

Afstudeerplan

Informatie afstudeerder en gastbedrijf

Afstudeerblok: 2015-2.2 (start uiterlijk 16 november 2015)
Startdatum uitvoering afstudeeropdracht: 16 november 2015
Inleverdatum afstudeerdossier volgens jaarrooster: 8 januari 2016

Studentnummer: 10007695
Achternaam: dhr. Brus
Voorletters: A.R.
Roepnaam: Aiken
Adres: Kenyattastraat 10
Postcode: 2622 GK
Woonplaats: Delft
Telefoonnummer: 015-2564677
Mobiel nummer: 06-40962270
Privé emailadres: aikenbrus@hotmail.com

Opleiding: Technische Informatica
Locatie: Delft
Variant: voltijd

Naam studieloopbaanbegeleider: Sjaak van Peski
Naam begeleidend examiner: Cobie van der Hoek
Naam tweede examiner: John Visser

Naam bedrijf: VDG Security B.V.
Afdeling bedrijf: R&D
Bezoekadres bedrijf: Platinastraat 65
Postcode bezoekadres: 2718 SZ
Postbusnummer:
Postcode postbusnummer:
Plaats: Zoetermeer
Telefoon bedrijf: 079 363 81 11
Telefax bedrijf:
Internetsite bedrijf: <https://vdgsecurity.com/>

Achternaam opdrachtgever: dhr. Grijpink
Voorletters opdrachtgever: T.A.
Titulatuur opdrachtgever:
Functie opdrachtgever: bestuurder
Doorkiesnummer opdrachtgever:
Email opdrachtgever:

Achternaam bedrijfsmentor: dhr. Hermann
Voorletters bedrijfsmentor: R
Titulatuur bedrijfsmentor:
Functie bedrijfsmentor: Lead Developer
Doorkiesnummer bedrijfsmentor: 079 3638111

Email bedrijfsmentor: r.hermann@vdgsecurity.com

Doorkiesnummer afstudeerder:

Functie afstudeerder (deeltijd/duaal): Stagiair R&D

Afstudeertitel

De VDG Sense software voorbereiden op de komst van de ARC-bus door middel van een nieuw te ontwikkelen plug-in bij VDG Security B.V.

1 Bedrijf

VDG Security B.V. staat voor Video Development Group Security B.V. VDG maakt innovatieve video management software oplossingen en heeft de applicatie VDG Sense ontwikkeld, dat een Video Management System is. Een Video Management System (VMS) wordt gebruikt om video's te monitoren, te analyseren en op te nemen. De focus van het bedrijf ligt voornamelijk op de verkeer infrastructuur, bouw en gezondheidszorg markt.

VDG Security is onderdeel van het technologie bedrijf TKH Group. De TKH Group bestaat uit een groot aantal bedrijven, met in totaal 5.337 werknemers. Een ander bedrijf dat onder TKH Group valt is iProtect.

De TKH Group bestaat uit drie hoofdgroepen, te weten: Telecom Solutions, Industrial Solution en Building Solutions.

VDG Security is werkzaam binnen Building Solutions. Binnen deze groep wordt nauw samengewerkt met de zuster bedrijven die ook actief zijn in de security branch. Tijdens deze samenwerking word zoveel mogelijk de technologie van de zuster bedrijven gecombineerd wordt met de video technologie van VDG Security.

2 Probleemstelling

Binnen de TKH group opereert een innovatieafdeling (TKH innovations) die zich onder andere bezighoudt met het bedenken van oplossingen waarmee de verschillende TKH ondernemingen en hun producten beter kunnen integreren. Projecten die door de TKH group uitgevoerd worden, bestaan in de meeste gevallen uit een samenstelling van een of meerdere TKH ondernemingen, die elk hun eigen product(en) leveren. Hierbij kan bijvoorbeeld aan een situatie gedacht worden dat VDG een camera-installatie met video analyse en opslag levert en iProtect een toegangscontrole oplossing dat samen moet gaan werken. Beide leveren hun eigen hardware- en software.

Binnen de TKH group is gebleken is dat het een intensieve klus is om de verschillende producten met elkaar te integreren om onder andere de volgende redenen:

- Elke TKH onderneming heeft in het verleden de vrije keuze gehad in welke techniek(en) en protocol(len) zij implementeerden in hun product(en) waarmee een extern systeem kan communiceren.
- De product- en softwareontwikkelingen staan niet stil en daarmee verandert dus ook regelmatig de wijze waarop met een product gecommuniceerd kan worden.

Om de samenwerking/integratie van verschillende producten op middellange termijn te verbeteren, heeft de TKH innovatieafdeling de ARC-bus bedacht. Dit is een softwarebus waarmee het mogelijk gemaakt moet worden dat verschillende producten middels een netwerkverbinding, gegevens met elkaar kunnen uitwisselen op een uniforme manier. Een ontvanger kan zich abonneren op berichten die verstuurd worden over de ARC-bus en is het mogelijk om zelf berichten te versturen waar andere zich weer op kunnen abonneren.

VDG wilt de software voorbereiden op de komst van de ARC-bus en dit integreren in hun interne systeem VDG Sense. De ARC-bus is een nieuw concept en het is voor de VDG dan ook nog niet volledig duidelijk op welke wijze dit gedaan moet worden.

3 Doelstelling

De doelstelling van de afstudeeropdracht is door middel van een onderzoek meer duidelijkheid te krijgen wat de ARC-bus is en kan. Met behulp van dit onderzoek worden requirements geïdentificeerd waarmee de ARC-bus geïntegreerd moet gaan worden in de VDG Sense software. Door middel van een plug-in moeten berichten over de ARC-bus uit gewisseld kunnen worden.

4 Resultaat

Een plug-in die als een middleware laag tussen de VDG Sense software en ARC-bus gaat fungeren. Naast de plug-in komt er een rapport waarin staat beschreven hoe de plug-in gebruikt kan worden. Via deze plug-in zal het mogelijk zijn om berichten over de ARC-bus uit te wisselen.

Aangezien het nog niet duidelijk is of VDG al toegang heeft tot de eerste officiële versie van de ARC-bus voor de afronding van de afstudeeropdracht, dient een simulatie tool gemaakt te worden. Deze simulatie tool dient zich hetzelfde als de ARC-bus te gedragen zodat de plug-in zonder tussenkomst van de officiële ARC-bus, volledig getest kan worden.

5 Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijhorende activiteiten

Systeemontwikkelingsmethode

VDG maakt gebruik van de systeemontwikkelingsmethode Scrumban, maar houden zich niet strikt aan een ontwikkelmethode. Een project opzet binnen VDG zou er als volgt uit kunnen zien: definitiestudie/analyse, basisontwerp, technisch ontwerp/detailontwerp, bouw, testen, integratie en beheer/onderhoud. Het lijkt op een watervalmethode en er wordt een Scrumban board gebruikt om het te visualiseren.

Voor het uitvoeren van de afstudeeropdracht zal ik ook gebruik maken van Scrumban zoals het binnen VDG wordt gebruikt. Verder is het ook de bedoeling dat de plug-in uiteindelijk in productie wordt genomen, nadat het gereviewed is door de senior developers en getest door de test engineer. Bij de fasering van mijlpalen en bijhorende activiteiten is er rekening gehouden met, deze twee punten. De activiteiten zullen worden uitgevoerd volgens de volgorde van de fasering.

Inwerken

Tijdens het inwerken zal ik me voornamelijk bezig houden met het verdiepen in de structuur plus code van VDG Sense en eerder ontwikkelde plug-ins voor VDG Sense. Verder zal ik ook meer ervaring opdoen over de betreffende onderwerpen waar ik niet voldoende kennis over heb. De tijdsduur van het inwerken zal 5 tot 10 dagen zijn.

Plan van Aanpak

Het Plan van Aanpak is bedoeld om een beeld te krijgen van hoe de huidige situatie eruit ziet en wat er allemaal gedaan moet worden voor het project. Ongeveer binnen 2 dagen zal er een Plan van Aanpak worden gemaakt.

Onderzoek

De onderzoeksperiode zal ik gaan gebruiken om te begrijpen hoe de TKH innovatieafdeling de ARC-bus heeft bedacht en hoe ik deze uiteindelijk kan gaan integreren in de software. De onderzoeksperiode zal ook gebruikt worden om de libraries en protocollen te onderzoeken die gebruikt worden in VDG Sense. De tijdsduur voor het onderzoek zal ongeveer 5 dagen zijn

Requirements discipline

Na het onderzoek zal ik in de requirements discipline functionele en niet functionele requirements SMART formuleren die voortgekomen zijn uit een of meerdere nader te bepalen stakeholders. Binnen 4 dagen zal de Requirements rapport worden opgesteld.

Analysis & Design discipline

Op basis van de Requirements discipline zal het Analysis & Design discipline worden gemaakt. De Analysis & Design discipline zal UML klassendiagrammen en sequence diagrammen bevatten. Binnen 4 dagen zal de Analysis & Design rapporty worden opgesteld.

Proof of concept

Er wordt een proof of concept gebouwd van de plug-in en simulatietool om te bewijzen dat de plug-in werd. Deze proof of concetp moet voldoen aan de requirements van de requirements discipline. Binnen 15 dagen zal er een proof of concept zijn gebouwd.

Testen

Er zal gebruik gemaakt moeten worden van tests om te zorgen dat de (kern)functionaliteit correct werkt. Naast de Unit test zal het eindproduct ook getest en gereviewed worden door senior developers van VDG Security B.V.. Voor het testen van het eindproduct zullen test cases geschreven worden. In deze test cases wordt geschreven wat er getest gaat worden en welke stappen gedaan moeten worden om dit te testen. Binnen 5 dagen zal er een testrapport zijn opgesteld.

Bug fixing

Op basis van de bugs die tegen gekomen zijn tijdens het uitvoeren van de testen zal er sprake zijn van bug fixing. Ongeveer 10 dagen zal ik bezig zijn met bug fixing.

Fine tuning

Op basis van de proof of concept zal er sprake zijn van fine tuning. Het fine tuning betreft het refactoren van de proof of concept code en het maken van extra functionaliteiten voor de proof of concept. Ongeveer 10 dagen zal ik bezig zijn met fine tuning.

6 Op te leveren (tussen) producten

- Plan van aanpak
- Onderzoeksrapport van de ARC-bus
- Requirements document
- Ontwerprapport
 - Use-case diagram

- UML klassendiagrammen
- UML sequence diagrammen
- Testrapport
 - Testplan
 - Test cases
 - Test scripts
 - Test log
 - Test resultaten
 - Evaluatie uitgevoerde testen
- Proof of concept
- Eindproduct
- Afstudeerverslag

7 Te demonstreren competenties en wijze waarop

A1 Analyseren van het probleemdomein

Het analyseren van het probleemdomein wordt gedaan door middel van een rapport te schrijven. In dit rapport word de werking van de ARC-bus beschreven en hoe dit gebruikt kan worden door VDG.

A5 Opstellen van systeemeisen (requirements)

Deze beroepstaak wordt uitgevoerd door een onderzoek te doen naar de ARC-bus. Dit onderzoek moet gedaan worden omdat het nog niet volledig duidelijk is hoe de ARC-bus precies werkt. Tijdens het onderzoek wordt onderzocht wat de mogelijkheden zijn van de ARC-bus. Uit dit onderzoek komt een onderzoeksrapport. In samenwerking met de opdrachtgever wordt besproken welke onderdelen van de ARC-bus geïmplementeerd gaan worden. De uitkomst hiervan zal in een requirements document opgenomen worden. Het requirements document zal alle functionele- en niet functionele requirements bevatten die nodig zijn om de plug-in te implementeren. De requirements worden SMART beschreven.

C8 Ontwerpen van een technisch informatie systeem

Het ontwerp wordt gemodelleerd in UML. De requirements worden opgenomen in UML diagrammen. De volgende UML diagrammen worden gemaakt:

- Use-case
- Klassendiagrammen
- Sequence diagrammen

In de ontwerpfase moet al rekening gehouden worden met de testbaarheid van de code en de codestandaarden die binnen VDG gelden. De plug-in komt te draaien in een softwareproduct wat al in productie is en waar dagelijks 13 man aan werken en voor al een aantal jaar. Het ontwerp zal besproken moeten worden met een senior ontwikkelaar voordat begonnen kan worden aan de implementatie.

D17 Testen van software systemen

Testdocumenten zullen gemaakt worden op basis van de requirements die opgesteld zijn. Tijdens het maken van het ontwerp zullen er ontwerpkeuzes gemaakt moeten worden om unit-test te kunnen uitvoeren. In het testdocument worden de testen beschreven. In deze beschrijving staat: wat er getest wordt, hoe dit getest wordt en welke middelen er nodig zijn om de bijbehorende testen uit te kunnen voeren. Voor alle requirements zullen test cases gemaakt worden.

G1 Praktische aspecten hanteren in (internationale) projecten

In het plan van aanpak worden de volgende hoofdstukken uitgewerkt:

- Aanleiding van de opdracht
- Probleemstelling
- Doelstelling
- Resultaten
- Opdrachtomschrijving
- Welke methodiek en waarom deze methodiek
- Aanpak
- Risicoanalyse

Aan het eind van het project kan in het afstudeerverslag gelezen worden of de planning gevolgd

Plan van aanpak

ARC-bus

Bedrijf: VD Security B.V.
Opdrachtgever: Timme Grippink
Plaats: Zoetermeer
Datum: 01-09-2015
Student: Aiken Brus (1007695)
Email student: a.r.brus@student.hhs.nl

Inhoudsopgave

1	Achtergrond	1
1.1	Het bedrijf	1
1.2	De achtergrond van het project	1
2	De projectopdracht	2
2.1	De aanleiding	2
2.2	De probleemstelling	2
2.3	De doelstelling	2
2.4	De resultaten	2
3	De projectactiviteiten	3
4	De grenzen van het project	4
5	De producten	5
6	Kwaliteit van het onderzoek/project	6
6.1	De methoden	6
6.2	Het plan van aanpak	8
6.3	Het onderzoeksrapport van de ARC-bus	9
6.4	Het ontwerp	9
6.5	Het testrapport	9
6.6	Het eindproduct	9
6.7	Het afstudeerverslag	9
7	De organisatie van het project	10
7.1	De organisatie	10
7.2	Informatie	10
8	De planning	11
9	De kosten en baten	12
9.1	De personeelskosten	12
9.2	De facilitaire kosten	12
9.3	De totale kosten	12
9.4	De baten	12
10	De risico's van het project	13
10.1	De interne risico's	13
10.2	De externe risico's	13
10.3	Conclusie	13
	Literatuurlijst	14

1 Achtergrond

In dit hoofdstuk wordt beschreven bij welke organisatie het project wordt uitgevoerd en wat de organisatie doet. In de tweede paragraaf wordt de achtergrond van het project beschreven.

1.1 Het bedrijf

VDG Security B.V. staat voor Video Development Group Security B.V.. VDG maakt innovatieve video management software oplossingen en heeft de applicatie VDG Sense ontwikkeld, dit is een Video Management System. Een Video Management System (VMS) wordt gebruikt om video's te monitoren, te analyseren en op te nemen. De focus van het bedrijf ligt voornamelijk op de verkeer infrastructuur, bouw en gezondheidszorg markt.

VDG Security is onderdeel van het technologie bedrijf TKH Group. De TKH Group bestaat uit een groot aantal bedrijven, met in totaal 5.337 werknemers. Een ander bedrijf dat onder TKH Group valt is iProtect. De TKH Group bestaat uit de volgende drie hoofdgroepen: Telecom Solutions, Industrial Solution en Building Solutions. VDG Security is werkzaam binnen Building Solutions. Binnen deze groep wordt nauw samen gewerkt met de zuster bedrijven die ook actief zijn in de security branch. Tijdens deze samenwerking wordt zoveel mogelijk de technologie van de zuster bedrijven gecombineerd met de video technologie van VDG Security. Het project dat wordt uitgevoerd binnen het bedrijf VDG Security B.V.. VDG bestaand uit verschillende afdelingen. Deze zijn:

- Research and Development
- Support
- Verkoop
- Magazijn

Het project is onderdeel van Research and Development. In deze afdeling werken elf mensen. Deze afdeling is onderverdeeld in een groep van server developers en client developers.

1.2 De achtergrond van het project

De opdrachtgever voor het project is de directeur van VDG Security Timme Grijpink. Dit project is niet een vervolg van een al eerder uitgevoerd project. Het uiteindelijk product gaat wel toegevoegd worden aan een product van VDG Security B.V. dat wordt doorontwikkeld in dezelfde periode als dit project. Op dit moment is er nog geen ARC-bus, dus moet voor elk device waar VDG Sense mee wil communiceren een speciaal ontworpen plug-in ontwikkeld worden.

2 De projectopdracht

In dit hoofdstuk wordt het probleemdomain beschreven en wat het doel is van het project.

2.1 De aanleiding

Binnen de TKH group opereert een innovatieafdeling (TKH innovations). Deze houdt zich, onder andere, bezig met het bedenken van oplossingen waarmee de verschillende TKH ondernemingen en hun producten beter kunnen integreren. Projecten die door de TKH group uitgevoerd worden, bestaan in de meeste gevallen uit een samenstelling van één of meerdere TKH ondernemingen, die elk hun eigen product(en) leveren. Hierbij kan bijvoorbeeld aan een situatie gedacht worden waarbij VDG een camera-installatie met video analyse en opslag levert en KeyProcessor een toegangscontrole oplossing. Deze twee producten zullen vervolgens samen moeten gaan werken. Beide zijn een TKH onderneming die hun eigen hardware en software leveren.

Uit het verleden is gebleken dat het een intensieve klus is om de verschillende producten met elkaar te integreren. Hierbij komen een tweetal problemen met name naar voren:

- Elke TKH onderneming heeft in het verleden de vrije keuze gehad in welke techniek(en) en protocol(len) zij implementeerden in hun product(en) waarmee een extern systeem kan communiceren.
- De product- en softwareontwikkelingen staan niet stil en daarmee verandert dus ook regelmatig de wijze waarop met een product gecommuniceerd kan worden.

2.2 De probleemstelling

Om de samenwerking/integratie van verschillende producten op de middellange termijn te verbeteren, heeft de TKH innovatieafdeling de ARC-bus bedacht. Dit is een softwarebus waarmee het mogelijk gemaakt moet worden dat verschillende producten middels een netwerkverbinding, gegevens met elkaar kunnen uitwisselen op een uniforme manier. Een ontvanger kan zich abonneren op berichten die verstuurd worden over de ARC-bus en het is mogelijk om zelf berichten te versturen waar andere zich weer op kunnen abonneren. VDG wilt de software voorbereiden op de komst van de ARC-bus en dit integreren in hun interne systeem VDG Sense. De ARC-bus is een nieuw concept en het is voor de VDG dan ook nog niet volledig duidelijk op welke wijze dit gedaan moet worden.

2.3 De doelstelling

De doelstelling van de afstudeeropdracht is door middel van een onderzoek meer duidelijkheid te krijgen wat de ARC-bus is en wat deze kan. Met behulp van dit onderzoek worden requirements geïdentificeerd waarmee de ARC-bus geïntegreerd moet gaan worden in de VDG Sense software. Door middel van een plug-in moeten berichten over de ARC-bus uit gewisseld kunnen worden.

2.4 De resultaten

Een van de resultaten is een plug-in die als een middleware laag tussen de VDG Sense software en ARC-bus gaat fungeren. Naast de plug-in komt er een rapport waarin staat beschreven hoe de plug-in gebruikt kan worden. Via deze plug-in zal het mogelijk zijn om berichten over de ARC-bus uit te wisselen.

3 De projectactiviteiten

In hoofdstuk 3 worden alle activiteiten genoemd waar aan gewerkt kan worden. Al deze activiteiten zijn onderverdeeld in de fases waarin ze uitgevoerd gaan worden. Het maken van het afstudeerverslag staat buiten de fases omdat daar in de loop van het project aan gewerkt gaat worden.

De analysefase

- Inwerken
- Plan van aanpak schrijven
- Onderzoek naar ARC-bus
- ‘Requirements’ SMART opstellen
- ‘Proof of concept’ ontwikkelen

De ontwerpfase

- Ontwerp moduleren
- Testdocument schrijven

De ontwikkelfase

- Simulatie tool ontwikkelen
- Eindproduct ontwikkelen

De testfase

- Eindproduct testen

Afstudeerverslag maken

4 De grenzen van het project

In dit hoofdstuk word er gekeken naar de grenzen van het project. Dit project gaat uitgevoerd worden in 20 weken in de vorm van een afstudeerstage. Het begint op 14-08-2015 en eindigt op 29-01-2015. Er word 40 uur per week gewerkt aan dit project. Van uit de Haagse Hogeschool zijn er ook nog een paar deadlines. In week 8 of 9 moet er een voortgangsverslag ingeleverd worden. In week 12 moet er een concept verslag ingeleverd worden. School stelt ook nog een paar eisen aan het afstudeerproject.

De eisen vanuit school zijn:

- Het project moet hbo niveau zijn
- Het moet een onderzoek onderdeel hebben
- Het moet een implementatie onderdeel hebben
- Er moeten een bepaald aantal beroepstaken uitgevoerd worden.

De beroepstaken die worden uitgevoerd zijn:

- A1 Analyseren van het probleemdomein
- A5 Opstellen van systeemeisen (requirements)
- C8 Ontwerpen van een technisch informatie systeem
- D17 Testen van software systemen
- G1 Praktische aspecten hanteren in (internationale) projecten

5 De producten

In de opsomming staan alle producten die gemaakt en meegeleverd gaan worden met het afstudeerverslag.

- Plan van aanpak
- Onderzoeksrapport van de ARC-bus
- Requirements document
- Ontwerprapport
 - Use-case diagram
 - UML klassendiagrammen
 - UML sequence diagrammen
- Testrapport
 - Testplan
 - Test cases
 - Test scripts
 - Test resultaten
 - Evaluatie uitgevoerde testen
- Simulatie tool
- Eindproduct
- Afstudeerverslag

6 Kwaliteit van het onderzoek/project

In dit hoofdstuk word beschreven hoe de kwaliteit van de (tussen)producten gewaarborgd kunnen worden. De kwaliteit wordt gewaarborgd door verschillende eisen te stellen aan de tussenproducten en het eindproduct.

6.1 De methoden

Om een project zo goed mogelijk uit te voeren is er gebruik gemaakt van een ontwikkelmethode. In dit hoofdstuk wordt uitgelegd welke ontwikkelmethode is gekozen en waarom deze is gekozen.

VDG maakt gebruik van de systeemontwikkelingsmethode Scrumban. Een projectopzet binnen VDG zou er als volgt uit kunnen zien: definitiestudie/analyse, basisontwerp, technisch ontwerp/detailontwerp, bouw, testen, integratie en beheer/onderhoud. Het lijkt op een watervalmethode en er wordt een Scrumban board gebruikt om het te visualiseren. Deze ontwikkelmethode zal ook worden gebruikt tijdens het project. Elke dag zal er een stand-up meeting gehouden worden met alle ontwikkelaars van VDG. Tijdens deze stand-up wordt besproken wat je de dag er voor hebt gedaan, wat je gaat doen vandaag en of je nog blokkades hebt.

Om te kijken welke ontwikkelmethode het beste is voor dit project zijn er eisen opgesteld waar de ontwikkelmethode aan moet voldoen. De eisen zijn opgesteld door te bestuderen wat de kenmerken van het project zijn. Tijdens het project worden er verschillende tussenproducten opgeleverd. De plug-in die ontwikkeld moet worden, moet geïntegreerd worden in de al bestaande software Sense. Het is nog niet zeker wat de ARC-bus allemaal kan. Hierdoor moet voor dit project een onderzoek gedaan worden naar de ARC-bus. De tussenproducten moeten getest worden. Het project duurt 20 weken. De eisen voor de ontwikkelmethode zijn:

- Het moet een softwareontwikkelmethode zijn.
- Er moet een mogelijkheid zijn om de product requirements tussentijds te kunnen aanpassen omdat het nog onzeker is hoe de ARC-bus gebruikt kan worden.
- De ontwikkel methode moet kunnen werken binnen de 20 weken die voor het project staan.
- Er moet een onderzoek element in zitten.

Tijdens het opstellen van de eisen is er gekeken naar de verschillende ontwikkelmethodes die bestaan en al bekend zijn bij de afstudeerder. Deze ontwikkelmethodes zijn:

- Rational Unified Process (RUP)
- De waterval methode
- Extreme programming (XP)
- Scrum
- Scrumban

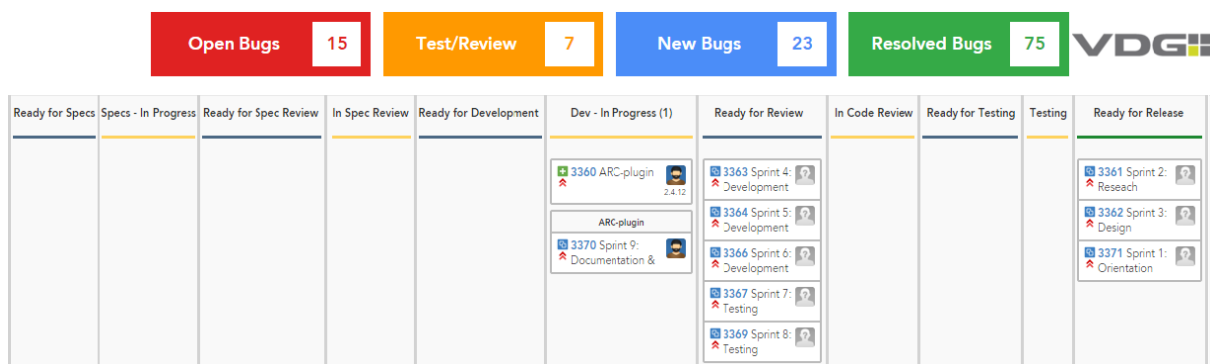
RUP is een ontwikkel methode voor systeemontwikkeling. Het maakt gebruik van iteraties en verschillende fases die je uitvoert in elke iteratie. Na elke iteratie is er een mijlpaal waar een onderdeel van het project af is. RUP is heel erg uitgebreid en is beter te gebruiken bij grote en langdurige projecten omdat RUP uitgaat van verschillende rollen binnen een project. RUP heeft voor verschillende onderdelen een 'best practice'. RUP maakt gebruik van vier fasen; inception, elaboration, construction en transition. Elk van deze fasen kunnen onderverdeeld worden in één of meer iteraties. Om RUP te kunnen gebruiken voor dit project moet het eerst aangepast worden. In dit project kunnen sommige milestone documenten vergeten worden. In de inception fase kan de business case over geslagen worden. De business case is een document waarin beschreven wordt waarom het systeem gemaakt moet worden, wat het systeem moet gaan doen en er komt een financieel overzicht in. Deze eerder genoemde onderdelen komen allemaal in het plan van aanpak te staan. Ook wordt het vision document niet gemaakt. In het vision document staat beschreven vanuit de ogen van de klant wat het product moet gaan doen en wat de risico's voor hun zijn. De prototypes van het architectuur kunnen nog niet gemaakt worden omdat er eerst onderzoek gedaan moet worden naar de ARC-bus. Tijdens de transition fase gaat er geen training gegeven worden over de werking van de plug-in. Het product wordt ook nog niet gedeployed. De verschillende disciplines worden in elke iteratie gedaan daar door kunnen de eisen van het project in elke iteratie worden aangepast.

De waterval methode is eenvoudig te gebruiken en te plannen. Alle eisen moeten in het begin bekend zijn en er is geen mogelijkheid om deze aan te passen later in het project. De waterval methode is ook niet iteratief.

Extreme programming is een iteratief ontwikkel methode. Tijdens deze ontwikkel methode is het mogelijk om de eisen aan te passen. Extreme programming heeft ook geen specificaties voor het ontwerpen van hardware.

Scrum maakt gebruik van sprints. Een sprint is een periode van een maand of minder en in deze periode wordt er aan een onderdeel van het project gewerkt dat aan het eind van de sprint af moet zijn. Scrum maakt gebruik van stand-up meetings. Dit betekent dat elke week of dag wordt besproken wat er gedaan is, wat er gedaan gaat worden en of er nog problemen ontstaan zijn. Tijdens scrum is het mogelijk om de eisen van het systeem aan te passen in het begin van elke sprint.

Scrumban is vergelijkbaar met Scrum. Dat betekent dat het ook gebruik maakt van sprints en stand-up meetings. Naast de Scrum onderdelen maakt het ook gebruik van een 'kanban board'. Een 'kanban board' wordt gebruikt om alle statussen van de activiteiten bij te houden en dit zichtbaar te maken. figure 1 is de door VDG Security B.V. zelf ontwikkelde 'kanban board'.

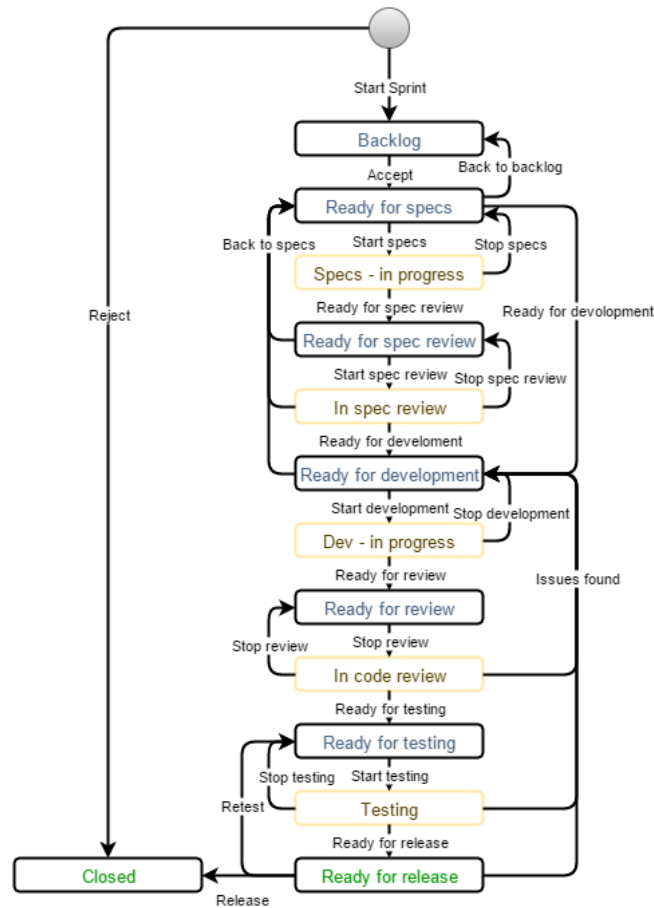


Figuur 1: Kanban board VDG Security B.V.[1]

De waterval methode valt af omdat deze niet flexibel is, aan het begin van de ontwikkeling moeten alle eisen bekend zijn en eerder kwam naar voren dat de eisen tussentijds aangepast moeten kunnen worden tijdens dit project. Er is geen mogelijkheid bij de waterval methode om een paar stappen terug te gaan en de eisen opnieuw te gaan verwoorden. Hiermee is deze methode dus niet geschikt. Extreme programming valt ook af omdat hier niet genoeg kennis van is. RUP, Scrum en Scrumban zijn alle drie goede mogelijkheden. Uiteindelijk is er gekozen om Scrumban te gebruiken want dit wordt ook gebruikt door VDG. Het project is onderverdeeld in negen sprints. Er is voor negen sprints gekozen omdat aan het eind van elke sprint tussen producten die onder de zelfde discipline vallen kan opleveren. Er is gekozen om de simulatie tool, C++ SDK en de plug-in in aparte sprints te doen omdat deze producten te groot zijn voor één sprint. Elke sprint is twee weken lang. Elke sprint volgt apart de workflow van figure 2 van start tot closed. In de backlog wordt van op geschreven wat er in de sprints gedaan gaat worden. Tijdens specs wordt worden de specificaties bepaald maar er wordt ook naar de al eerder gedachten specificaties gekeken. In spec review worden de specificaties door de bedrijfsbegeleider door genomen. Tijdens de stap Dev - in progress worden de tussenproducten van de sprint ontwikkelt. In code review wordt de gemaakte code gereviewd door de bedrijfsbegeleider. Bij de stap testing worden de tussenproducten getest. wanneer alles gedaan is komt het in ready for release. De workflow wordt ook gebruikt door het bedrijf.

Tijdens dit project worden er een paar standaard scrumban specificaties niet gebruikt. Er is geen WIP (work in progress) limit. WIP is de hoeveelheid taken die per activiteit gedaan kan worden. Er is geen WIP omdat het project door één persoon gedaan wordt en een persoon kan maar aan één taak tegelijkertijd werken. Scrumban kent twee verschillende meetings de dagelijkse stand-up en wekelijkse timeblock. In de dagelijkse stand-up verteld elke ontwikkelaar wat er de vorige dag gebeurt is, wat er vandaag gedaan gaat worden en of er nog blokkades zijn. De VDG Security developers hebben een retrospectieve na elke release.

In de eerste sprint is een oriëntatie sprint. In deze sprint wordt er ingewerkt en wordt het plan van aanpak gemaakt. In de tweede sprint wordt het onderzoek gedaan. Tijdens dit onderzoek wordt ook een 'proof of concept' ontwikkeld. Deze proof of concept wordt gebruikt om te onderzoeken wat de functionaliteit van de



Figuur 2: workflow sprints[Jira]

SDK is en hoe het is opgebouwd. SDK is een software development kit. In deze kit zitten de meeste hulpmiddelen om een programma te kunnen schrijven. Dit onderzoek geeft een onderzoeks-document als resultaat. In dit document staat beschreven waar de ARC-bus uit bestaat, hoe dit gebruikt kan worden en dit ook bruikbaar gaat zijn voor VDG Security B.V. Na het onderzoek van de ARC-bus is er een lijst met mogelijkheden van de ARC-bus deze mogelijkheden kunnen omgezet worden naar requirements voor de plug-in. In deze sprint wordt daarom ook nog een requirements document gemaakt. De derde sprint is de ontwerp sprint. Tijdens deze sprint wordt een design van de plug-in gemaakt. Na het ontwerpen wordt een testdocument geschreven. Vanaf sprint 4 word er ontwikkeld. Als eerste wordt er in sprint 4 een simulatie tool ontwikkeld. In dit testdocument komt te staan welke tests uitgevoerd gaan worden. Deze simulatie tool gaat alle onderdelen van de ARC-bus geïntegreerd krijgen. In sprint 5 en 6 gaat de plug-in ontwikkeld worden. Tegelijkertijd met het ontwikkelen gaan er unit test uitgevoerd worden. Sprint 7 gaat gebruikt worden om het hele product te testen met en zonder de simulatie tools. Na de test sprint kunnen er bugs te voorschijn zijn gekomen. In sprint 8 worden deze bugs weg ontwikkeld. De laatste Sprint is sprint 9, deze wordt gebruikt om aan het eindverslag te werken. Tijdens het project word er elke week aan het eindverslag gewerkt. In het volgende hoofdstuk zullen alle sprints uitgebreid toegelicht worden en beschreven worden hoe deze ingepast worden in dit onderzoek.

6.2 Het plan van aanpak

In het plan van aanpak wordt beschreven hoe het project aangepakt gaat worden en welke activiteiten uitgevoerd moeten gaan worden om tot een goed eindproduct te komen. De kwaliteit van het plan van aanpak wordt gewaarborgd door de checklist te gebruiken van Roel Grit[2]. In die checklist staat welke hoofdstukken in een plan van aanpak moeten en welke punten in de hoofdstukken besproken moeten worden om een goed plan van aanpak te maken. Het plan van aanpak zal gelezen worden door de begeleider van het project.

6.3 Het onderzoeksrapport van de ARC-bus

Voordat het onderzoek begint, wordt eerst met het bedrijf afgesproken welke onderdelen onderzocht moeten worden. Er is communicatie mogelijkheid met de developers van de ARC-bus. Door deze communicatie kunnen er nog extra vragen gesteld worden over de onderdelen indien die niet helemaal duidelijk zijn na het lezen van de documentatie. De ARC-bus kan fysiek getest en bekeken worden. Daarnaast is er ook nog documentatie over de ARC-bus aanwezig. Ook kan er naar de source code van de ARC-bus gekeken worden. Tijdens het onderzoek word er een proof of concept ontwikkeld. De proof of concept wordt gebruikt om de werking van de SDK te kunnen onderzoeken.

6.4 Het ontwerp

Voor het ontwerp wordt UML gebruikt. UML diagrammen worden gebruikt voor de architectuur en om te bepalen welke onderdelen van het eindproduct ontwikkelt moeten worden. De kwaliteit van de UML diagrammen worden gewaarborgd door de UML standaard te gebruiken. De benamingen van de klassen, functies en variabelen moeten hetzelfde zijn in de analyse-diagrammen en de ontwerp-diagrammen. De UML diagrammen moeten voldoen aan UML 2.5 standaard.

6.5 Het testrapport

Om te kijken of het eindproduct werkt volgens de requirements die worden opgesteld in de analysefase wordt het eindproduct getest. De tests die uitgevoerd gaan worden, worden uitbundig beschreven in het testrapport. Dit testrapport wordt onderverdeeld is een paar tussen onderdelen. Het eerste onderdeel is een beschrijving van wat het doel is van het testen, wat voor test er uitgevoerd gaat worden en welke middelen nodig zijn om de test te kunnen uitvoeren. Het tweede onderdeel zijn alle tests stap voor stap uitgeschreven. Voor elke test wordt de verwachte uitkomst opgeschreven zodat aan het eind van het testen vergeleken kan worden of de daadwerkelijke uitkomst klopt. Als er nog scripts worden geschreven voor het automatisch testen worden deze als bijlage toegevoegd aan het testrapport. Als laatste wordt de verwachte uitkomst en de daadwerkelijk uitkomst vergeleken met elkaar en daar komt een conclusie uit. In de conclusie staat wat nog verder getest moet worden en welk onderdelen nog verder ontwikkeld moeten worden.

6.6 Het eindproduct

Het eindproduct is het product dat opgeleverd gaat worden aan VDG. De eisen waar het aan moet voldoen, worden in de analysefase gedefinieerd. Deze eisen kunnen in de loop van het project aangepast worden als dat nodig is. De kwaliteit van het eindproduct wordt gewaarborgd door Google codestandaarden te gebruiken[3]. Het eindproduct word getest zodat er gezien kan worden of het product werkt volgens de eisen. Dit testen gebeurt aan de hand van de eerder gemaakte testrapport. Om de code aanpassingen overzichtelijk te houden wordt er gebruik gemaakt van een Git repository genaamd Bitbucket [4]. Git word ook gelijk gebruikt als versie beheer.

6.7 Het afstudeerverslag

Het afstudeerverslag is een verslag dat gemaakt wordt voor school. In dit verslag wordt beschreven wat er tijdens de afstudeerstage is uitgevoerd, waarom en welke keuzes zijn gemaakt tijdens het project. De opbouw van het verslag is afkomstig van het document Criteria procesverslag[5]. Het afstudeerverslag zal niet op het eind in een keer gemaakt worden. Tijdens het project zal er elke twee weken een dag aan het verslag gewerkt worden. Om de maand zal dit verslag doorgelezen worden door een extern persoon. Deze externe persoon zal de grammaticale fouten uit het document halen.

7 De organisatie van het project

In het hoofdstuk Projectorganisatie word een stukje geschreven over de personen die aan het project werken en de begeleiders. Daarnaast wordt er ook nog een stukje geschreven over de communicatie tussen het project lid, de begeleiders en de opdrachtgever.

7.1 De organisatie

Dit project wordt uitgevoerd door één projectlid. De naam van het projectlid is Aiken Brus. Naast het projectlid zijn er ook nog twee examinatoren van de Haagse Hogeschool betrokken bij dit project. De eerste examinerator is Cobie van der Hoek en de tweede is John Visser. Omdat dit een afstudeerstage is, is er ook een bedrijfsmentor vanuit VDG betrokken. De bedrijfsmentor is Robin Hermann. In tabel 1 staat de persoonsinformatie.

Naam	Adres	Telefoonnummer	e-mailadres
Aiken Brus	Kenyattstraat 10, Delft	06-40962270	a.r.brus@student.hhs.nl
Cobie van der Hoek	Rotterdamseweg 137, Delft lokaal 2.016	070-4458888	J.J.vanderHoek@hhs.n
John Visser	Rotterdamseweg 137, Delft lokaal 2.012	015-2606281	J.J.Visser@hhs.nl
Robin Hermann	Platinastraat 65, Zoetermeer	079-3638111	r.hermann@vdgsecurity.com
Timme Grijpink	Platinastraat 65, Zoetermeer	079-3638111	t.grijpink@vdgsecurity.com

Tabel 1: persoons-informatie

7.2 Informatie

De opdrachtgever is de directeur van VDG. Doordat dit project op de locatie van VDG wordt uitgevoerd is er geen probleem met communicatie tussen de opdrachtgever en het project lid. Er word een urenregistratie bijgehouden. Elke dag is er een stand-up meeting met alle developers van VDG. Hierin wordt besproken wat er de dag ervoor is gedaan, wat er die dag gedaan gaat worden en of er nog blokkades zijn.

8 De planning

De planning is in de vorm van een strokenplanning. In deze planning staat in welke weken aan welke activiteit gewerkt gaat worden. De strokenplanning kan je vinden als Bijlage 1.

9 De kosten en baten

Tijdens een project is het belangrijk om de kosten en baten op te schrijven, omdat als het project te veel geld gaat kosten moet er misschien een overweging gemaakt worden om het project toch niet te doen.

9.1 De personeelskosten

Omdat dit project een stage opdracht is, wordt de opdrachtnemer een stagiaire loon gegeven. Dit bedraagt 350 euro per maand, Dat is 1750 euro in totaal.

9.2 De facilitaire kosten

Bij dit project wordt er gebruik gemaakt van faciliteiten die al in bezit zijn van het bedrijf waardoor deze kosten nul bedragen.

9.3 De totale kosten

De totale kosten die gemaakt worden tijdens dit project bedraagt 1750 euro. Dit zijn alleen de personeelskosten.

9.4 De baten

Als het project na de 20 weken af is dan kan de opdrachtgever de plug-in verder gaan ontwikkelen en in de toekomst is er dan een betere communicatiemiddel tussen de diverse producten binnen de TKH group.

10 De risico's van het project

Om een project te beginnen is het handig om de risico's op te stellen, want als deze te groot zijn is het een mogelijkheid om het project te stoppen voordat het geld gaat kosten. De kans en ernst word bepaald door een cijfer van een tot vijf. Het Effect wordt bepaald door een cijfer van een tot tien. Waarbij een het laagst is en vijf of tien het hoogst.

10.1 De interne risico's

De interne risico's zijn te vinden in tabel 2.

Risico	Kans	Ernst	Effect	Maatregel
Er is niet genoeg kennis over de protocollen die worden gebruikt door VDG	2	1	2	Beter gaan inlezen over de protocollen
Deadlines voor tussenproducten worden niet gehaald	3	2	4	Zorgen voor een goede planning en de voortgang van het project regelmatig controleren.
3	2	6	Goede kwaliteit standaards opstellen en controleren of deze gevolgd worden.	

Tabel 2: Interne risico's

10.2 De externe risico's

De externe risico's zijn te vinden in tabel 3.

Risico	Kans	Ernst	Effect	Maatregel
De documentatie van de ARC-bus is niet voldoende om daar een plug-in voor te kunnen ontwikkelen	2	3	8	Er is een mogelijkheid tot communicatie met de developers van de ARC-bus. Aan de developers kunnen genoeg vragen gesteld worden.
De ARC-bus werkt niet	2	5	10	Doormiddel van de documentatie word er een simulatie tool ontwikkeld. Deze simulatie tool kan gebruik worden in plaats van de ARC-bus.

Tabel 3: Externe risico's

10.3 Conclusie

Hoewel er een aantal risico's zijn, is het niet nodig om het project stop te zetten. Mocht er toch een groot probleem ontstaan, dan wordt de opdrachtgever direct geïnformeerd.

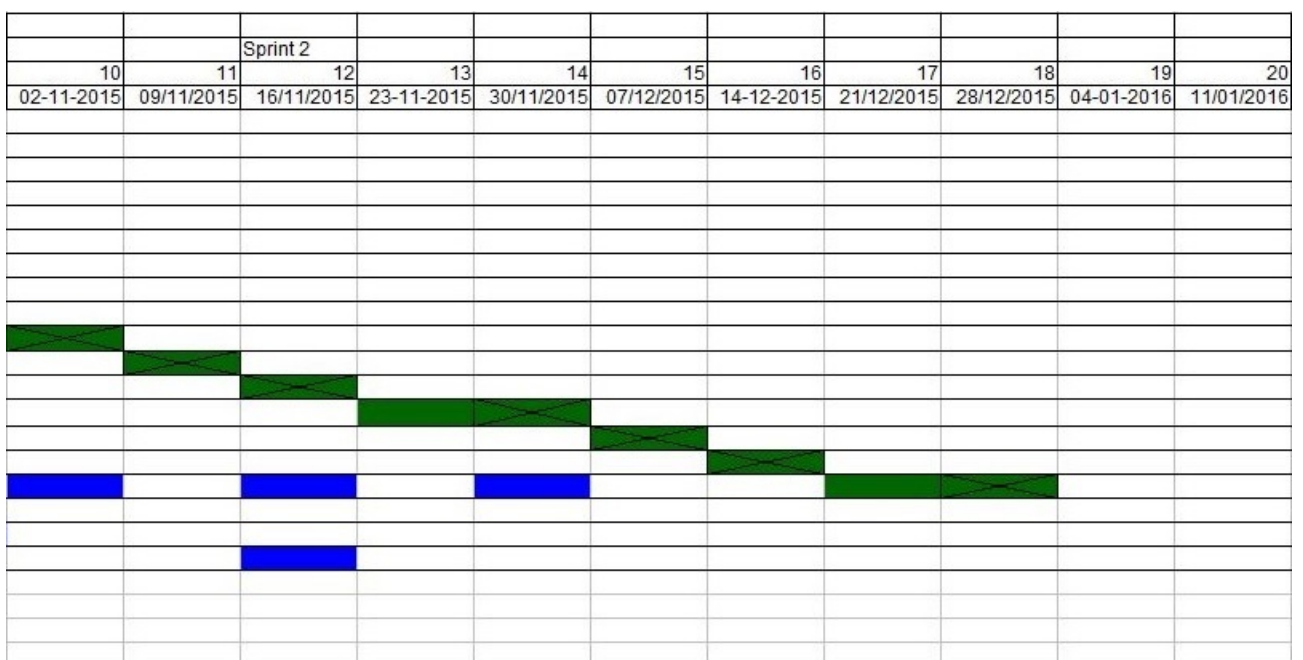
Literatuurlijst

- [1] VDG. *VDG Security B.V.* URL: <https://vdgsecurity.atlassian.net/> (bezocht op 21-09-2015).
- [2] Roel Grit. *Projectmanagement: Projectmatig Werken in De Praktijk*. 2008. URL: http://hoadd.noordhoff.nl/sites/7720/_assets/7720d24.pdf (bezocht op 07-09-2015).
- [3] Google. *Google C++ Style Guide*. URL: https://google-styleguide.googlecode.com/svn/trunk/cppguide.html#Ownership_and_Smart_Pointers (bezocht op 07-09-2015).
- [4] Atlassian. *Bitbucket*. URL: <https://bitbucket.org/> (bezocht op 07-09-2015).
- [5] h de Vreught en E van der weijden blast. *Criteria Procesverslag Team TI Den Haag versie 2*. HHS, 2013.

Bijlage



Figuur 3: Strokenplanning part 1



Figuur 4: Strokenplanning part 2

ARC-bus onderzoek

Student: Aiken Brus (10007695)

Datum: 21-03-2016

Plaats: Zoetermeer

Afstudeerverslag Haagse Hogeschool

Opleiding: Technische informatica

Bedrijf: VD Security B.V.

Opdrachtgever: Timme Grippink

Begeleidende examiner: Cobie van der Hoek

Tweede examiner: John Visser

Inhoudsopgave

1	Inleiding	1
2	ARC-Bus	2
3	DeviceRegistry	4
4	SDK	6
5	Toekomst	8
6	Conclusie	9
	Literatuurlijst	10
	Bijlage	11
7.1	ARC-RPC	11
	Request message	11
	Event message	11
	Reply message	11
	Error message	11
	Device Registry Messages	13
	Register device	13
	Device unregister	13
	Device update	14
	Get all devices	15

1 Inleiding

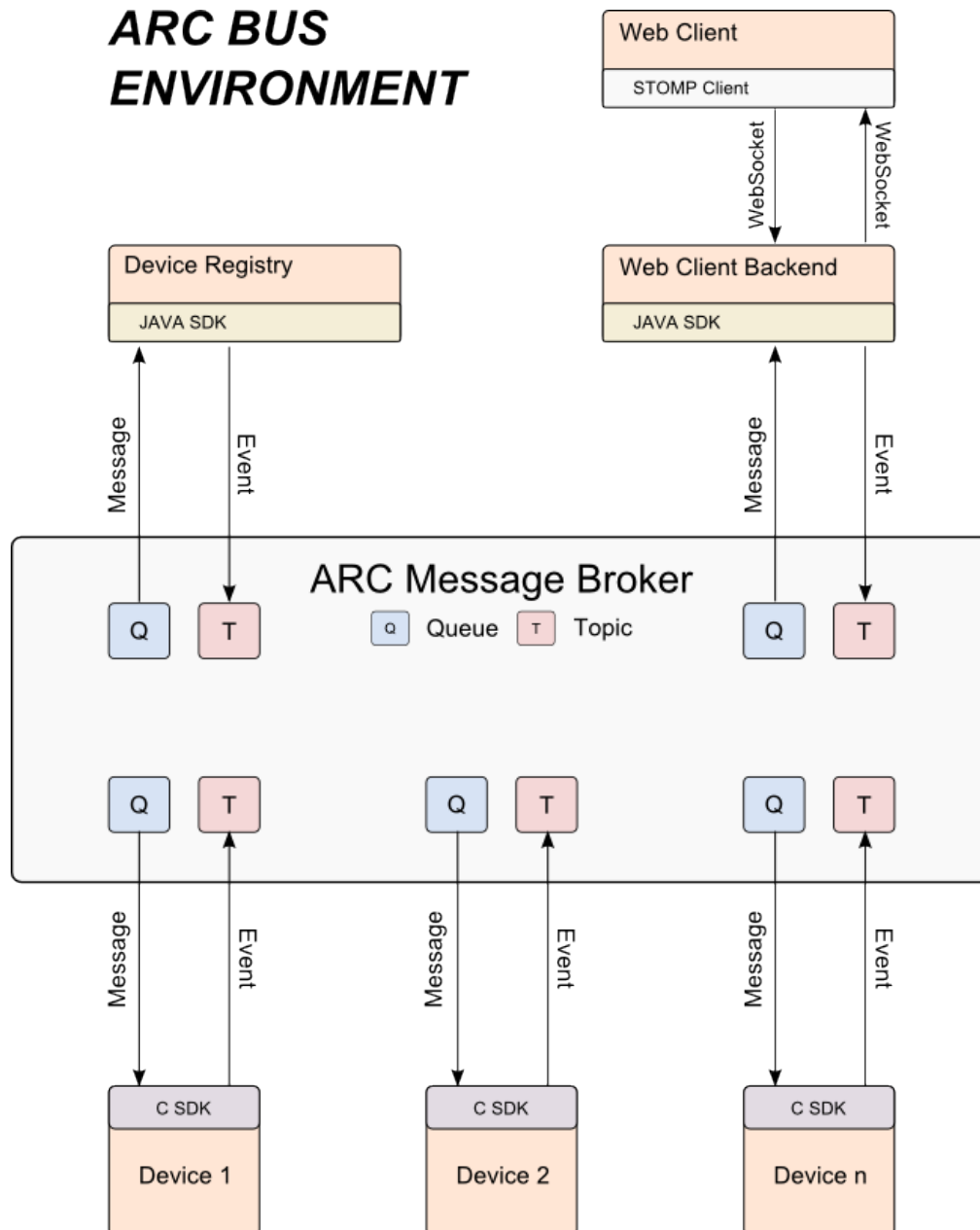
Dit document is een bijlage van het eindverslag ARC-bus plug-in. In dit document wordt het onderzoek van de ARC-bus beschreven. Dit onderzoek is gedaan om te kijken of de ARC-bus een optie is om te gebruiken voor de communicatie tussen verschillende devices en servers. Om aan een antwoord te komen voor dit onderzoek zijn er een paar punten onderzocht wat er allemaal onderzocht zou moeten worden. Deze punten zijn opgesteld door te overleggen met de begeleider. Deze zijn:

- Wat is de ARC-bus?
- Welke protocol wordt er gebruikt?
- Uit welke onderdelen bestaat de ARC-bus
- Hoe kan de ARC-bus gebruikt worden
- Wat zijn de toekomst plannen voor de ARC-bus
- Kan de ARC-bus gebruikt worden door VDG Security B.V.

Tijdens dit onderzoek wordt er gebruik gemaakt van de documentatie van de ARC-bus. Ook wordt er gekeken naar verschillende source code om te kijken hoe dit gebruikt kan worden. Er is ook nog een kans op een interview met de lead developer van de ARC-bus.

2 ARC-Bus

De ARC-bus is gemaakt door de afdeling TKH-innovations binnen TKH group. De bedoeling van de ARC-bus is om de communicatie tussen de producten binnen de TKH group te verbeteren. Bij de huidige situatie wordt voor elk product een eigen plug-in gemaakt. Dit wordt gedaan omdat de bedrijven verschillende protocollen gebruiken in hun producten. De bedoeling van de ARC-bus is om berichten van de een device naar de andere te krijgen. Het idee is om alle devices te connecten via een ARC-bus. In figuur 1 is een tekening hoe dit er schematisch uit komt te zien. Met dit idee hoeven de berichten niet meer apart verstuurd te worden naar elk device die het nodig heeft, maar kan het naar de bus gestuurd worden en die regelt dat het bij de goede eindbestemming komt.



Figuur 1: ARC-bus environment[1]

De ARC-bus bestaat uit drie verschillende onderdelen. De eerste is de ARC-bus backbone of wel message broker. De message broker wordt uitgevoerd op één server binnen een netwerk.

De bedoeling van de message broker is om berichten die hij heeft gekregen van cliënt of server door te sturen naar een andere cliënt of server die dit bericht hoort te ontvangen. Als de berichten niet verstuurd kunnen worden dan moet de verzendende cliënt een notificatie krijgen. De clients sturen een bericht naar de message broker. Als de message broker een bericht binnen krijgt wordt bepaald door middel van de zender en aan de hand van de registraties waar het bericht naar toe gaat.

Voor de backbone is er ActiveMQ gekozen. ActiveMQ is een open source message broker geschreven in Java. Een paar specificaties van ActiveMQ zijn:

- Cross platform
- Cross language voor clients
 - Java
 - C
 - C++
 - C#
 - Ruby
 - Perl
 - Python
 - PHP
- ondersteund vier verschillende protocols
 - Openwire
 - Stomp
 - AMQP
 - MQTT

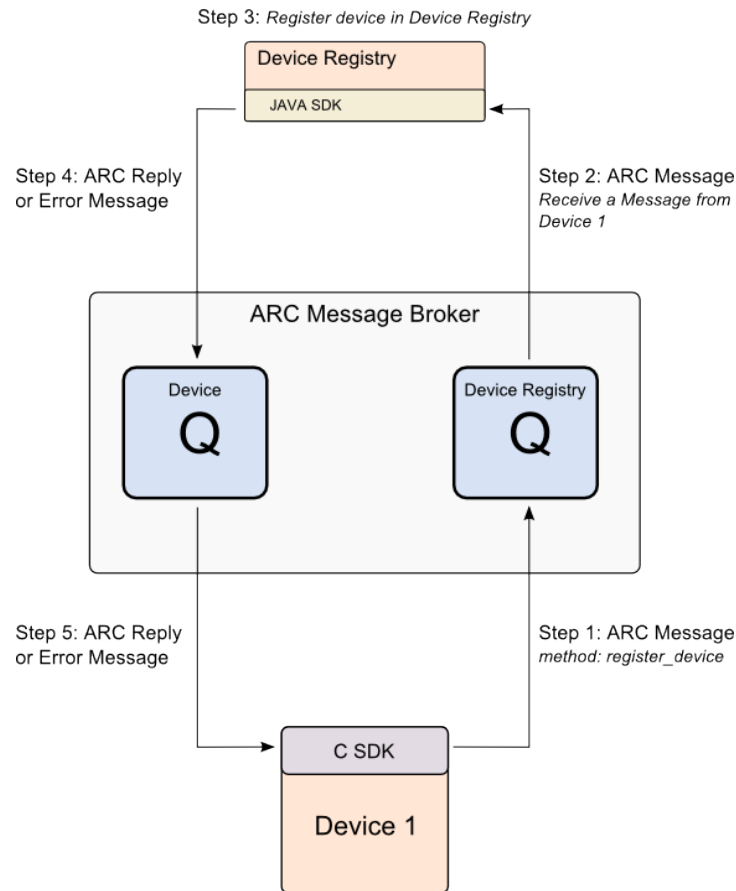
De developers van de ARC-bus hebben ook nog gekeken naar:

- **ØMQ Majordomo broker:** Volgens de ontwikkelaars was dit geen goede keus omdat het product nog niet helemaal af is.
- **Qpid broker:** Maakt gebruik van de protocol AQMP 0.10 en 1.0 standaard. Op het moment is het voornamelijk een wire protocol (overbrengen van data point to point) en niet multicast.
- **RabbitMQ:** RabbitMQ was volgens de developers ook een goede keus, maar ze dachten dat de veranderingen die ze moesten maken moeilijker zouden worden dan bij ActiveMQ
- **HornetMQ:** lijkt heel erg op ActiveMQ, maar is meer gericht op Java dan cross language.

Bij ActiveMQ kan er ook nog gebruik gemaakt worden van ssl. Secure Sockets Layer (SSL) is een cryptographic protocols die is ontworpen voor communicatie beveiliging over computer netwerken. Met dit protocol kunnen de berichten te verstuurd worden beveiligd worden zodat niet een onbevoegd persoon die in het netwerk zit de berichten kan af luisteren.

3 DeviceRegistry

Het tweede onderdeel van de ARC-bus is de DeviceRegistry. De DeviceRegistry is een aparte server die draait naast de message broker. De DeviceRegistry is een database met alle geregistreerden devices. Een device kan zich registreren op de DeviceRegistry via de ARC-bus. Een schematische afbeelding van het verzenden naar de DeviceRegistry vind je in figuur 2. De inhoud van de messages kan je vinden in de bijlage



Figuur 2: Device registreren[1]

De deviceregistry heeft verschillende functionaliteiten die aangeroepen kunnen worden. Deze zijn:

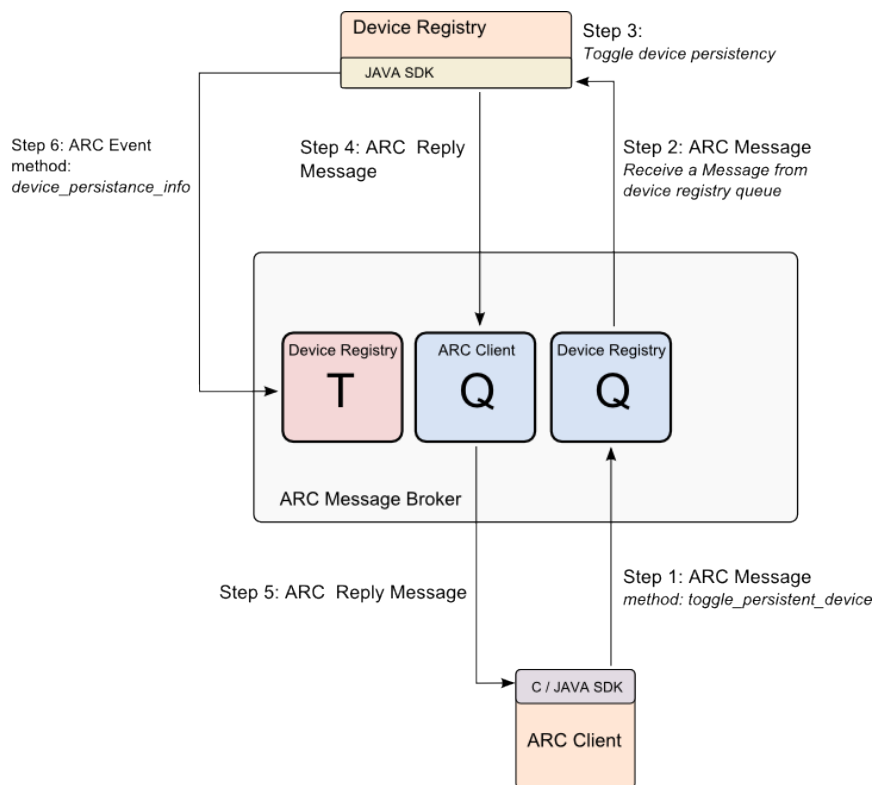
- Device registreren in de database.
- Device uit de database halen.
- Alle devices opvragen die in de database zitten.
- De informatie van het device update in de database.
- Een uuid op vragen.
- De persistent value op true zetten.

Een register device message is een ARC-bus request message. Als method wordt "register_device" mee gegeven. Om de request volledig te maken moeten er ook nog een paar parameters meegegeven worden aan het request. Als het registreren goed is gegaan krijgt het device een ARC Reply message als het fout is gegaan komt er een ARC Error Message. Als het device registreren geslaagd is, wordt er ook nog een "device_registered" event gestuurd. Dit event wordt gestuurd naar elk device die een connectie heeft met de ARC-bus message broker. Als er een device wordt geregistreerd die al geregistreerd is, ontstaat er een 'device_status_info' event, omdat de deviceregistry dan de values van het device update.

Om een device af te melden moet er een ARC request message gestuurd worden met "unregister_device" als method. Als parameter hoeft alleen de UUID meegeleverd te worden. Als dit lukt komt er een reply message met response één. Als dit fout gaat komt er een ARC Error message. Nadat een device af gemeld is bij de DeviceRegistry dan kan hij niet meer in de database gevonden worden. Naast een reply message wordt er ook nog een ARC event message gestuurd met de method "device_unregisterd". Als er een "device_status_info" event message gestuurd wordt door de DeviceRegistry dan is het device niet afgemeld.

Als het nodig is dat de deviceregistry het device blijft onthouden in de database als het device disconnected is. Dan kan dit gedaan worden door een request message te sturen met de methode genaamd "Toggle_persistent_device". De data van het device blijft permanent in de database bestaan. Op het moment dat het device zich afmeldt bij de deviceregistry blijft hij als offline in de database te staan.

Als dit gelukt is dan krijg het device een reply message met response één. In figuur 3: zie je hoe het bericht loopt voor de "Toggle_persistent_device" methode.



Figuur 3: 'Toggle persistent device' methode[1]

4 SDK

Als derde onderdeel van de ARC-bus moeten er SDK gemaakt worden die geïmplementeerd kan worden in de producten van de TKH group. De SDK moet beschikbaar zijn in verschillende programmeer talen. Er is gekozen om AMQP te implementeren in de SDK. AMQP is een open standard application layer protocol voor message-oriented middleware. Ook bij dit protocol was er eerst naar andere gekeken.

ØMQ: ØMQ is een networking library/framework die ook werkt in embedded omgevingen. ØMQ maakt gebruik van sockets die gestuurd kunnen worden net als bij TCP. Het is geschreven in C++. Er zijn wrappers voor elke populaire programmeer taal en besturingssysteem. Het is een framework waardoor er nog veel geschreven moet worden. Het is niet een standaard zoals AMQP

Stomp: Stomp is een messaging protocol gebaseerd op ascii. Het design idee komt van HTTP. Stomp is heel erg gebruikersgericht en wordt alleen uitgebreid en verbeterd door de gebruikers. Stomp kan vanuit elke programmeer taal berichten sturen naar een message broker die Stomp support. Het maakt niet uit in welke programmeertaal de message broker is geschreven.

BACnet: Bacnet is een communicatie protocol speciaal ontworpen voor gebouwbeheer en regelapplicaties zoals verwarming-, ventilatie-, airconditioning-, verlichting-, toegangscontrole- en branddetectie-systemen, waardoor dit niet geschikt is voor de ARC-bus.

Voor AMQP wordt Qpid proton gebruikt. Dit is een library van apache.

De developers van de ARC-bus hebben al twee SDK's geschreven. Deze zijn geschreven in Java en in C.

De huidige SDK's kunnen:

- Verbinden met de ARC-bus
- Registeren op de device registry
- Versturen van berichten
- Ontvangen van berichten

Er zijn twee verschillende berichten die de cliënt kan sturen naar de ARC-bus. Een Request message en een Event message. Een request message wordt gebruikt als een device informatie nodig heeft van een andere device. Het event message wordt gebruikt als er informatie naar een andere device gestuurd moet worden zonder dat er een request voor geweest is of als het andere device iets moet uitvoeren zonder dat er een response terug hoeft te komen. Qpid-proton wordt als library gebruikt voor het protocol AMQP. De SDK implementeert deze library. In de bijlage kan je de opbouw van de ARC-RPC vinden en maakt gebruik van de AMQP message format. RPC staat voor Remote procedure call. RPC is een technologie die toestaat om van een client code te laten uitvoeren op een andere client die hier door de ontwikkelaars niet voor ontwikkelt is.

De source is ook beschikbaar van de SDK's. Er is een Java en een C versie. Voor dit onderzoek is er alleen naar de C versie gekeken. De reden hiervoor is dat VDG Sense zelf geschreven is in C++ waardoor het makkelijker is om de C SDK te gebruiken. Van de C SDK is een class diagram gemaakt. Dit classen diagram is niet gemaakt door de ontwikkelaar van de ARC-bus zelf. Dit is tijdens het onderzoek reverse engineered. Dit kan je vinden in bijlage x op pagina. Voor het configureren van de SDK en daarna verzenden van een bericht is een sequence diagram gemaakt om dit makkelijk te visualiseren en te begrijpen.

Tijdens het uitvoeren draaien er twee devices op één pc. De message broker wordt uitgevoerd op een virtuele machine op een server. Na het uitvoeren van de source code zijn er een paar dingen uitgekomen.

1. De SDK wordt in een aparte thread uitgevoerd dan de example code.
2. Het bericht dat verzonden wordt van device één naar device twee wordt ontvangen door device twee.
3. Het adres van het bericht kan worden opgevraagd. Hier komt de verwachte data uit.
4. De type kan worden opgevraagd, hier komt de juiste data uit.
5. De zender kan worden opgevraagd, hier komt de correcte naam van de zender uit.

6. De parameters kunnen worden opgevraagd.
7. Error messages die binnen komen worden nog niet verwerkt.

Test met drie devies op één pc. De message broker is gerund op een virtuale machine op een server. Tijdens het testen van Events en Requests sturen zijn dit de dingen die na voren gekomen zijn:

1. Een request word maar naar één device gestuurd
2. Als een device event berichten wilt ontvangen moet er apart voor aangemeld worden.
3. Elk device die zich aangemeld heeft op event van een device krijgt zijn event berichten.
4. Als je op een event registreert van de zender ">" dan krijg je alle events binnen.
5. Als je op een event registreert van de zender "tkh.>" dan krijg je alle berichten waar de zender naam begint met tkh.

Alle soorten datatypes kunnen in een ARC-bus message zitten.

Volgens de documentatie kunnen er meerderen message broker aan elkaar gekoppeld worden. Devices die aan de verschillende message brokers gekoppeld zijn, kunnen nog steeds onderling met elkaar kunnen communiceren. Na het testen bleek dit nog niet geïmplementeerd te zijn. Tijdens het gesprek met de deal developer van TKH innovations bleek het dat wel zou moeten werken, maar hij zou dit nog uitzoeken. Na dat dit onderzocht was melde de lead developer dat dit inderdaad nog niet werkte.

In de huidige SDK messenger wordt er tijdens het instellen van de messenger automatisch geregistreerd op request berichten die voor de cliënt bedoeld zijn.

De berichten worden onvangen door de SDK maar wordt voor de rest niks mee gedaan. Tijdens het implementeren van de SDK kan een callback functie worden ingesteld. Deze callback functie wordt dan aangeroepen op het moment dat er een bericht ontvangen wordt door de SDK.

5 Toekomst

De ARC-bus is nog niet helemaal uitontwikkeld. De TKH lead developer heeft al een paar plannen voor in de toekomst. Een paar van de plannen zijn voor.

- Meerderen Message brokers aan elkaar koppelen.
- Een C# SDK ontwikkelen.
- UPnP[2], Devices kunnen de message broker vanzelf vinden.
- Een filter bij de Get devices methode.
- Een installer maken zodat de message broker geïnstalleerd kan worden op een pc.

6 Conclusie

In het hoofdstuk conclusie wordt antwoord gegeven om de vragen die in de inleiding worden gesteld.

De ARC-bus is ontwikkeld door de afdeling TKH-innovations dit is een afdeling die valt onder de TKH group. De bedoeling van de ARC-bus is om de communicatie tussen de producten binnen de TKH group te verbeteren. Dit wordt gedaan door middel van een messenger broker. De clients melden zich aan bij de message broker en registreren zich op event berichten van een bepaalde zender en op request berichten die voor hem bedoeld zijn. Alle berichten worden eerst naar de message broker gestuurd en deze bepaald door middel van de registraties naar welke client de berichten naar toe moeten worden gestuurd. De ARC-bus gebruikt de protocol AMQP voor de berichten. AMQP is een open standard application layer protocol voor message-oriented middleware. De ARC-bus bestaand uit de eerder vernoemde message broker die er voor zorgt dat de berichten bij de juiste clients uitkomen. Naast de message broker is er ook nog een deviceregistry. De deviceregistry is een database waar de clients zicht kunnen registreren. De client kan ook alle devices opvragen die in die database zitten. Voor sommige embedded devices is er ook een optie om voor een UUID te vragen. De lead developer heeft nog een paar onderdelen die in de planning staan voor de toekomst bijvoorbeeld een mogelijkheid dat de message broker aan elkaar gekoppelt kunnen worden en UPnP.

Uit dit onderzoek bleek het dat de ARC-bus gebruikt kan worden voor de communicatie tussen de verschillende Sense servers. Er moet alleen nog wat verder ontwikkeld worden aan de ARC-bus.

Literatuurlijst

- [1] TKH innovations. *ARC-bus documentatie*. URL: <https://tkhinnovations.atlassian.net/wiki/display/EB/Architecture> (bezocht op 28-09-2015).
- [2] Wikipedia. *Universal Plug and Play*. URL: https://en.wikipedia.org/wiki/Universal_Plug_and_Play (bezocht op 30-09-2015).
- [3] Apache. *ActiveMQ*. URL: <http://activemq.apache.org/> (bezocht op 29-09-2015).
- [4] Apache. *Qpid proton*. URL: <https://qpid.apache.org/proton/> (bezocht op 29-09-2015).
- [5] TKH innovations. *arc-bus-sdk git*. URL: <https://git.tkhinnovations.com/development-center/arc-bus-sdk> (bezocht op 28-09-2015).

Bijlage

7.1 ARC-RPC

Request message

Een request message is een AMQP 1.0 message met extra header en extra parameters in de body.

ARC RPC: Een application-properties in de header die specificeert de versie van de ARC-RPC protocol. Dit moet een string zijn en de waarde 1.0 hebben.

Method: Een application-properties in de header die specificeert the naam van de methode die uitgevoerd moet worden. Dit moet een string zijn.

Message-type: Een application-properties in de header die specificeert dat dit bericht een request is. Dit moet een string zijn met de waarde één.

Correlation-id: Een property in de header die bepaald wordt door de cliënt dit moet een string zijn.

Reply-to: Een property in de header die bepaald door de client dit moet een string bevatten. De string moet het adres zijn waar de reply van de request naar toe gestuurd moet worden.

Method parameters: Een object in de body dat de parameters bevat die gebruikt worden voor het uitvoeren van de method. Dit mag weg gelaten worden.

Event message

Een event message is een AMQP 1.0 message met extra header en extra parameters in de body.

ARC RPC: Een application-properties in de header die specificeert de versie van de ARC-RPC protocol. Dit moet een string zijn en de waarde 1.0 hebben.

Method: Een application-properties in de header die specificeert the naam van de methode die uitgevoerd moet worden. Dit moet een string zijn.

Message-type: Een application-properties in de header die specificeert dat dit bericht een request is. Dit moet een string zijn met de waarde drie.

Method parameters: Een object in de body dat de parameters bevat die gebruikt worden voor het uitvoeren van de method. Dit mag weg gelaten worden.

Reply message

Wanneer er een request bericht is ontvangen moet er met een reply gestuurd worden. Behalve als het een notificatie is. De Reply message is een AMQP 1.0 message met extra header en extra parameters in de body.

ARC RPC: Een application-properties in de header die specificeert de versie van de ARC-RPC protocol. Dit moet een string zijn en de waarde 1.0 hebben.

Message-type: Een application-properties in de header die specificeert dat dit bericht een request is. Dit moet een string zijn met de waarde twee.

Correlation-id: Een property in de header. Deze property moet in de header aanwezig zijn en moet de zelfde value hebben als de correlation-id van de request message. Als er een error was bij het uitlezen van de correlation-id van de request message (parse error of Ongeldig verzoek), Dan moet dit leeg zijn.

Response: Een object in de body dat de response value bevat. De value is bepaald door de method van de request message. Deze member is verplicht.

Error message

Wanneer een RPC aanroep een error tegenkomt moet er er beantwoord worden met een error message. Behalve wanneer de aanroep een notificatie is.

When a rpc call encounters an error, the Server MUST reply with a Error message , except for in the case of Notifications. De error message is een AMQP 1.0 message met extra header en extra parameters in de body.

arcrpc: Een application-properties in de header die specificeert de versie van de ARC-RPC protocol. Dit moet een string zijn en de waarde 1.0 hebben.

Message-type: Een application-properties in de header die specificeert dat dit bericht een request is. Dit moet een string zijn met de waarde vier.

Correlation-id: Een eigenschap in de header. Deze eigenschap moet in de header aanwezig zijn en moet de zelfde waarde hebben als de correlation-id van de request message. Als er een error was bij het uitlezen van de correlation-id van de request message (parse error of Ongeldig verzoek), Dan moet dit leeg zijn.

Error: Een object in de message body dat definieert de error object. Dit is verplicht.

Error Object De error member is een AMQP map de de volgende members:

Code: Een nummer dat de type fout die opgetreden is aangeeft. Dit moet een integer zijn.

Message: Een string met daar in een korte beschrijving van de error. Het bericht moet worden beperkt tot een beknopte enkele zin.

Data: Een Primitieve of Structured waarde die extra informatie over de fout bevat. Dit kan worden weggelaten. De waarde van deze waarde wordt bepaald door de server (bv gedetailleerde foutinformatie, geneste fouten etc.).

De error code's van -32768 tot -32000 zijn gereserveerd.

Code	Message	Meaning
-32700	Parse error	Ongeldig AMQP bericht is ontvangen door de server. Er is een fout opgetreden op de server, terwijl het AMQP bericht werd verwerkt.
-32600	Invalid Request	De message is niet een geldige request object.
-32601	Method not found	De method bestaat niet of is niet beschikbaar.
-32602	Invalid params	ongeldige method parameter(s).
-32603	Internal error	ongeldige method parameter(s).
-32000 to -32099	Server error	Gereserveerd voor implementatie gedefinieerde server errors.

Device Registry Messages

Al Deze messages zijn messages zijn gehaald uit de documentatie.

Register device

Request

Type message	ARC request Message
method	register_device

Params Deze Params zitten in een map. Waarbij de Name de key is

Naam	Verplicht	Type	Beschrijving
_id	Ja	UUID	Unieke identificatie binnen in de ARC omgeving.
Address	Ja	String	Adres van het device in de ARC Bus omgeving.
State	Nee	String	Device status - online / offline
Persistent	Nee	Boolean	Behouden apparaat status na verbreken van de connectie - true / false
Vendor	Nee	String	Device specifieke informatie
Product	Nee	String	Device specifieke informatie
Category	Nee	String	Device specifieke informatie
Registration	Nee	String	Device specifieke informatie
IPv6	Nee	String	Device specifieke informatie
IPv4	Nee	String	Device specifieke informatie
Mac	Nee	String	Device specifieke informatie
Nee	Nee	Object	Device specifieke informatie
{custom param 2}	Nee	Object	Device specifieke informatie

Als er een device word geregistreerd die al geregistreerd is ontstaat er een 'device_status_info' event. Dit event wordt gestuurd naar alle devices die aan de ARC bus message broker vast zitten. In dit event zitten de volgende parameters:

Type message	ARC request Message
method	register_device

De parameters die ik de body zitten zijn de zelfde als die in de request message. Naast een event krijgt het device die zicht geregistreerd ook een reply message terug. het bericht ziet er als volgt uit:

Type message	ARC reply Message
_id	Unieke identificatie binnen in de ARC omgeving.

Device unregister

Bij dit bericht wordt de geregistreerde device uit de database verwijderd.

Request

Type message	ARC request Message
method	unregister_device

Params

Name	Required	Type	Description
_id	Yes	UUID	Unieke identificatie binnen in de ARC omgeving.

Response

Als het device uitgeschreven geslaagd is dan komt het volgende bericht als een reply terug.

Type message	ARC Reply Message
response	1

Als er een fout is opgetreden tijdens het uit uitschrijven komt er een error message terug.

Mislukt

Type message	ARC Error Message
--------------	-------------------

Device update

Het bericht om de device settings te kunnen updaten in de database van de deviceregistry is nog niet gereleased. Op het moment als je je device informatie wil updaten dan moet je een request bericht sturen met "register_device" als method.

Request

Type message	ARC request Message
method	update_device

Params

Name	Required	Type	Description
_id	Ja	UUID	Unieke identificatie binnen in de ARC omgeving.
persistent	Nee	boolean	Behouden Device status na verbreken van de connectie - true / false
vendor	Nee	String	Device specifieke informatie
product	Nee	String	Device specifieke informatie
category	Nee	String	Device specifieke informatie
registration	Nee	String	Device specifieke informatie
ipv6	Nee	String	Device specifieke informatie
ipv4	Nee	String	Device specifieke informatie
mac	Nee	String	Device specifieke informatie

Response

Als het update van de device settings gelukt is dan stuurt de deviceregistry een de volgende reply message terug.

Successfully

Type message	ARC Reply Message
--------------	-------------------

Params

Name	Type	Value
result	String	success

Als het update van de settings van een device mislukt stuurt de deviceregistry een error message terug.

Mislukt

Type	ARC Error Message
------	-------------------

Get all devices

Als de deviceregistry een message binnen krijgt met "get_all_devices" als method dan stuurt de deviceregistry een reply terug met de data die in de database zitten.

Request

Type	ARC Message
method	get_all_devices

Response

Type	ARC Reply Message
response	List<Map<String, Object>> devices

ARC-bus Requirements document

Student: Aiken Brus (10007695)

Datum: 21-03-2016

Plaats: Zoetermeer

Afstudeerverslag Haagse Hogeschool

Opleiding: Technische informatica

Bedrijf: VD Security B.V.

Opdrachtgever: Timme Grippink

Begeleidende examiner: Cobie van der Hoek

Tweede examiner: John Visser

Inhoudsopgave

1	Inleiding	1
1.1	grenzen van het project	1
2	Het product	2
2.1	De aanleiding	2
2.2	De probleemstelling	2
2.3	De doelstelling	2
2.4	De resultaten	2
2.5	Use-case	3
3	De eisen	4
3.1	Simulatie tool	4
3.2	Plug-in	5
	Literatuurlijst	6
	Bijlage	7
	Bijlage 1: Use-case beschrijving	7
	Bijlage 1.1: use-case beschrijvingen plug-in	7
	Bijlage 1.2: use-case beschrijvingen simulatie tool	11

1 Inleiding

In dit document worden de requirement op gesteld voor het ARC-bus project. De bedoeling van de requirement document is de lezer te informeren welke producten gemaakt worden en wat deze producten gaan doen. Dit document is niet bedoeld om informatie te geven hoe dit gedaan gaat worden in de implementatie. Dit document is opgebouwd in twee kern hoofdstukken ‘Het product’ en ‘De eisen’. In het hoofdstuk ‘Het product’ word uitgelegd waarom deze producten gemaakt worden, Wat er met de producten bereikt wil worden en wat de producten nou precies inhouden. In het tweede hoofdstuk van de kern zijn de eisen van de producten daad werkelijk beschreven.

1.1 grenzen van het project

De afstudeerder is alleen verantwoordelijk voor de plug-in en niet voor de onderdelen van de VDG Sense software. De verantwoordelijkheid van de code van VDG Sense ligt bij de ontwikkelaars van VDG security. De afstudeerder is ook niet verantwoordelijk voor de functie van de ARC-bus.

2 Het product

2.1 De aanleiding

Binnen de TKH group opereert een innovatieafdeling (TKH innovations). Deze houdt zich, onder andere, bezig met het bedenken van oplossingen waarmee de verschillende TKH ondernemingen en hun producten beter kunnen integreren. Projecten die door de TKH group uitgevoerd worden, bestaan in de meeste gevallen uit een samenstelling van één of meerdere TKH ondernemingen, die elk hun eigen product(en) leveren. Hierbij kan bijvoorbeeld aan een situatie gedacht worden waarbij VDG een camera-installatie met video analyse en opslag levert en KeyProcessor een toegangscontrole oplossing. Deze twee producten zullen vervolgens samen moeten gaan werken. Beide zijn een TKH onderneming die hun eigen hardware en software leveren.

Uit het verleden is gebleken dat het een intensieve klus is om de verschillende producten met elkaar te integreren. Hierbij komen een tweetal problemen met name naar voren:

- Elke TKH onderneming heeft in het verleden de vrije keuze gehad in welke techniek(en) en protocol(len) zij implementeerden in hun product(en) waarmee een extern systeem kan communiceren.
- De product- en softwareontwikkelingen staan niet stil en daarmee verandert dus ook regelmatig de wijze waarop met een product gecommuniceerd kan worden.

2.2 De probleemstelling

Om de samenwerking/integratie van verschillende producten op de middellange termijn te verbeteren, heeft de TKH innovatieafdeling de ARC-bus bedacht. Dit is een softwarebus waarmee het mogelijk gemaakt moet worden dat verschillende producten middels een netwerkverbinding, gegevens met elkaar kunnen uitwisselen op een uniforme manier. Een ontvanger kan zich abonneren op berichten die verstuurd worden over de ARC-bus en het is mogelijk om zelf berichten te versturen waar andere zich weer op kunnen abonneren. VDG wilt de software voorbereiden op de komst van de ARC-bus en dit integreren in hun interne systeem VDG Sense. De ARC-bus is een nieuw concept en het is voor de VDG dan ook nog niet volledig duidelijk op welke wijze dit gedaan moet worden.

2.3 De doelstelling

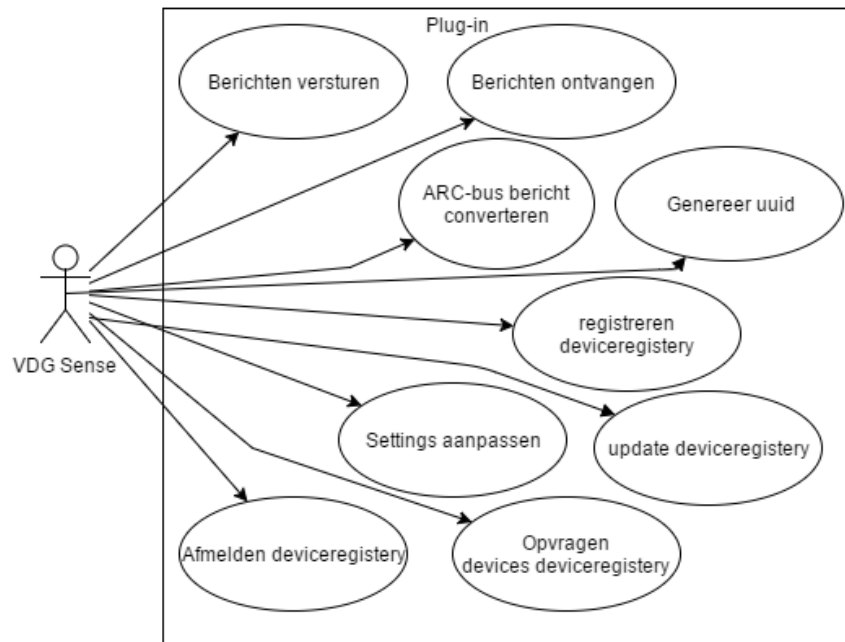
De doelstelling van de afstudeeropdracht is door middel van een onderzoek meer duidelijkheid te krijgen wat de ARC-bus is en wat deze kan. Met behulp van dit onderzoek worden requirements geïdentificeerd waarmee de ARC-bus geïntegreerd moet gaan worden in de VDG Sense software. Door middel van een plug-in moeten berichten over de ARC-bus uit gewisseld kunnen worden.

2.4 De resultaten

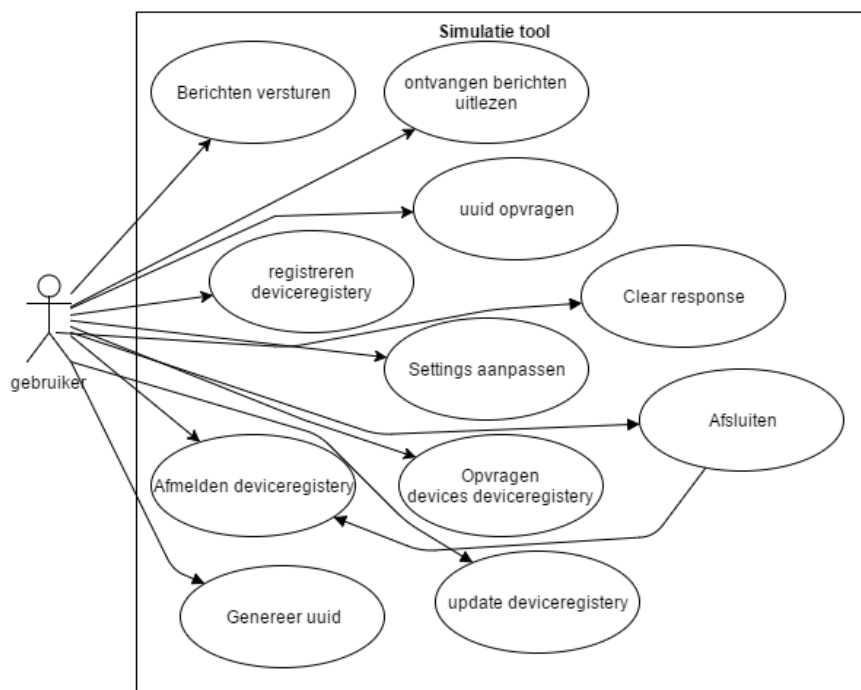
Een van de resultaten is een plug-in die als een middleware laag tussen de VDG Sense software en ARC-bus gaat fungeren. Naast de plug-in komt er een rapport waarin staat beschreven hoe de plug-in gebruikt kan worden. Via deze plug-in zal het mogelijk zijn om berichten over de ARC-bus uit te wisselen.

In het volgende hoofdstuk wordt beschreven welke ontwikkelmethode gekozen is om de opdracht te voltooien en hoe deze is gekozen.

2.5 Use-case



Figuur 1: Use-case Plug-in ARC-bus



Figuur 2: Use-case simulatie tool

3 De eisen

In dit hoofdstuk worden de eisen van het eind product opgeschreven. De eisen zijn verzameld door in gesprek te gaan met de lead developer van VDG security, developer van de ARC-bus en door een onderzoek te doen naar de werking en specificaties van de ARC-bus. Tijdens dit onderzoek zijn de functies van de SDK en ARC-bus onderzocht. Uit dit onderzoek zijn verschillende functies ontstaan die geïmplementeerd moesten worden in de ARC-bus. De eisen zijn onderverdeeld in verschillende categorieën. Deze categorieën zijn:

- Functionele eisen dit zijn de functies waar het product aan moet voldoen.
- Usability eisen dit zijn de eisen om het voor de gebruiker makkelijker te maken om de plug-in te gebruiken.
- Technische eisen dit zijn de eisen waar het product aan moet voldoen op een technisch vlak.

3.1 Simulatie tool

Eerst zijn de eisen van de simulatie tool opgesteld. **Functionele eisen**

- Door middel van een knop in de simulatie tool moet er een request berichten kunnen verzenden.
- Er moet een invoerveld komen om te bepalen waar de request bericht naar toe gestuurd wordt.
- Als de simulatie tool een request bericht heeft ontvangen moet er automatisch een reply bericht verstuurd worden naar de zender van het request bericht.
- Door middel van een knop in de simulatie tool moet een event berichten kunnen verzenden naar de ARC-bus.
- De simulatie tool moet berichten kunnen ontvangen van devices zonder dat ze een specifieke naam moeten hebben.
- De simulatie tool moet ontvangen berichten converteren naar een xml formaat en die dan uitprinten wordt.
- De simulatie tool moet zich bij de ARC-bus kunnen abonneren wanneer de plug-in wordt opgestart.
- De simulatie tool moet zich bij de ARC-bus kunnen afmelden wanneer de plug-in wordt afgesloten.
- De simulatie tool moet een IP-adres setting hebben en deze moet je kunnen instelling in een aparte venster van de simulatie tool.
- De Simulatie tool moet een port setting hebben en deze moet je kunnen instelling in een aparte venster van de simulatie tool.
- De simulatie tool moet een device naam hebben en deze moet je kunnen instelling in een aparte venster van de simulatie tool.
- De simulatie tool moet een uuid hebben en deze moet je kunnen instelling in een aparte venster van de simulatie tool.
- De simulatie tool moet opnieuw gaan connecten wanneer de settings zijn aangepast.
- De simulatie tool moet zich kunnen registreren bij de DeviceRegistry met de eerder opgegeven settings.
- De simulatie tool moet zich automatisch bij het opstarten registreren bij de deviceregistry.
- De simulatie tool moet door middel van een knop zich kunnen afmelden bij de DeviceRegistry.
- De simulatie tool moet zich automatisch bij het afsluiten afmelden bij de deviceregistry.
- De simulatie tool moet automatisch zijn eigen configuraties kunnen update als de configuraties aangepast zijn.
- De simulatie tool moet met een druk op een knop alle devices die aangemeld zijn op de DeviceRegistry kunnen opvragen.
- De simulatie tool moet met een druk op een knop kunnen instellen bij de DeviceRegistry of hij de zelfde uuid wil behouden als hij reconnect na dat hij disconnected was.

- Er moet een invoerveld zijn om de waarde van de persistent te kunnen bepalen met één of nul.
- Er moet een invoerveld zijn waar de naam van het device instaat waar de request bericht naar toe wordt verstuurt.
- Er moet een veld zijn waar de binnen gekomen berichten in worden geprint.
- Er moet een knop zijn maar de response veld mee kan leeg gemaakt worden.
- Er moet een knop zijn om de simulatie tool mee af te sluiten.

Technische eisen

- De simulatie tool moet asynchrone berichten kunnen verzenden en ontvangen.

3.2 Plug-in

Als tweede zijn de eisen van de plug-in opgesteld. Een deel van de eisen die in de simulatie tool zitten komen ook weer terug in de plug-in.

Functionele eisen

- De plug-in moet een request berichten kunnen verzenden naar de ARC-bus.
- De plug-in moet een event berichten kunnen verzenden naar de ARC-bus.
- De plug-in moet een reply bericht kunnen verzenden naar de ARC-bus.
- De plug-in moet berichten kunnen ontvangen van de ARC-bus.
- De plug-in moet ontvangen berichten kunnen converteren naar een Sense bericht.
- De plug-in moet een Sense berichten kunnen converteren naar een ARC-bus bericht.
- De plug-in moet zich bij de ARC-bus kunnen abonneren op event berichten van elk apparaat apart.
- De plug-in moet zich bij de ARC-bus kunnen afmelden op de event berichten.
- De plug-in moet een IP-adres setting hebben en deze moet je kunnen instelling in VDG Sense.
- De plug-in moet een port setting hebben en deze moet je kunnen instelling in VDG Sense.
- De plug-in moet zich registreren bij de DeviceRegistry na het aanzetten van de plug-in.
- De plug-in moet zich afmelden bij de DeviceRegistry na het afsluiten van Sense pluginmanager of als de ARC-bus plug-in wordt uitgezet.
- De plug-in moet zijn eigen configuraties kunnen update in de DeviceRegistry.
- De plug-in moet alle devices die aangemeld zijn op de DeviceRegistry kunnen opvragen.
- De plug-in moet kunnen instellen bij de DeviceRegistry of hij de zelfde id wil behouden als hij reconnect na dat hij disconnected was.

Usability eisen

- De gebruiker moet via Sense de plug-in settings de ARC-bus plug-in settings kunnen aanpassen.

Technische eisen

- De plug-in moet asynchrone berichten kunnen verzenden en ontvangen.

Technische Niet functionele eisen

- Maak gebruik van GoogleTest[1] voor de unit tests
- Maak gebruik van GoogleMock[2] voor de unit tests

Literatuurlijst

- [1] Google. *Google Test*. URL: <https://github.com/google/googletest> (bezocht op 04-11-2015).
- [2] Google. *What Is Google C++ Mocking Framework?* URL: <https://github.com/google/googletest/blob/master/googlemock/docs/ForDummies.md> (bezocht op 07-12-2015).
- [3] A Brus. *Ontwikkelen van een ARC-bus plug-in voor VDG Sense*. Zoetermeer, 2016.
- [4] Google. *Introduction: Why Google C++ Testing Framework?* URL: <https://github.com/google/googletest/blob/master/googletest/docs/Primer.md> (bezocht op 04-11-2015).
- [5] Google. *CheatSheet*. URL: <https://github.com/google/googletest/blob/master/googlemock/docs/CheatSheet.md> (bezocht op 07-12-2015).

Bijlage

Bijlage 1: Use-case beschrijving

Bijlage 1.1: use-case beschrijvingen plug-in

Naam	Berichten versturen
Aanleiding	VDG Sense wilt een event sturen naar de ARC-bus
Actoren	VDG Sense
Doel	Een bericht verzenden naar de ARC-bus
Beschrijving	<ol style="list-style-type: none">1. Request of Event Bericht wordt gemaakt.2. Inhoud word toegevoegd aan het bericht.3. Bericht word verzonden.
Resultaat	Een bericht is gestuurd naar de ARC-bus

Naam	Bericht ontvangen
Aanleiding	Plug-in ontvangt een bericht van de ARC-bus
Actoren	VDG Sense
Doel	Een bericht ontvangen en verwerken
Beschrijving	<ol style="list-style-type: none">1. Bericht wordt ontvangen.2. In het bericht word gekeken waar het bericht vandaan komt.3. Bericht word gegeven aan de bijbehorende handler.4. Bericht word omgezet naar een VDG bericht.5. ARC-bus plug-in stuurt een callback met het bericht naar VDG Sense manager
Resultaat	VDG Sense heeft een bericht ontvangen van de ARC-bus

Naam	registeren ARC-bus
Aanleiding	VDG Sense software moet berichten versturen naar de ARC-bus
Actoren	VDG Sense
Doel	VDG Sense wil berichten van bepaalde devices ontvangen.
Beschrijving	<ol style="list-style-type: none"> 1. Door middel van de Device name kan er aangemeld worden voor event of request berichten
Resultaat	VDG Sense is aangemeld op de ARC-bus en ontvangt request en event berichten van de devices waar VDG Sense is op aangemeld.

Naam	Afmelding ARC-bus
Aanleiding	VDG Sense wil afmelden op berichten van andere devices berichten.
Actoren	VDG Sense
Doel	VDG Sense wil geen berichten meer ontvangen van bepaalde devices.
Beschrijving	<ol style="list-style-type: none"> 1. Door middel van de Device name kan er aangemeld worden voor event of request berichten
Resultaat	Plug-in ontvangt geen berichten meer van de devices waar op afgemeld zijn.

Naam	Opvragen devices ARC-bus
Aanleiding	VDG Sense wil graag weten welke devices er aan de ARC-bus vast zitten.
Actoren	VDG Sense
Doel	Alle geregistreerde devices opvragen
Beschrijving	<ol style="list-style-type: none"> 1. Plug-in maakt een ARC-bus message. 2. Set de methode met 'get_all_devices' 3. Stuur bericht naar ARC-bus. 4. Plug-in ontvangt een reply message van de ARC-bus met alle devices in de body. 5. Data van de reply message omzetten naar een VDG Sense bericht
Resultaat	Een VDG Sense bericht met alle devices in een map

Naam	Setings aanpassen
Aanleiding	De gebruiker van VDG Sense heeft de settings van de ARC-bus verandert in VDG Sense.
Actoren	VDG Sense
Doel	De settings van de ARC-bus plug-in aanpassen
Beschrijving	<ol style="list-style-type: none"> 1. De nieuwe settings komen binnen in de ARC-bus plug-in. 2. De oude settings worden vervangen door de nieuwe settings. 3. Als de IP of port is verandert kan word er opnieuw geconnect met de ARC-bus 4. De veranderingen worden gesaved naar een ‘ini’ file
Resultaat	De settings zijn verandert en er is een nieuwe connectie gemaakt met de ARC-bus

Naam	ARC-bus bericht converteren
Aanleiding	Er is een ARC-bus bericht ontvangen of VDG Sense systeem wil een ARC-bus bericht versturen
Actoren	VDG Sense
Doel	Een ARC-bus bericht omzetten van of naar een Sense Bericht
Beschrijving	<p>Als er een bericht ontvangen wordt vanaf de ARC-bus kan het Sense Systeem dit nog niet gebruiken dus wordt de data uit het bericht gehaald en omgezet naar een xml waarde en die wordt doorgegeven naar het Sense systeem.</p> <p>Als het Sense systeem een bericht wil versturen krijgt de plug-in een xml waarde binnen. Dit kan niet verzonden worden via de ARC-bus. De data word uit het xml structuur gehaald en als data in een ArcBusMessage gestopt.</p>
Resultaat	

Naam	Genereer uuid
Aanleiding	De ARC-bus plug-in wordt opgestart
Actoren	VDG Sense
Doel	Het krijgen van een uuid van de deviceregistry
Beschrijving	<ol style="list-style-type: none"> 1. Plug-in heeft geen uuid 2. maak een request ARC-bus message bedoeld voor de DeviceRegistry met method GenerateUUID 3. Verstuur het message
Resultaat	Er wordt een uuid ontvangen van de DeviceRegistry

Naam	update deviceregistry
Aanleiding	Setting van de plug-in zijn aangepast
Actoren	VDG Sense
Doel	Settings aanpassing in de database van de DeviceRegistry
Beschrijving	<ol style="list-style-type: none"> 1. Plug-in heeft geen uuid 2. maak een request ARC-bus message bedoeld voor de DeviceRegistry met method GenerateUUID 3. Verstuur het message 4. ontvang een reply van DeviceRegistry of het gelukt is
Resultaat	De settings zijn aangepast in de database van de DeviceRegistry

Bijlage 1.2: use-case beschrijvingen simulatie tool

Naam	Berichten versturen
Aanleiding	VDG Sense wilt een event sturen naar de ARC-bus
Actoren	Gebruiker
Doel	Een bericht verzenden naar de ARC-bus
Beschrijving	<ol style="list-style-type: none">1. Gebruiker klikt op button "SendEvent" of "SendRequest"2. Request of Event Bericht wordt gemaakt.<ol style="list-style-type: none">1. Inhoud word toegevoegd aan het bericht.2. Bericht word verzonden.
Resultaat	Een bericht is gestuurd naar de ARC-bus

Naam	Bericht ontvangen
Aanleiding	Plug-in ontvangt een bericht van de ARC-bus
Actoren	Gebruiker
Doel	Een bericht ontvangen en verwerken
Beschrijving	<ol style="list-style-type: none">1. Bericht wordt ontvangen.2. In het bericht word gekeken waar het bericht vandaan komt.3. Bericht word gegeven aan de bijbehorende handler.4. Bericht word omgezet naar een VDG bericht.5. ARC-bus plug-in stuurt een callback met het bericht naar VDG Sense manager
Resultaat	VDG Sense heeft een bericht ontvangen van de ARC-bus

Naam	registeren ARC-bus
Aanleiding	VDG Sense software moet berichten versturen naar de ARC-bus
Actoren	Gebruiker
Doel	VDG Sense wil berichten van bepaalde devices ontvangen.
Beschrijving	Door middel van de Device name kan er aangemeld worden voor event of request berichten
Resultaat	VDG Sense is aangemeld op de ARC-bus en ontvangt request en event berichten van de devices waar VDG Sense is op aangemeld.

Naam	Afmelding ARC-bus
Aanleiding	VDG Sense wil afmelden op berichten van andere devices berichten.
Actoren	Gebruiker
Doel	VDG Sense wil geen berichtne meer ontvangen van bepaalde devices.
Beschrijving	Door middel van de Device name kan er aangemeld worden voor event of request berichten
Resultaat	Plug-in ontvangt geen berichten meer van de devices waar op afgemeld zijn.

Naam	Opvragen devices ARC-bus
Aanleiding	VDG Sense wil graag weten weten welke devices er aan de ARC-bus vast zitten.
Actoren	Gebruiker
Doel	Alle geregistreerde devices opvragen
Beschrijving	<ol style="list-style-type: none"> 1. Gebruiker klikt op button "GetAllDevices" 2. Plug-in maakt een ARC-bus message. 3. Set de methode met 'get_all_devices' 4. Stuur bericht naar ARC-bus. 5. Plug-in ontvangt een reply message van de ARC-bus met alle devices in de body. 6. Data van de reply message omzetten naar een VDG Sense bericht
Resultaat	Een VDG Sense bericht met alle devices in een map

Naam	Setings aanpassen
Aanleiding	De gebruiker van VDG Sense heeft de settings van de ARC-bus verandert in VDG Sense.
Actoren	Gebruiker
Doel	De settings van de ARC-bus plug-in aanpassen
Beschrijving	<ol style="list-style-type: none"> 1. Gebruiker klikt op button "Save" 2. De oude settings worden vervangen door de nieuwe settings. 3. Als de IP of port is verandert kan word er opnieuw geconnect met de ARC-bus 4. De veranderingen worden gesaved naar een 'ini' file
Resultaat	De settings zijn verandert en er is een nieuwe connectie gemaakt met de ARC-bus

Naam	ARC-bus bericht converteren
Aanleiding	Er is een ARC-bus bericht ontvangen of VDG Sense systeem wil een ARC-bus bericht versturen
Actoren	Gebruiker
Doel	Een ARC-bus bericht omzetten van of naar een Sense Bericht
Beschrijving	<p>Als er een bericht ontvangen wordt vanaf de ARC-bus kan het Sense Systeem dit nog niet gebruiken dus wordt de data uit het bericht gehaald en omgezet naar een xml waarde en die wordt doorgegeven naar het Sense systeem.</p> <p>Als het Sense systeem een bericht wil versturen krijgt de plug-in een xml waarde binnen. Dit kan niet verzonden worden via de ARC-bus. De data word uit het xml structuur gehaald en als data in een ArcBusMessage gestopt.</p>
Resultaat	

Naam	Genereer uuid
Aanleiding	De ARC-bus plug-in wordt opgestart
Actoren	Gebruiker
Doel	Het krijgen van een uuid van de deviceregistry
Beschrijving	<ol style="list-style-type: none"> 1. Gebruiker klikt op button "GenerateUUID" 2. maak een request ARC-bus message bedoeld voor de DeviceRegistry met method GenerateUUID 3. Verstuur het message
Resultaat	Er wordt een uuid ontvangen van de DeviceRegistry

Naam	update deviceregistry
Aanleiding	Setting van de plug-in zijn aangepast
Actoren	Gebruiker
Doel	Settings aanpassing in de database van de DeviceRegistry
Beschrijving	<ol style="list-style-type: none"> 1. Gebruiker klikt op button "UpdateDevice" 2. maak een request ARC-bus message bedoeld voor de DeviceRegistry met method GenerateUUID 3. Verstuur het message 4. ontvang een reply van DeviceRegistry of het gelukt is
Resultaat	De settings zijn aangepast in de database van de DeviceRegistry

Naam	clear response
Aanleiding	Setting van de plug-in zijn aangepast
Actoren	Gebruiker
Doel	Settings aanpassing in de database van de DeviceRegistry
Beschrijving	<ol style="list-style-type: none"> 1. Gebruiker klikt op button "Clear" 2. Simulatie tool clear het response veld
Resultaat	De settings zijn aangepast in de database van de DeviceRegistry

Naam	Afsluiten
Aanleiding	Setting van de plug-in zijn aangepast
Actoren	Gebruiker
Doel	Settings aanpassing in de database van de DeviceRegistry
Beschrijving	<ol style="list-style-type: none"> 1. Gebruiker klikt op button "Close" 2. Simulatie tool verstuurd een unregisterdevice message naar de DeviceRegistry
Resultaat	De settings zijn aangepast in de database van de DeviceRegistry

Testdocument

Student: Aiken Brus (10007695)

Datum: 21-03-2016

Plaats: Zoetermeer

Afstudeerverslag Haagse Hogeschool

Opleiding: Technische informatica

Bedrijf: VD Security B.V.

Opdrachtgever: Timme Grippink

Begeleidende examiner: Cobie van der Hoek

Tweede examiner: John Visser

Inhoudsopgave

1	Inleiding	1
2	Het Doel	2
	Literatuurlijst	3
	Bijlage	4
	Test2 integratie test	4
	Test2.1 Simulatie tool test	4
	Test2.2 Plug-in test	14
	Test3 System test	18

1 Inleiding

Om te bepalen of een product dat ontwikkelt is gaat werken op de manier dat het bedacht is moet dit product getest worden. Het testen van een product kan op veel verschillende manieren. In dit document wordt beschreven hoe dit testen gedaan gaat worden voor de opdracht ARC-bus plug-in. Wat een product uiteinde-

lijk moet gaan doen staat beschreven in het requirements document en de UML klassen diagrammen, daarom is dit gebruikt om de testen te schrijven. Het eerste hoofdstuk dat in dit document in het doel. In dit hoofd-

stuk wordt beschreven waarom er getest gaat worden en welke test uitgevoerd gaan worden. In dit hoofdstuk wordt ook beschreven welke benodigdheden gebruikt gaan worden om de testen in de test fase. In de

bijlage staat de verschillende test cases die uitgevoerd gaan worden. De test cases zijn onderverdeeld in eerst de scripts voor unittests. Daarom de integratie tests. Van deze tests zijn er die van de simulatie tool en die van de plug-in zelf. Als laatst zijn de systeem tests cases beschreven.

2 Het Doel

Op basis van alle diagrammen is er een testdocument geschreven. In dit testdocument staan verschillende testen beschreven om de eisen uit het requirements document te kunnen testen. Een testdocument is nodig om er voor te zorgen dat het systeem compleet werkt volgens de opgestelde eisen. Elke test die beschreven is maakt gebruik van de zelfde template. Het testdocument is zo beschreven dat iemand die niet gewerkt heeft aan het systeem de test kan uitvoeren. Dit is gedaan om een beter zicht te hebben of de ARC-bus plug-in daadwerkelijk doet wat de opdrachtgever wil. Voor het testen van de plug-in zullen verschillende test worden uitgevoerd. Als eerste de unit test. De unit test is een test die zo klein mogelijke onderdelen van de plug-in test. De test word gebruikt om de functies van de classes te testen door middel van test scripts. Deze test is een white box test. white box test is een test die wordt uitgevoerd door middel van de kennis van de code. Deze test worden gemaakt en uitgevoerd tijdens het ontwikkelen van de plug-in. De bedrijfs-begeleider zal ook nog een code review doen. Voor de unit test wordt “GoogleTest” gebruikt. “GoogleTest” is een frame-

work om in c++ test te kunnen schrijven. Voor het testen van de interfaces wordt de test framework “GoogleMock” gebruikt. Mocks zijn objecten die voorgeprogrammeerd zijn met verwachtingen, deze verwachtingen vormen de specificaties van class functies waar van verwacht wordt dat ze aangeroepen worden en bijvoorbeeld een specifieke return waarde geven. Deze mocks kunnen gebruik worden in het testen. Bij deze tests wordt getest of de functie verwachtingen die opgesteld zijn in de mock ook in de werkelijkheid zo voor komen. Het is de bedoeling dat deze tests gerund gaan worden elke keer als er iets verandert aan de SDK. Om

de samenwerking van de verschillende classes te kunnen testen, wordt er een integration test uitgevoerd. Deze test maakt gebruik van een simulatie tool. Deze tests maken gebruik van een ‘Specification-based testing’. Dit betekent dat de stappen die uitgevoerd moeten worden beschreven staan in een test case. Deze test cases staan beschreven in de bijlage onder de hoofdstuk ‘De test cases’. Tijdens de integratie worden de twee producten simulatie tool en plug-in getest. Als eerste wordt de simulatie tool getest. De uitkomst van deze test zou moeten zijn dat de simulatie tool werkt. Daarna kan de simulatie tool gebruikt worden om de plug-in te testen. Elke test case wordt enkele keren uitgevoerd. Als laatst word er een ‘System test’ uitgevoerd. Deze

test maakt ook gebruik van ‘Specification-based testing’. De ‘System test’ wordt uitgevoerd door een tester van VDG en door de afstudeerder. Deze test worden niet uitgevoerd met de simulatie tool maar met VDG Sense zelf. Doordat deze test worden uitgevoerd door de tester is deze test een black box test. Voor sommige van de system tests zijn automatisch tests geschreven. De benodigheden bij deze drie testen verschillen. Bij

test één is ‘Qt Creator’ nodig voor het runnen van de test scripts en om onderdelen van VDG Sense te runnen. Bij test twee is de simulatie tool nodig. Hier bij zijn ook de test cases nodig. Bij test drie is VDG Sense nodig met de plug-in. voor deze test zijn ook de geschreven test cases nodig.

Literatuurlijst

- [1] Google. *Google Test*. URL: <https://github.com/google/googletest> (bezocht op 04-11-2015).
- [2] Google. *Introduction: Why Google C++ Testing Framework?* URL: <https://github.com/google/googletest/blob/master/googletest/docs/Primer.md> (bezocht op 04-11-2015).
- [3] Google. *What Is Google C++ Mocking Framework?* URL: <https://github.com/google/googletest/blob/master/googlemock/docs/ForDummies.md> (bezocht op 07-12-2015).
- [4] Google. *CheatSheet*. URL: <https://github.com/google/googletest/blob/master/googlemock/docs/CheatSheet.md> (bezocht op 07-12-2015).

Bijlage

Test2 integratie test

Test2.1 Simulatie tool test

Test Case	Request bericht versturen
Beschrijving	Verstuur een request bericht over de arcbus. use-case bericht versturen.
Benodigdheden	Twee simulatie tools, message broker, deviceregistry
preconditie	Instellingen zijn ingesteld
Uitgevoerd door	Aiken Brus
Datum van uitvoering	12-07-2015

Test2.1 1.1	Verwachte uitkomst	Officiële uitkomst
<p>1. Vul de naam van de twee simulatie tool in de invoer veld bij "Event / Request"</p> <p>2. Druk op de knop "Sendevent"</p>	<p>In de response veld komt de volgende message te staan:</p> <pre><messagetype> request </messagetype> <timesend>(datum en tijd) </timesend> <sender> tkh.vdg.device.simulation.1 </sender> <method> RequestData </method> <Body> { "data" (string) "TestData" (string) } </Body></pre> <p>en bij de simulatie tool die het bericht verzond komt de volgende reply terug.</p> <pre><messagetype> reply </messagetype> <timesend> (datum en tijd) </timesend> <sender> tkh.vdg.device.simulation.2 </sender> <Body> { "data" (string) "TestData" (string) } </Body></pre>	<pre><messagetype> request </messagetype> <timesend> 12/07/15 10:23:04</timesend> <sender> tkh.vdg.device.simulation.1 </sender> <method> RequestData </method> <Body> { "data" (string) "TestData" (string) } </Body></pre> <p>en bij de simulatie tool die het bericht verzond komt de volgende reply terug.</p> <pre><messagetype> reply </messagetype> <timesend> 2/07/15 10:23:04 </timesend> <sender> tkh.vdg.device.simulation.2 </sender> <Body> { "data" (string) "TestData" (string) } </Body></pre>

Test Case	Request bericht versturen
Beschrijving	Verstuur een request bericht over de arcbus naar een niet bestaande naam. use-case bericht versturen.
Benodigdheden	Één simulatie tools, message broker, deviceregistry.
preconditie	Instellingen zijn ingesteld.
Uitgevoerd door	Aiken Brus
Datum van uitvoering	12-07-2015

Test2.1 2.1	Verwachte uitkomst	Officiële uitkomst
1. Vul een random naam in bij de invoerveld bij "Event / Request" 2. Druk op de knop "SendRequest"	Het bericht wordt niet verstuurd en ook niet ontvangen door de tweede simulatie tool	Het bericht wordt verstuurd naar de ARC-bus maar niet naar een andere client

Test Case	Event bericht versturen
Beschrijving	Verstuur een event bericht over de arcbus. use-case bericht versturen.
Benodigdheden	Twee simulatie tools, message broker
preconditie	Instellingen zijn ingesteld
Uitgevoerd door	Aiken Brus
Datum van uitvoering	

Test2.1 3.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop "Sendevent"	In de output van de tweede message komt te staan: <messagetype> event </messagetype> <timesend> (datum en tijd) </timesend> <sender> tkh.vdg.device.simulation.1 </sender> <method> EventData </method> <Body> { "data" (string) "TestData" (string) } </Body>	<messagetype> event </messagetype> <timesend> 2/07/15 10:28:17 </timesend> <sender> tkh.vdg.device.simulation.1 </sender> <method> EventData </method> <Body> { "data" (string) "TestData" (string) } </Body>

Test Case	Berichten kunnen ontvangen van devices zonder dat ze een specifieke naam moeten hebben.
Beschrijving	Verstuur een request bericht over de arcbus naar een niet bestaande naam. use-case bericht ontvangen. use-case bericht versturen.
Benodigdheden	Twee simulatie tools, message broker, deviceregistry
preconditie	Instellingen zijn ingesteld
Uitgevoerd door	Aiken Brus
Datum van uitvoering	12-07-2015

Test2.1 4.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> 1. Druk op de knop "Settings" 2. Verander de Device Name naar een willekeurige naam bijvoorbeeld: "sdhjewdf" 3. Druk op de knop "Sendevent" 	<p>In de output komt de volgende message te staan:</p> <pre> <messagetype> event </messagetype> <timesend> (datum en tijd) </timesend> <sender> sdhjewdf </sender> <method> EventData </method> <Body> { "data" (string) "TestData" (string) } </Body> </pre>	<pre> <messagetype> event </messagetype> <timesend> 12/07/15 10:32:45 </timesend> <sender> sdhjewdf </sender> <method> EventData </method> <Body> { "data" (string) "TestData" (string) } </Body> </pre>

Test Case	Verander het IP-adres en De simulatie tool moet opnieuw gaan connecten wanneer de settings zijn aangepast.
Beschrijving	Verander van het IP-adres in de settings menu. Use-case settings aanpassen.
Benodigdheden	Simulatie tool, twee message brokers.
preconditie	Allebei de message brokers hebben verschillende IP-adressen.
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 5.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> 1. Open de settings dialog met de knop "Settings" 2. Vul het IP-adres van de tweede message broker in het invoerveld 3. druk daarna op de knop "Save" om de settings op te slaan 	<p>Op de website <IP-adres>:8161/admin/connections.jsp staat je simulatie tool als een connectie</p>	<p>De simulatie tool staat tussen de connecties</p>

Test Case	Verander het IP-adres
Beschrijving	Test de grenswaardes die geaccepteerd wordt. Use-case settings aanpassen.
Benodigdheden	Simulatie tool
preconditie	
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 6.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> Open de settings dialog met de knop "Settings" vul de volgende grenswaardes in: <ol style="list-style-type: none"> (a) 0.0.0.1 (b) 255.255.255.254 druk op de knop "Save" 	De simulatie tool probeert connectie te maken	De simulatie tool probeert connectie te maken met de message broker op IP-adres a en b

Test Case	Verander het IP-adres foute waardes
Beschrijving	Test foute waardes. Settings aanpassen.
Benodigdheden	Simulatie tool
preconditie	
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 7.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> Open de settings dialog met de knop "Settings" vul de volgende foute waardes in: <ol style="list-style-type: none"> (a) random cijfers (b) 255.255.255.255 (c) 0.0.0.0 (d) random letters (e) Vul niks in druk op de knop "Save" 	je kan niet een groter getal invullen dan 255. Er ontstaat een error message bij foute waardes b, c en e. Bij foute waarde d wordt er niks in de ingevuld.	Hoger dan 255 kan inderdaad niet ingevuld worden. optie b, c en e heeft een error wanneer er op save word gedrukt. Er kunnen geen letters ingevuld worden.

Test Case	Verander port
Beschrijving	Verander de port nummer in de settings dialog. Use-case settings aanpassen.
Benodigdheden	Simulatie tool, twee message broker
preconditie	Allebei de message brokers hebben verschillende porten ingesteld
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 8.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> Open de settings dialog met de knop "Settings" Vul de port nummer van message broker 2 in druk op de knop "Save" 	Op de website <IP-adres>:8161/admin/connections.jsp staat je simulatie tool als een connectie	De verbinding word verbroken met message broker één en er word een connectie gemaakt met message broker twee.

Test Case	Verander port grenswaardes
Beschrijving	Test grenswaardes van de port nummers. Use-case settings aanpassen.
Benodigdheden	Simulatie tool
preconditie	
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 9.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> Open de settings dialog met de knop "Settings" Vul de port nummer met de volgende grens waardes: <ol style="list-style-type: none"> -1 0 1 65534 65535 65536 Vul niks in druk op de knop "Save" 	<p>De port nummer c, d, e doen het. De port nummers a, b en f kunnen niet ingevuld worden. g geeft een error message.</p>	<p>Als de port nummers van c, d en e worden ingevuld probeert de simulatie tool te connecten met de message broker. De port nummers van a, b en f kunnen niet ingevuld worden en bij g is er een error message zichtbaar.</p>

Test Case	Verander Device naam
Beschrijving	Verander de Device naam van de simulatie tool. Use-case settings aanpassen.
Benodigdheden	Simulatie tool
preconditie	
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 10.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> Open de settings dialog met de knop "Settings" Vul een random device naam in druk op de knop "Save" test dit door bijvoorbeeld een event bericht te sturen naar een andere message broker. 	<p>De simulatie tool heeft dat nu als naam</p>	<p>De simulatie tool saved the naam en gebruikt die. Dit is getest voor een event bericht te sturen naar een tweede simulatie tool.</p>

Test Case	Change uuid
Beschrijving	Verander de uuid van de simulatie tool. Use-case settings aanpassen.
Benodigdheden	Simulatie tool
preconditie	
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 11.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> 1. open de settings dialog met de knop "Settings" 2. Vull de volgende uuid in: 283add21-5412-4b5a-9e99-dce542692222 3. druk op de knop "Save" 	De simulatie tool heeft dat nu als uuid	De nieuwe uuid word geaccepteerd.

Test Case	Change uuid foute uuid
Beschrijving	Test een foute uuid. Use-case settings aanpassen.
Benodigdheden	Simulatie tool
preconditie	
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 12.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> 1. open de settings dialog met de knop "Settings" 2. Probeer de volgende configuraties in de uuid invoerveld: <ol style="list-style-type: none"> (a) Vul random cijfers en letters in (b) Laat de invoerveld leeg 3. druk op de knop "Save" 	Simulatie tool geeft een error message.	Bij allebij de opties geeft de simulatie tool een error message.

Test Case	Register device
Beschrijving	Simulatie tool registeren bij de deviceregistry
Benodigdheden	Simulatie tool, message broker, deviceregistry. Use-case registeren ARC-bus.
preconditie	Instellingen zijn ingesteld
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 13.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop "RegisterDevice"	De deviceregistry geeft the melding Device (device name) added to database en in de simulatie tool output komt een event message uitgeprint met de method device_registered.	Device tkh.vdg.device.simulation.1 283add21-5412-4b5a-9e99-dce566661111 added to database

Test Case	Update device
Beschrijving	Simulatie tool updaten bij de deviceregistry. Use-case update ARC-bus.
Benodigdheden	Simulatie tool, message broker, deviceregistry
preconditie	Instellingen zijn ingesteld
Uitgevoerd door	
Datum van uitvoering	

Test2.1 14.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop "UpdateDevice"	De deviceregistry geeft the melding Device <device name> updated in database en in de simulatie tool output komt een event message uitgeprint met de method device_status_info.	Device 283add21-5412-4b5a-9e99-dce566661111 updated in database

Test Case	Unregister device
Beschrijving	Simulatie tool updaten bij de deviceregistry
Benodigdheden	Simulatie tool, message broker, deviceregistry. Use-case afmelden ARC-bus.
preconditie	Instellingen zijn ingesteld
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 15.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop "UnRegisterDevice"	De deviceregistry geeft the melding Device <device name> removed from database en in de simulatie tool output komt een event message uitgeprint met de method device_unregister.	Device 283add21-5412-4b5a-9e99-dce566661111 removed from database

Test Case	Get All Devices
Beschrijving	Vraagt al de devices die in de database van de deviceregistry staat op. Use-case Opvragen devices deviceregistry.
Benodigdheden	Simulatie tool, message broker, deviceregistry
preconditie	Instellingen zijn ingesteld en er zijn devices geregistreerd bij de deviceregistry.
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 16.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop "GetAllDevices"	In de simulatie tool output komt een reply message uitgeprint met in de body alle devices die in de database staan.	<pre> <messagetype> reply </messagetype> <timesend> 12/07/15 13:23:45</timesend> <sender> (null) </sender> <Body> [{ "_id" (string) {283add21-5412-4b5a-9e99-dce566661111} (uuid) "product" (string) "Simulationtool" (string) "address" (string) "tkh.vdg.device.simulation.1" (string) "ipv4" (string) "172.21.243.86" (string) "vendor" (string) "VDG" (string) "name" (string) "Simulationtool" (string) "state" (string) "online" (string) "category" (string) "devices" (string) "persistent" (string) false (bool) "mac" (string) "08:62:66:4D:3A:CF" (string) }] </Body> </pre> <p>Er zit maar één device in de database dus de data klopt</p>

Test Case	Genereer uuid
Beschrijving	Vraag een uuid op van de deviceregistry
Benodigdheden	Simulatie tool, message broker, deviceregistry. Use-case Genereer uuid.
preconditie	Instellingen zijn ingesteld
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 17.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop "Generate-UUID"	In de simulatie tool output komt een reply message uitgeprint met in de body de uuid.	<pre><messagetype> reply </messagetype> <timesend>12/07/15 14:33:15 </timesend> <sender> (null) </sender> <Body> {bb7453bc-9233-4a19-906d-b25bbe4ae4a9} (uuid) </Body></pre>

Test Case	Clear response invoerveld
Beschrijving	Het leeg maken van de response invoerveld. Use-case Clear response
Benodigdheden	Simulatie tool
preconditie	er staat tekst in de response invoerveld
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 18.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop "Clear"	Het invoerveld is leeg.	Het invoerveld wordt leeg gemaakt.

Test Case	Afsluiten simulatie tool en de simulatie tool moet zich automatisch bij het afsluiten afmelden bij de eviceregistry.
Beschrijving	Afsluiten van de simulatie tool. Use-case Afsluiten. Use-case Afmelden device-registry.
Benodigdheden	Twee simulatie tools, één message broker, DeviceRegistry met database
preconditie	Alles is opgestart en hebben de juiste configuraties en een lege database
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.1 19.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> 1. Druk op de knop "Close" van simulatie tool 1 2. druk op de knop "GetAll-Devices" 	De simulatie tool is afgesloten en bij simulatie tool 2 staat de data van simulatie tool 1 niet in de reply.	<p>Na het close van simulatie tool 1: kwam bij de eerste test het zelfde uit als toen simulatie tool 1 nog aan stond.</p> <p>na bug fixing was de reply:</p> <pre> <messagetype> reply </messagetype> <timesend> 01/04/15 10:41:01 </timesend> <sender> (null) </sender> <Body> [{ "_id" (string) {283add21-5412-4b5a-9e99-dce566661122} (uuid) "product" (string) "Simulationtool" (string) "address" (string) "tkh.vdg.device.simulation.2" (string) "ipv4" (string) "172.21.243.86" (string) "vendor" (string) "VDG" (string) "name" (string) "Simulationtool" (string) "state" (string) "online" (string) "category" (string) "devices" (string) "persistent" (string) false (bool) "mac" (string) "08:62:66:4D:3A:CF" (string) }] </Body> </pre>

Test2.2 Plug-in test

Test Case	Bericht versturen
Beschrijving	Deze test is om te kijken of er een ARC-bus message gestuurd kan worden naar de ARC-bus broker. Use-case Berichten versturen.
Benodigdheden	een computers met VDG Sense, één computer met daar op de message broker, web plug-in tool, ARC-bus plug-in, simulatie tool.
preconditie	Senseen ARC-bus plug-zijn hebben al goede settings en simulatie tool is gestart
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.2 1.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop 'SendEvent' of 'SendRequest' van de web plug-in tool.	<p>Bij de simulatie tool wordt het ontvangen bericht ontvangen en geprint op het scherm</p> <pre><messagetype> event </messagetype> <timesend> (tijd en datum)</timesend> <sender> sense naam </sender> <method> RequestData </method> <Body> null </Body></pre> <p>of</p> <pre><messagetype> request </messagetype> <timesend> (tijd en datum)</timesend> <sender> sense naam </sender> <method> RequestData </method> <Body> { "data" (string) "TestData" (string) } </Body></pre>	<p>De uitkomst van SendEvent is:</p> <pre><messagetype> event </messagetype> <timesend> 12/07/15 15:25:15 </timesend> <sender> tkh.vdg.device.7ae0552a-ab1c-4b06-9af7-9012a9d2a22e </sender> <method> RequestData </method> <Body> null </Body></pre> <p>en de uitkomst van SendRequest is:</p> <pre><messagetype> request </messagetype> <timesend> 12/07/15 15:25:16 </timesend> <sender> tkh.vdg.device.7ae0552a-ab1c-4b06-9af7-9012a9d2a22e </sender> <method> RequestData </method> <Body> { "data" (string) "TestData" (string) } </Body></pre>

Test Case	Bericht ontvangen
Beschrijving	Deze test is om te kijken of een bericht ontvangen kan worden en de juiste data uit het bericht gelezen kan worden. Use-case Berichten ontvangen.
Benodigdheden	VDG sense, ARC-bus plug-in, simulatie tool, web plug-in tool
preconditie	Sense en ARC-bus plug-zijn hebben al goede settings en simulatie tool is gestart met de goede settings
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.2 2.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop 'SendEvent' van de simulatie tool van de web plug-in tool. Al je op 'SendRequest' drukt met de naam van Sense in de invoerveld gebeurt er niks	In de console en de log file van de sensepluginmanager wordt een melding gegeven wanneer het bericht is ontvangen. Als de console niet draait dan kan ik het log bestand van de plug-in terug gevonden worden dat er een bericht ontvangen is.	in de consle en log file staat SenseMessageParser event ontvangen from tkh.vdg.device.simulation.1 with method EventData en als op 'SendRequest' drukt gebeurt er niks.

Test Case	Registeren deviceregistry
Beschrijving	Deze test is om te kijken of er op de ARC-bus geregistreerd kan worden via de plug-in. Use-case Registeren deviceregistry.
Benodigdheden	VDG Sense, simulatie tool, ARC-bus plug-in, web plug-in tool
preconditie	VDG Sense en ARC-bus plug-zijn hebben al goede settings en simulatie tool is gestart
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.2 3.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop 'RegisterDevice' van de web plug-in tool.	De simulatie tool zou een device_status_info moeten krijgen als de VDG Sense systeem al staat geregistreerd of anders device_registered	De simulatie tool krijgt een device_status_info bericht dat VDG Sense al geregistreerd is

Test Case	Afmelden deviceregistry
Beschrijving	Deze test is om te kijken of er afgemeld kan worden bij de deviceregistry. Use-case Afmelden deviceregistry.
Benodigdheden	VDG Sense, simulatie tool, ARC-bus plug-in, web plug-in tool
preconditie	VDG Sense, en simulatie tool is gestart
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.2 4.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop 'UnRegisterDevice' van de web plug-in tool.	In de log bestand van ArcBus komt te staan dat er een event is ontvangen in de DeviceRegistryMessageParser met de method device_unregistered	Er is een bericht ontvangen in de DeviceRegistryMessageParser met de method device_unregistered

Test Case	UpdateDevice
Beschrijving	Deze test is om te kijken of er een update device bericht gestuurd kan worden naar de DeviceRegistry. Use-case update deviceregistry.
Benodigdheden	VDG Sense, simulatie tool, ARC-bus plug-in, web plug-in tool
preconditie	VDG Sense, en simulatie tool is gestart
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.2 5.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop 'UpdateDevice' van de web plug-in tool.	In de log bestand van ArcBus komt te staan dat er een event is ontvangen in de DeviceRegistry-MessageParser met de method device_status_info	Er is een bericht ontvangen in de DeviceRegistryMessageParser met de method device_status_info

Test Case	GetAllDevices
Beschrijving	Deze test is om te kijken of alle devices opgevraagd kunnen worden bij de deviceregistry, Use-case opvragen devices deviceregistry.
Benodigdheden	VDG Sense, simulatie tool, ARC-bus plug-in, web plug-in tool
preconditie	VDG Sense, en simulatie tool is gestart
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.2 6.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop 'GetAllDevices' van de web plug-in tool.	In de log bestand van ArcBus komt te staan dat er een reply is ontvangen in de DeviceRegistryMessageParser van de zender (null)	Er is een reply ontvangen in de DeviceRegistryMessageParser van de zender (null)

Test Case	GenerateUUID
Beschrijving	Deze test is om te kijken of er een UUID opgevraagd kan worden aan de DeviceRegistry. Use-case genereer uuid.
Benodigdheden	VDG Sense, simulatie tool, ARC-bus plug-in, web plug-in tool
preconditie	VDG Sense, en simulatie tool is gestart
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test2.2 7.1	Verwachte uitkomst	Officiële uitkomst
Druk op de knop 'Generate-UUID' van de web plug-in tool.	In de log bestand van ArcBus komt te staan dat er een reply is ontvangen in de DeviceRegistryMessageParser van de zender (null)	Er is een reply ontvangen in de DeviceRegistryMessageParser van de zender (null)

Test3 System test

Test Case	Settings aanpassen via Sense GUI
Beschrijving	Deze test is om te kijken of de settings aangepast kunnen worden via de Sense GUI binnen 25 seconden. Use-case Settings aanpassen.
Benodigdheden	Sense met ARC-bus plug-in
preconditie	Sense is opgestart
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test3 1.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> 1. Ga naar de setup tab 2. Ga naar de Plugins tab 3. click op de plugin ArcBus 4. Hij hij nog niet active is maak hem active 5. Vul in de text box 'IP-adres' de IP-adres van de message broker 6. Vul in de text box 'port' de port adres van de message broker 7. druk op de knop 'save' 	In de ArcBus.xml file in de settings staan de nieuwe settings.	De ingevulde settings staan in de ArcBus.ini file.

Test Case	Switch layout 2 van systeem twee
Beschrijving	Stuur een event switch layout event van systeem één naar systeem twee. Use-cases berichten versturen, berichten ontvangen, ARC-bus berichten converteren.
Benodigdheden	twee systemen met VDG Sense geïnstalleerd, één message broker.
preconditie	Allebei de VDG Sense systemen hebben de ARC-Bus settings van de message broker, twee layouts één met de naam secondlayout, de tweede systeem is live modus met de andere layout zichtbaar
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test3 2.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> 1. Open de link http://127.0.0.1/ArcBus/test.html in je browser 2. Druk op de knop "VMs-switchlayout2" 	De secondlayout moet zichtbaar zijn in de live modus.	De secondlayout is zichtbaar op de live modus.

Test Case	Switch layout 1 van systeem twee
Beschrijving	
Benodigdheden	Stuur een event switch layout event van systeem één naar systeem twee. Use-cases berichten versturen, berichten ontvangen, ARC-bus berichten converteren.
preconditie	Allebei de VDG Sense systemen hebben de ARC-Bus settings van de message broker, twee layouts één met de naam Firstlayout, de tweede systeem is live modus met de andere layout zichtbaar
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test3 3.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> 1. Open de link <code>http://127.0.0.1/ArcBus/test.html</code> in je browser 2. Druk op de knop "VM-switchlayout1" 	De secondlayout moet zichtbaar zijn in de live modus.	De secondlayout is zichtbaar op de live modus.

Test Case	Execute een macro van systeem twee
Beschrijving	Stuur een event van systeem één naar systeem twee waarmee een macro uitgevoerd wordt. Use-cases berichten versturen, berichten ontvangen, ARC-bus berichten converteren.
Benodigdheden	twee systemen met VDG Sense geïnstalleerd, één message broker.
preconditie	Allebei de VDG Sense systemen hebben de ARC-Bus settings van de message broker, Een macro die zichtbaar uitgevoerd wordt.
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test3 4.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> 1. Open de link <code>http://127.0.0.1/ArcBus/test.html</code> in je browser 2. Druk op de knop "macroexecute" 	De gemaakte macro moet uitgevoerd zijn.	De macro word inderdaad uitgevoerd.

Test Case	Send signal SenseNotClosedGracefully
Beschrijving	Stuur een event bericht met SenseNotClosedGracefully. Use-cases berichten versturen, berichten ontvangen, ARC-bus berichten converteren.
Benodigdheden	twee systemen met VDG Sense geïnstalleerd, één message broker.
preconditie	Allebei de VDG Sense systemen hebben de ARC-Bus settings van de message broker, Een server macro met als signal source type system en signal name Sense Not closed gracefully, Met action sourcetype device en action name Send generic event met parameter 1, Een layout met een event panel
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test3 5.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> 1. Open de link <code>http://127.0.0.1/ArcBus/test.html</code> in je browser 2. Druk op de knop "SenseNotClosedGracefully" 	Er ontstaat een generic event met value 1 en deze is zichtbaar in de event panel	De generic event is ontstaan en is zichtbaar in de event panel

Test Case	Send signal ConnectionLost
Beschrijving	Stuur een event bericht met ConnectionLost. Use-cases berichten versturen, berichten ontvangen, ARC-bus berichten converteren.
Benodigdheden	twee systemen met VDG Sense geïnstalleerd, één message broker.
preconditie	<ul style="list-style-type: none"> • Allebei de VDG Sense systemen hebben de ARC-Bus settings van de message broker • Een device met id 1, location test • De tweede VDG Sense systeem heeft een • Een layout met een event panel.
Uitgevoerd door	Aiken Brus
Datum van uitvoering	7-12-2015

Test3 6.1	Verwachte uitkomst	Officiële uitkomst
<ol style="list-style-type: none"> 1. Open de link <code>http://127.0.0.1/ArcBus/test.html</code> in je browser 2. Druk op de knop "Connectionlosttest" 3. Na enkele seconden druk op de knop "Stop_Connectionlosttest" 	Enkele ConnectionLost event in de event panel	Een connectionlost event van het device dummydevice. Dit device heeft device id 1

User manual C++ SDK

Student: Aiken Brus (10007695)

Datum: 21-03-2016

Plaats: Zoetermeer

Afstudeerverslag Haagse Hogeschool

Opleiding: Technische informatica

Bedrijf: VD Security B.V.

Opdrachtgever: Timme Grippink

Begeleidende examiner: Cobie van der Hoek

Tweede examiner: John Visser

Inhoudsopgave

1	Inleiding	1
2	SDK	2
	Bijlage	4

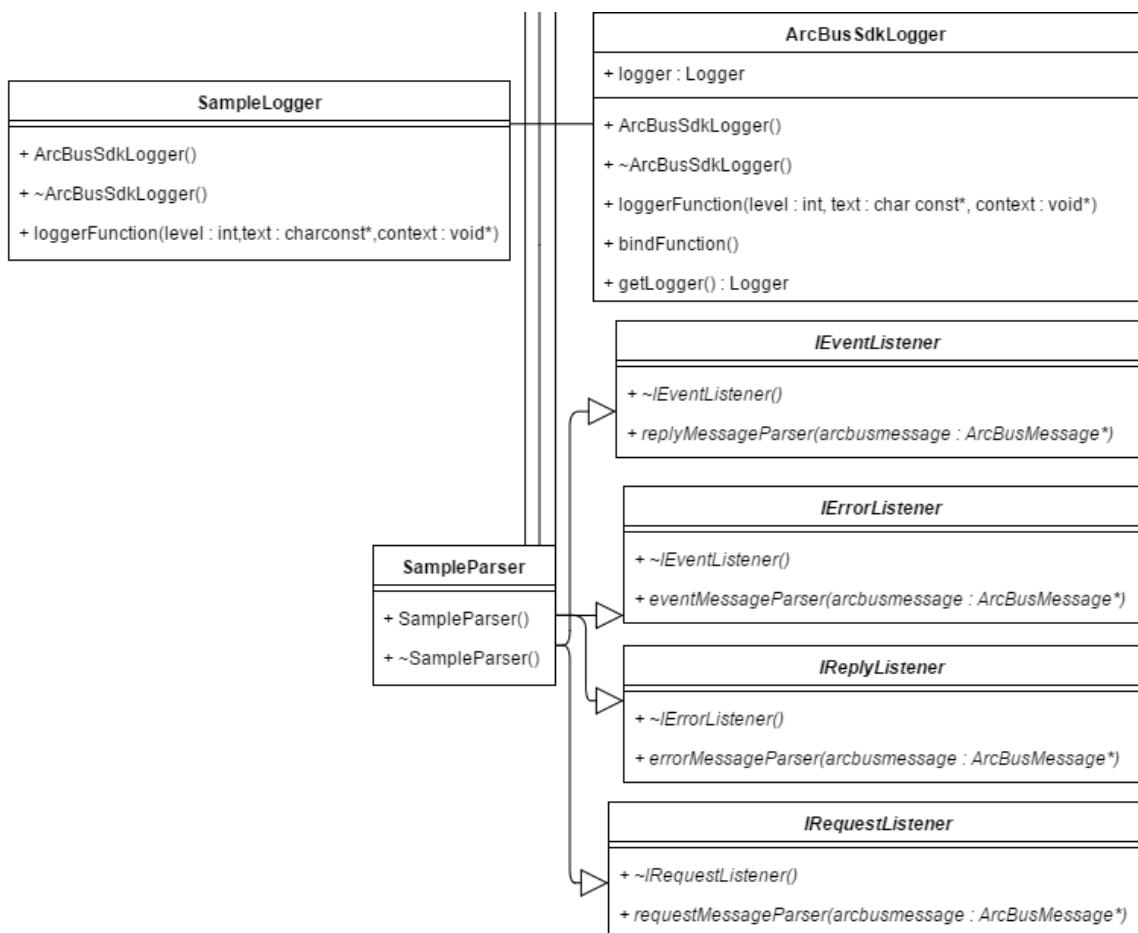
1 Inleiding

Het doel van dit document is om een TKH developer te informeren hoe de C++ SDK geïntegreerd kan worden in hun eigen code. Er is maar één hoofdstuk in dit verslag en dat is gelijk het hoofdstuk waar in staat wat er gedaan moet worden om de C++ SDK te integreren. In de Bijlage staat de Klassendiagram van de C++ SDK weergegeven.

2 SDK

De C++ SDK is een wrapper gebouwd om de C SDK heen. Dit zorgt er voor dat de C SDK omgezet wordt naar object georiënteerd. De C++ SDK maakt gebruik van de C SDK om berichten te verzenden. Om de

C++ SDK te kunnen gebruiken moeten er een paar dingen bedacht worden tijdens het ontwerpen van je product. Je moet nadenken of je een zelf gemaakte logger wil gebruiken of de huidige in de C++ SDK. De huidige logger print alles naar cout. Net als in figuur 1 kan je een logger klasse maken. Om er voor te zorgen dat de C++ SDK jouw logger gebruikt moet je de regel van tabel 1 gebruiken. waarbij SampleLogger de naam van jouw klasse is en loggerFunction de naam van de functie waar je mee logt. Om berichten te kunnen verwerken moeten er parser klassen gemaakt worden. Er zijn vier listeners waar de parser van kan overerven (zie figuur 1). Je hoeft alleen die listeners te overerven waar van je de soort berichten wil ontvangen. Om de ontvangen berichten binnen te krijgen in de parser moet je parser klassen geregistreerd worden bij de messenger. Er kunnen oneindig parsers geregistreerd worden. Dit wordt gedaan in tabel 3. Voor dat de parsers geregistreerd kunnen worden moet er eerst een messenger gemaakt worden. Dit wordt gedaan in tabel 2. Met de “registerEventListener” kan er aangemeld worden om de eventberichten te ontvangen. Bij deze functie worden er twee parameters mee gegeven. De eerste is net als bij de rest het object van de parser klassen. De tweede parameter is de naam van het apparaat of server. In de eventparser functie komen alleen berichten van die specifieke zender. De > in het voorbeeld betekent dat hij alle berichten wilt hebben die beginnen met wat er voor > staat. In dit voorbeeld staat er niks voor dus ontvangt hij alle berichten. Bijvoorbeeld bij device.> ontvangt de functie alleen berichten van zenders waarvan de naam beginnen met device. Amqp is het protocol dat gebruikt wordt, 127.0.0.1 is het ip-adres van de message broker, 5672 de port nummer van de message broker, device.name.example die naam die je wilt geven aan je device, 00000000-0000-0000-0000-000000000000 plaats hier een random UUID, "certificate.der" "privatekey.der" zijn files voor ssl.



Figuur 1: Parser en Logger part

```
stdFunctionLogger = std::bind(&SampleLogger::loggerFunction, this, _1, _2, _3);
```

Tabel 1: Bind logger functie

```

arcBusMessenger = ArcBusMessenger::getArcBusMessenger();
arcBusMessenger->init("amqp"
                    , "127.0.0.1"
                    , "5672"
                    , "device.name.example"
                    , "00000000-0000-0000-0000-000000000000"
                    , "certificate.der"
                    , "privatekey.der");
return arcBusMessenger->start();

```

Tabel 2: Load messenger

```

SampleParser = new GeneralMessageParser();
arcBusMessenger->registerErrorListener(SampleParser);
arcBusMessenger->registerEventListener( SampleParser, ">");
arcBusMessenger->registerReplyListener( SampleParser);
arcBusMessenger->registerRequestListener( SampleParser);

```

Tabel 3: Load sampleparser

In de bijlage vindt je de klassendiagram van de C++ SDK.

Bijlage

