

Afstudeerdossier



ONVIF Plug-in

Het implementeren van de ONVIF camera plug-in voor VDG Security
om de nieuwe plug-in architectuur in gebruik te nemen

Auteur: Tim Kesteloo

Datum: 13/06/2016

Opleiding: Technische Informatica

Faculteit: Academie voor ICT & Media, Haagse Hogeschool Delft

Begeleiders: J.J. van der Hoek, A. Andrioli

Referaat

Titelbeschrijving

Tim Kesteloo, "ONVIF Plug-in". Afstudeerverslag Technische Informatica, Haagse Hogeschool te Delft, 2016.

Samenvatting

Dit document is het procesverslag van de afstudeeropdracht die Tim Kesteloo heeft uitgevoerd voor de opleiding Technische Informatica bij VDG Security B.V.

Tijdens de afstudeerperiode is voor VDG Security B.V. onderzoek gedaan naar het implementeren van het ONVIF protocol in de nieuwe plug-in architectuur.

Tijdens de afstudeerperiode is in de nieuwe architectuur de ONVIF Profile S Device Management-, Media- en PTZService, WS-UsernameToken, en WS-Discovery volledig geïmplementeerd.

Descriptoren

Afstudeerverslag, VMS, ONVIF, ONVIF Profile S, C++

Voorwoord

Voor u ligt het afstudeerverslag 'ONVIF Plug-in - Het implementeren van de ONVIF camera plug-in voor VDG Security om de nieuwe plug-in architectuur in gebruik te nemen'. Hierin komen het onderzoek naar ONVIF en de implementatie van de plug-in aan bod.

Dit verslag is geschreven voor mijn afstuderen van de opleiding Technische Informatica aan de Haagse Hogeschool te Delft. Hiervoor ben ik van februari 2016 tot juni 2016 werkzaam geweest bij VDG Security B.V. (VDG) te Zoetermeer.

Met veel hulp van anderen is deze afstudeerperiode tot stand gekomen. Ik wil graag de tijd nemen om deze hier expliciet te benoemen.

Ik wil graag Wilco Gorthuis, collega bij VDG, bedanken voor het enorme enthousiasme om mij te helpen met ONVIF, te ondersteunen op het gebied van keuzes en beslissingen, en altijd tijd voor mij vrij te maken om in overleg te gaan over de mogelijkheden en beperkingen.

Ik bedank Robin Hermann, collega en mentor bij VDG, die mij heeft geadviseerd en begeleid tijdens het hele traject en bij de totstandkoming van dit document.

Ook wil ik alle andere collega's van het R&D team bedanken, die mij altijd hebben geholpen wanneer ik deze nodig heb gehad.

En natuurlijk ben ik erg dankbaar dat VDG mij de mogelijkheid heeft geboden bij hen mijn afstudeeropdracht uit te mogen voeren, mij het vertrouwen te geven om dit zowel zelfstandig als in teamverband uit te kunnen voeren, en mij hierbij op te nemen als collega.

Tim Kesteloo

Zoetermeer

03/06/2016

Inhoudsopgave

1.	Inleiding	5
2.	Afkortingenlijst	6
3.	Organisatie	7
3.1.	Algemeen	7
3.2.	Ontwikkelmethode	7
4.	Opdrachtoomschrijving	9
4.1.	Situatie bij aanvang	9
4.2.	Mogelijke oplossingsmethoden	9
4.3.	Doelstelling	9
4.4.	Werkzaamheden	9
5.	Kennismaking en Plan van Aanpak	10
5.1.	VDG Sense	10
5.2.	ONVIF	10
5.3.	Resultaat kennismaking	12
6.	Sprint 1: Onderzoek	15
6.1.	Vervolg ONVIF onderzoek	15
6.2.	Proof of Concept	17
6.3.	Sprint resultaat	20
7.	Sprint 2: Design	23
7.1.	Ontwerp	23
7.2.	Veranderingen ONVIF Profile S Specification	26
7.3.	Sprint resultaat	26
8.	Sprint 3: Ontwikkeling DeviceService	27
8.1.	Aanpassingen ontwerp	27
8.2.	Voorstel verandering architectuur	27
8.3.	Testplan	28
8.4.	Sprint resultaat	29
9.	Sprint 4: Ontwikkeling MediaService	31
9.1.	Aanpassingen ontwerp	31
9.2.	Discovery	34
9.3.	Sprint resultaat	35
10.	Sprint 5: Ontwikkeling PTZService	37
10.1.	Aanpassingen ontwerp	37
10.2.	Sprint resultaat	38
11.	Sprint 6: Ontwikkeling Events & Discovery	40
11.1.	Aanpassingen ontwerp	40
11.2.	Sprint resultaat	41
12.	Sprint 7: Testing	43
12.1.	Proces	43
12.2.	VDG Plug-in Test Tool	43
12.3.	ONVIF Client Test Tool	43
12.4.	Sprint resultaat	44
13.	Conclusie en aanbeveling	46
14.	Evaluatie	47
15.	Literatuurlijst	50
16.	Bijlagen	51

1. Inleiding

In onze huidige samenleving zijn camera's bijna niet meer weg te denken. Iedereen heeft er te allen tijde één bij zich op z'n mobiele telefoon. Daarnaast zijn er ook veel camera's die minder mobiel zijn, zoals beveiligingscamera's op en in gebouwen of meetcamera's bij een snelweg of in een fabrieksomgeving.

Een aspect dat snel over het hoofd gezien kan worden, is de communicatie tussen de camera en het systeem dat deze bestuurt en/of gebruikt. Dit komt omdat de gebruiker hier vaak niet mee in aanraking komt; een webpagina of applicatie die alle opties van de camera met één druk op de knop laat besturen is vaak de oplossing.

Echter, deze oplossing is vaak specifiek gemaakt voor het model waarmee gewerkt wordt. Elke camerafabrikant schrijft voor zijn eigen camera's een eigen communicatieprotocol, wat ervoor zorgt dat wanneer zo'n camera gebruikt moet worden, kennis van het protocol aanwezig moet zijn.

VDG is een massagebruiker van dit soort camerasystemen door deze op te nemen in haar Video Management System (VMS) en zij beperkt zich niet tot enkele fabrikanten of modellen. Hierdoor komt het vaak voor dat een nieuw protocol geïmplementeerd moet worden, wat erg veel tijd kost.

In dit verslag wordt duidelijk hoe tijdens de duur van de afstudeerperiode dit probleem is aangepakt door in de nieuwe architectuur van VDG de ONVIF specificatie te implementeren.

Dit document is een chronologische vertelling van de gebeurtenissen die hebben plaatsgevonden en keuzes die zijn gemaakt. Hierbij is onderscheid gemaakt tussen de verschillende sprints zoals deze beschreven zijn in de planning.

2. Afkortingenlijst

API:	Application Programming Interface; verzameling definities op basis waarvan een programma kan communiceren met een ander programma
Dome:	Dome-camera; Koepelvormige camera die PTZ ondersteunt
DRY:	Don't Repeat Yourself; programmeerprincipe om herhaling van code te minimaliseren
GPIO:	General Purpose Input/Output; Een pin die zowel als input als output geconfigureerd kan worden
H264:	Advanced Video Coding (AVC); Videocodec met sterke compressie van videobeelden
Kanban:	Iteratieve ontwikkelmethodiek gefocust op productcycles
MJPEG:	Motion JPEG; Videocodec die bestaat uit opvolgende JPEG plaatjes
MPEG4:	MP4; videocodec voor compressie van audio en video naar een videoformaat.
NTP:	Network Time Protocol; Protocol gebruikt voor tijdsynchronisatie tussen systemen
ONVIF:	Open Network Video Interface Forum
PACS:	Physical Access Control System; Systemen die mensen of objecten authenticeren en toegang verlenen tot een fysieke locatie.
PTZ:	Pan, Tilt, Zoom; Het besturen van een camera over de x-as, y-as en de lens zoomen.
Qt:	QtCreator; een IDE voor het ontwikkelen code.
R&D:	Research and Development (afdeling)
Scrum:	Iteratieve ontwikkelmethodiek gefocust op sprints en backlog items
Scrumban:	Iteratieve ontwikkelmethodiek die aspecten van Scrum en Kanban combineert.
Sense:	Het hoofdproduct van VDG, haar VMS
SOAP:	Simple Object Access Protocol; protocol voor communicatie tussen verschillende protocollen m.b.v XML berichten, veelal over HTTP verstuurd.
SRP:	Single Responsibility Principle; programmeerprincipe om elk onderdeel één verantwoordelijk te geven, en dat dit het enige onderdeel is die hier verantwoordelijk voor is
TKH Group:	Technologiebedrijf TKH Group NV; Groep ondernemingen waar VDG deel van is
VCA:	Video Content Analysis; automatisch video analyseren en events detecteren
VDG:	Video Development Group; VDG Security B.V.; Het bedrijf waar de afstudeeropdracht is uitgevoerd
VMS:	Video Management System; Systeem waar videobeelden in worden bekeken, voornamelijk gebruikt in beveiligingssituaties
WIP:	Work in Progress
WS-....:	OASIS Web Service Security standaarden. De standaarden die aan bod komen zijn WS-Security (WS-UsernameToken) , WS-Addressing , WS-BaseNotification en WS-Discovery .
WSDL:	Web Services Description Language; Een platform-onafhankelijke taal voor het beschrijven van de functies van een Web Service.
XML:	Extensible Markup Language; Standaard voor de syntax van formele opmaaktalen in de vorm van platte tekst.

3. Organisatie

3.1. Algemeen

VDG Security B.V. (VDG) staat voor Video Development Group Security B.V. Haar hoofdproduct is de applicatie VDG Sense, wat 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 gezondheidszorgmarkt.

VDG is onderdeel van het technologiebedrijf TKH Group. De TKH Group bestaat uit een groot aantal bedrijven, met in totaal meer dan 5.000 werknemers. Zij bestaat uit drie groepen: Telecom Solutions, Industrial Solutions en Building Solutions. VDG Security valt onder de Building Solutions groep. Binnen deze groep wordt nauw samengewerkt met bedrijven die ook actief zijn in de security branche. Tijdens deze samenwerking wordt zoveel mogelijk de technologie van de zusterbedrijven gecombineerd met de video technologie van VDG.

VDG bestaat uit vier afdelingen; Research and Development, Sales, Support en Magazijn. De opdracht zal uitgevoerd worden binnen het team Research and Development (R&D).

3.2. Ontwikkelmethode

VDG R&D maakt gebruik van Scrumban, een methodiek die van origine was bedacht als overstap van Scrum naar Kanban, maar tegenwoordig als losstaande methode gebruikt kan worden.^[1] Waar Scrum erg gericht is op het indelen van sprints en deze te vullen met backlog items, is Scrumban vrijer op het gebied van planningsroutine en meer gefocust op het uitvoeren van taken volgens een workflow. VDG heeft ervoor gekozen om deze taken te verbinden aan de planning van de software release, die maandelijks plaatsvindt.

VDG heeft binnen Scrumban een eigen workflow ontwikkeld, waarbij elke feature of bug de volgende fases doorloopt:

- Ready for Specs: Bugs zijn omschreven door Support en features zijn voorgesteld door een klant of collega in de backlog. Een korte omschrijving is gegeven.
- Specs in Progress: Een ontwikkelaar wordt toegekend aan de taak, die nu verantwoordelijk is voor het uitwerken van de specificaties voor deze taak. Wanneer dit volbracht is, wordt de taak naar Ready for Spec Review gezet.
- In Spec Review: Een andere ontwikkelaar wordt toegewezen aan de taak, die nu de gemaakte specs evalueert. Als deze zijn goedgekeurd, wordt de taak Ready for Development. Zo niet, dan gaat deze terug naar Ready for Specs.
- Dev in Progress: Een ontwikkelaar is toegewezen om de taak volgens de goed-gekeurde specs uit te voeren. Wanneer dit volbracht is, is deze Ready for Review.
- In Code Review: Een andere ontwikkelaar neemt de gemaakte code door. Als deze wordt goedgekeurd, wordt hij Ready for Testing. Indien er issues worden gevonden, wordt deze teruggeplaatst naar Ready for Development.
- Testing: Een externe tester bekijkt of alles werkt zoals beschreven. Als dit het geval is, dan is de taak Ready for Release. Als hier issues worden gevonden, gaat deze terug naar Ready for Development.
- Ready for Release: Aan het einde kan een re-test plaatsvinden, om te controleren of de combinatie van afgeronde taken ook goed samenwerken. Indien hier nog issues uitkomen, kan de taak wederom terugkeren naar Ready for Development.

Dit proces is visueel uitgewerkt in [figuur 1](#), zoals deze in de management software die VDG gebruikt, JIRA, beschreven staat. Hierin geven de pijlen aan wat de mogelijke transities zijn tussen de verschillende fases.



Figuur 1: VDG Workflow

VDG houdt een work-in-progress (WIP) van 1 aan. Dit betekent dat een ontwikkelaar te allen tijde maar één taak actief mag hebben. Wanneer deze aan een andere taak wil werken, dient de huidige taak te worden afgerond of teruggezet naar de vorige fase. Dit wordt weergegeven via de pijlen die terug naar boven wijzen.

VDG streeft ernaar elke maand een nieuwe release uit te brengen. Hierbij ligt in de eerste twee weken de focus op specs en ontwikkeling. De twee opvolgende weken prioriteren testen en bugfixes.

4. Opdrachtomschrijving

Aan het begin van de afstudeerperiode is het afstudeerplan [\[Bijlage A. Afstudeerplan\]](#) opgezet. In dit document is de initiële opdracht als volgt omschreven.

4.1. Situatie bij aanvang

Regelmatig krijgt de R&D afdeling van VDG Security de opdracht om een nieuwe camera te integreren in de VDG Sense software. Elke camerafabrikant ontwikkelt zijn eigen API (application program interface) in de camera waar een integrator (zoals VDG Security) zich aan moet conformeren. Aangezien VDG Security een groot aantal klanten bedient, met een breed scala aan verschillende type camera's, is het een tijdrovende klus om al deze drivers zelf te ontwikkelen en bij te houden.

VDG wil dit proces verbeteren, zodat minder van de kostbare ontwikkeltijd wordt gebruikt voor camera-integratie. Zo wil men dit proces kunnen uitbesteden aan derden of een makkelijkere manier ontwikkelen voor het integreren van camera's met het VMS.

4.2. Mogelijke oplossingsmethoden

Om het proces van camera integratie te versnellen zijn twee oplossingen bedacht:

- Ten eerste heeft VDG een nieuwe architectuur bedacht om het integreren van nieuwe cameradrivers te versimpelen.
- Ten tweede heeft VDG interesse in ONVIF, een specificatie die zowel voor de 'Client' als het 'Device' beschrijft hoe deze moeten communiceren.

4.3. Doelstelling

Er is gekozen om deze twee oplossingen te combineren. Dit resulteert in de opdracht om in de nieuw ontworpen architectuur een ONVIF plug-in te maken.

Het doel van deze plug-in is tweeledig; zij dient als test voor de architectuur, wat de mogelijkheid biedt om deze te verbeteren en/of uit te breiden. Dit moet leiden tot een definitieve architectuur die vervolgens gebruikt kan worden voor het implementeren van nieuwe camera drivers. Hierin zal de ONVIF plug-in de eerste functionele driver worden, waardoor ONVIF camera's via Sense aangestuurd kunnen worden.

4.4. Werkzaamheden

De afstudeeropdracht omvat de volgende activiteiten die uitgevoerd dienen te worden:

- Onderzoek naar ONVIF
- Onderzoek naar wensen VDG m.b.t. camera integratie en ONVIF
- Onderzoek nieuwe plug-in-architectuur
- Ontwerpen plug-in implementatie
- Plug-in implementeren
- Testen plug-in implementatie
- Documenteren van gemaakte keuzes

5. Kennismaking en Plan van Aanpak

De eerste fase (gedurende twee weken) van het afstudeertraject is ingericht als kennismaking. Hieronder vallen het opstarten binnen het bedrijf, kennismaken met haar systemen en doelen, een eerste blik op ONVIF werpen en de huidige implementatie, en aan de hand van deze informatie het opzetten van een Plan van Aanpak.

5.1. VDG Sense

Om een beeld te krijgen van de organisatie en haar product, heeft een introductie tot VDG Sense (Sense), het VMS van VDG, plaatsgevonden. Dit is belangrijk voor het vaststellen van het probleemdomain van de afstudeeropdracht.

Hiervoor zijn meerdere mediums gebruikt.

Ten eerste heeft een training plaatsgevonden. Hierbij heeft een support medewerker de afstudeerder de basistraining gegeven voor het configureren en gebruiken van Sense zoals deze aan klanten wordt verstrekt. Hierbij zijn onder andere aan bod gekomen: het verbinden met een camera, configureren van een camera, Video Content Analysis (VCA) en de videowall. Deze aspecten worden uitgelegd in [\[Bijlage B. Introductie VDG Sense\]](#).

Daarnaast is samen met de ONVIF expert een bug opgepakt betreffende een ONVIF camera. Hierbij is getest of alle functionaliteiten van de camera goed werken in de huidige ONVIF implementatie van VDG. Tijdens het testen van de camera is kennis gemaakt met de ONVIF Device Manager applicatie (een losstaande applicatie van ONVIF die ONVIF camera's kan aansturen) en hoe deze gebruikt kan worden voor testdoeleinden. Daarnaast is in Sense kennisgemaakt met de huidige ONVIF implementatie en de manier hoe een camera wordt geconfigureerd en aangestuurd.

5.2. ONVIF

5.2.1. Methode

Bij de start van het afstudeertraject was er geen kennis van ONVIF. Om hier meer over te leren is besloten tijdens de kennismaking een klein onderzoek naar ONVIF te doen.

Met het onderzoek wil men de volgende vragen beantwoorden:

1. Wat is ONVIF?
 - a. Door wie is het opgezet en vanuit welk probleem/wens?
2. Hoe is ONVIF opgebouwd?
3. Hoe wordt ONVIF toegepast?

Bovenstaande vragen zijn beantwoord door bronnenonderzoek te doen. Hiervoor is voornamelijk gebruik gemaakt van de documentatie die ONVIF aanbiedt op haar [website](#), zoals de ONVIF Programmer's Guide [\[5\]](#) en de ONVIF Specifications. Daarnaast zijn er externe bronnen gebruikt om informatie over ONVIF te verkrijgen en de opinies over ONVIF te peilen. De bronnen die hiervoor zijn gebruikt en de resultaten worden uitgebreid behandeld in [\[Bijlage C. ONVIF Research Document\]](#). Een kort resultaat van het onderzoek en het antwoord op de gestelde vragen zal worden gegeven in het volgende hoofdstuk [\[5.2.2\]](#).

Naar aanleiding van het onderzoek is gebleken dat er met veel systemen gewerkt zal worden waar de afstudeerder nog geen ervaring mee heeft. Er is geconcludeerd dat het noodzakelijk is om een proof of concept in te plannen om hier vóór de start van het ontwikkeltraject mee bekend te raken. Wanneer hier

ervaring in is opgedaan kan een betere inschatting gemaakt worden van het werk, waar in de planning rekening mee kan worden gehouden.

Tijdens het proof of concept wil men de volgende taken uitvoeren:

- Het kennismaken met de nieuwe plug-in architectuur
- Web service functionaliteit
 - Berichten versturen via een SOAP service
 - WSDL conversie naar code
 - Aanroepen gegenereerde WSDL code
- Het testen van camera functionaliteit

Daarnaast zal er een vervolgonderzoek moeten plaatsvinden. Hierin moeten de volgende vragen beantwoord worden:

- Welke requirements stelt VDG aan de ONVIF plug-in?
- Welke ONVIF features wil VDG ondersteunen?
- Welke ONVIF versie (specification set) zal ontwikkeld worden?

5.2.2. Onderzoek resultaten

[\[Bijlage C. ONVIF Research Document\]](#)

Wat is ONVIF?

Door wie is het opgezet en vanuit welk probleem/wens?

ONVIF is opgezet in 2008 door drie grote bedrijven; Axis, Bosch en Sony. Zij merkten op dat het grote verschil in communicatieprotocollen dat geschreven wordt door fabrikanten voor hun camera's, zoals toen en nu heel normaal is, enorm veel tijd kost. Dit geldt voor zowel de ontwikkelaar van het protocol als de integrator van het protocol. Vanuit de wens om hier één standaard voor te ontwikkelen is ONVIF ontstaan. Deze beschrijft hoe de communicatie plaatsvindt en welke functionaliteiten een apparaat ("Device") of applicatie ("Client") hiervoor moet implementeren.

Hoe is ONVIF opgebouwd?

De functionaliteiten zijn onderverdeeld in verschillende profielen, die elk bestaan uit meerdere features. Momenteel zijn dit drie bestaande profielen en twee in de testfase, respectievelijk profielen S,C,G en profielen A en Q.

- Profiel S is het meest voorkomende profiel. Deze is onder andere verantwoordelijk voor video & audio streaming, PTZ, relay outputs, configuratie en multicast.
- Profiel C is gericht op systemen die onderdeel zijn van een Physical Access Control System (PACS). Hieronder valt het aansturen van deuren, alarmen en events afhandelen.
- Profiel G beschrijft geavanceerde video functionaliteiten, zoals edge-recording en storage.

Alhoewel ONVIF vaak gebruikt wordt voor het aansturen van camera's, hoeft dit niet altijd zo te zijn. Zo is het mogelijk dat een apparaat dat geen video ondersteunt alsnog de overige communicatie via het ONVIF protocol laat plaatsvinden. Alleen Profiel S verplicht de implementatie van video streaming, maar niet elk apparaat implementeert Profiel S.

Elk profiel bestaat uit een set van features die geïmplementeerd moet worden ("MANDATORY"), zoals beveiligingsprotocollen, en optionele features ("OPTIONAL"), zoals PTZ en Relay Outputs. Daarnaast zijn er verschillen tussen wat verplicht is voor een Device en voor een Client. Dit betekent dat elk apparaat of applicatie die ONVIF implementeert een specifieke feature-set implementeert. Dit wordt een specificatie genoemd.

Hierbij moet echter rekening gehouden worden met de evolutie van het protocol. De features worden veranderd en verbeterd, waardoor nieuwe versies uitgebracht worden. ONVIF streeft er tevens naar om

volledig backwards compatible te zijn, wat betekent dat wanneer een Device of Client een feature implementeert, dat het uitbrengen van een nieuwe versie geen effect heeft op het functioneren hiervan. De set van features die een apparaat implementeert wordt de “Specification Set” genoemd. Elke versie van deze lijst beschrijft welke specificatie (versie) van de features geïmplementeerd zijn wanneer een applicatie deze set gebruikt.

Hoe wordt ONVIF toegepast?

De functionaliteiten van elke specificatie staan beschreven in een WSDL (Web Service Description Language) omschrijving. Deze XML beschrijving kan gebruikt worden door applicaties om code te genereren.

ONVIF heeft twee Test Tools gemaakt zodat ontwikkelaars die een Device- of Client-implementatie gemaakt hebben deze kunnen testen. De Device Test Tool kan een ONVIF Device over het netwerk benaderen en alle bestaande functies controleren op het apparaat door deze direct aan te roepen. De Client Test Tool leest netwerkverkeer uit een “network-trace”-bestand en controleert vervolgens of de communicatie tussen de Client en de Devices waarmee gecommuniceerd is correct is verlopen.

Beide Test Tools worden tegenwoordig gebruikt voor het automatisch vaststellen of een nieuw Device of Client “ONVIF Conformant” is. Het “Document of Conformance” dat gegenereerd wordt door de tool moet meegeleverd worden bij het indienen van een aanvraag van “Conformance” aan ONVIF.

5.3. Resultaat kennismaking

Aan het eind van de kennismaking is de eerste versie van het Plan van Aanpak [\[Bijlage D. Plan van Aanpak\]](#) opgeleverd als uitbreiding van het Afstudeerplan [\[Bijlage A. Afstudeerplan\]](#).

5.3.1. Plan van Aanpak

Het Plan van Aanpak beschrijft in grote lijnen hoe de opdracht zal worden aangepakt. In het afstudeerplan is de opdrachtomschrijving vastgesteld, die niet is veranderd sinds het Afstudeerplan. Wel is men verder ingegaan op de randvoorwaarden van de uitvoer door het maken van een risicoanalyse, het vaststellen van de scope, een keuze te maken voor de ontwikkelmethode en een globale planning op te zetten voor de gehele periode.

Risicoanalyse

Voor het analyseren van de risico's van de afstudeeropdracht is een combinatie gebruikt van brainstormen (checklist) en de ‘wat-als’ vraag. Op basis hiervan is een vergelijking gemaakt tussen de kans dat een risico plaatsvindt en de impact die het zou hebben. Indien de combinatie van deze factoren hoog zou zijn, is besloten om hier op voorhand handelingen voor te ondernemen, om de kans te verkleinen, of om de impact te verkleinen.

Naar aanleiding van het ONVIF onderzoek [\[5.2\]](#) is tijdens de risicoanalyse één belangrijke observatie naar voren gekomen. Om een beter beeld te krijgen van het formaat van de opdracht, en dus een betere scope te kunnen bepalen, is het van belang een proof of concept te maken. Deze zal worden opgenomen in de planning.

Scope

Voor het definiëren van de scope is de informatie gebruikt die verzameld is tijdens het ONVIF onderzoek [\[5.2\]](#) in samenspraak met de opdrachtgever en begeleider. Hiermee is de opdracht zodanig afgebakend dat de opdrachtgever tevreden is met het eindproduct en dat de beide partijen de opdracht als haalbaar ervaren.

Naar aanleiding van het ONVIF onderzoek [\[5.2\]](#) is gebleken dat het ONVIF Profiel waar VDG interesse in heeft Profiel S is, namelijk camerabeeld en -geluid ophalen, PTZ, outputs aansturen en events ophalen. Hierdoor vallen Profiel C, G, A en Q buiten de scope.

Er moet nu onderzocht worden in welke onderdelen van Profiel S VDG interesse heeft, naast de features die ONVIF verplicht binnen Profiel S. De volgende prioriteitenlijst is in overleg met de begeleider opgesteld om de prioriteiten van ontwikkeling weer te geven zolang de features nog niet bekend zijn. Deze lijst is gebaseerd op de verplichte en optionele features in ONVIF en zal worden ingevuld naar aanleiding van het vervolgonderzoek.

1. De verplichte functies van verplichte features **en** de optionele functies van verplichte features die VDG specifiek wil gebruiken
2. De verplichte functies van optionele features die VDG specifiek wil gebruiken
3. De optionele functies van optionele features die VDG specifiek wil gebruiken
4. De verplichte functies van optionele features die VDG in de toekomst wil gebruiken
5. De optionele functies van optionele features die VDG in de toekomst wil gebruiken

Ontwikkelmethodiek

Alhoewel VDG met Scrumban werkt, is aangegeven dat de afstudeeropdracht in een zelfgekozen ontwikkelmethode opgepakt mag worden. Daarom zijn er enkele ontwikkelmethodes vergeleken. Hierbij is gekeken naar de voordelen, nadelen, en eerdere ervaring die de afstudeerder al heeft met de methode. Waar veel waarde aan is gehecht zijn de flexibiliteit van planning en een iteratieve methodiek vanwege de relatief korte periode waarin de opdracht zal plaatsvinden. Dit is belangrijk omdat veel eisen nog niet bekend zijn of pas later in het traject duidelijk zullen worden. Dit komt omdat de plug-in architectuur ook nog volop in ontwikkeling is, waardoor veranderingen in het ontwerp geen uitzondering zullen zijn. Wanneer deze eisen veranderen is het noodzakelijk dat hier op snel ingespeeld kan worden tijdens de afstudeerperiode.

Hierdoor zijn twee methodieken overgebleven: Scrum en Scrumban.

Scrum is al meerdere keren gebruikt door de afstudeerder en dus erg bekend. Scrumban is erg vergelijkbaar met Scrum, maar is minder gericht op sprints en meer op een continu proces en workflow van een product.

De doorslaggevende factoren voor de keuze voor Scrumban waren als volgt:

- De werkmethode van VDG [\[3.2\]](#), die bewezen effectief is volgens hun workflow. Hierdoor is ook een hogere kwaliteitscontrole mogelijk vanuit VDG op het product.
- Leren en gebruiken van een (relatief) nieuwe ontwikkelmethode, om ervaring op te doen met nieuwe manieren om een probleem aan te pakken.

Desondanks is besloten om enkele aspecten anders uit te voeren dan VDG's toepassingen van Scrumban, namelijk:

- Alle taken, m.u.v. reviews, worden uitgevoerd door de afstudeerder.
- De Work-In-Progress (WIP) is niet 1; er mag aan meerdere taken tegelijkertijd gewerkt worden. Dit biedt meer flexibiliteit in het relatief korte ontwikkeltraject.
- Er zullen sprints worden ingedeeld. Dit biedt meer duidelijkheid en planningsmogelijkheden voor het relatief korte ontwikkeltraject.

Planning

De periode van 19 weken is opgedeeld in sprints van 2 weken en één sprint van 3 weken. Er is gekozen voor korte sprints om sneller te kunnen reageren op veranderende eisen [\[5.3.1 Ontwikkelmethodiek\]](#). Sprints van 1 week zou te kort zijn en te veel planning overhead geven. Sprints van 3 weken zouden te lang zijn en niet de flexibiliteit bieden die gewenst is, aangezien aan het begin van elke sprint de resultaten van de vorige sprint worden gebruikt om de volgende sprint in te delen. Wanneer de sprints langer worden zal er minder vaak naar de feedback kunnen worden gekeken.

Er is een inschatting gemaakt dat het onderzoek zal kunnen worden afgerond in 1 sprint, in combinatie met het proof of concept. Vervolgens zijn er 2 sprints gereserveerd voor het ontwerp, respectievelijk het functioneel ontwerp en technisch ontwerp per sprint. Aansluitend zou men starten met de ontwikkeling van de plug-in gedurende 3 sprints. Wat er ontwikkeld zal worden tijdens deze sprints is nog onbekend en zal tijdens het onderzoek in de volgende sprint vastgesteld worden. Tevens zal in de eerste ontwikkelsprint het testplan worden opgeleverd. Dit gebeurt hier omdat op deze manier de informatie van het ontwerp en van de ontwikkelinfrastructuur gebruikt kan worden om te bepalen hoe getest zal worden. Hoewel tijdens deze ontwikkelsprints ook al getest wordt, is ervoor gekozen om na de ontwikkelfase een sprint te wijden aan het testen van alles dat geproduceerd is. Men verwacht dat deze meer tijd nodig zal hebben, dus zal hier de 3 weken sprint in worden opgenomen. Hier kunnen meer geavanceerde tests plaatsvinden zoals regressie-, load- en stresstests. Tot slot zal een sprint worden gewijd aan de overdracht van alle documentatie en code (zie [tabel 1](#) of [\[Bijlage D. Plan van Aanpak\]](#) voor de inhoudelijke planning).

Sprint	Titel	Taken
0	Kennismaking	VDG Sense en ONVIF kennismaking, oplevering Plan van Aanpak
1	Onderzoek	ONVIF onderzoek, requirements opstellen, proof of concept, oplevering onderzoeksrapport
2-3	Ontwerp	Oplevering Functioneel- en technisch ontwerp
4-6	Ontwikkeling	Ontwikkelen ONVIF plug-in (& eerste ontwikkelsprint oplevering testplan)
7	Testen (3w)	Testen plug-in, oplevering testrapport
8	Overdracht	Overdracht documentatie & code

Tabel 1: Globale planning Afstudeerperiode in sprints v1

6. Sprint 1: Onderzoek

In deze fase zijn twee belangrijke taken uitgevoerd:

- Naar aanleiding van het onderzoek dat heeft plaatsgevonden is gebleken dat vervolgonderzoek moet plaatsvinden. [\[5.2.1\]](#) De volgende vragen wil men beantwoorden:
 - Welke requirements stelt VDG aan de ONVIF plug-in?
 - Welke ONVIF features wil VDG ondersteunen?
 - Welke ONVIF versie (specification set) zal ontwikkeld worden?
- Er wordt een proof of concept gemaakt. [\[5.3.1\]](#) Deze moet beter inzicht geven in de complexiteit van de taken die uitgevoerd zullen gaan worden in de toekomst.

6.1. Vervolg ONVIF onderzoek

Tijdens de kennismaking is vooral gefocust op het verkrijgen van een globaal beeld van ONVIF. Als vervolg hierop, is de focus verplaatst naar het vaststellen van de eisen van VDG en deze te koppelen aan de mogelijkheden in ONVIF die binnen de scope zijn gedefinieerd.

Met de voorkennis van het eerder uitgevoerde onderzoek zijn deze aspecten onderzocht door in gesprek te gaan met de verantwoordelijken.

6.1.1. Requirements VDG

Tijdens een bespreking met de opdrachtgever, begeleider en de ONVIF expert zijn de eisen aan de ONVIF plug-in opgesteld. Met name is hierbij gekeken naar de ONVIF features die men wil ondersteunen. [Figuur 2](#) is het resultaat hiervan, dat weergeeft welke ONVIF Profile S features men nu al ondersteunt en welke men graag wil kunnen ondersteunen in de toekomst.

Hieruit volgt de volgende lijst van features die VDG graag in de plug-in functioneel ziet die binnen de scope van de afstudeeropdracht passen:

- Streaming: MJPEG, MPEG4, H264, Audio, Multicast
- PTZ: ContinuousMove, Presets, Home, Auxiliary Commands
- Relay Outputs
- Events
- Discovery

De functionele en technische eisen zullen in het ontwerpdocument worden opgenomen wat zal gebeuren in de ontwerpsprint [\[7.1\]](#): Ontwerp.

6.1.2. Specification set

ONVIF maakt gebruik van specificatie-sets. Een specificatie-set is een combinatie van meerdere specificaties van een service in een bepaalde versie. Dit biedt de mogelijkheid om één specificatie een nieuwe versie te geven, zonder de andere producten te bewerken, zodat er een nieuwe versie uitgebracht kan worden.

Aangezien ONVIF is opgezet zodat zij backwards compatible is, loont het om de meest recent uitgebrachte specificatie-set te implementeren. Dit is specificatie-set versie 2.6.1 [\[3\]](#).

De services die binnen specificatie-set 2.6.1 vallen zijn:

- | | |
|---------------------------------------|---|
| - Media Version 2.6.1 / December 2015 | - PTZ Version 2.6.1 / December 2015 |
| - Media2 Version 1.0 / December 2015 | - Streaming Version 2.6 / June 2015 |
| - Core Version 2.6.1 / December 2015 | - Imaging Version 2.5 / December 2014 |
| - Event Version 2.6 / June 2015 | - Device IO Version 2.6 / December 2015 |

Feature	Currently Supported	Desired Implementation	Explanation (complete explanation can be found in the Drive Research document or ONVIF Profile S Specification (links below))
Profile Mandatory Features			
Video Streaming	✓	✓	Listing of media profiles & streaming of video using RTSP
Video Encoder Configuration	✗	✓	Listing and modification of video encoder configurations on the device
User Authentication	✓	✓	WS-UsernameToken using timestamps and nonce & Http Digest
Capabilities	?	✓	Querying device for capabilities
Profile Conditional Features			Both conditional to device and client (us). GetCapabilities() will return a value to see if it is enabled on the device.
Video Streaming - MPEG4 (sub Video Streaming)	✓	✓	Streaming of MPEG4 video using RTSP
Video Streaming - H264 (sub Video Streaming)	✓	✓	Streaming of H264 video using RTSP
PTZ	✓	✓ Continuous move is currently preferred	Using and moving a PTZ device using continuous move
PTZ - Absolute Positioning (sub PTZ)	✗	P2 ✓	Moving a PTZ device to an absolute position
PTZ - Relative Positioning (sub PTZ)	✓	✓	Moving a PTZ device to a relative position
PTZ - Presets (sub PTZ)	✓	✓	Listing of presets & Moving a PTZ device to a preset
PTZ - Home Position (sub PTZ)	✓	✓	Moving a PTZ device to its home position
PTZ - Auxiliary Command (sub PTZ)	✗	✓	Support for PTZ specific auxiliary commands
Audio Streaming	✓	✓	Streaming of Audio
Audio Streaming - G726 (sub Audio Streaming)	✓	✓	Streaming of G726 audio using RTSP
Audio Streaming - AAC (sub Audio Streaming)	✓	✓	Streaming of AAC audio using RTSP
Multicast Streaming (sub Video Streaming)	✓	✓	Streaming video over multicast
Relay Outputs	✓	✓	Listing, configuration and triggering of relay outputs
NTP	✗	✓	Synchronization of time using NTP servers
Dynamic DNS	✗	✗	Configuration of Dynamic DNS
Zero Configuration	✗	✗	Configuration of Zero Configuration
IP Address Filtering	✗	P2 ✓	Configuration of IP Address Filters
Device Mandatory Features			Mandatory for devices, conditional for Client (us)
Discovery	✓	✓	Discovery of a device on the network & Setting of discovery mode & listing, adding modifying and removing of discovery scopes
Network Configuration	✗	✗	Configuration of network settings on the device
System	(/ partially)	✓ (/partially)	Configuration of system settings & Device Information
User handling	✗	✗	Manage users on the device
Event handling	✓ partially (Push)	✓ Push	Retrieving and filtering of events from a device (either with "pull" functions, "push" functions or both)
Media profile configuration	✗	P2 ✓ (GUI redesign)	Creation, retrieval and deletion of media profiles
Video source configuration	✗	✓	Listing and modification of video source configurations on the device
Metadata configuration	✗	P2 ✓	Listing and modification of metadata configuration on the device

Figuur 2: Lijst van gewenste ONVIF Profile S features VDG Plug-in tegenover de huidige implementatie van ONVIF in Sense (uitleg [\[6.1.1\]](#))

6.2. Proof of Concept

Het doel van het proof of concept is geweest om in een vroege fase van het onderzoek beeld te krijgen van de technieken die zullen worden gebruikt. Aan de hand van deze kennis is het mogelijk om een technisch onderbouwde planning op te kunnen zetten, aangezien er meer bekend is over problemen die zich kunnen voordoen en hoeveel tijd een taak in kan nemen.

Het proof of concept zal worden gemaakt in C++(11) Qt, wat dezelfde omgeving is als wat VDG gebruikt voor het ontwikkelen van Sense. Later zal uitgezocht worden of de plug-in hier ook in zal worden ontwikkeld, maar voor het proof of concept levert deze keuze het snelste resultaat op.

De doelen voor het proof of concept zijn:

- Het kennismaken met de nieuwe plug-in architectuur
- Web service functionaliteit
 - Berichten versturen via een SOAP service
 - WSDL conversie naar code
 - Aanroepen gegenereerde WSDL code
- Het testen van camera functionaliteit via ONVIF

Deze aspecten zullen in dit hoofdstuk apart beantwoord worden.

6.2.1. Plug-in architectuur

De nieuwe architectuur, ontwikkeld door Wilco Gorthuis, is uitgebreid uitgelegd. Hierbij zijn drie aspecten van belang geweest:

- Het probleem van de huidige architectuur
- De visie van de functionaliteit van de nieuwe architectuur
- De huidige implementatie van de nieuwe architectuur

Huidige architectuur

Er is op dit moment geen architectuur voor het toevoegen van plug-ins. Momenteel worden deze drivers in de source code opgenomen. Dit betekent dat alle drivers te allen tijde aanwezig zijn in de applicatie. Dit komt de schaalbaarheid en leesbaarheid van de code niet ten goede, dus wil VDG hier een nieuwe architectuur voor opzetten.

Visie nieuwe architectuur

Men wil dat de cameradrivers aparte plug-ins worden die in Sense kunnen worden ingeladen. Zo zou het mogelijk zijn dat klanten alleen toegang hebben tot drivers die zij daadwerkelijk willen gebruiken. Dit zou het makkelijker maken om nieuwe cameradrivers toe te voegen aan Sense of bestaande drivers te updaten.

Die nieuwe architectuur definieert aan de serverkant de verwachte functionaliteiten die de plug-in implementeert door middel van interface-klassen. In de plug-in wordt van deze interfaces overgeërfd en de implementatie voor de functies uitgewerkt. Op deze manier hoeft de server maar één functie aan te roepen, bijvoorbeeld ZetAudioOpnameAan. Als een plug-in dit ondersteunt (dit is afhankelijk van de capaciteiten van het apparaat), dan beschrijft de plug-in hoe dit naar het apparaat gecommuniceerd moet worden.

Ontwerp nieuwe architectuur

[\[Bijlage E. plug-in architectuur\]](#) laat zien hoe het eerste ontwerp van de nieuwe architectuur eruit ziet. Een belangrijk onderdeel hiervan is de “Device Driver” klasse. Deze is verantwoordelijk voor het inladen van de plug-in en de server te informeren over zijn features.

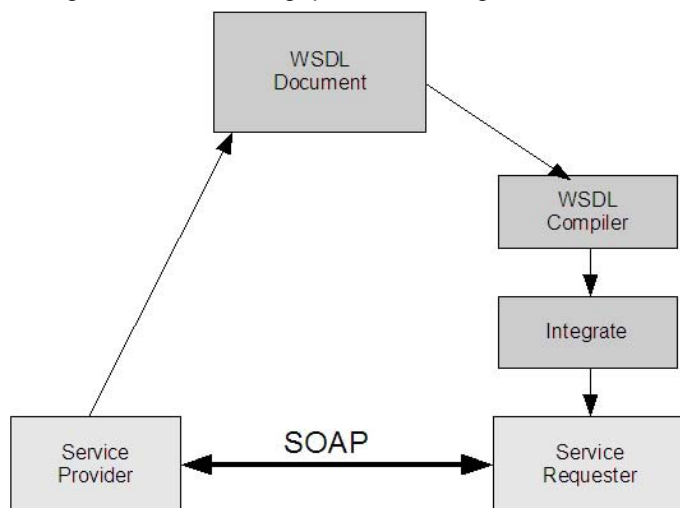
Dit gebeurt door één of meerdere “Device Features” te retourneren. Deze zijn gespecificeerd in de “Dome”, “Media”, “GPIO” en “Discovery” klassen. Elk van deze features beschrijft een set functionaliteiten in de vorm van functies.

Het idee achter de architectuur is dat de plug-in één of meer “Device Features” kan implementeren. De server wordt geïnformeerd over de features die de plug-in ondersteunt door deze als een lijst terug te geven. De server leest dit uit en kan vervolgens direct de implementatie aanroepen die in de plug-in aanwezig is voor de verwachte functies.

6.2.2. Web Service

ONVIF maakt gebruik van Web Services; een gestandaardiseerde manier om applicaties te integreren met behulp van open source, platform onafhankelijke standaarden zoals XML, SOAP en WSDL over een IP-netwerk. Hierbij is XML verantwoordelijk voor het beschrijven van de syntax, SOAP voor het versturen van berichten, en WSDL voor het beschrijven van de functionaliteiten van de services.

Alle ONVIF services zijn beschreven in WSDL met HTTP als het onderliggende transport mechanisme. Alhoewel andere transport protocollen ook mogelijk zijn bij SOAP communicatie, zoals SMTP, wordt over het algemeen HTTP toegepast door het gemak die het biedt in de huidige internetinfrastructuren. [\[4\]](#)



Figuur 3: Web Service based development principles (source:ONVIF Core Specification [\[2\]](#))

[Figuur 3](#) geeft het standaard principe van softwareontwikkeling op basis van Web Services weer. In dit geval is de Service Provider een ONVIF camera en de Service Requester de plug-in die tijdens de afstudeerperiode geïntegreerd zal worden. De services die een camera ondersteunt, staan beschreven in de ONVIF specificaties die geschreven zijn in WSDL. Deze services kunnen vervolgens via SOAP aangeroepen worden. [\[2\] 4.1 Webservices](#)

kdSOAP

Er is gekozen om tijdens het proof of concept de SOAP service kdSOAP van KDAB te gebruiken. Hier is om meerdere redenen voor gekozen.

Ten eerste is VDG ontevreden over de SOAP service die momenteel gebruikt wordt, gSOAP. Zij wordt ervaren als onleesbaar waardoor zij lastig te debuggen is, wat veel tijd kost. Daarnaast heeft VDG alleen toegang tot de “public license” van gSOAP en wil niet dat haar code publiek wordt gemaakt. Het kopen van een licentie is momenteel geen optie.

Daarom zijn enkele alternatieven gezocht ter vervanging van gSOAP. Samen met de begeleider zijn eisen gesteld aan de service. De eisen die hieraan gesteld worden zijn:

- De service is goedkoop of gratis en mag worden gebruikt zonder dat eigen code publiek moet worden gemaakt.
- De service kan snel in gebruik worden genomen zodat dit geen negatieve invloed heeft op de afstudeeropdracht.
- De service ondersteunt minimaal alle protocollen die ONVIF hanteert (bijv. WS).

Alternatieven zoals de Windows SOAP API, SvcUtil & WsUtil, en Windows Web Service vallen af aangezien zij niet gemakkelijk in gebruik zijn en dus de opdracht in gevaar brengen.

Er is gekozen voor kdSOAP, ontwikkeld door KDAB, die veel software maakt voor C++ Qt.

Hiervoor is gekozen omdat men verwacht dat, omdat zij optimaal gebruik maakt van de Qt Core, de service erg efficiënt is, makkelijker leesbaar en zij het snelst resultaat oplevert met de minste werklust.

WSDL conversie naar C++

kdSOAP levert bij hun SOAP service ook een programma dat van een WSDL beschrijving C++ code kan maken; kdwsdl2cpp (lees als KD, WSDL, to, C++). Deze code kan gebruikt worden om vanuit een programma direct de gedefinieerde functies aan te roepen met de SOAP client. Een voorbeeld hiervan is zichtbaar in [\[Bijlage F. kdwsdl2cpp voorbeeld\]](#).

6.3. Sprint resultaat

Tijdens deze sprint is het ONVIF Onderzoek afgerond [\[Bijlage C. ONVIF Research Document\]](#) en is het proof of concept uitgevoerd. Daarnaast is de planning aangepast naar aanleiding van de bevindingen van het onderzoek.

6.3.1. ONVIF Onderzoek

Tijdens het onderzoek is duidelijk geworden welke features VDG graag geïmplementeerd wil zien in de ONVIF plug-in. Naar aanleiding hiervan is de in het Plan van Aanpak (Scope) [\[5.3.1\]](#) gedefinieerde prioriteitenlijst ingevuld:

1. *Video Streaming, Video Encoder Configuration, User Authentication, Capabilities, System-GetSystemDateAndTime (belangrijk, want wordt gebruikt als "Hallo")*
2. *PTZ (excluding Auxiliary & AbsoluteMove), Audio Streaming, Multicast, Relay Outputs, Discovery*
3. *Video Streaming (MPEG & H264), PTZ-Auxiliary*
4. *PTZ-AbsoluteMove, PTZ-RemovePresets, NTP, IP address filtering, System, Event Handling, Media Profile Configuration, Media Source Configuration, Metadata Configuration*

In de volgende sprint zullen de requirements voor deze features worden uitgewerkt.

Daarnaast is de ONVIF Client Test Tool getest door ONVIF berichten van het proof-of-concept te traceren met Wireshark en deze aan de applicatie te geven.

Hierdoor is duidelijk geworden dat de Test Tool kan controleren of de SOAP berichten (syntactisch) valide zijn, maar niet of de berichten functioneel zijn. Daarnaast is de gebruiker ervoor verantwoordelijk dat tijdens het testen alle functionaliteiten die men wil testen worden aangeroepen, want alleen dan kan de Test Tool hierover oordelen.

Hieruit is geconcludeerd dat er een testomgeving moet komen die alle functionaliteiten kan aanroepen die men wil testen. De ONVIF Client Test Tool bepaalt of de berichten valide zijn, terwijl een persoon beoordeelt of de acties de gewenste resultaten oplevert.

6.3.2. SOAP Review

[\[Bijlage G. kdSOAP Review\]](#)

Als alternatief voor de gebruikte SOAP service is tijdens het proof of concept een vervanger getest: kdSOAP. Deze SOAP service blijkt echter op bepaalde aspecten onderontwikkeld te zijn. Een belangrijk onderdeel hiervan is de authenticatie van de gebruiker aan het apparaat. ONVIF verplicht hier WS-Security voor te gebruiken, maar zij wordt niet ondersteund door kdSOAP.

Tijdens de sprint is getracht om in de huidige situatie van kdSOAP een oplossing hiervoor te maken. Hierin zijn 3 stappen doorgenomen:

- Is het mogelijk om zelf de WS-Security headers te bouwen met de bestaande code?
- Kan KDAB helpen met het ontwikkelen van de WS-Security headers?
- Kan een andere bibliotheek gebruikt worden om alleen de authenticatie te verzorgen?

Elk van deze mogelijkheden is negatief uitgevallen. Het zelf maken van headers is niet flexibel genoeg om deze volgens de WS-Security standaard te kunnen opzetten. KDAB heeft aangegeven dit te kunnen ontwikkelen, maar vraagt hiervoor om financiering. Tot slot is het erg ongewenst om verschillende SOAP services te gebruiken voor verschillende doeleinden en is besloten dat dit geen goede oplossing voor het probleem is.

Vervolgens is een beslissing gemaakt uit de volgende opties:

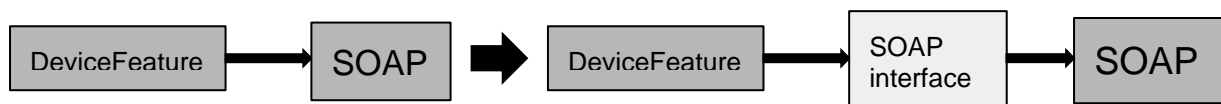
1. Teruggaan naar de huidige SOAP service die gebruikt wordt door VDG.
2. Zelf een oplossing ontwikkelen binnen kdSOAP en deze committen (pull-request) naar de Github van KDAB.
3. Een andere SOAP service onderzoeken.

Om sneller verder te kunnen gaan met het ontwikkelen van de plug-in is in overleg met de begeleider besloten om de huidige SOAP service van VDG te gebruiken; gSOAP.

Hierbij zijn de volgende punten doorslaggevend geweest:

- kdSOAP lijkt dusdanig ver achter te lopen, dat VDG niet geïnteresseerd is om hier nu functionaliteiten aan toe te voegen.
- gSOAP is al bekend bij VDG. Dit betekent dat ontwikkeling sneller kan plaatsvinden.
- Alternatieven bieden nog minder functionaliteiten dan gSOAP of kdSOAP.

De architectuur zal worden voorbereid op het wisselen van het communicatieprotocol (andere SOAP service, ofwel een verandering van SOAP naar een ander protocol). Dit gebeurt door het construeren van een extra laag tussen de code die een bericht wil versturen en de daadwerkelijke implementatie van de communicatie (zie [figuur 4](#)).



Figuur 4. Schets SOAP afhankelijkheid

6.3.3. Planning

Naar aanleiding van het vervolgonderzoek is de planning aangepast. Men verwacht dat het ontwerp veel verandert tijdens de ontwikkeling, waardoor het niet nodig is om 2 sprints te gebruiken voor de ontwerpfase, maar 1. De overgebleven sprint zal worden gebruikt als ontwikkelsprint, zodat er een sprint is voor elke ONVIF feature die ontwikkeld zal worden.

In combinatie met de opgestelde prioriteitenlijst [\[5.3.1 Scope\]](#) is een planning gemaakt voor de komende vier ontwikkelsprints m.b.t. het implementeren van de ONVIF functionaliteiten in de plug-in. De eerste sprint dient als opzet van de architectuur en de DeviceService, aangezien deze aan de basis staat van alle communicatie. De tweede ontwikkelsprint focust op MediaService, die verantwoordelijk is voor audio en video. Vervolgens kan in sprint 3 de PTZService gestart worden, die afhankelijk is van de MediaService. Tot slot worden de laagste prioriteiten aangepakt zoals Events, NTP en specifieke configuratie (zie [tabel 2](#)).

ONVIF Features implemented by Sprint			
Sprint 3 - DeviceService and start MediaService	Sprint 4 - MediaService	Sprint 5 - PTZ	Sprint 6 - Events & extra
Capabilities: - GetCapabilities System: - GetDeviceInformation - GetSystemDateAndTime Relay Outputs: - GetRelayOutputs - SetRelayOutputSettings - SetRelayOutputState Video Streaming: - GetProfiles - GetStreamUri - GetVideoEncoder Configuration - SetVideoEncoder Configuration - GetVideoEncoder ConfigurationOptions User Authentication: - WS-UsernameToken Authentication - HTTP Digest	Video Streaming: - Media Streaming using RTSP - Media Streaming using RTSP - JPEG RTP header extension - Video Streaming (MPEG4 & H264) - SetSynchronization Point Multicast: - StartMulticast Streaming - StopMulticast Streaming Audio Streaming: - AddAudioSource Configuration - GetCompatibleAudio SourceConfigurations - AddAudioEncoder Configuration - GetCompatibleAudio EncoderConfigurations Discovery: - WS-Discovery	PTZ: - GetNodes - GetNode - Get Configurations - AddPTZ Configuration - ContinuousMove - Stop Presets: - GetPresets - GotoPresets - RemovePreset Home: - GotoHome Position - SetHomePosition Auxiliary: - SendAuxiliary Command	Event Handling: - Notify - Subscribe - Renew Media Profile Configuration: - GetProfiles - CreateProfile Video Source Configuration: - GetCompatible VideoSource Configurations - GetVideoSource Configuration Options - SetVideoSource Configuration - AddVideoSourceConfiguration NTP: - GetNTP - SetNTP

Tabel 2: ONVIF planning per ontwikkelsprint

7. Sprint 2: Design

Tijdens deze sprint is het eerste ontwerp van de plug-in tot stand gekomen. Hierbij zijn de functionele en niet-functionele eisen van het probleemdomein tevens verder uitgewerkt.

7.1. Ontwerp

Tijdens deze sprint zijn de eerste ontwerpen voor de plug-in gemaakt.

Vóór het uitbrengen van de eerste versie van het ontwerp is er meerdere malen overlegd over o.a. gekozen implementaties en gebruikte patronen. Dit heeft geresulteerd in [\[Bijlage H. Design Document\]](#), die de functionele en niet-functionele eisen beschrijft van de plug-in en UML diagrammen die deze eisen vertalen naar de beoogde implementatie.

7.1.1. Requirements

In de ontwerpfase zijn de eisen aan het product vanuit twee perspectieven benaderd, namelijk vanuit VDG en vanuit ONVIF. Beiden hebben eisen gesteld aan het product op functioneel en technisch gebied. De technische eisen van ONVIF liggen vast in de specifications en de WSDL beschrijvingen van de Services. De eisen gesteld door VDG zijn verzameld door het interviewen van de opdrachtgever en de ONVIF expert. De volledige lijst van eisen is beschikbaar in [\[Bijlage H. Design Document\] Ch3. Requirements](#), waarvan hieronder enkele interessante zijn uitgelicht.

Functionele eisen:

1. De plug-in moet kunnen detecteren of een camera op ONVIF draait
2. De plug-in moet SOAP (ONVIF) berichten geautoriseerd kunnen versturen
3. De plug-in moet kunnen detecteren welke features een ONVIF device bevat
4. Als een camera Media ondersteunt, moet Sense beeld en audio kunnen streamen
 - a. Als een camera de protocollen H264 en/of MPEG4 ondersteunt, kan Sense dit via de plug-in configureren en streamen
 - b. Als een camera multicast streamen ondersteunt, kan Sense via de plug-in een multicast stream ophalen

Technische (niet-functionele) eisen:

1. De plug-in zal worden ontwikkeld in C++(11) in QtCreator.
2. Voor de SOAP communicatie en vertaling van de WSDL services zal gSOAP worden gebruikt (n.a.v. [\[6.3.2\]](#))

7.1.2. Ontwerpkeuzes

In het eerste ontwerp zijn bepaalde keuzes gemaakt die invloed hebben op de implementatie van de plug-in. De belangrijkste hiervan zijn als volgt:

- Er is gekozen om de verantwoordelijkheid van features aanmaken te verschuiven naar een aparte klasse: `OnvifFeatureFactory`. Hiervoor is gekozen om een betere scheiding te maken tussen wat de `DeviceDriver` klasse voor verantwoordelijk heeft en waar wordt gedefinieerd welke features een camera heeft. Dit is gedaan vanuit de gedachte van het Single Responsibility Principle (SRP). Deze volgt voornamelijk geen Factory-pattern, omdat nog niet duidelijk is waar deze effectief zou kunnen worden toegepast. Er zullen binnen de plug-in niet meerdere variaties of instanties zijn van een specifieke feature. Daarnaast heeft zij ook veel kennis over hoe de features opgebouwd moet worden, zoals de *Capabilities* van een bepaalde feature. Het Factory-pattern biedt veel voordelen wanneer de gebruikende klasse geen kennis hoeft te hebben over hoe het object gemaakt wordt, maar `OnvifFeatureFactory` heeft dit wel, waardoor dit nu niet interessant lijkt.
- Naar aanleiding van de SOAP Review [\[6.3.2\]](#) is besloten om een scheiding te maken tussen de SOAP Service en de features die deze services gebruiken. Deze `FunctionMap` beschrijft zelfgedefinieerde objecten en functies, wat het mogelijk maakt om de features intact te houden bij een wisseling van SOAP Service.
- Er is besloten om geen Decorator Pattern te gebruiken voor de SOAP implementatie. Dit leek interessant omdat de SOAP communicatie veel gemeen heeft, dus dat dit opgebouwd zou kunnen worden uit één Concrete SOAP besturing, en elke unieke service hier gebruik van maakt. Echter wordt dit al opgelost door de `wsdl2cpp` conversie en wil men niet de mogelijkheid bieden om één object te maken die een combinatie van meerdere services gebruikt.

7.1.3. Klassediagram

Voor het ontwerp van de ONVIF plug-in is een klassediagram opgezet. Deze is voornamelijk gebruikt voor twee doeleinden: Ten eerste als visualisatie van de plug-in architectuur en ten tweede als 'praatplaatje' voor het ontwerp van de ONVIF plug-in. Dit is van belang geweest bij het bespreken van de architectuur met de begeleider.

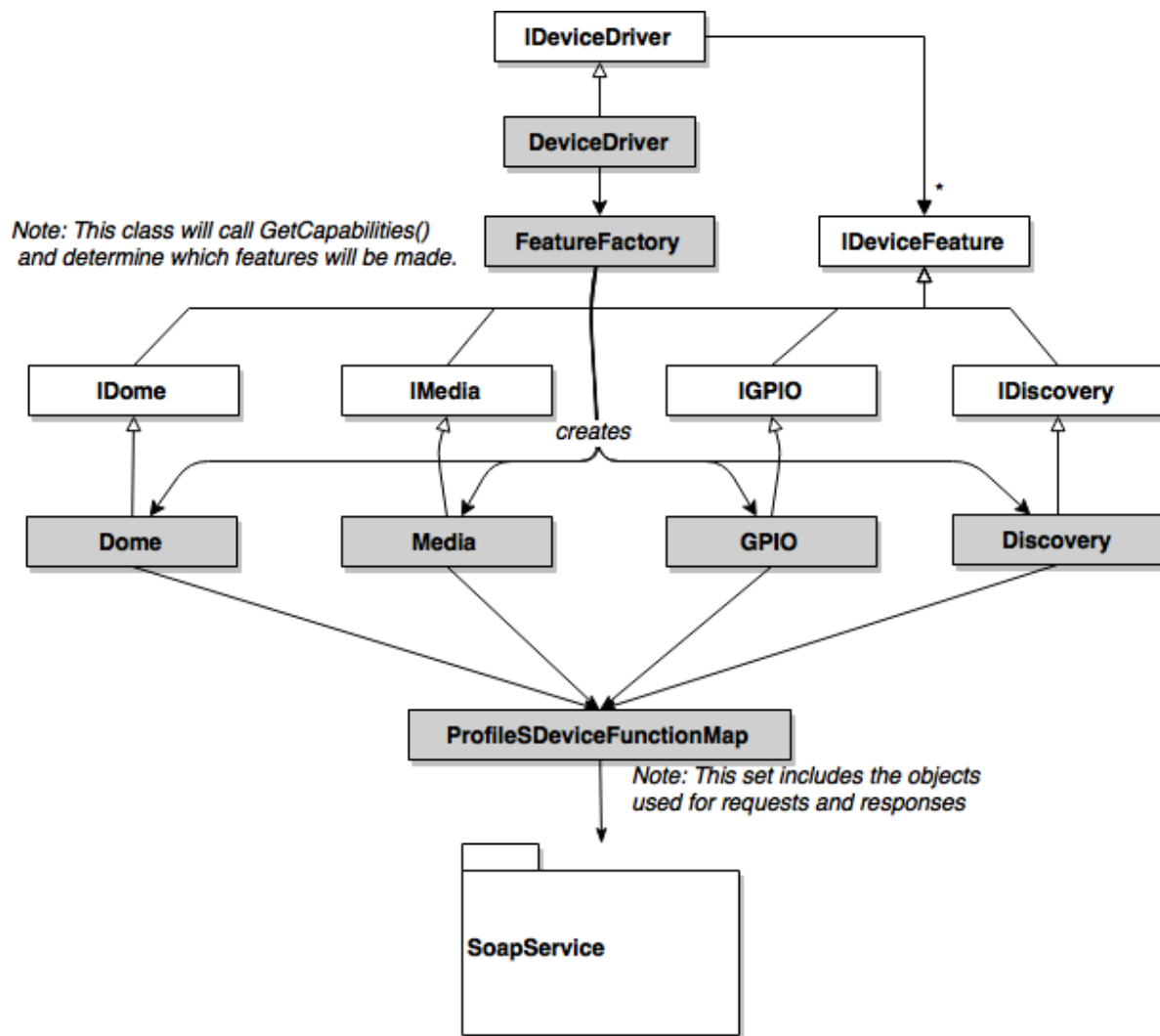
[Figuur 5](#) beschrijft de opzet van de plug-in architectuur en de ONVIF implementatie hiervan.

De witte klassen `IDeviceDriver`, `IDeviceFeature`, `IDome`, `IMedia`, `IGPIO` en `IDiscovery` zijn *pure virtual interfaces*. Deze kunnen worden gezien als omschrijving die de plug-in zal moeten ondersteunen. Sense gebruikt deze interfaces in de Sense Core en bestuurt zo de camera's.

De grijze klassen zijn de implementaties van deze architectuur in de ONVIF plug-in. Daarnaast zijn enkele componenten toegevoegd. De eerste hiervan is de `FeatureFactory`.

Vanuit het Single Responsibility Principle (SRP) is besloten om de `DeviceDriver` niet de verantwoordelijkheid te geven over het bepalen welke features een camera ondersteunt en deze aan te maken. De `FeatureFactory` roept de ONVIF functie `GetCapabilities` aan, waarna een apparaat zal aangeven welke componenten deze heeft. Op basis van deze informatie kan de `FeatureFactory` bepalen welke Features moeten worden aangemaakt.

Ten tweede is de `ProfileSDeviceFunctionMap` toegevoegd, als resultaat van de SOAP Review [\[6.3.2\]](#). Aangezien VDG niet tevreden is over het gebruik van gSOAP, is besloten het ontwerp flexibel te maken m.b.t het wisselen van de SOAP Service. De `ProfileSDeviceFunctionMap` beschrijft de functies die de features kunnen aanroepen, die vervolgens vertaald worden naar de specifieke SOAP implementatie.



Figuur 5: Klassediagram ONVIF Plug-in (simpel) (Uitgebreid Plug-in Architectuur in [Bijlage E](#))

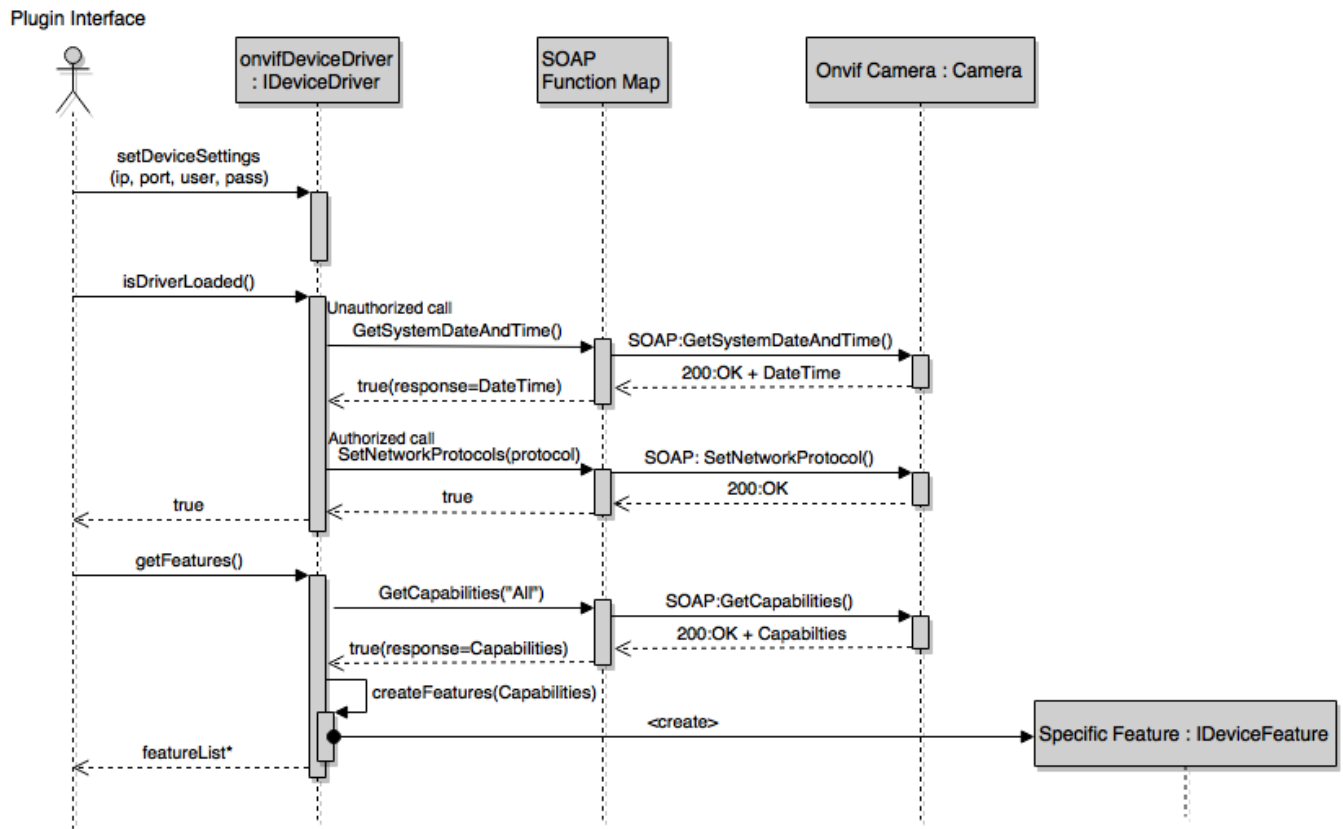
7.1.4. Sequencediagram

Om een beter beeld te krijgen van de communicatie tussen Sense en de plug-in zijn sequencediagrammen gemaakt. Een belangrijke interactie is Sense voorzien van de features via de plug-in. In ONVIF is dit extra interessant omdat dit dynamisch wordt opgebouwd op basis van antwoorden van de camera, namelijk welke features hij ondersteunt.

[Figuur 6](#) beschrijft Sense als de gebruiker van de plug-in via de Plug-in Interface. Zij voorziet de plug-in van de camera gegevens en wacht vervolgens tot de plug-in verbinding heeft gemaakt. Een belangrijke functie binnen ONVIF is GetSystemDateAndTime, aangezien deze altijd zonder autorisatie uitgevoerd kan worden. Vervolgens test de plug-in ook of een geautoriseerd bericht slaagt.

Vervolgens zal de plug-in de featurelijst opvragen, die bestaat uit instanties van IDeviceFeatures zoals een Media- of Dome-implementatie. Om dit te bepalen zal de ONVIF plug-in de Capabilities van een camera opvragen. Deze respons kan worden gebruikt voor het initialiseren van de features zodat deze aan de featurelijst kunnen worden toegevoegd.

Tot slot wordt de lijst teruggegeven aan de plug-in zodat deze direct van de individuele componenten gebruik kan maken.



Figuur 6: Sequencediagram initialisatie plug-in

7.2. Veranderingen ONVIF Profile S Specification

Gedurende deze sprint is een nieuwe versie van de Profile S Specification uitgebracht door ONVIF. Deze is van v1.0 naar v1.1 gegaan en zou alleen visuele veranderingen omvatten, maar bij nadere inspectie is gebleken dat bepaalde functies veranderd zijn van 'mandatory' naar 'optional'. Er is in overleg besloten om deze nieuwe set aan te houden voor compliance, wat als gevolg heeft dat er enkele functies niet meer moeten worden geïmplementeerd.

De functielijst is bijgewerkt in [\[Bijlage I. Compliance & Functions\]](#) die is gebruikt in het [\[Bijlage C. ONVIF Research Document\]](#).

7.3. Sprint resultaat

Tijdens deze sprint is [\[Bijlage H. Design Document\]](#) opgeleverd. In het ontwerp is rekening gehouden met de opgestelde requirements, resultaten uit het ONVIF onderzoek, en eerder genomen ontwerpkeuzes. Dit document zal worden gereviewd. Naar aanleiding van de review en ontwerpkeuzes die verder in het traject zullen worden genomen zal dit document verder evolueren.

8. Sprint 3: Ontwikkeling DeviceService

In sprint 3 is begonnen met de ontwikkeling van de ONVIF plug-in driver.

De volgende taken zullen uitgevoerd worden in deze sprint:

- Eerste communicatie tussen de server en plug-in realiseren in de ontworpen architectuur
- Implementeren van de volgende Device Service aspecten:
 - Capabilities
 - System
 - Relay Outputs
 - Authenticatie (WS-Security & HTTP Digest)
- Implementeren van de volgende Media Service aspecten:
 - Mediaprofielen ophalen
 - StreamUri opvragen van bestaand Mediaprofiel
- Testplan opzetten

8.1. Aanpassingen ontwerp

[\[Bijlage H: Design Document\]](#)

Tijdens deze sprint zijn bepaalde aspecten van het ontwerp toegevoegd of veranderd. De belangrijkste veranderingen zijn als volgt:

- Omdat er veel gelijkenissen zijn tussen de ONVIF Services is ervoor gekozen om één klasse te ontwikkelen (BaseService) die een standaard set taken kan uitvoeren die voor elke service gelijk is, zoals het toevoegen van autorisatie-headers. Dit centraliseert gelijksoortige taken volgens het DRY principe.
- Autorisatie is pas mogelijk wanneer het verschil in tijd tussen Client en Device niet te groot is. De fabrikant bepaalt zelf hoe groot het tijdverschil mag zijn. Om toch met de camera te kunnen communiceren, vindt in de BaseService een truc plaats waardoor communicatie altijd mogelijk is. Er is een extra klasse ontworpen voor dit authenticatieproces. Deze *pure virtual interface* definieert hoe het tijdverschil berekend wordt; ISystemTimeService. Dit is logisch volgens SRP.

8.2. Voorstel verandering architectuur

Daarnaast heeft er overleg plaatsgevonden met betrekking tot de plug-in architectuur. Na onderzoek [\[Bijlage J. Dome Research\]](#) is gebleken dat de huidige verwachtingen van bepaalde klassen in het ontwerp mogelijk onrealistisch zijn. Zo is bijvoorbeeld de IDome klasse verantwoordelijk voor zaken die niet specifiek aan Dome of PTZ besturing gerelateerd zijn.

Er is voorgesteld om de architectuur nogmaals te bekijken en waar mogelijk de verantwoordelijkheden te verspreiden naar nieuwe klassen, specifiek gekeken naar de IDome klasse. In overleg is besloten om dit op te pakken tijdens de PTZService sprint.

8.3. Testplan

In de planning is besloten om in de eerste ontwikkelsprint, [\[Bijlage K. Testplan\]](#) op te zetten. Hiervoor is gekozen aangezien tijdens het ontwikkeltraject de geproduceerde code al getest zal worden. Dit document beschrijft op welke manieren getest zal worden, wat getest moet worden en in welke sprint dit plaatsvindt.

8.3.1. Methode

Het testplan beschrijft het testtraject vanuit twee testmethoden:

- De ONVIF Client Test Tool voor het automatisch testen van de Profile S Compliance. Deze tool wordt door ONVIF verplicht voor het vaststellen van Compliance. Zij genereert een document wat aangeleverd moet worden aan ONVIF wanneer de applicatie klaar is. De tests die worden uitgevoerd staan beschreven in de Profile S Client Test Specification [\[6\]](#), waarbij alleen de tests uitgevoerd zullen worden voor features die geïmplementeerd zullen worden.
- Handmatig testen en/of unit-testen van de plug-in implementatie in Sense. VDG zal de plug-in architectuur implementeren in Sense, wat betekent dat de ONVIF camera's via de plug-in kunnen worden gebruikt. Via deze implementatie is te controleren of de camera's de gewenste resultaten geven. Wat het gewenste resultaat is zal per test worden gedefinieerd in samenspraak met de begeleider.

Deze tests zijn onderscheiden in twee situaties:

- Tijdens de ontwikkeling van de driver: Aan het eind van elke sprint wordt gecontroleerd of het doel voor die sprint behaald is. Dit gebeurt door via beide methoden de ontwikkelde features te testen.
- Na de ontwikkeling van de driver: Wanneer alle features ontwikkeld zijn (en dus getest zijn), is het noodzakelijk om een regressietest uit te voeren, om te controleren of alle features nog steeds voldoen. Vervolgens is het mogelijk om extra tests uit te voeren, vooral gericht op hoe de Sense implementatie omgaat met de ONVIF plug-in, zoals Stress Tests & Load Tests (hoe functioneert het systeem met veel camera's?), en Stability Tests (kan het systeem een langere periode draaien zonder problemen?).

Ten tijde van de testsprint zal gespecificeerd worden welke tests dit precies omvat en hoe deze uitgevoerd zullen worden. Het is onnodig om dit op voorhand te specificeren zolang de eindsituatie niet bekend is. Daarnaast is er in de testsprint voldoende ruimte om te onderzoeken wat voor tests van belang zijn om de kwaliteit van de plug-in te waarborgen.

8.4. Sprint resultaat

[\[Bijlage M. Testrapport\] Sprint 3](#)

Tijdens het inplannen van de ontwikkel-sprints is voor elke sprint een einddoel gesteld. De volgende ONVIF doelen zijn gedefinieerd:

DeviceService functions:

- Capabilities: - GetCapabilities
- System: - GetDeviceInformation
- GetSystemDateAndTime
- Relay Outputs: - GetRelayOutputs
- SetRelayOutputSettings
- SetRelayOutputState

MediaService functions:

- Video Streaming: - GetProfiles
- GetStreamUri
- Video Encoder Configuration: - GetVideoEncoderConfiguration
- GetVideoEncoderConfigurations
- SetVideoEncoderConfiguration
- GetVideoEncoderConfigurationOptions

Core functions:

- User Authentication: - WS-UsernameToken Authentication
- HTTP Digest

De resultaten van de testtool zijn zichtbaar in [figuur 7](#).

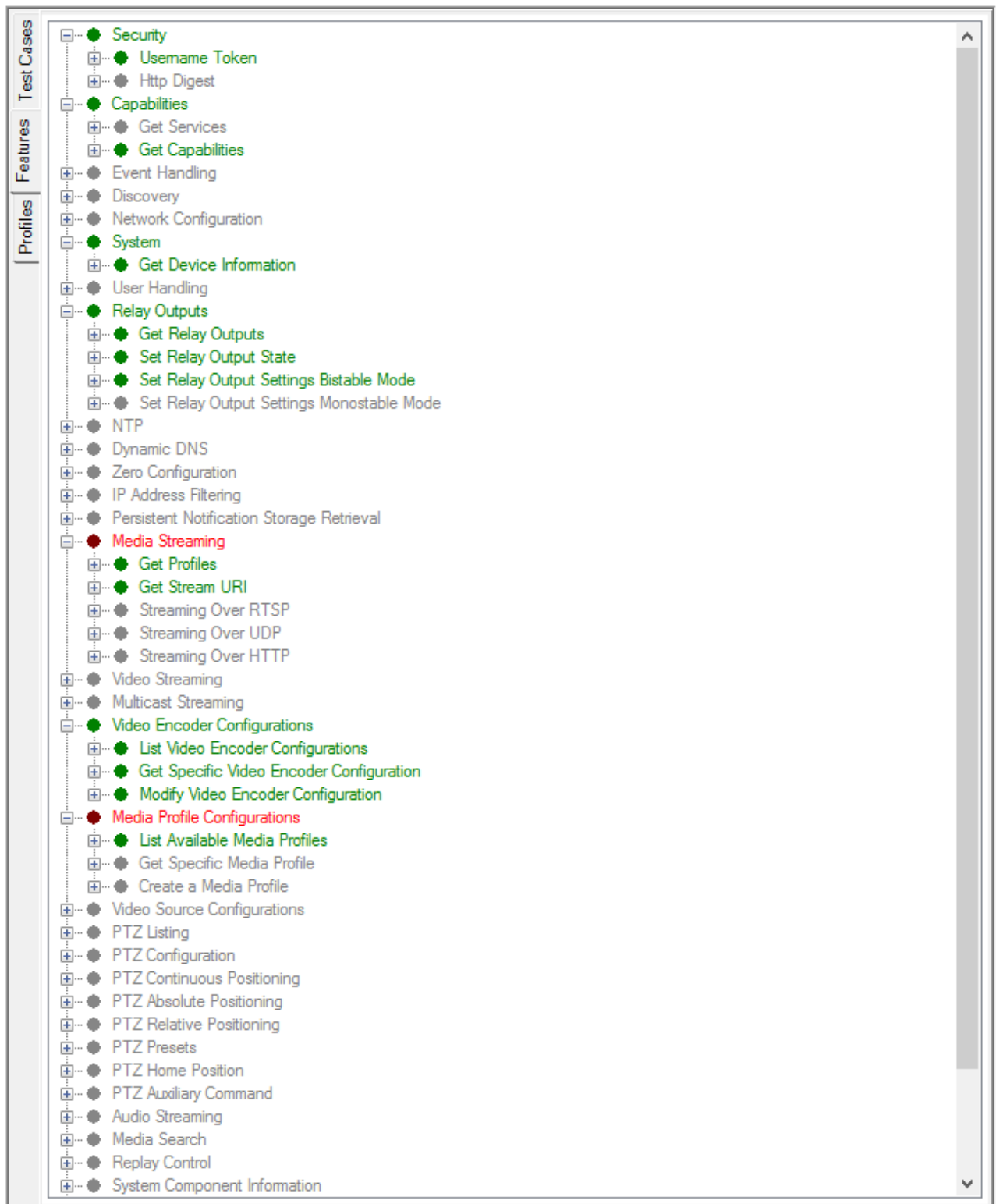
Hierbij zijn enkele punten van belang om te benadrukken:

HTTP Digest: Deze methode van autorisatie wordt niet door elke camera ondersteund. Er is op dit moment wel code voor ontwikkeld om via deze methode te autoriseren, maar er is momenteel geen camera beschikbaar om dit op te testen.

GetServices: Deze functie is niet beschreven in eerdere documenten. Er zal in de volgende sprints uitgezocht worden of deze functionaliteit dient te worden geïmplementeerd (aangezien de testtool de "Capabilities" kolom momenteel als "SUPPORTED" beschouwd).

Set Relay Output Settings Monostable Mode: Enkele camera's in de huidige testset ondersteunen deze modus niet of beperkt. Er is geprobeerd om hier een volledige test voor te schrijven, maar dit is verplaatst aangezien er belangrijkere taken geprioriteerd worden.

Hiermee is de sprint succesvol afgerond en kan de volgende sprint gestart worden zonder veranderingen of aanpassingen naar aanleiding van de resultaten van deze sprint.



Figuur 7: Sprint 3 resultaat - ONVIF Client Test Tool (Features)

9. Sprint 4: Ontwikkeling MediaService

In deze sprint is verder gewerkt aan de MediaService, waar in de vorige sprint een start mee is gemaakt. Hierbij zijn de volgende doelen gespecificeerd:

- Via de ONVIF plug-in het opzetten van een video- en audiostream in Sense in de architectuur
 - Videostreamen in MJPEG, MPEG4 en H264
 - Multicast videostream configureren, starten en stoppen
 - Audiostreamen in G.711
- Vinden van camera's in het netwerk volgens het Discovery protocol zoals deze beschreven is in ONVIF

Dit hoofdstuk beschrijft de veranderingen die aan het ontwerp hebben plaatsgevonden tijdens deze sprint, hoe de ontwikkeling van de bovenstaande doelen is opgepakt en waar dit tot heeft geresulteerd in deze sprint.

9.1. Aanpassingen ontwerp

Tijdens de ontwikkeling van deze sprint zijn er ontwerpkeuzes gemaakt of veranderd. De belangrijkste worden hier uitgelegd. Dit is opgenomen in de nieuwe versie van het ontwerp document [\[Bijlage H. Design Document\]](#).

9.1.1. ISystemTimeService

Zoals beschreven in "Sprint 3: Ontwerp aanpassingen" [\[8.1\]](#) is besloten om een BaseService toe te voegen aan het ontwerp, die het configureren van de SOAP Services onder één kap brengt. Een belangrijk aspect hiervan is het opvragen van de systeemtijd van het apparaat. Deze wordt gebruikt voor het berekenen van de autorisatiegegevens, en het is niet toegestaan om met een apparaat te communiceren dat meer dan 10 seconden verschilt van degene die de aanvraag doet.

Door het aanmaken van de MediaService is duidelijk geworden dat de huidige implementatie (een static variable in de BaseService) ongedefinieerd gedrag oplevert wanneer meerdere services actief zijn. Vandaar is er gekozen om de BaseService een nieuwe afhankelijkheid te geven: de ISystemTimeService. Deze interface beschrijft de functie waarmee het tijdverschil tussen apparaat en server opgehaald kan worden.

In de praktijk betekent dit dat de DeviceService uit Sprint 3 nu mede een ISystemTimeService is, waar elke BaseService afhankelijk van is, zo ook een DeviceService zelf. Het berekenen van dit tijdverschil is nu de verantwoordelijkheid van de ISystemTimeService die aan BaseService verbonden is (zie [figuur 8](#) voor de simpele weergave of [\[Bijlage L. Redesign ClassDiagram Sprint 4\]](#) voor het volledige klassediagram).

9.1.2. Redesign Functionmap

In het originele ontwerp is de functionmap bedacht [\[7.1.2\]](#). Deze zou beschrijven welke SOAP functies uitgevoerd kunnen worden door de plug-in. Deze functionmap zou gebruik maken van de individuele services die werden gegenereerd uit de ONVIF WSDL. Echter is gebleken dat deze manier van generalisatie erg veel tijd kost. Daarom is in overleg met de begeleider besloten om een andere aanpak te kiezen. Elke *DeviceFeature* zal gebruik gaan maken van één of meerdere SOAP Serviceklassen. Echter, om wel aan de gewenste eisen van makkelijke wisseling van SOAP implementatie te voldoen, zal elke gSOAP Service worden beschreven door een *pure virtual interface*: een *I<type>Service*.

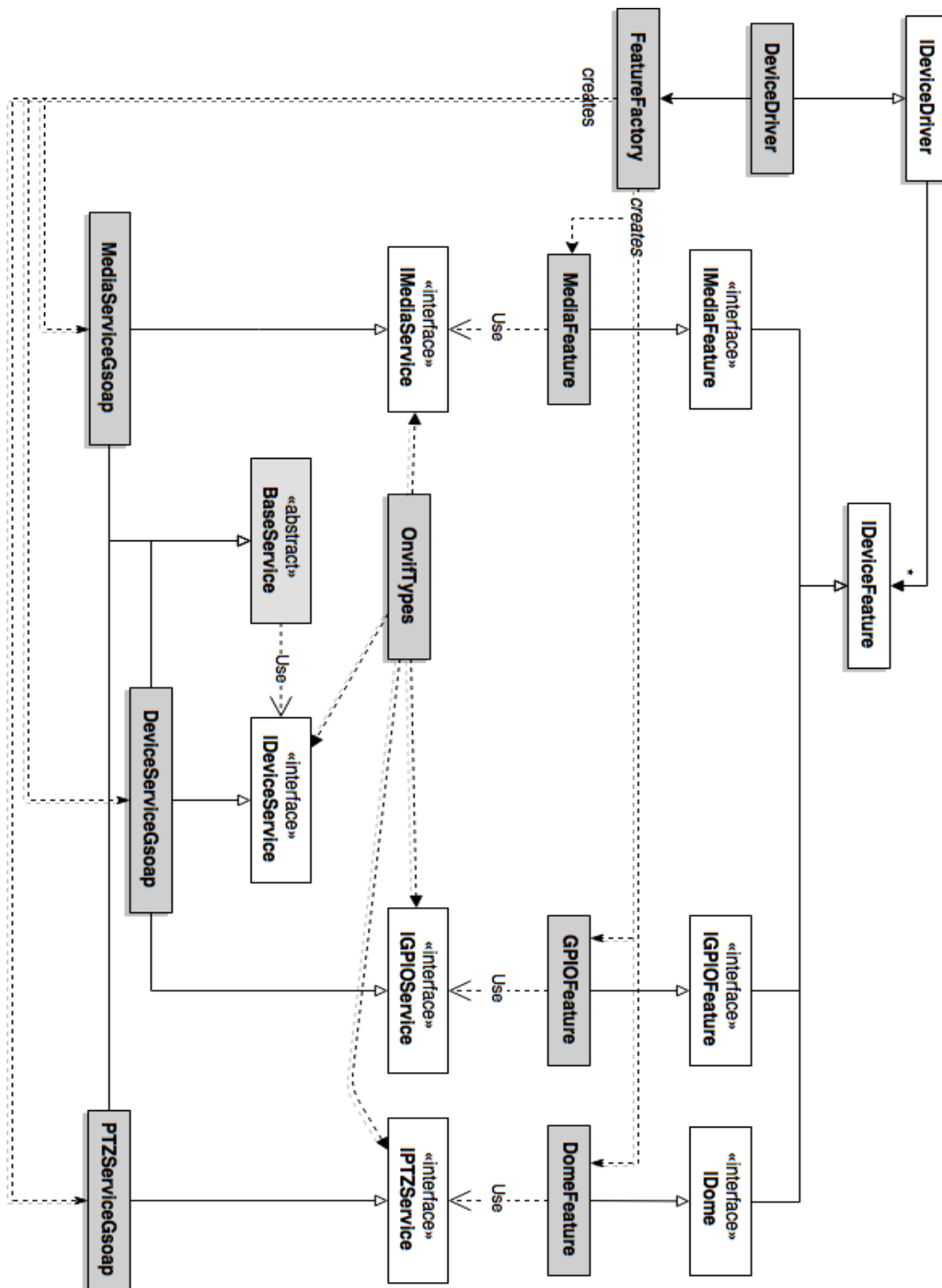
Deze *IServices* definiëren de informatie die uit een service kan komen via twee types: simpele types (integers, strings, etc.) en complexe types. Deze complexe types zijn grotere objecten die uit meerdere simpele types en/of complexe types bestaan en kan gevisualiseerd worden als zijnde klassen. Deze complexe types zijn gedefinieerd als de *OnvifTypes* (zie [figuur 8](#) voor de simpele weergave of [\[Bijlage L. Redesign ClassDiagram Sprint 4\]](#) voor het volledige klassediagram).

Vanaf dit moment wordt in de plug-in onderscheid gemaakt tussen twee aspecten. Ten eerste zijn er de *Features*, die implementaties zijn van de *DeviceFeatures* zoals *IDome* en *IMediaFeature*. Daarnaast zijn er de *Services* zoals *DeviceServiceGSoap* en *MediaServiceGSoap*. Dit is de SOAP communicatie, die generaliseerd is door de *IServices* zoals *IDeviceService* en *IMediaService*.

9.1.3. *ISystemTimeService* wordt *IDeviceService*

De *ISystemTimeService* die is ontwikkeld in deze sprint [\[9.1.1\]](#) heeft het gewenste resultaat opgeleverd. Door de opsplitsing is het makkelijker geworden om voor elke service de autorisatie te regelen op basis van het tijdsverschil. Echter is gebleken dat de services er baat bij hebben om meer dan alleen de tijd op te kunnen vragen.

In combinatie met het redesign is gekozen om de *ISystemTimeService* te vervangen voor de nieuwe interfaceklasse *IDeviceService*. Vanuit plug-in perspectief is dit een logische stap, aangezien elke service onderdeel is van het apparaat en de informatie over zichzelf moet kunnen opvragen (zie [figuur 8](#) voor de simpele weergave of [\[Bijlage L. Redesign ClassDiagram Sprint 4\]](#) voor het volledige klassediagram).



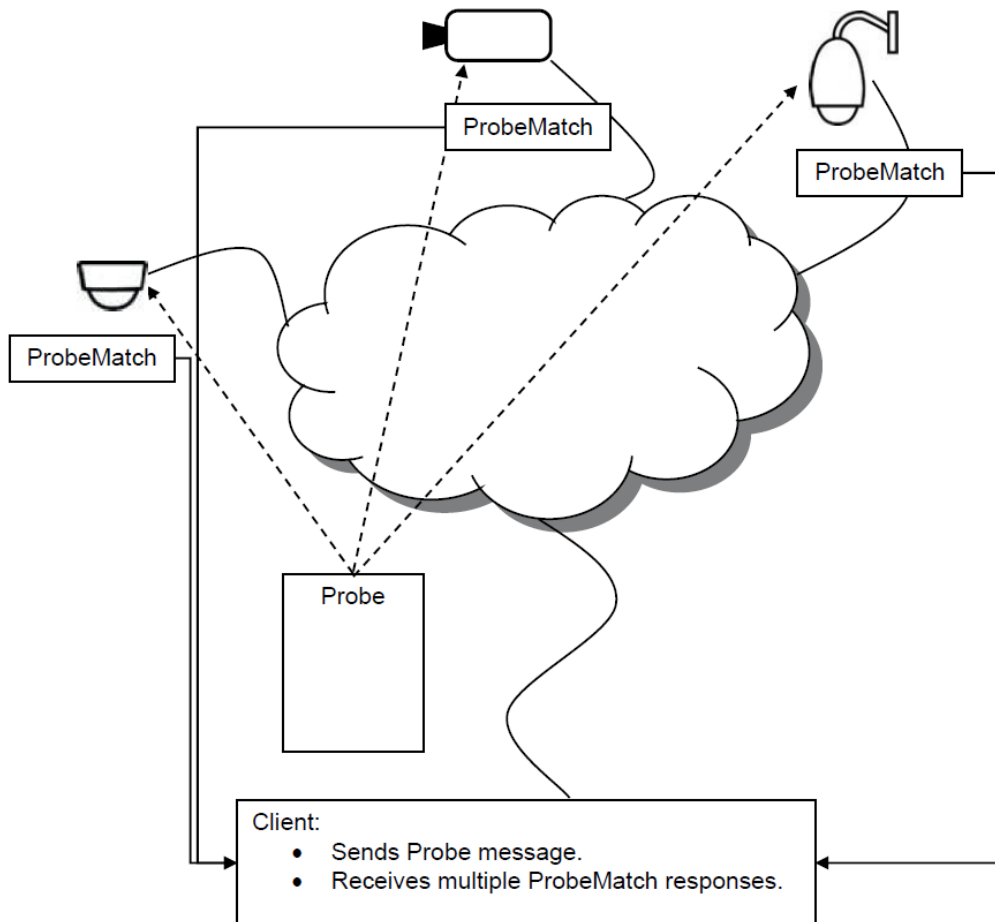
Figuur 8: Redesign ProfileSFunctionMap naar aparte IServices met OnvifTypes (simpel)

9.2. Discovery

9.2.1. WS-Discovery

Discovery is het proces van apparaten in een netwerk vinden zonder kennis van hen te hebben. ONVIF beschrijft hiervoor het WS-Discovery protocol. Voor de plug-in zijn we geïnteresseerd in de implementatie vanuit de Client.

Die Client is verantwoordelijk voor twee dingen: Het versturen van de *Probe Message* als broadcast in het netwerk en de reacties van camera's ontvangen: de *Probe Matches*. Dit algemene principe wordt weergegeven in [figuur 9](#).



Figuur 9: Visualisatie Discovery principe (source: ONVIF Programmer's Guide [\[5\]](#))

ONVIF verplicht het gebruik van de standaard WS-Discovery. Zij beschrijft onder andere hoe de *Probe Messages* en *Probe Matches* geformatteerd zijn.

9.2.2. Uitstel

Vanwege het redesign [\[9.1.1 Redesign Functionmap\]](#) is in overleg met de begeleider besloten om het ontwikkelen van Discovery uit te stellen naar de laatste sprint. Dit besluit is tevens genomen door de onzekerheid of de architectuur wel correct voorbereid is op het implementeren van het protocol in de plug-in.

9.3. Sprint resultaat

[\[Bijlage M. Testrapport\] Sprint 4](#)

Voor deze sprint waren de volgende ONVIF doelen opgezet:

Video Streaming:

- Media Streaming using RTSP
- Media Streaming using RTSP - JPEG RTP header extension
- Video Streaming (MPEG4 & H264)
- SetSynchronizationPoint

Multicast:

- StartMulticastStreaming
- StopMulticastStreaming

Audio Streaming:

- AddAudioSourceConfiguration
- GetCompatibleAudioSourceConfigurations
- AddAudioEncoderConfiguration
- GetCompatibleAudioEncoderConfigurations

Discovery:

- WS-Discovery

Met als einddoel dat het mogelijk is om via de ONVIF plug-in in Sense een camera video- en audiostream te kunnen weergeven.

[Figuur 10](#) laat het resultaat zien van de ONVIF Client Test Tool aan het eind van deze sprint. Hier wordt het volgende zichtbaar:

Discovery: Zoals beschreven in [\[9.2\]](#) is het niet gelukt om Discovery deze sprint te implementeren, en zal deze opgepakt worden in een latere sprint.

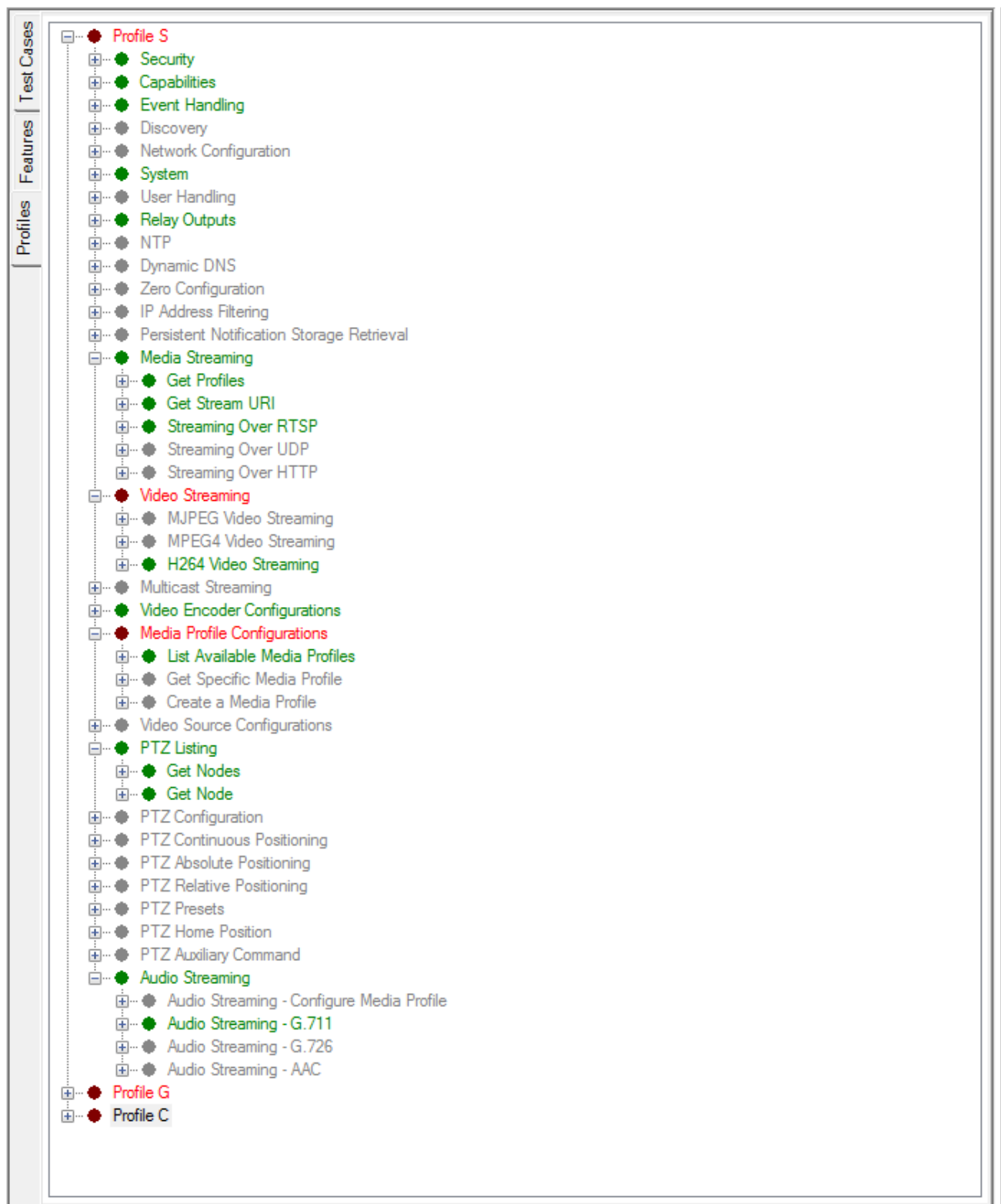
Media/Video Streaming: Het is gelukt om een Media Stream over RTSP te configureren met de plug-in en dit in Sense te laten zien. Hierbij is een H264 stream van de testcamera naar Sense gestuurd en is het beeld getoond in Sense. Echter, de test van Video Streaming is nog niet compleet, aangezien er minimaal een MJPEG stream verzonden moet worden. Het is al mogelijk dit te configureren in de plug-in, maar dit is hier niet getest door gebrek aan implementatie van de plug-in architectuur voor dit protocol in Sense. Hetzelfde geldt voor de MPEG4 configuratie.

Multicast: Er is gebleken dat het configureren van Multicast op meer dan alleen de MediaService betrekking heeft. Het is niet direct gelukt om dit goed te configureren op de gebruikte testcamera en zal in de volgende sprint worden meegenomen.

Audio Streaming: Het is gelukt om een G.711 audio stream op te halen via de plug-in. Echter bevat Sense momenteel nog geen plug-in-architectuur-implementatie van de audiostream, waardoor deze alleen op ONVIF niveau is getest.

PTZ: Tijdens deze sprint is een begin gemaakt aan de volgende service, namelijk PTZService. Deze kan nu alleen nog de aanwezige PTZNodes ophalen.

Hiermee is de sprint naar omstandigheden goed afgerond. Multicast Streaming en Discovery zijn verplaatst naar latere sprints naar aanleiding van het redesign dat in deze sprint heeft plaatsgevonden. Het einddoel van de sprint is gehaald, door audio en camerabeeld via de plug-in in Sense weer te geven.



Figuur 10: Sprint 4 resultaat - ONVIF Client Test Tool (Profiles)

10. Sprint 5: Ontwikkeling PTZService

Het doel voor sprint 5 is het ontwikkelen van de PTZ Feature geweest. De volgende ONVIF doelen zijn gedefinieerd:

PTZ:

- Een camera bewegen via ContinuousMove
- Een camera bewegen door deze naar een preset te sturen
- Een camera bewegen door deze naar zijn home positie te sturen
- De Auxiliary commando's van een camera aanroepen

Dit hoofdstuk beschrijft de ontwerpkeuzes en aanpassingen die tijdens deze sprint hebben plaatsgevonden en het eindresultaat van de sprint.

10.1. Aanpassingen ontwerp

Tijdens de ontwikkeling van deze sprint zijn er ontwerpkeuzes gemaakt of veranderd. De belangrijkste worden hier uitgelegd. [\[Bijlage H. Design Document\]](#).

10.1.1. Niet opsplitsen IDome

In hoofdstuk [\[8.2 Voorstel verandering architectuur\]](#) is beschreven dat er twijfel is of bepaalde functionaliteiten horen onder de gedefinieerde *DomeFeature*. Net zoals toentertijd is besloten om dit te laten zoals het is, naar aanleiding van de volgende punten:

- De functies die niet met Dome besturing te maken hebben (zoals auto-focus en iris) [\[8.2\]](#) verplaatsen naar een andere heeft grote gevolgen voor de manier hoe VDG omgaat met camerabesturing. Hier komt meer aan te pas dan alleen het veranderen van de *feature* interface en dient dus meer tijd te krijgen om dit goed uit te werken, zodat deze verandering juist in Sense kan worden opgenomen.
- De ONVIF implementatie hoeft hier nu nog niet op voorbereid te zijn. Dit zijn functionaliteiten die buiten het Profiel S gebied vallen, namelijk de configuratie van *Imaging Settings*.

10.1.2. Sense implementatie en testen

Tijdens sprint 4 is het VDG duidelijk geworden dat het implementeren van de plug-in architectuur in Sense meer tijd zal kosten. Aangezien men dit netjes wil kunnen implementeren, is aangegeven dat het niet mogelijk zal zijn om tijdens de afstudeerperiode alle *DeviceFeatures* te implementeren in Sense. Dit betekent dat het niet meer mogelijk is om in Sense te testen of een geïmplementeerde feature werkt of niet.

Als vervanging hiervoor heeft VDG aangegeven een testtool te zullen ontwikkelen. Deze ontwikkelprocedure zal worden gestart zodra hier tijd voor is en er wordt verwacht tijdens de testsprints deze te kunnen gebruiken. Wanneer het mogelijk is dat externe bedrijven plug-ins kunnen maken, wil VDG deze testtool ook kunnen gebruiken voor het valideren van plug-ins.

De VDG Plug-in Testtool zal de plug-in inladen, controleren of alle functionaliteiten de verwachte resultaten geven en hier een visueel rapport van aan de gebruiker tonen.

Dit betekent dat het [\[Bijlage K. Testplan\]](#) m.b.t. het testen van de plug-in architectuur en de Sense implementatie niet meer van toepassing is. Er zal een nieuw plan gemaakt worden die zal beschrijven wat en hoe de eindsituatie getest zal worden met behulp van de testtool. Dit zal gebeuren in de testsprint [\[12.1\]](#).

10.2. Sprint resultaat

[\[Bijlage M. Testrapport\] Sprint 5](#)

Voor deze sprint waren de volgende ONVIF doelen opgezet:

PTZService:

- GetNodes
- GetNode
- GetConfigurations
- AddPTZConfiguration
- ContinuousMove
- Stop

Presets:

- GetPresets
- GotoPresets
- RemovePreset

Home:

- GotoHomePosition
- SetHomePosition

Auxiliary:

- SendAuxiliaryCommand

Met als einddoel om de camera te kunnen besturen via de Sense besturing, hem naar een preset of home te sturen, en het aanroepen van de Auxiliary Commands. Voor het testen van deze functies is een nieuwe Dome-camera aan de testcameraset toegevoegd.

[Figuur 11](#) laat het resultaat zien van de ONVIF Client Test Tool aan het eind van deze sprint. Hier wordt het volgende zichtbaar:

Relay Outputs: Door het toevoegen van verschillende nieuwe camera's aan de testset is gebleken dat niet elke camera goed reageert op de huidige testcases voor relays. Dit betekent niet dat de implementatie niet werkt, maar dat er functies aangeroepen worden waarvan de camera het bestaan niet weet. De testcases zullen moeten worden verbeterd in de testfase zodat dit eerst gecontroleerd wordt.

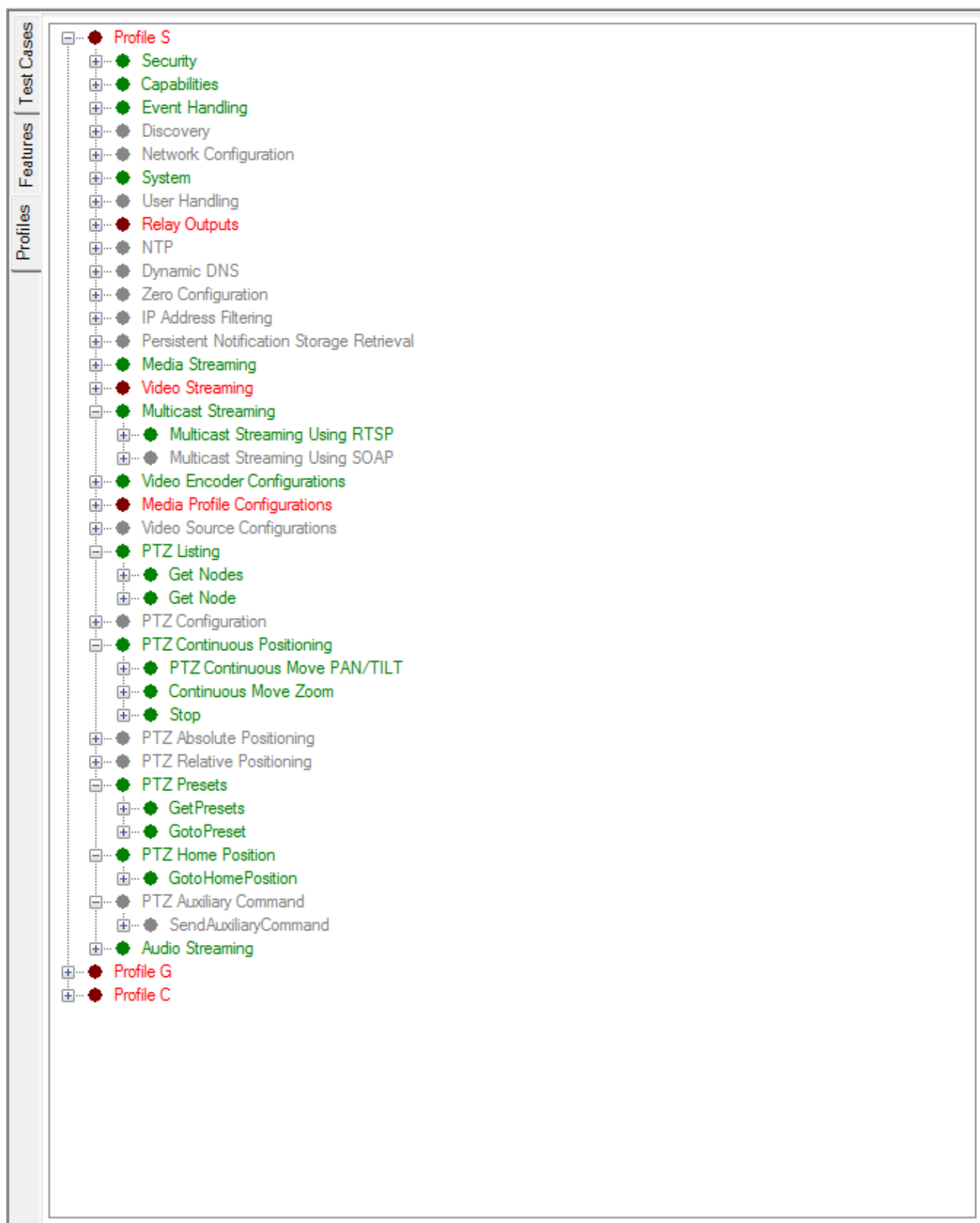
Multicast: In deze testopstelling is men erin geslaagd de nieuwe Dome-camera te configureren voor multicast, waardoor de RTSP Multicast Streaming test afgerond is.

Video Streaming: Net zoals in Sprint 4 is er geen camera geconfigureerd voor het versturen van een MJPEG stream, waardoor de testcase niet compleet is.

PTZ: De testcase geeft weer dat men erin is geslaagd om ContinuousMove te gebruiken en te stoppen, presets op te halen en deze te gebruiken, en home te gebruiken.

Auxiliary Commands: Alhoewel er code is geschreven om de auxiliary commands op te halen en aan te roepen, is gebleken dat de gekozen Dome-camera geen auxiliary commands heeft. In de testfase zal dit getest worden op een andere Dome-camera die dit wel kan.

Hiermee is de sprint succesvol afgerond. Alleen 'auxiliary commands' is niet door de testtool goedgekeurd, maar de code is gereviewd en er wordt verwacht dat dit bijna geen tot geen problemen zal opleveren.



Figuur 11: Sprint 5 resultaat - ONVIF Client Test Tool (Profiles)

11. Sprint 6: Ontwikkeling Events & Discovery

Deze sprint, die van origine als “uitloopsprint” was gepland voor het toevoegen van functionaliteiten of taken die niet volbracht waren wanneer gepland, heeft de volgende doelen:

- Event Handling (Push): Het inschrijven bij een camera voor het ontvangen van Events en bij het ontvangen van een Notify het bericht doorgeven aan Sense.
- WS-Discovery: het in Sprint 4 geplande [\[9.2 Discovery\]](#) implementeren zodat ONVIF apparaten in het netwerk kunnen worden gevonden en teruggegeven aan Sense.
- NTP
- Media Profile- en Video Source Configuratie

Hierbij heeft de ontwikkeling van Discovery een hogere prioriteit dan de ontwikkeling van Events, op basis van de voorkeur van VDG en de voortgang van de architectuur van deze features. Door de verwachte werkdruk zijn Media Profile Configuratie en Video Source Configuratie van de planning verwijderd, en zal NTP behandeld worden indien er tijd over is.

11.1. Aanpassingen ontwerp

Tijdens de ontwikkeling van deze sprint zijn er ontwerpkeuzes gemaakt of veranderd. De belangrijkste worden hier uitgelegd. [\[Bijlage H. Design Document\]](#).

11.1.1. Afhankelijkheid Dome van MediaFeature

Tijdens de ontwikkeling van de PTZ functionaliteiten is gebleken dat de ONVIF berichten in de PTZService erg afhankelijk zijn van het Media-profiel van de MediaFeature en MediaService. Hiervoor is tijdens Sprint 5 een quickfix gemaakt, maar er is besloten om dit in deze sprint netter te maken. Hiervoor zijn twee oplossingen bedacht:

1. PTZFeature mag ook informatie ophalen over de keuze van de gebruiker (die verantwoordelijk is voor het kiezen van het juiste profiel) en bepaalt aan de hand hiervan welk MediaProfileToken gebruikt wordt.
2. Dome kan het gekozen MediaProfile opvragen bij de MediaFeature, waar hij dan van afhankelijk zou worden.

Er is gekozen voor optie twee om de volgende redenen:

- In realiteit kan PTZ alleen plaatsvinden als er beeld is geconfigureerd. Dit betekent dat er nooit een DomeFeature kan bestaan zonder dat er een MediaFeature bestaat, net als in de realiteit.
- Door de afhankelijkheid is het mogelijk om één klasse verantwoordelijk te maken voor de keuze, namelijk de MediaFeature, die een natuurlijke controle heeft over Media profielen. Hierdoor wordt DomeFeature ook niet belast met taken waar hij niet voor verantwoordelijk is.

11.2. Sprint resultaat

[\[Bijlage M. Testrapport\] Sprint 6](#)

Voor deze sprint waren de volgende ONVIF doelen opgezet:

Event Handling:

- Notify
- Subscribe

Discovery:

- WS-Discovery

NTP

Met als einddoel om camera's te kunnen vinden in het netwerk en deze terug te geven aan Sense, en te kunnen 'subscriben' bij een camera om events te ontvangen en deze via Notify berichten uit te lezen en door te geven aan Sense.

Sinds deze sprint is een nieuwe versie van de ONVIF Client Test Tool uitgebracht, namelijk v16.01. Deze zal moeten worden gebruikt om conformance te behalen, dus zal vanaf nu ook gebruikt worden in de tests, totdat een nieuwere versie uitgebracht wordt.

[Figuur 12](#) laat het resultaat zien van de ONVIF Client Test Tool v16.01 aan het eind van deze sprint. Hier wordt het volgende zichtbaar:

Discovery: Het is gelukt om in deze sprint Discovery te implementeren. Hierdoor kan de plug-in ONVIF apparaten in het netwerk ontdekken en deze aan Sense retourneren.

Events: Het is niet gelukt om de Event Handling Push methode te ontwikkelen. Door tijdgebrek naar aanleiding van het Discovery protocol is besloten om deze taak voor nu achterwege te laten. Daarnaast is het ook duidelijk geworden dat ONVIF-Devices zelf kunnen aangeven welke events deze kunnen teruggeven en hoe zij geformatteerd zijn. Dit is een lastig proces en vereist meer tijd om goed uitgewerkt te worden.

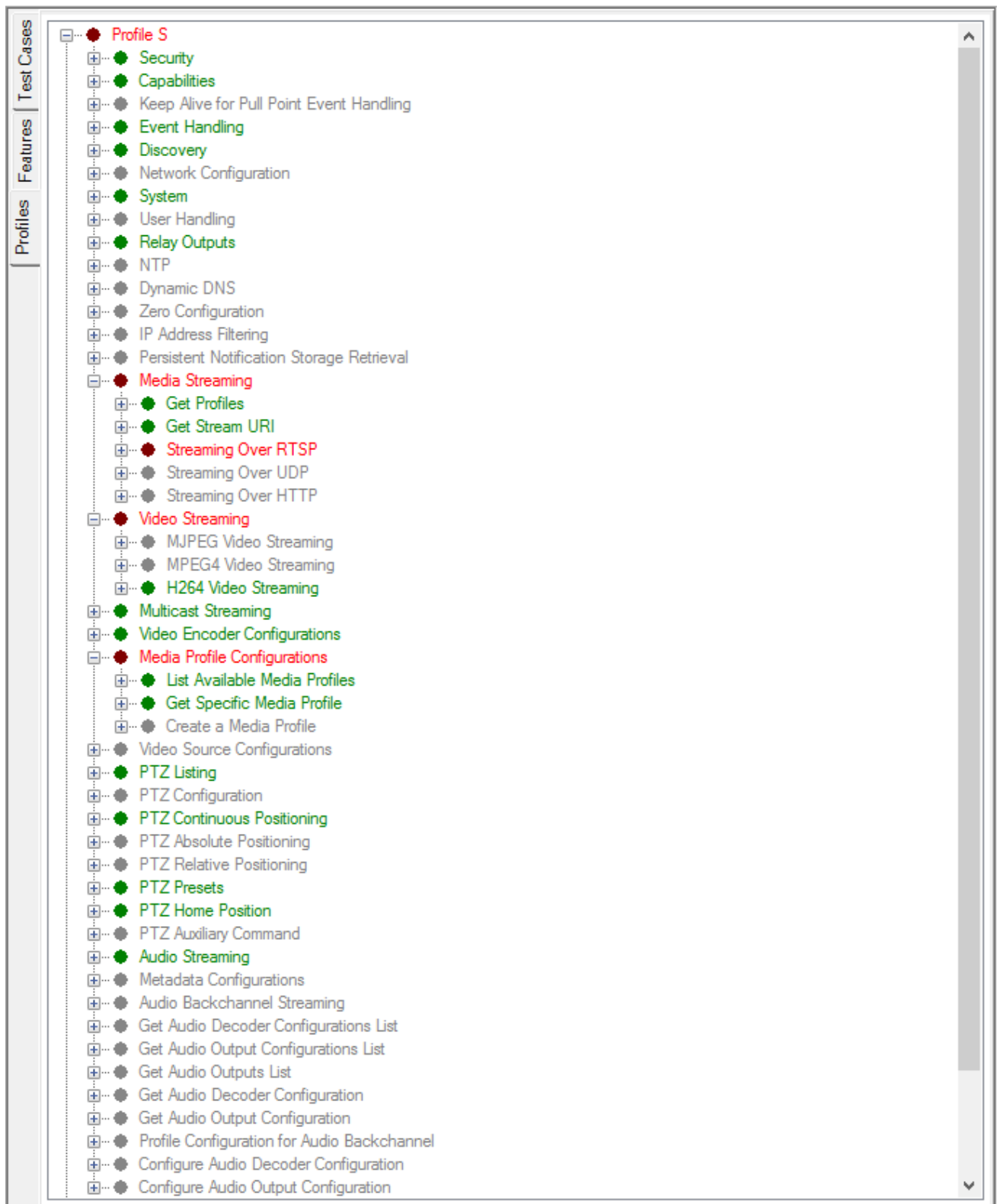
De opdrachtgever is ervan op de hoogte dat 'Events' niet meer zal worden ontwikkeld en is hiermee akkoord gegaan.

RTSP Streaming: Deze test faalt in deze testcase omdat er geen RSTP Breakdown heeft plaatsgevonden. Aangezien deze sprint niet focust op Media Streaming is dit niet verder onderzocht op dit moment. Dit zal in de testfase wel gedaan worden.

MediaProfileConfigurations: Deze set faalt omdat Create MediaProfile niet wordt uitgevoerd maar wel verplicht is. Hiervoor zal een aparte testcase worden geschreven in de testfase.

NTP: De geplande NTP implementatie is voorbereid in de plug-in, maar er is nog niet besloten hoe men deze wil toepassen. Het gevaar van de NTP server instellen op een camera wanneer deze toegevoegd wordt aan Sense, betekent dat een camera per definitie geherconfigureerd wordt. VDG moet besluiten of men dit gedrag wil forceren vanuit de ONVIF driver, hier een verandering in de architectuur voor moet plaatsvinden, of dat dit eigenlijk nooit moet gebeuren.

Hiermee is de laatste ontwikkelsprint afgerond. In de testfase zullen alle functionaliteiten nogmaals grondig getest worden. Kleine bugs die gevonden worden zullen direct opgelost worden. Grotere bugs worden genotuleerd en zullen moeten worden opgepakt door VDG na de afstudeerperiode. Aan het eind van de testfase zal het eindproduct geëvalueerd worden en duidelijk worden of alle requirements behaald zijn.



Figuur 12: Sprint 6 resultaat - ONVIF Client Test Tool (Profiles)

12. Sprint 7: Testing

Tijdens de ontwikkeling van de plug-in is al veelvuldig getest. Echter is dit telkens gebeurd over het product van de sprint. Hierdoor is niet bekend of de plug-in als geheel functioneert zoals gewenst.

Dit hoofdstuk beschrijft de testsprint die heeft plaatsgevonden. Het testonderdeel wordt uit twee perspectieven bekeken: de architectuur zal worden getest door de VDG Plug-in Test Tool en de ONVIF implementatie zal worden getest door de ONVIF Client Test Tool.

12.1. Proces

Tijdens sprint 5 is de manier van validatie en testen aangepast [\[10.1.2\]](#). VDG heeft ervoor gekozen om niet direct een implementatie van de architectuur in Sense te maken, maar eerst een aparte tool te ontwikkelen die de plug-in kan inladen en testen: de VDG Plug-in Test Tool. Dit heeft als gevolg dat de opgestelde eisen niet meer functioneel in Sense kunnen worden getest, maar dat deze eisen door de tool getest worden.

Voor de testsprint zijn drie taken gedefinieerd:

1. Het uitbreiden van de testset camera's, en deze configureren zodat alle testcases uitgevoerd kunnen worden. Door het testen van meer ONVIF implementaties van camera's is de kans groter om problemen in de plug-in te achterhalen.
2. Het oplossen van kleine bugs die in de nieuwe testset zich voordoen. Grote bugs worden gedocumenteerd en dienen na de afstudeerperiode te worden opgelost.
3. Laatste test, die het eindresultaat van de afstudeerperiode vastlegt.

12.2. VDG Plug-in Test Tool

Hoewel gepland was dat deze applicatie tijdens de testfase gebruikt zou kunnen worden, is VDG door omstandigheden er niet in geslaagd om de testtool af te maken.

De consequentie hiervan is dat het niet mogelijk is om de plug-in te testen via de plug-in architectuur. In overleg met de opdrachtgever is vastgesteld dat het niet nodig is om een andere manier te zoeken om dit aspect te testen. VDG zal de testtool afronden en zal dan de ONVIF plug-in kunnen testen via de plug-in architectuur. Tot die tijd zal de ONVIF plug-in alleen getest worden op zijn ONVIF implementatie.

12.3. ONVIF Client Test Tool

Aan het eind van elke sprint zijn met de ONVIF Client Test Tool de implementaties die tijdens de sprint ontwikkeld zijn getest. Tijdens deze sprint zal dit nogmaals gebeuren, maar zal de plug-in uitgebreider getest worden. Dit zal gebeuren door de testset van camera's uit te breiden. Aangezien elke camera een eigen implementatie van ONVIF Profile S kan hebben, zijn er verschillen tussen hoe camera's reageren op een ONVIF functie. Daarom is het noodzakelijk om in deze testfase zoveel mogelijk problemen te achterhalen zodat deze nog opgelost kunnen worden.

Dit is de enige manier waarop de tests met de ONVIF Client Test Tool uitgebreid kunnen worden. Overige tests zijn erg afhankelijk van de VDG Plug-in Test Tool, waar niet mee kan worden getest [\[12.2\]](#).

Het volgende hoofdstuk [\[12.4\]](#) beschrijft het eindresultaat met behulp van de ONVIF Client Test Tool.

12.4. Sprint resultaat

[\[Bijlage M. Testrapport\] Sprint 7](#)

Tijdens de testperiode zijn enkele kleine bugs opgelost. Er zijn geen grote bugs gevonden tijdens het testen met de ONVIF Client Test Tool.

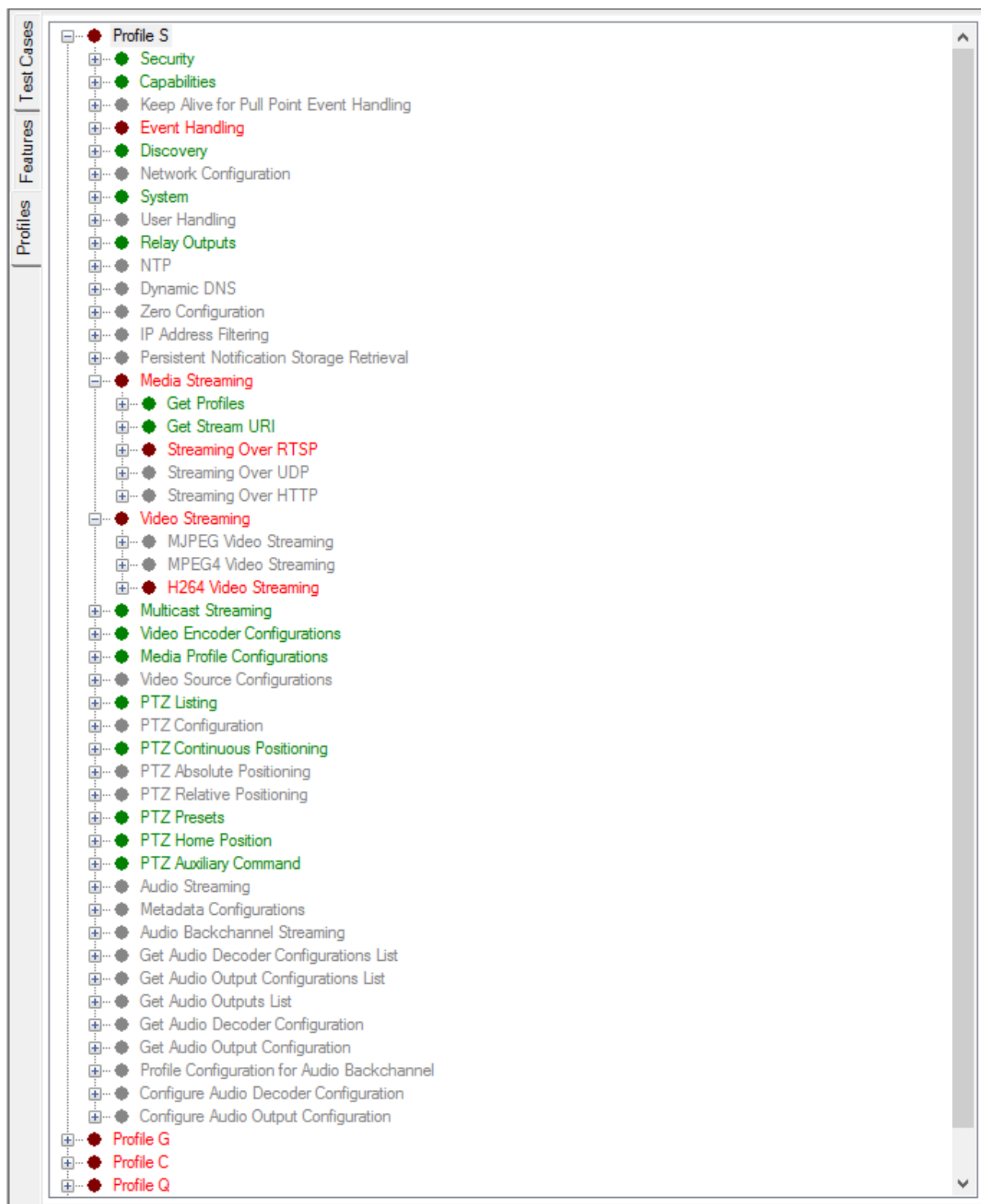
Bij het uitbreiden van de testset camera's is een camera opgenomen die wel **Auxiliary Commands** geconfigureerd had. Deze zijn getest en zijn geslaagd.

Door de afwezigheid van de Sense implementatie of van de VDG Plug-in Test Tool is het niet mogelijk geweest om een correcte **Media-** en **VideoStream** op te zetten, aangezien dit de verantwoordelijkheid is van de architectuur. Hierdoor is tevens **Metadata Streaming** van **Event Handling** niet geslaagd, omdat dit een onderdeel is van **RTSP Streaming**.

Aangezien de Test Tool niet is afgerond en er dus geen Media- en Videostreaming kan plaatsvinden is besloten om alle camera's in de standaard RTSP-H264 configuratie te laten. Wanneer de Test Tool klaar is, zal alsnog een testset opgezet moeten worden waar ook de UDP en HTTP protocollen getest worden, en de videocodecs MPJEG en MPEG4.

In [figuur 13](#) is het eindresultaat van de ONVIF Client Test Tool zichtbaar.

Dit betekent dat de huidige implementatie nog niet ONVIF Profile S Compliant is, aangezien het streaming aspect niet aanwezig is. Tijdens de ontwikkeling van de MediaService [\[9.3\]](#) is duidelijk geworden dat VDG hier wel toe in staat is, maar heeft simpelweg nog niet de tijd gehad om dit te implementeren in de testtool. Alle andere tests zijn geslaagd voor alle camera's in de testset (wanneer de feature door de camera ondersteund werd). Dit betekent dat wanneer het streaming aspect door VDG geïmplementeerd wordt, dat de ONVIF plug-in Profile S Compliant zal zijn.



Figuur 13: Sprint 7 resultaat - ONVIF Client Test Tool (Profiles)

13. Conclusie en aanbeveling

Het doel van de afstudeeropdracht bestond uit twee doelen door het ontwikkelen van één product, namelijk (1) het ontwikkelen van de ONVIF plug-in in de nieuwe plug-in architectuur om tevens (2) te bewijzen dat de architectuur voldoet en in Sense doorontwikkeld kan worden [\[4.3\]](#).

Gedurende de afstudeerperiode is de ONVIF plug-in ontwikkeld in de nieuwe plug-in architectuur. Zoals geconcludeerd is in de testsprint [\[12.4\]](#) is de ONVIF plug-in nog niet Profile S Compliant. Dit komt omdat, by design, de plug-in niet verantwoordelijk is voor streaming, en VDG er niet in is geslaagd om de tool te maken die dit zou moeten afhandelen.

Alle andere tests zijn uitgevoerd op de testset camera's en zijn als SUPPORTED beschouwd. De enige uitzondering hierop zijn de 'Events'. Door de hogere prioriteit van 'Discovery' is men hier niet meer aan toe gekomen.

Als het streaming aspect wordt geïmplementeerd in de VDG Plug-in Test Tool of in Sense, dan zal de ONVIF plug-in Profile S Compliant zijn. De features die zij dan ondersteund zijn:

- Mediastreaming (MJPEG, MPEG4, H264, Multicast)
- Audiostreaming (G.711)
- WS-Discovery
- PTZ (ContinuousMove, Presets, Home, Auxiliary)
- Relay Outputs
- System (GetSystemDateAndTime, Authentication (WS-Security), Capabilities)

Hiermee heeft de afstudeeropdracht de twee gestelde doelen [\[4.3\]](#) behaald, namelijk:

- Het testen van de plug-in architectuur door de ONVIF plug-in te ontwikkelen en te controleren of de architectuur voldoet om deze in Sense op te nemen en door te ontwikkelen.
- Het onderzoeken van ONVIF en een volgens de ONVIF Profile S specificatie correcte ONVIF plug-in ontwikkelen binnen de nieuwe architectuur.

Aanbevelingen

Er zijn enkele aspecten waar VDG naar moet kijken na de afstudeeropdracht.

Ten eerste zal de test tool afgerond moeten worden, zodat de streaming tests uitgevoerd kunnen worden.

Vervolgens moet er een besluit worden genomen wat betreft NTP [\[11.2\]](#). De ONVIF code hiervoor is opgezet maar niet getest, om camera's zo min mogelijk te herconfigureren. Als een cameratijd niet gelijk is aan de servertijd, mag de server dan verantwoordelijk zijn voor het herconfigureren van de cameratijd? Biedt de impact (zoals een tijdsprong bij herconfiguratie) die NTP heeft genoeg voordelen?

Een ander belangrijk aspect dat tijdens de afstudeerperiode niet verder is uitgezocht is hoe men wil omgaan met ONVIF feedback. Een normale driver in de architectuur zou simpelweg informatie teruggeven, zoals de StreamUri of het aantal outputs die de camera heeft. Echter, aangezien een ONVIF camera hier een netwerkcall voor moet doen, kan het zijn dat de aanvraag niet slaagt, bijvoorbeeld bij incorrecte authenticatie. Voldoet het om een leeg antwoord te geven, of loont het om via de plug-in architectuur errors te kunnen opvragen?

Tot slot wordt aanbevolen om de Sense Client User Interface voor te bereiden op specifieke ONVIF camera's. De extra informatie die ONVIF over een camera kan geven blijkt vaak erg waardevol tijdens configuratie en het zou installateurs kunnen helpen bij het configureren van de camera.

14. Evaluatie

In dit hoofdstuk zal het opgeleverde werk kritisch gereflecteerd worden.

Wanneer men over de 'begeleider' spreekt, is dit de begeleider binnen VDG. De begeleider van de studie zal specifiek de 'studiebegeleider' genoemd worden.

14.1. Producten

ONVIF Research: [\[Bijlage C\]](#) Het ONVIF onderzoek is opgedeeld geweest in twee onderdelen: de initiële kennismaking met ONVIF en het onderzoek naar de wensen van VDG m.b.t. ONVIF. Deze verdeling heeft het mogelijk gemaakt om eerst een overzicht te krijgen van de mogelijkheden en niet direct beïnvloed te worden door wat het bedrijf wil.

Achteraf was het misschien handiger geweest om deze onderdelen sneller elkaar te laten opvolgen. In dat geval zou het mogelijk zijn geweest om de Scope in het Plan van Aanpak direct te kunnen notuleren i.p.v. deze later in te vullen.

De resultaten van het onderzoek zijn tijdens het onderzoek teruggekoppeld aan de begeleider en de ONVIF expert, waardoor de juiste vragen zijn onderzocht en beantwoord.

Plan van Aanpak: [\[Bijlage D\]](#) Het Plan van Aanpak is opgezet op basis van de "Checklist volledigheid Plan van Aanpak", zoals naar gerefereerd wordt in het document. Alle aspecten van dit document zijn gereviewd door de begeleider. Naar aanleiding hiervan is enkel de planning aangepast op aanraden van de begeleider.

Design document: [\[Bijlage H\]](#) Het ontwerpdocument is een belangrijk document geweest gedurende de ontwikkeling en is ook als zodanig behandeld volgens de werkmethodiek. Na de planningsverandering om hier initieel maar één sprint aan te besteden, is deze tijdens elke opvolgende sprint behandeld en geüpdatet om de nieuwe beslissingen te reflecteren. In de eerste versie is zij gebruikt als visualisatie van het doel. Naarmate het afstudeertraject vorderde, is deze aangepast tot zij de huidige implementatie van de plug-in reflecteert.

Testplan: [\[Bijlage K\]](#) Het testplan is in eerste instantie opgezet om te beschrijven welke tests uitgevoerd zouden worden, zowel tijdens als na de ontwikkelperiode. Naar aanleiding van de keuze van VDG om niet meer in Sense verder te gaan, maar in de Test Tool, is het doel van de architectuur-tests aangepast. Echter, aangezien VDG er niet in is geslaagd om de Test Tool af te ronden, is het niet meer mogelijk geweest om de architectuur te testen.

Voor het testen van ONVIF is besloten om de ONVIF Client Test Tool te gebruiken, waarvan alle testcases gedocumenteerd zijn in de Profile S Test documentatie.

Testrapport: [\[Bijlage M\]](#) Alle tests die zijn uitgevoerd aan het einde van de sprints en in de testsprint zijn verzameld in het testrapport. Door het gebrek aan de VDG Plug-in Test Tool of een alternatief is deze beperkt tot de ONVIF Client Test Tool resultaten. Alhoewel dit erg beperkt is, reflecteert dit wel de situatie die zich heeft voorgedaan tijdens de testsprint.

14.2. Proces

In het [\[Bijlage D. Plan van Aanpak\]](#) is gekozen om Scrumban te gebruiken. Hiermee is met enkele kleine aanpassingen de werkmethode van VDG gevolgd. Dit heeft erg geholpen bij de uitvoering van het afstudeerproject.

De consequenties van veel reviews en de impact die een 'slechte' review meedraagt heeft bijgedragen aan de kwaliteit van alle producten die opgeleverd zijn, zowel in geschrift als in code. Hierdoor is er vertrouwen dat gedurende de afstudeerperiode de juiste keuzes gemaakt zijn en dat het eindproduct als goed product kan worden overgedragen aan VDG.

Ook heeft de ontwikkelmethode de mogelijkheid geboden tot de gewenste flexibiliteit. Zo is meerdere malen de planning aangepast n.a.v. een verandering van prioriteiten of wensen [\[6.3.3\]](#) [\[9.2.2\]](#) [\[11.0\]](#). Als gevolg hiervan zijn de meest belangrijke aspecten uitgevoerd en is uiteindelijk maar één aspect niet geslaagd: de 'Events'. Achteraf gezien was deze sowieso te groot geweest om volledig te implementeren, dus kan beoordeeld worden dat voor de voortgang van het traject hier de juiste keuze is gemaakt.

14.3. Eindproduct

Het probleem dat VDG had aan het begin van de afstudeeropdracht was als volgt: "Het integreren van een nieuwe camera in Sense is een veel voorkomende klus die veel tijd kost". [\[4.1\]](#)

De ONVIF plug-in is erin geslaagd om volgens de Profile S specificaties alle berichten te kunnen versturen en is dus, los van de media- en videostreaming (die volgens de plug-in architectuur geen verantwoordelijkheid is van de plug-in, maar van Sense), Profile S Compliant. [\[12.4\]](#)

De plug-in architectuur is hierbij deels getest. Door het maken van de ONVIF plug-in en het ontwikkelen van de VDG Plug-in Test Tool zijn aanpassingen gedaan aan de architectuur en is VDG er nu van overtuigd dat de nieuwe architectuur een belangrijke toevoeging zal zijn aan de bestaande applicatie. Echter, aangezien de test tool nog niet af is, is niet te controleren of de plug-in volledig voldoet aan de opgezette architectuur. [\[12.2\]](#)

Hiermee heeft de afstudeeropdracht de twee gestelde doelen [\[4.3\]](#) behaald, namelijk:

- Het testen van de plug-in architectuur door de ONVIF plug-in te ontwikkelen en te controleren of de architectuur voldoet om deze in Sense op te nemen en door te ontwikkelen.
- Het onderzoeken van ONVIF en een volgens de ONVIF Profile S specificatie correcte ONVIF plug-in ontwikkelen binnen de nieuwe architectuur.

Hiermee is de eerste stap gezet in het oplossen van het probleem. Als de plug-in architectuur daadwerkelijk zal worden opgenomen in Sense, zal het makkelijker zijn om nieuwe drivers toe te voegen aan Sense. Daarnaast biedt de ONVIF plug-in de mogelijkheid om camera's die ONVIF Compliant zijn via deze driver aan te sturen, waardoor het niet nodig is om een aparte plug-in te ontwikkelen.

14.4. Beroepstaken

In [\[Bijlage A. Afstudeerplan\]](#) zijn enkele beroepstaken gekozen voor de afstudeeropdracht.

H4 – Problemen analyseren: Het probleem is geanalyseerd tijdens het onderzoek [\[Bijlage C. ONVIF Research\]](#) en is beschreven in het onderzoekverslag en het Plan van Aanpak [\[Bijlage D. Plan van Aanpak\]](#).

J12 – Adviesproces uitvoeren: Op enkele punten in het afstudeertraject zijn belangrijke keuzes gemaakt in overleg met de begeleider en opdrachtgever. Een voorbeeld hiervan is de [\[Bijlage G. kdSOAP Review\]](#) waar het probleem is uitgelicht, de opties zijn geanalyseerd, een voorkeur voor een optie is uitgesproken (het advies), en hierover een besluit is genomen.

C10 – Ontwerpen van een systeemarchitectuur: Tijdens de initiële ontwerpfase en vervolgens gedurende elke sprint is het [\[Bijlage H. Design Document\]](#) tot stand gekomen. Hierin wordt de architectuur beschreven met behulp van modelleertalen en ontwerpkeuzes.

D17 – Testen van een softwaresysteem: Zoals in het [\[Bijlage K. Testplan\]](#) beschreven, zou er getest worden met behulp van de ONVIF Client Test Tool en de VDG Plug-in Test Tool. Helaas is de VDG Plug-in Test Tool niet afgerond tijdens de afstudeerperiode, waardoor de architectuur niet expliciet getest is [\[12.2\]](#). Wel is aan het eind van elke sprint en tijdens de testsprint met de ONVIF Client Test Tool de validiteit en voortgang getest [\[Bijlage M. Testrapport\]](#). Deze resultaten zijn tevens besproken aan het eind van elke ontwikkelsprint in het hoofdstuk 'Sprint resultaat'.

15. Literatuurlijst

- [1] Title: Scrum vs. Kanban vs. Scrumban: Planning, Estimation, and Performance Metrics
Author: Dalia Lasaite
Version: 02/05/2013
Link: <http://www.eylean.com/blog/2013/05/scrum-vs-kanban-vs-scrumban-planning-estimation-and-performance-metrics/>
Date of reference: 26/01/2016

- [2] Title: ONVIF Core Specification
Author: ONVIF
Version: V2.1 -06/2011
Link: <http://www.onvif.org/specs/core/ONVIF-Core-Spec-v210.pdf>
Date of reference: 08/02/2016

- [3] Title: ONVIF Network Interface Specification Set
Author: ONVIF
Version: V2.6.1
Link: <http://www.onvif.org/specs/DocMap-2.6.1.html>
Date of reference: 08/02/2016

- [4] Title: SOAP (Transport Methods)
Author: Wikipedia
Version: 04/02/2016
Link: https://en.wikipedia.org/wiki/SOAP#Transport_methods
Date of reference: 11/02/2016

- [5] Title: ONVIF Programmer's Guide
Author: ONVIF White Paper
Version: v1.0 - 05/2011
Link: http://www.onvif.org/Portals/0/documents/WhitePapers/ONVIF_WG-APG-Application_Programmer's_Guide.pdf

- [6] Title: ONVIF Profile S Client Test Specification
Author: ONVIF White Paper
Version: v16.0 - 01/2016
Link: http://www.onvif.org/Portals/0/documents/testspecs/v16_01/ONVIF_Profile_S_Client_Test_Specification_v16.01.pdf

16. Bijlagen

Inhoudsopgave

Bijlage A. Afstudeerplan
Bijlage B. Introductie VDG Sense
Bijlage C. ONVIF Research Document
Bijlage D. Plan van Aanpak
Bijlage E. Plug-in architectuur
Bijlage F. kdwsdl2cpp (voorbeeld)
Bijlage G. kdSOAP Review
Bijlage H. Design Document
Bijlage I. Compliance & Functions (normalized to requirements)
Bijlage J. Dome Research
Bijlage K. Testplan
Bijlage L. Redesign Classdiagram Sprint 4 (A3)
Bijlage M. Testrapport

Bijlage X. Feedback bespreking conceptverslag
Bijlage Y. Feedback Tussentijds Assessment
Bijlage Z. Evaluatie bedrijfsmentor