd kan worden aan de opdrachtgever, is het aan te raden om dit door te ontwikkelen. Waar tijdens dit project niet naar gekeken hoeft te worden, is het gebruik van andere methodes dan de blockchain. Dit zou eventueel in een vervolgactie opgenomen kunnen worden.

# HOOFDSTUK 8

## REFLECTIE

---

In dit hoofdstuk reflecteer ik op mijn beroepsmatig handelen in het afstudeerproject. Deze reflectie voer ik uit aan de hand van situatiebeschrijvingen die volgens de STARRT methode zijn uitgewerkt. Deze situatiebeschrijvingen zijn opgesteld in het kader van:

- Mijn werkwijze en gemaakte keuzes ten aanzien van de projectuitvoering;
- Het toegepaste theoretisch kader;
- Het resultaat en de consequenties van het resultaat voor de organisatie.

Deze reflectie wordt gestructureerd gedaan op 4 kritische situaties in de volgende situatiebeschrijving zijn volgens de STARRT methode.

## 8.1 Situatiebeschrijvingen

### 8.1.1 Projectaanpak keuze

<b>Situatie</b>	Een situatie doet zich voor in de definitiefase waar een algemeen beeld duidelijk is over het project, maar met relatief gezien weinige informatie een juiste projectaanpak gekozen moet worden.
<b>Taak</b>	Mijn taak is op dat moment het schrijven van het Plan Van Aanpak document, waarin de keuze van projectaanpak en planning aanbod kwam.  Mijn doel is een projectaanpak te kiezen die makkelijk omgaat met veranderingen, onduidelijkheden en resultaatgericht is. Daarnaast moet het een projectaanpak zijn voor één projectlid, waarin vorige projecten er altijd in groepsverband is gewerkt. Verder heb ik als eis voor de projectaanpak dat deze mij voldoende de mogelijkheid geeft om mijn proces aan te tonen en momenten van retrospective om het steeds te verbeteren.
<b>Aanpak</b>	Mijn eerste aanpak van dit probleem is de overweging om het project aan te pakken op de manier die ik vanuit school heb geleerd. Zo heb ik gekeken naar een waterval of RUP aanpak, maar door de onzekerheid en onduidelijkheden in het project is een agile aanpak meer gewenst. Vanuit school is dan alleen Scrum over, echter voerde ik dit project alleen uit en ging dit tegen het hele principe van Scrum in. Uiteindelijk heb ik om advies gevraagd van mijn bedrijfsbegeleider, die Kanban heeft voorgesteld. Hier heb ik al eerder ervaring mee gehad, met een project buiten school, maar ik miste hier vooral de retrospective momenten en iteraties.  Ik heb vervolgens nog contact opgenomen met Jille Treffers, een docent vanuit de HAN waarvan ik weet dat hij mij hiermee goed kan begeleiden. Uit het advies en vervolgens nog kort onderzoek naar Kanban heb ik mijn besluit hierover genomen.
<b>Resultaat</b>	Het resultaat is een agile projectaanpak met Kanban, waar ik de analyse- en uitwerkingsfase heb opgesplitst in iteraties van 2 weken. Hierdoor hoef ik niet vooraf al te weten wat er per iteratie gedaan hoeft te worden en dit zorgt voor flexibiliteit, waarbij stukje voor stukje het proces verbeterd wordt en het inzichtelijk blijft voor de begeleiders en klant.
<b>Reflectie</b>	Achteraf gezien is het een verstandig besluit geweest. Ik heb hier de verschillende alternatieve overwogen en advies gevraagd van experts en zelf onderzoek gedaan naar de consequenties voor dit type project. Ik ben daarom van mening dat ik juist heb gehandeld en hiermee de geschikte methoden en technieken toegepast waarmee ik competenties: SD-6, SD-8 aanton.

### 8.1.2 Communicatie met de opdrachtgever

<b>Situatie</b>	De situatie speelt zich af in een van de eerste momenten dat er input nodig was vanuit de opdrachtgever.
<b>Taak</b>	Mijn taak op dat moment was bij afgerond werk dit op te sturen voor feedback en acceptatie. Echter nam het antwoord op deze vragen te veel tijd.
<b>Aanpak</b>	<p>Ik heb dit overlegd met mijn bedrijfsbegeleider en samen zijn we tot de conclusie gekomen dat dit op te lossen is als hij de rol als product owner neemt. Hiermee kan intern feedback gegeven worden op werk en is er minder communicatie nodig met de opdrachtgever.</p> <p>Het steeds wachten op antwoord is geen optie met de weinig hoeveelheid tijd en ik heb iemand anders nodig die vooral bij besluiten zijn meningen over het te maken systeem kan los laten.</p>
<b>Resultaat</b>	Het project wordt nadat Daniel product owner is geworden niet verder vertraagd door de opdrachtgever.
<b>Reflectie</b>	Ik heb naar mijn gevoel juist gehandeld en het probleem direct aangekaart en het ook samen opgelost.

### 8.1.3 Oplossingsrichtingen planmatig vergelijken

<b>Situatie</b>	Tijdens het realiseren zijn er een aantal momenten geweest waarop ik een besluit voor oplossingsrichting moest kiezen. Denk hierbij aan bijvoorbeeld de keuze van Blockchain project. Of de keuze van framework voor de back-end of front-end.
<b>Taak</b>	Mijn taak op dat moment was een besluit nemen op bijvoorbeeld de keuze van Blockchain project. Of de keuze van framework voor de back-end of front-end.
<b>Aanpak</b>	Ik heb dezelfde aanpak gebruikt als datgene wat mij is geleerd tijdens de Software Architectuur lessen. Het maken van decision forces, oftewel criteria waar een oplossing aan moet voldoen. Vervolgens heb ik kort onderzoek gedaan naar de verschillende oplossingen en mijn keuze kort naast een decision forces view beargumenteerd.
<b>Resultaat</b>	Een geschikte oplossingsrichting, waarbij ik de verschillende alternatieve heb overwogen en een geïnformeerd beargumenteerde heb gemaakt.
<b>Reflectie</b>	Ik heb in sommige gevallen door de tijdsdruk te weinig tijd gehad om nog naar meer dan 1 andere alternatief te kijken. Ook merkte ik dat dit enorm belangrijk was tijdens het afstuderen, dat ik mijn besluiten goed kon aantonen en ik miste die in de studie. Hiermee toon ik competenties: SD-8 aan.

#### 8.1.4 Analyse software design documenteren

<b>Situatie</b>	Na het kiezen van de projectaanpak kom ik erachter dat, in tegenstelling tot vorige schoolprojecten, ik met het huidige plan weinig tot geen UML of zware software design documenten zou maken en hiermee mijn competenties voor Software Design en Software Architectuur niet inzichtelijk maakte.
<b>Taak</b>	Het zoeken naar een oplossing, het liefste iets wat aansluit bij de rest of wat ik heb geleerd tijdens de studie waarmee ik wel agile kan blijven ontwikkelen en ik niet teveel tijd steeds kwijt heb aan het digitaliseren van UML diagrammen.
<b>Aanpak</b>	Het gebruik van C4 kwam terug tijdens de software architectuur lessen die precies mijn probleem aankaarten.
<b>Resultaat</b>	Ik heb tijdens de analysefase gebruik gemaakt van C4 waarmee het statisch en dynamisch modelleren is gemaakt door gestructureerd te maken in code.
<b>Reflectie</b>	Ik ben erg tevreden met de diagrammen en het feit dat ik resultaat gericht kan werken en daarnaast een formele manier had om de software design en architectuur uit te leggen aan de verschillende stakeholders.

## BIBLIOGRAFIE

---

- [1] Calum Iain Munro. *Onderzoeksverslag - Blockchain*. 2018.
- [2] Calum Iain Munro. *Plan van Aanpak - Afstudeerproject: Allianz*. 2018.
- [3] Uwe van Heesch. *Software Architecture Course*. 2016.

## HOOFDSTUK 9

### BIJLAGEN

---

# BIJLAGE A

## SMART CONTRACT: CLAIMS

```
1 pragma solidity ^0.4.15;
2
3 contract Claims {
4
5     // Claim struct
6     struct Claim {
7         bytes32 id;
8         string policyId;
9         ClaimTypes claimType;
10        uint amount;
11        State state;
12        bool exists;
13        bytes32[] insurers;
14    }
15
16    struct Vote {
17        bool value;
18        bool hasVoted;
19        bool canVote;
20    }
21
22    enum State {New, Pending, Approved, Rejected, AutoApproved}
23    enum ClaimVote {Approved, Rejected}
24    enum ClaimTypes {NA, GlasDamage, FireDamage, StormDamage, Theft}
25
26    // Who owns this contract
27    address private authority;
28
29    // Claim[] claims;
30
31    mapping(bytes32 => Claim) claims;
32    mapping(bytes32 => mapping(bytes32 => Vote)) claimVotes;
33
34    // Map of claims for addresses
35    mapping(bytes32 => uint) claimsIndexMapping;
36    mapping(address => bytes32[]) addressClaimsMapping;
37
38    event newClaim(bytes32 id, string policyId, ClaimTypes claimType, uint
39                    amount);
40    event claimApproved(bytes32 id);
41    event claimRejected(bytes32 id);
42    event claimAutoApproved(bytes32 id);
43    event logVote(bool value, bool hasVotes, bool canVote);
```

```

43
44     // Construct me
45     function Claims() public {
46         authority = msg.sender;
47     }
48
49     function createClaim(bytes32 id, string policyId, ClaimTypes claimType, uint
50         amount, bytes32[] insurers) public {
51         require(!hasClaim(id));
52         var claim = Claim(
53             id,
54             policyId,
55             claimType,
56             amount,
57             State.New,
58             true,
59             insurers
60         );
61
61         for (uint i = 0; i < insurers.length; i++) {
62             claimVotes[id][insurers[i]] = Vote(false, false, true);
63         }
64
65         claims[id] = claim;
66         addressClaimsMapping[msg.sender].push(id);
67
68         newClaim(id, policyId, claimType, amount);
69         if (shouldAutoApprove(claim)) {
70             claim.state = State.AutoApproved;
71             claims[id] = claim;
72
73             claimAutoApproved(claim.id);
74         }
75     }
76
77     function getClaim(bytes32 id) public constant returns (
78         string policyId,
79         ClaimTypes claimType,
80         uint amount,
81         uint state
82     ) {
83         require(hasClaim(id));
84         var claim = claims[id];
85
86         policyId = claim.policyId;
87         claimType = claim.claimType;
88         amount = claim.amount;
89         state = uint(claim.state);
90     }
91
92     function vote(bytes32 claimId, bytes32 insurerId, bool vote) public {

```

```

93     require(hasClaim(claimId));
94
95     var claim = claims[claimId];
96     claim.state = State.Pending;
97     claims[claimId] = claim;
98
99     var voteObject = claimVotes[claimId][insurerId];
100    require(!voteObject.hasVoted);
101
102    voteObject.value = vote;
103    voteObject.hasVoted = true;
104    claimVotes[claimId][insurerId] = voteObject;
105
106    if (hasVoteFinished(claimId)) {
107        var approved = true;
108        for (uint i = 0; i < claim.insurers.length; i++) {
109            voteObject = claimVotes[claimId][claim.insurers[i]];
110
111            if (!voteObject.value) {
112                approved = false;
113                i = claim.insurers.length + 1;
114            }
115        }
116
117        if (approved) {
118            claim.state == State.Approved;
119            claimApproved(claimId);
120        } else {
121            claim.state == State.Rejected;
122            claimRejected(claimId);
123        }
124
125        claims[claimId] = claim;
126    }
127 }
128
129 function setClaimState(bytes32 id, State state) internal {
130     var claim = claims[id];
131
132     claim.state = state;
133     claims[id] = claim;
134 }
135
136 function shouldAutoApprove(Claim claim) internal constant returns (bool) {
137     return (
138         (claim.claimType == ClaimTypes.Theft && claim.amount < 1000)
139     );
140 }
141
142 function hasClaim(bytes32 id) internal constant returns (bool) {
143     Claim storage claim = claims[id];

```

```
144     return claim.exists == true;
145 }
146
147 function hasVoteFinished(bytes32 claimId) internal constant returns (bool) {
148     var claim = claims[claimId];
149     uint votes = 0;
150     for (uint i = 0; i < claim.insurers.length; i++) {
151         var vote = claimVotes[claimId][claim.insurers[i]];
152
153         if (vote.hasVoted) {
154             votes++;
155         }
156     }
157
158     return votes == claim.insurers.length;
159 }
160 }
```