

Afstudeerplan

Informatie afstudeerder en gastbedrijf *(structuur niet wijzigen)*

Afstudeerblok: 2016-1.1 (start uiterlijk 8 februari 2016)

Startdatum uitvoering afstudeeropdracht: 25 januari 2016

Inleverdatum afstudeerdossier volgens jaarrooster: 3 juni 2016

Studentnummer: 11116552

Achternaam: dhr Kesteloo

Voorletters: T.V.

Roepnaam: Tim

Adres: Ottoburgstraat 20

Postcode: 2282EJ

Woonplaats: Rijswijk

Telefoonnummer: 0681025119

Mobiel nummer: 0681025119

Privé emailadres: kestelootim@gmail.com

Opleiding: Technische Informatica

Locatie: Delft

Variant: voltijd

Naam studieloopbaanbegeleider: A. Andrioli

Naam begeleidend examiner: J.J. van der Hoek

Naam tweede examiner: A. Andrioli

Naam bedrijf: VDG Security

Afdeling bedrijf: Research & Development

Bezoekadres bedrijf: Platinastraat 65

Postcode bezoekadres: 2718SZ

Postbusnummer:

Postcode postbusnummer:

Plaats: Zoetermeer

Telefoon bedrijf: +31793638111

Telefax bedrijf:

Internetsite bedrijf: www.vdgsecurity.com

Achternaam opdrachtgever: dhr van Deutekom

Voorletters opdrachtgever: D.

Titulatuur opdrachtgever: B.A.Sc

Functie opdrachtgever: Manager R&D

Doorkiesnummer opdrachtgever:

Email opdrachtgever: d.vandeutekom@vdgsecurity.com

Achternaam bedrijfsmentor: dhr Hermann

Voorletters bedrijfsmentor: R.

Titulatuur bedrijfsmentor:

Functie bedrijfsmentor: Lead Developer

Doorkiesnummer bedrijfsmentor:

Email bedrijfsmentor: r.hermann@vdgsecurity.com

NB: bedrijfsmentor mag dezelfde zijn als de opdrachtgever

Doorkiesnummer afstudeerder: -

Functie afstudeerder (deeltijd/duaal): Developer

Titel afstudeeropdracht:

Het implementeren van de Onvif camera plug-in voor VDG Security om zo de nieuwe pluginarchitectuur in gebruik te nemen.

Opdrachtomschrijving

1. Bedrijf

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

VDG Security is onderdeel van het technologiebedrijf TKH Group. De TKH Group bestaat uit een groot aantal bedrijven, met in totaal 5.337 werknemers. Zij bestaat uit drie hoofdgroepen: Telecom Solutions, Industrial Solution en Building Solutions. VDG Security valt onder de Building Solutions groep. Binnen deze groep wordt nauw samengewerkt met de zuster bedrijven die ook actief zijn in de security branch. Tijdens deze samenwerking wordt zoveel mogelijk de technologie van de zuster bedrijven gecombineerd met de video technologie van VDG Security.

2. Probleemstelling

Regelmatig krijgt de R&D afdeling van VDG Security de opdracht om een nieuwe camera te integreren in de VDG Sense software. Wanneer een nieuwe camera geïntegreerd dient te worden, dan bestaat de ontwikkeling van zo'n driver doorgaans uit de volgende functionaliteiten:

- Camera discovery: Het proces wat er voor nodig is om het IP- en MAC-adres van een camera te achterhalen in een netwerk.
- Video ontvangen: Opbouwen van een videoverbinding met de camera
- Audio ontvangen: Opbouwen van een audioverbinding met de camera
- Alarm inputs detecteren: Detecteren wanneer er input pinnen getriggerd worden
- Alarm outputs zetten: Output pinnen kunnen aansturen
- Pan, Tilt en Zoom (PTZ) commando's sturen: Voor het aansturen van de camera
- Camera specifieke events ontvangen: zoals gezichts- of kentekenherkenning, beweging detectie, etc

Elke camera fabrikant 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. Om dit proces te versnellen zijn de volgende twee initiatieven bedacht:

Pluginarchitectuur

Wanneer een nieuwe camera in de VDG Sense software is toegevoegd, dan moet in de huidige situatie een nieuwe versie van de software uitgebracht worden voordat klanten hiervan gebruik kunnen maken. VDG Security ziet liever dat de verschillende camera drivers in een aparte plugin (DLL) gebouwd kunnen worden met de volgende voordelen:

1. Indien een nieuwe camera-integratie is gedaan of een wijziging heeft plaatsgevonden, dan hoeft alleen de plugin uitgebracht of geüpdatet te worden.
2. Indien de plugin een duidelijk interface heeft met de VDG Sense software, dan kunnen derde partijen zelf een plugin ontwikkelen die in de VDG Sense software ingeladen kan worden. Dit zal dan als een Software Development Kit (SDK) aangeleverd kunnen worden.

VDG Security heeft een plan en een ontwerp klaar liggen om zo'n dergelijke pluginarchitectuur met bijhorende SDK te ontwikkelen, maar is nog niet aan de ontwikkeling van een plugin begonnen. Door middel van het implementeren van een plugin in de nieuwe pluginarchitectuur kan proefondervindelijk duidelijk worden of de pluginarchitectuur en SDK voldoende generiek zijn.

Onvif

Ten tweede is VDG Security erg geïnteresseerd in Onvif. Dit is een protocol/standaard die door een aantal vooraanstaande camerafabrikanten is ontwikkeld, waarmee het voor integrators makkelijker moet worden om met minimale tijd investering meerdere type camera's te ondersteunen. De gedachte hier achter is dat als een integrator het Onvif protocol in hun VMS heeft geïmplementeerd, automatisch alle camera's ondersteunt die ook het Onvif protocol ondersteunen. Middels dit protocol wordt bijvoorbeeld het opbouwen van een videoverbinding op een uniforme manier afgehandeld en maakt de fabrikant en type van de camera niet meer uit.

3. Doelstelling van de afstudeeropdracht

De doelstelling van de afstudeeropdracht is om een Onvif driver te implementeren als plugin in de nieuwe pluginarchitectuur. Hoewel VDG verantwoordelijk is voor het implementeren van de SDK architectuur (C-interface waar de DLL's (C++ modules) mee zullen communiceren), zal veel overleg noodzakelijk zijn zodat de implementatie van de plugin zo optimaal mogelijk zal kunnen plaatsvinden.

Alhoewel VDG Security al een aantal Onvif camera's ondersteunt, bestaat er twijfel of deze implementatie verschillende Onvif versies aankan en of alle features (op de juiste manier) ondersteund worden. Daarmee is ook de doelstelling van de afstudeeropdracht om het Onvif protocol door middel van een onderzoek goed te onderzoeken en vervolgens zoveel mogelijk features in de nieuwe plugin te implementeren met de daarbij horende tests.

Omdat dit de eerste plugin zal zijn in de nieuwe pluginarchitectuur zou dit ertoe kunnen leiden dat blijkt dat pluginarchitectuur niet voldoende generiek is opgezet en dus niet alle functionaliteiten kan ondersteunen. In dit geval is het mogelijk om aanbeveling te doen tot het aanpassen of uitbreiden van de pluginarchitectuur.

4. Resultaat

Het traject wordt binnen VDG beschouwd als een testcase voor de nieuwe pluginarchitectuur. Het einde van dit traject zal dus uitwijzen of de door VDG geïmplementeerde pluginarchitectuur goed genoeg werkt om deze aan haar klanten te kunnen aanbieden.

Als bewijs hiervan zal het aan het eind van de stageperiode mogelijk zijn om in de testomgeving (simplistische versie van de Core / testapplicatie om core-functionaliteiten te imiteren) een Onvif camera aan te sturen en de beoogde features te kunnen benutten.

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

Onderzoek

Om juiste aanbevelingen te kunnen doen aan VDG dienen de volgende aspecten nog te worden uitgezocht:

- VDG heeft al iets met Onvif gedaan. Wat houdt dit in en wat zijn de verdere wensen van VDG?
- De pluginarchitectuur is in een relatief ver gevorderd stadium; Zo is de designfase zo goed als klaar en zou deze gebouwd kunnen worden. Wat zijn de huidige ideeën en waar denkt men problemen te kunnen tegenkomen?

Deze taken zullen in het begin van het afstudeertraject (+- 4 weken) duidelijk moeten worden om een juiste inschatting te kunnen maken voor de planning. Naar aanleiding van deze onderzoeken zullen de belangrijkste wensen en requirements duidelijk worden, waarna het **Plan van Aanpak** af te ronden is.

Design

Naar aanleiding van het voorgaande onderzoek kan een design worden gemaakt voor de verdere ontwikkeling van de plug-in. Hierbij moet gekeken worden welke Onvif versies nu geïmplementeerd zullen worden en welke in de toekomst. Tijdens de designfase zal hier in zekere mate rekening mee moeten worden gehouden.

Dit zal leiden tot één of twee designdocumenten, afhankelijk van de grote en complexiteit van de plug-in, namelijk het **Functional Design** en het **Technical Design**.

Aangezien aanpassingen aan het design tijdens de constructiefase verwacht worden, zullen deze documenten niet definitief zijn tot het eindproduct naar wens is.

Constructie

Hierin zal de daadwerkelijke plug-in worden geschreven. Nadat de meest belangrijke requirements en technische keuzes vastgelegd zijn, is het mogelijk om hier een **proof-of-concept** voor te maken. Wanneer deze geslaagd is, zal de code uitgebreid worden en **gereviewed**, zodat niet alleen de functionaliteit van de plug-in, maar ook de code aansluit bij de wensen van VDG.

Testen

Tijdens het afstudeertraject zal er geen vaste testperiode worden aangewezen. Juist zal ernaar worden gestreefd dat dit een veelvoorkomend aspect is tijdens de constructiefase. Op deze manier is het mogelijk om op korte termijn de verschillen tussen de architectuur en plug-in te **bespreken** en zo te verbeteren.

Overdracht

Om zeker te zijn dat VDG gebruik kan gaan maken van alle opgeleverde eindproducten zal er tijd gereserveerd worden om deze bestanden over te dragen. Dit houdt in dat alle producten nogmaals beschreven worden en doorgenomen, zodat deze in de toekomst gebruikt kunnen worden.

6. Op te leveren (tussen)producten

- Onderzoek Onvif, features en implementatie
- Plan van Aanpak
- Functional Design
- Technical Design
- Testplan & Testrapport
- Productdocumentatie
- Onvif plug-in

7. Te demonstreren competenties en wijze waarop

H4 – Problemen analyseren

In het kader van de onderzoeksvraag dient tijdens de afstudeerperiode Onvif te worden geanalyseerd. Het is nog onduidelijk hoe Onvif binnen het bestaande VMS van VDG past en hoe dit toegepast kan worden op het bestaande ontwerp van de pluginarchitectuur. Deze beschrijving van het probleemdomein zal terug te zien zijn in het Onvif **onderzoeksrapport**.

C10 – Ontwerpen van een systeemarchitectuur

Nadat de analyse voorbij is, dient de Onvif plugin ontworpen te worden. Een voordeel van deze stap uitgebreid uitwerken zou kunnen zijn dat dit kan dienen als werkmethode voor externe bedrijven wanneer het implementeren van plugins aan derden wordt overgelaten. Wanneer dit proces duidelijk is uitgeschreven, kan het gehele traject nauwkeurig nagelezen worden en gereviewd worden. Dit ontwerp zal resulteren in het **Functional Design** en **Technical Design**.

D17 – Testen van een softwaresysteem

Om zeker ervan te zijn dat de plugin functioneert zoals verwacht, dient deze getest te worden. Dit betekent dat aan het einde van het traject te controleren is of alle eisen die vooraf aan plugin gesteld zijn meetbaar te verifiëren zijn. Een groot onderdeel hiervan zal de door VDG geleverde testapplicatie/stripped-down Core omvatten, maar hier dient alsnog een **Testplan** voor te worden geschreven. De resultaten van de uitgevoerde tests zullen terug te lezen zijn in het **Testrapport**.

J12 – Adviesproces uitvoeren

Indien tijdens het ontwikkeltraject blijkt dat er een verandering of toevoeging aan de pluginarchitectuur moet plaatsvinden, dient dit onderbouwd te zijn naar aanleiding van de wensen, requirements en het voordeel van deze functionaliteit. Deze aanbevelingen zullen terug te lezen zijn in de **Productdocumentatie**.

Introductie VDG Sense

Sense is verantwoordelijk voor het opzetten van cameraverbindingen, het configureren van de camera's via de applicatie, het ophalen en opslaan van video- en audiodata, en het analyseren van de beelden. Aan deze beeldanalyse, ook wel de Video Content Analysis (VCA) genoemd, kunnen regels worden gegeven. Wanneer een beeld voldoet aan de gegeven regels (zoals het overschrijden van een lijn door een object of beweging detecteren) wordt een event afgegeven. Deze events kunnen vervolgens gekoppeld worden aan een actie binnen Sense, zoals een email sturen of het camerabeeld centreren in de applicatie.

Verbinden

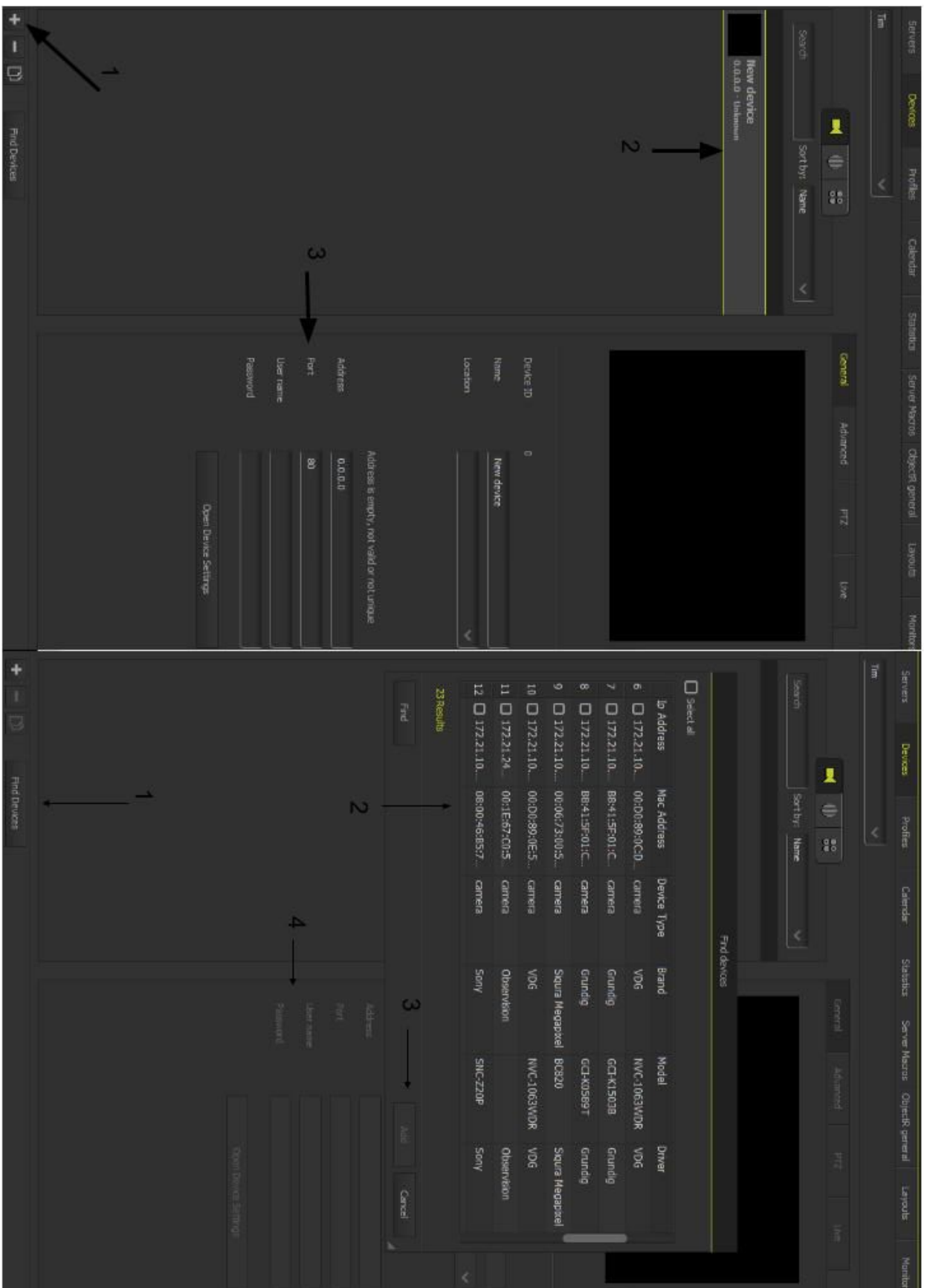
Alle camera's waar VDG gebruik van maakt zijn IP camera's. Dit betekent dat zij via een netwerkverbinding worden aangestuurd. Het vinden van een nieuwe camera kan in Sense op twee manieren: handmatig of automatisch.

Bij het handmatig selecteren van een camera wordt het "New Device" scherm geopend. Hier dient het IP-adres van de camera te worden ingevuld, het poortnummer, en de login gegevens. (zie [figuur 1](#); Linkerzijde; 1: Klik voor nieuw device; 2: Selecteer het nieuwe device; 3: vul de gegevens (IP, poort, login) in).

Voor het automatisch verbinden met een camera wordt de knop "Find Devices" aangeklikt. Hierbij wordt het netwerk afgezocht voor IP-camera's. Deze worden in een lijst gezet en kunnen vervolgens toegevoegd worden aan de eigen lijst. Hierna moeten alleen de login gegevens nog ingevuld worden. (zie [figuur 1](#); Rechterzijde; 1: Find Devices; 2: Selecteer een Device en 3: voeg deze toe; 4: Het IP-adres en poort is ingevuld. Vul logingegevens in.) Vervolgens wordt het camerabeeld live opgehaald en opgeslagen.

Configureren

Elke camera heeft verschillende features geïmplementeerd. Deze zijn uitgewerkt in de drivers geschreven door VDG en kunnen worden geselecteerd in Sense. Zo is het bijvoorbeeld mogelijk om 'Dual Streaming' (twee verschillende kwaliteiten beeld uit de camera) of Audio aan te zetten. Via Sense zijn hier maar enkele muisklikken voor nodig. In het geval dat PTZ (pan, tilt, zoom) aanwezig is, kan de camera ook bestuurd worden.



Figuur 1: Handmatig toevoegen van camera vs automatisch toevoegen

VCA

Vervolgens is het mogelijk om het beeld te analyseren. VDG heeft verschillende modules die hiervoor gebruikt kunnen worden, zoals nummerbordherkenning of object tracking. Deze algoritmes kunnen een event triggeren, die vervolgens gekoppeld kan worden aan een macro. Deze macro beschrijft wat er moet gebeuren, afhankelijk van de detectie die gemeten is. Zo kan er bijvoorbeeld een alarm afgezet worden, of het camerabeeld oplichten in Sense.

VCA wordt zelden door VDG zelf ontwikkeld. Het komt vaak voor dat de algoritmes van een extern bedrijf hiervoor gebruikt worden.

Videowall

Al deze functionaliteiten komen samen in de videowall. Dit scherm biedt de mogelijkheid aan de gebruiker om alle geconfigureerde camera's op één of meerdere schermen weer te geven. Hier kunnen tevens extra knoppen of besturingspanelen aan worden toegevoegd, zodat de 'operator' volledige controle heeft over de camera's of andere systemen die verbonden zijn met Sense. Hierbij kan gedacht worden aan een deur die via een knop kan worden geopend of gesloten.

Introduction

This document describes the research that has been done on ONVIF by Tim Kesteloo. The focus for this research was to get a general understanding of what ONVIF is and how it works. Secondly, we we're interested in what the goal of reaching Profile S Compliance with our application would mean: What are the minimum requirements and the maximum requirements. This in turn leads to the question which of these requirements are of interest to VDG.

When this information has been obtained, we can specify which actions will be taken.

[Introduction](#)

[General](#)

[VDG Requirements](#)

[Profile S](#)

[ONVIF Client Test Tool](#)

[Conformance Process - Client](#)

[Concerns](#)

[Profile S Specification](#)

[Profile Mandatory Features](#)

[Profile Conditional Features](#)

[Device Mandatory Features](#)

[Reference](#)

General

ONVIF is a specification that describes a protocol for the communication between “Devices”, mainly (but not limited to) camera’s, and “Clients”, mainly Video Management Service (VMS) like VDG Sense. Though ONVIF is not a standard (yet), the protocol does push for a more standardized approach to communication between Devices and Clients. [\[3\]](#)

It does this by describing multiple profiles, which each contains a set of functionalities that it implements.

Profile S is responsible for IP based video and audio streaming, and it’s configuration, like multicast. It also contains features like Pan-Tilt-Zoom (PTZ) control and relay output.

Profile G is for IP-based video storage systems. This includes edge-recording and storage, and retrieving this historic video-, audio- and metadata stream.

Profile C does not control camera systems, but is setup for IP-based access control systems. This includes door access control and event & alarm management.

Profile Q’s goal is to make the interoperability between ONVIF devices as easy and Zero-configuration as possible. This is described as easy set-up and after set-up security, data-integrity & privacy and discovery, configuration & control of conformant devices.

Profile A is an improvement to the Access Control. This includes enhancement in Physical Access Control Systems (PACS) and retrieving & configuring PACS related entities like access rules, credentials and schedules. This is mainly used in unison with Profile C. [\[4\]](#) [\[5\]](#)

Because of the separation of profiles, it’s not possible to be ONVIF Compliant. Instead, a Device or Client can become compliant to an ONVIF profile, e.g. Profile S Compliance. If a Device and Client are both compliant to this profile, it can be stated that all implemented functionality can be used between the two systems. [\[2\]](#)

However, a profile specification consists of Mandatory, Conditional and Optional features. This means that the developer implementing the ONVIF profile on the Device or Client has a choice whether to implement a feature if it’s not mandatory, e.g. a camera that can’t move won’t have to implement Pan, Tilt, Zoom (PTZ) features. Because of this, it’s necessary for a Client to check which features are implemented by the Device. This is done by using the “GetCapabilities” function. Whenever a Device implements a feature, he must set the appropriate value which will be returned when this function is called. [\[6\]](#)

In order to test whether a Device or Client is indeed compliant to the profile, ONVIF has developed a Device- and Client Test tool. These will test some simple usecases and check if correct messages are sent at the right time, if the messages are formatted correctly and if it contains data. The actual data inside isn’t tested, which means that the Testtool won’t determine if the implementation actually works. [\[5\]](#) [\[7\]](#)

ONVIF works based on Web Service technology. This was mainly done because of it’s cross-platform nature and the ease of implementation. Because of this, there are some components that must be used. These are XML, which is a mechanism that is used to represent and exchange data, SOAP, for web-communication and WSDL, which is a description of the services offered. [\[5\]](#)

VDG Requirements

VDG's VMS can be described as an ONVIF Client. This means that it will support an ONVIF implementation which supports the communication with ONVIF devices.

VDG's primary goal is to implement Profile S. In doing so, it is expected that each Profile S compliant camera can be used in the VMS without having to write extra implementation.

Profile S

We've listed all the features that are present in Profile S. VDG has made a quick implementation of ONVIF, so we want to know how far they got. Next, we're interested in which conditional features VDG supports or want to support in the future (see figure XXX).

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

Figure XXX: Featurelist for Profile S (JIRA)

On this basis, we've made a list of functions that'll have to be implemented for each feature. Whenever it's chosen to implement a conditional feature, we get a new list of functions that are either mandatory or conditional to the implementation of that feature. Because of this, we looked into VDG's interest in the conditional functions for the chosen conditional features. This leads to the following list of functions that are desired:

	Mandatory	Conditional	Device Mandatory
Getter	8	12	16
Setter (Add and Remove commands included)	4	15	14
Action (neither a real get nor set)	4	11	6

Figure XXX: Legend for Functionlist (including total # of functions)

In the "Requirement" tab, we differentiate Mandatory and Optional functions. Whenever it is red, it means that it is mandatory, or VDG wants this function to be implemented. Gray blocks are sub-services that are defined as "won't be implemented".

Profile (S) Mandatory Functions	Service (WSDL)	Requirement	Sub-service
GetProfiles	Media	Mandatory	Video Streaming
GetStreamUri	Media	Mandatory	Video Streaming
Media Streaming using RTSP	Streaming	Mandatory	Video Streaming
Media Streaming using RTSP - JPEG RTP header extension	Streaming (MJPEG)	Mandatory	Video Streaming
GetVideoEncoderConfiguration	Media	Mandatory	Video Encoder Configuration
GetVideoEncoderConfigurations	Media	Mandatory	Video Encoder Configuration
AddVideoEncoderConfiguration	Media	Optional	Video Encoder Configuration
RemoveVideoEncoderConfiguration	Media	Optional	Video Encoder Configuration
SetVideoEncoderConfiguration	Media	Mandatory	Video Encoder Configuration
GetCompatibleVideoEncoderConfigurations	Media	Optional	Video Encoder Configuration
GetVideoEncoderConfigurationOptions	Media	Mandatory	Video Encoder Configuration

GetGuaranteedNumberOfVideoEncoderInstances	Media	Optional	Video Encoder Configuration
WS-UsernameToken Authentication	Core	Mandatory	User Authentication
HTTP Digest	Core	Mandatory	User Authentication
GetCapabilities	Device	Mandatory	Capabilities
GetWsdUrl	Device	Optional	Capabilities
Profile (S) Conditional Functions	Service	Requirement	Sub-chapter
SetSynchronizationPoint	Media	Optional	Video Streaming (MPEG4)
SetSynchronizationPoint	Media	Optional	Video Streaming (H264)
AddPTZConfiguration	Media	Mandatory	PTZ
RemovePTZConfiguration	Media	Optional	PTZ
GetNodes	PTZ	Mandatory	PTZ
GetNode	PTZ	Mandatory	PTZ
GetConfigurations	PTZ	Mandatory	PTZ
GetConfiguration	PTZ	Optional	PTZ
GetConfigurationOptions	PTZ	Optional	PTZ
SetConfiguration	PTZ	Optional	PTZ
ContinuousMove	PTZ	Mandatory	PTZ
Stop	PTZ	Mandatory	PTZ
GetStatus	PTZ	Optional	PTZ
AbsoluteMove	PTZ	Mandatory	PTZ (Absolute Positioning)
RelativeMove	PTZ	Mandatory	PTZ (Relative Positioning)
GetPresets	PTZ	Mandatory	PTZ (Presets)
GotoPresets	PTZ	Mandatory	PTZ (Presets)
SetPresets	PTZ	Optional	PTZ (Presets)
RemovePresets	PTZ	Optional	PTZ (Presets)
GotoHomePosition	PTZ	Mandatory	PTZ (Home Position)
SetHomePosition	PTZ	Optional	PTZ (Home Position)

SendAuxiliaryCommand	PTZ	Mandatory	PTZ (Auxiliary Command)
AddAudioSourceConfiguration	Media	Mandatory	Audio Streaming
GetAudioSourceConfiguration	Media	Optional	Audio Streaming
GetAudioSourceConfigurations	Media	Optional	Audio Streaming
	Streaming	Mandatory	Audio Streaming (G726)
	Streaming	Mandatory	Audio Streaming (AAC)
StartMulticastStreaming	Media	Mandatory	Video Streaming (Multicast)
StopMulticastStreaming	Media	Mandatory	Video Streaming (Multicast)
GetRelayOutputs	Device	Mandatory	Relay Outputs
SetRelayOutputSettings	Device	Mandatory	Relay Outputs
SetRelayOutputState	Device	Mandatory	Relay Outputs
GetNTP	Device	Mandatory	NTP
SetNTP	Device	Mandatory	NTP
GetDynamicDNS	Device	Mandatory	Dynamic DNS
SetDynamicDNS	Device	Mandatory	Dynamic DNS
GetZeroConfiguration	Device	Mandatory	Zero Configuration
SetZeroConfiguration	Device	Mandatory	Zero Configuration
GetIPAddressFilter	Device	Mandatory	IP Address Filtering
SetIPAddressFilter	Device	Mandatory	IP Address Filtering
AddIPAddressFilter	Device	Mandatory	IP Address Filtering
RemoveIPAddressFilter	Device	Mandatory	IP Address Filtering
Device Mandatory Functions	Service	Requirement	Sub-chapter
WS-Discovery	Core	Mandatory	Discovery
GetDiscoveryMode	Device	Optional	Discovery
SetDiscoveryMode	Device	Optional	Discovery
GetScopes	Device	Optional	Discovery
SetScopes	Device	Optional	Discovery

AddScopes	Device	Optional	Discovery
RemoveScopes	Device	Optional	Discovery
GetNetworkInterfaces	Device	Mandatory	Network Configuration
SetNetworkInterfaces	Device	Mandatory	Network Configuration
GetNetworkDefaultGateway	Device	Mandatory	Network Configuration
SetNetworkDefaultGateway	Device	Mandatory	Network Configuration
GetHostname	Device	Optional	Network Configuration
SetHostname	Device	Optional	Network Configuration
GetDNS	Device	Optional	Network Configuration
SetDNS	Device	Optional	Network Configuration
GetNetworkProtocols	Device	Optional	Network Configuration
SetNetworkProtocols	Device	Optional	Network Configuration
GetDeviceInformation	Device	Mandatory	System
GetSystemDateAndTime	Device	Optional	System
SetSystemDateAndTime	Device	Optional	System
SetSystemFactoryDefault	Device	Optional	System
Reboot	Device	Optional	System
GetUsers	Device	Mandatory	User handling
CreateUsers	Device	Mandatory	User handling
SetUser	Device	Mandatory	User handling
DeleteUsers	Device	Mandatory	User handling
Notify	Event	Mandatory*	Event handling
Subscribe	Event		Event handling
Renew	Event		Event handling
Unsubscribe	Event		Event handling
SetSynchronizationPoint	Event	Mandatory*	Event handling
CreatePullPointSubscription	Event		Event handling
PullMessage	Event		Event handling

GetEventProperties	Event	Optional	Event handling
TopicFilter	Event	Optional	Event handling
MessageContentFilter	Event	Optional	Event handling
GetProfiles	Media	Mandatory	Media Profile Configuration
GetProfile	Media	Mandatory	Media Profile Configuration
CreateProfile	Media	Mandatory	Media Profile Configuration
DeleteProfile	Media	Optional	Media Profile Configuration
GetVideoSources	Media	Optional	Media Source Configuration
GetVideoSourceConfiguration	Media	Mandatory	Media Source Configuration
GetVideoSourceConfigurations	Media	Mandatory	Media Source Configuration
AddVideoSourceConfigurations	Media	Mandatory	Media Source Configuration
RemoveVideoSourceConfiguration	Media	Optional	Media Source Configuration
SetVideoSourceConfiguration	Media	Mandatory	Media Source Configuration
GetCompatibleVideoSourceConfigurations	Media	Mandatory	Media Source Configuration
GetVideoSourceConfigurationOptions	Media	Mandatory	Media Source Configuration
GetMetadataConfiguration	Media	Mandatory	Metadata configuration
GetMetadataConfigurations	Media	Mandatory	Metadata configuration
AddMetadataConfiguration	Media	Optional	Metadata configuration
RemoveMetadataConfiguration	Media	Optional	Metadata configuration
SetMetadataConfiguration	Media	Mandatory	Metadata configuration
GetCompatibleMetadataConfigurations	Media	Optional	Metadata configuration
GetMetadataConfigurationOptions	Media	Mandatory	Metadata configuration

Figure XXX: Spreadsheet Functionlist

In the Plan of Approach, a prioritylist was proposed for this exact situation. The chosen priority was as followed:

1. Mandatory features' mandatory functions & VDG specifcly requested Mandatory features' conditional functions
2. Optional features that VDG already supported in her previous versions' mandatory functions
3. #2 Optional features' VDG specifcly requested optional functions
4. Optional features that VDG wants to support in the future' mandatory functions.
5. Optional features that VDG wants to support in the future' optional functions.

This leads to the following **sub-service function-prioritylist**:

1. Video Streaming, Video Encoder Configuration, User Authentication, Capabilities, System-GetSystemDateAndTime (important method often used as "Hello")
 $4 + 4 + 2 + 1 + 1 = 12$ functions
2. PTZ (excluding Auxiliary & AbsoluteMove), Audio Streaming, Multicast, Relay Outputs, Discovery
 $12 + 3 + 2 + 4 + 1 = 22$ functions
3. Video Streaming (MPEG & H264)
 $1 + 1 = 2$ functions
4. PTZ-AbsoluteMove, PTZ-RemovePresets, NTP, IP address filtering, System, Event Handling, Media Profile Configuration, Media Source Configuration, Metadata Configuration
 $1 + 1 + 2 + 4 + 1 + 4 + 3 + 6 + 4 = 26$
5. None defined

ONVIF Client Test Tool

This program was obtained in the second week. We started testing it on the current Sense implementation of ONVIF. An example result of this is shown in figure XXX:

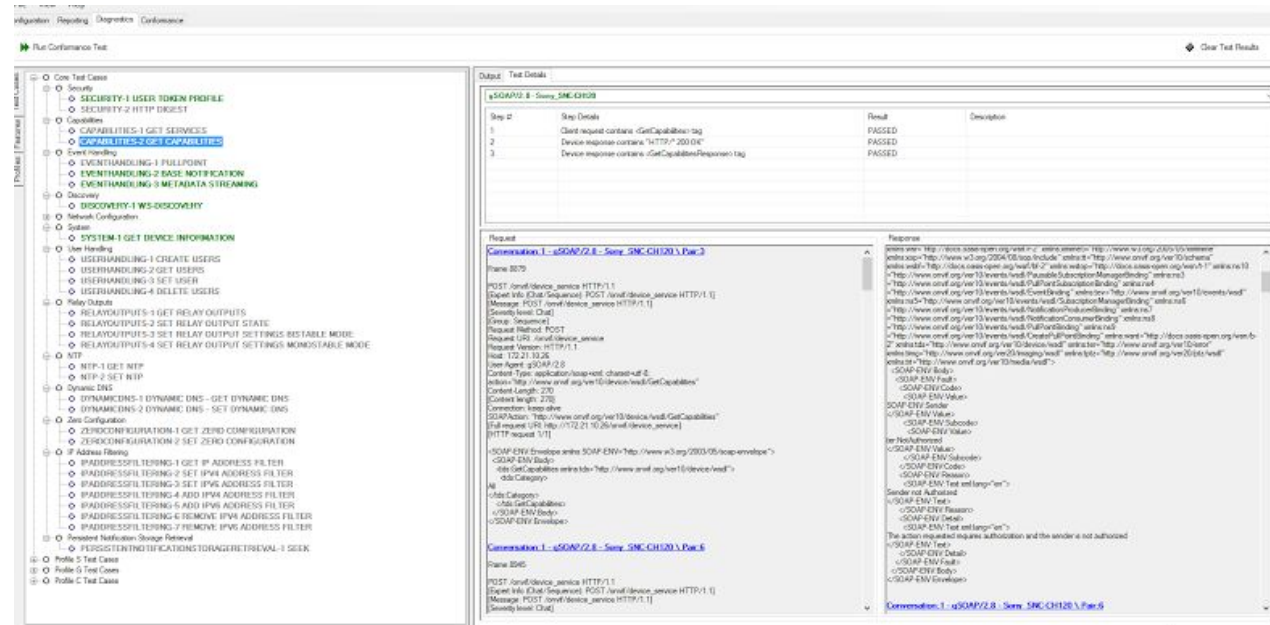


Figure XXX: ONVIF Client Test Tool example

There are two main flaws in this design.

1. The Test Tool does not test every function. We expected to be able to ask for profile S compliance, and that it would tell us whether or not a feature implementation was missing. This is not the case. If a function is not performed in the log file, it is not tested.
2. The program doesn't check for a correct usage. For example, the PTZ could not work on a camera even though it passed the conformance test. This could happen because the correct tags are send, but the values inside are not correct (which aren't checked by the tool).

This means that the following has to be done:

A testprogram must be made that calls for all functions that we (want) implemented.

This will ensure that all actions that need to be tested are in the logfile. When this is succesfull, it could be that a camera does nothing. Because of this we will have to **manually test** if the functions actually lead to the expected result.

Note: The tool tests the CLIENT (our implementation). This means that if we have a set of camera's, and only 1 will respond correctly, the CLIENT has succeeded it's task. This is because we expect that not every camera will have implemented a feature (correctly) or can malfunction.

Conformance Process - Client

[8] In order to be conformant as a client to a profile, the following steps must have been taken (in our case only applicable for the “Client” path. The “Device” path doesn’t apply to us): Figure 2.

This shows that the conformance process requires 2 documents:

1. Generated DoC (Declaration of Compliance) by the Client Testtool
2. [ONVIF Interface Guide](#) (in XML format, conforming to DocBook v5.x standard)

The 3rd document, Features List, is only needed for ONVIF Devices, thus not applicable to us as Client implementation.

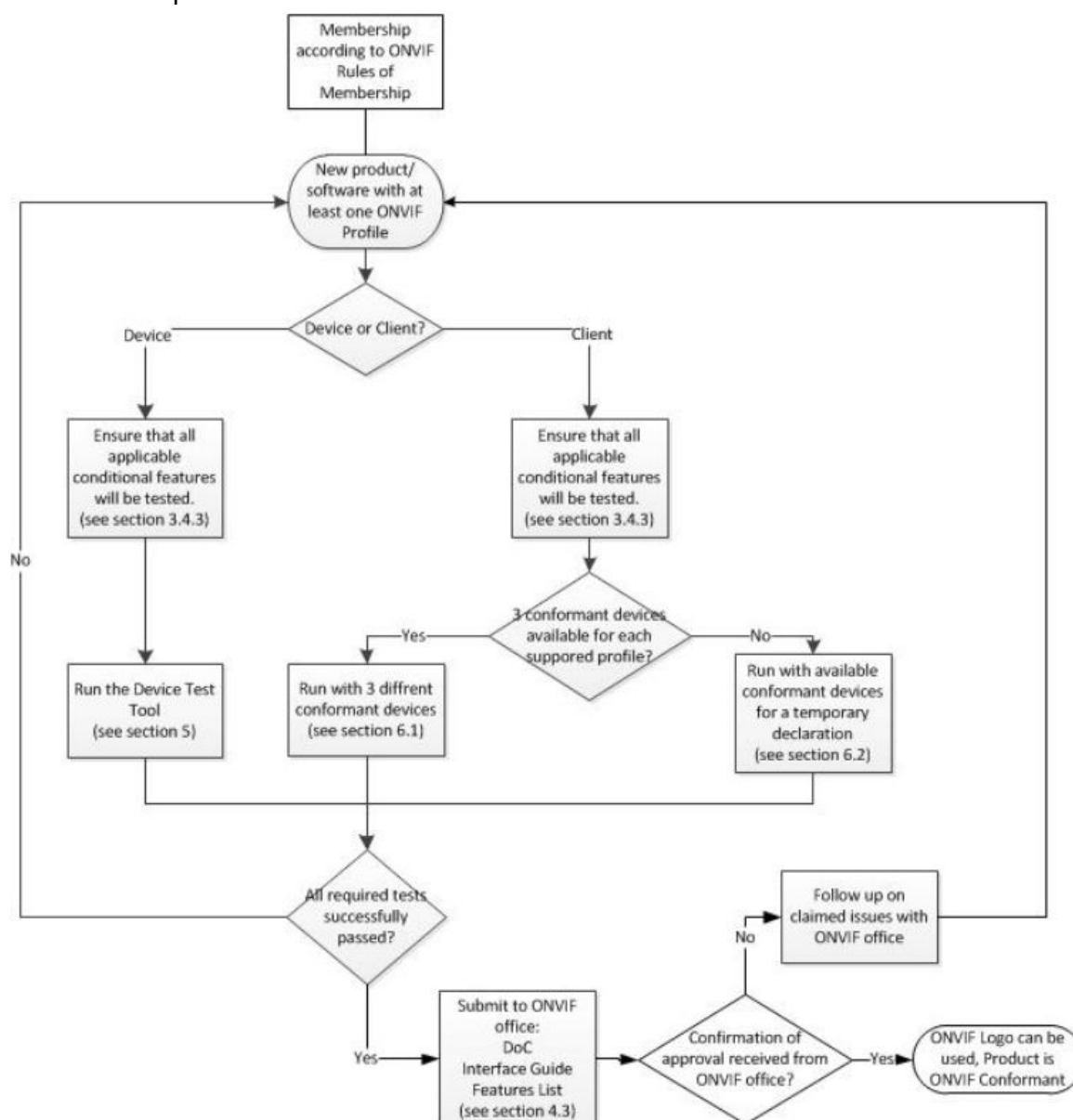


Figure XXX: ONVIF Conformance Process Flow chart (from the [ONVIF Conformance Process Specification](#))

Concerns

What is the impact of the different versions? Does it require extra work to support all versions?

As described in the [ONVIF Technical FAQ \[7\]](#), there is indeed a big difference between the versions. With a new version, the XML is often changed to remove deprecated information or add new parameters. This difference makes the version an entire new set of commands. Each of these versions is placed in a new namespace. For example the difference between a version 1.0 and 2.0 GetOptions in the Imaging Service (timg) will be timg10:GetOptions and timg:20 GetOptions respectively. Each function has a different implementation.

However, ONVIF is always focussed on backwards compatibility.

There is a Media and Media2 wsdl. What are the differences and which should we use?

Media2 and DeviceIO are new additions since Specification Set 2.6.1, as described in the SpecSet 2.6.1 release notes. [\[10\]](#)

This states that Media2 is a preparation for implementing new Video and Audio Codecs, mainly H265. It also states that it's backwards compatible with the Media Service, which would suggest that camera's that implemented the original Media Service can still be controlled when Media 2 is implemented on the Client.

Profile S Specification

[\[6\]](#)

Introduction

This is a simplified version of the [Profile S Specification](#) document by ONVIF. We've removed all references to the Device features and focussed on the Client features, both mandatory and optional (there are no conditional defined in the document).

Chapter "Profile Mandatory Features" describes the features that must be implemented to be conformant to ONVIF.

"Profile Conditional Features" describes the features that shall be implemented if the client supports the feature. The requirements represent the minimum required to be implemented for conformance.

"Device Mandatory Features" describes features that are mandatory for devices (not important in our case) and conditional for the Client in order to be conformant.

References

Specifications:

- Core > v1.02
- Media Service
- PTZ Service
- Streaming

Contents

[Profile S Specification](#)

[Introduction](#)

[References](#)

[Contents](#)

[Profile Mandatory Features](#)

[Video Streaming](#)

[Video Encoder Configuration](#)

[User Authentication](#)

[Capabilities](#)

[Profile Conditional Features](#)

[Video Streaming - MPEG4 \(sub Video Streaming\)](#)

[Video Streaming - H264 \(sub Video Streaming\)](#)

[PTZ](#)

[PTZ - Absolute Positioning \(sub PTZ\)](#)

[PTZ - Absolute Positioning \(sub PTZ\)](#)

[PTZ - Presets \(sub PTZ\)](#)

[PTZ - Home Position \(sub PTZ\)](#)

[PTZ - Auxiliary Command \(sub PTZ\)](#)

[Audio Streaming](#)

[Audio Streaming - G726 \(sub Audio Streaming\)](#)

[Audio Streaming - AAC \(sub Audio Streaming\)](#)

[Multicast Streaming \(sub Video Streaming\)](#)

[Relay Outputs](#)

[NTP](#)

[Dynamic DNS](#)

[Zero Configuration](#)

[IP Address Filtering](#)

[Device Mandatory Features](#)

[Discovery](#)

[Network Configuration](#)

[System](#)

[User handling](#)

[Event handling](#)

[Media profile configuration](#)

[Video source configuration](#)

[Metadata configuration](#)

Profile Mandatory Features

List of features that are guaranteed to be supported by the client.

Video Streaming

- Client shall be able to list media profiles from the device using the GetProfiles operation.
- Client shall be able to get the stream URI for the selected profile using the GetStreamURI operation.
- Client be able to stream video on RTP/UDP or RTP/RTSP/HTTP/TCP using the selected profile over RTSP.
- Client shall be able to receive a stream and decode MJPEG video using the selected profile over RTSP.
- Client shall be able to understand RTSP - JPEG RTP header extension.

Function	Service	Requirement
GetProfiles	Media	Mandatory
GetStreamUri	Media	Mandatory
Media Streaming using RTSP	Streaming	Mandatory
Media Streaming using RTSP - JPEG RTP header extension	Streaming (MJPEG)	Mandatory

Video Encoder Configuration

- Client shall be able to list available video encoder configurations using GetVideoEncoderConfigurations operation.
- Client shall be able to modify video encoder configurations using the GetVideoEncoderConfiguration, GetVideoEncoderConfigurationOptions and SetVideoEncoderConfiguration operations.
- Client may be able to add video encoder configurations to media profiles using the GetCompatibleVideoEncoderConfigurations and AddVideoEncoderConfiguration operations.
- Client may be able to query maximum number of concurrent stream using the GetGuaranteedNumberOfVideoEncoderInstances operation.
- Client may be able to remove video encoder configurations from media profiles using the RemoveVideoEncoderConfiguration operation.

Function	Service	Requirement
GetVideoEncoderConfiguration	Media	Mandatory
GetVideoEncoderConfigurations	Media	Mandatory
AddVideoEncoderConfiguration	Media	Optional
RemoveVideoEncoderConfiguration	Media	Optional
SetVideoEncoderConfiguration	Media	Mandatory
GetCompatibleVideoEncoderConfigurations	Media	Optional
GetVideoEncoderConfigurationOptions	Media	Mandatory
GetGuaranteedNumberOfVideoEncoderInstances	Media	Optional

User Authentication

- Client shall implement WS-UsernameToken according to WS-security as covered by the core specification.
- Client shall implement HTTP Digest as covered by the core specification.

Function	Service	Requirement
WS-UsernameToken Authentication	Core	Mandatory
HTTP Digest	Core	Mandatory

Capabilities

- Client shall be able to get Capabilities of a device using GetCapabilities.

Function	Service	Requirement
GetCapabilities	Device	Mandatory
GetWsdUrl	Device	Optional

Profile Conditional Features

The Profile Conditional Features section list the features that shall be implemented if the client supports the feature. For instance a client that has PTZ support shall implement the ONVIF PTZ interface. The requirements represents the minimum required to be implemented for conformance.

Video Streaming - MPEG4 (sub Video Streaming)

- Client shall be able to receive a stream and decode MPEG4 video using the selected profile over RTSP.

Function	Service	Requirement
SetSynchronyzationPoint	Media	Optional

Video Streaming - H264 (sub Video Streaming)

- Client shall be able to receive a stream and decode H264 video using the selected profile over RTSP.

Function	Service	Requirement
SetSynchronyzationPoint	Media	Optional

PTZ

- Client shall be able to read PTZ capabilities from a PTZ node using GetNodes or GetNode operations.
- Client shall be able to add a PTZ configuration to a profile using GetConfigurations and AddPTZConfiguration.
- Client shall be able to move a PTZ device using the ContinuousMove operation.
- Client shall be able to stop a PTZ device using the Stop operation.
- Client may be able to query a device PTZ status using the GetStatus operation.
- Client may be able to modify a PTZ configuration using the GetConfiguration, GetConfigurationOptions and SetConfiguration operations.

Function	Service	Requirement
AddPTZConfiguration	Media	Mandatory
RemovePTZConfiguration	Media	Optional
GetNodes	PTZ	Mandatory
GetNode	PTZ	Mandatory
GetConfigurations	PTZ	Mandatory
GetConfiguration	PTZ	Optional
GetConfigurationOptions	PTZ	Optional
SetConfiguration	PTZ	Optional
ContinuousMove	PTZ	Mandatory
Stop	PTZ	Mandatory
GetStatus	PTZ	Optional

PTZ - Absolute Positioning (sub PTZ)

- Client shall be able to move a PTZ device using the AbsoluteMove operation.

Function	Service	Requirement
AbsoluteMove	PTZ	Mandatory

PTZ - Relative Positioning (sub PTZ)

- Client shall be able to move a PTZ device using the RelativeMove operation.

Function	Service	Requirement
RelativeMove	PTZ	Mandatory

PTZ - Presets (sub PTZ)

- Client shall be able to list the presets using the GetPresets operation.
- Client shall be able to move a PTZ device to a specific preset using the GotoPreset operation.
- Client may be able to set and remove presets using SetPreset and RemovePreset operations.

Function	Service	Requirement
GetPresets	PTZ	Mandatory
GotoPresets	PTZ	Mandatory
SetPresets	PTZ	Optional
RemovePresets	PTZ	Optional

PTZ - Home Position (sub PTZ)

- Client shall be able to move a PTZ device to its home position using the GotoHomePosition operation.
- Client may be able to set home position of a PTZ device using the SetHomePosition operation.

Function	Service	Requirement
GotoHomePosition	PTZ	Mandatory
SetHomePosition	PTZ	Optional

PTZ - Auxiliary Command (sub PTZ)

- Client shall be able to send auxiliary command to a PTZ device using the SendAuxiliaryCommand operation.

Function	Service	Requirement
SendAuxiliaryCommand	PTZ	Mandatory

Audio Streaming

- Client shall be able to receive a stream and playback audio in G.711 μ Law (Simplex-Camera Microphone Only, 1ch) codec.
- Client shall be able to configure a media profile for audio streaming using the GetCompatibleAudioSourceConfigurations, AddAudioSourceConfiguration, GetCompatibleAudioEncoderConfigurations and AddAudioEncoderConfiguration operations.
- Client may be able to modify an audio encoder configuration using the GetAudioEncoderConfigurationOptions, GetAudioEncoderConfiguration and SetAudioEncoderConfiguration operations.
- Client may be able to list audio sources using the GetAudioSources operation.
- Client may be able to modify an audio source configuration using the GetAudioSourceConfigurationOptions, GetAudioSourceConfiguration and SetAudioSourceConfiguration operations.

Function	Service	Requirement
AddAudioSourceConfiguration	Media	Mandatory
GetAudioSourceConfiguration	Media	Optional
GetAudioSourceConfigurations	Media	Optional
GetAudioSources	Media	Optional
RemoveAudioSourceConfiguration	Media	Optional
SetAudioSourceConfiguration	Media	Optional
GetCompatibleAudioSourceConfigurations	Media	Mandatory
GetAudioSourceConfigurationOptions	Media	Optional
GetAudioEncoderConfiguration	Media	Optional
GetAudioEncoderConfigurations	Media	Optional
AddAudioEncoderConfiguration	Media	Mandatory
RemoveAudioEncoderConfiguration	Media	Optional
SetAudioEncoderConfiguration	Media	Optional
GetCompatibleAudioEncoderConfigurations	Media	Mandatory
GetAudioEncoderConfigurationOptions	Media	Optional

Audio Streaming - G726 (sub Audio Streaming)

- Client shall be able to receive a stream and decode G726 audio using the selected profile over RTSP.

Function	Service	Requirement
No specific function requirement		

Audio Streaming - AAC (sub Audio Streaming)

- Client shall be able to receive a stream and decode AAC audio using the selected profile over RTSP.

Function	Service	Requirement
No specific function requirement		

Multicast Streaming (sub Video Streaming)

- Client shall be able to control multicast streaming using RTSP or the StartMultiCastStreaming and StopMultiCastStreaming operations;
- Client shall be able to receive a multicast stream by a device.

Function	Service	Requirement
StartMulticastStreaming	Media	Mandatory
StopMulticastStreaming	Media	Mandatory

Relay Outputs

- Client shall be able to list available relay outputs using the GetRelayOutputs operation.
- Client shall be able to set the settings of a relay output using the SetRelayOutputSettings operation.
- Client shall be able to trigger a relay output using the SetRelayOutputState operation.

Function	Service	Requirement
GetRelayOutputs	Device	Mandatory
SetRelayOutputSettings	Device	Mandatory
SetRelayOutputState	Device	Mandatory

NTP

- Client shall be able to configure NTP servers on a device using GetNTP and SetNTP operations.

Function	Service	Requirement
GetNTP	Device	Mandatory
SetNTP	Device	Mandatory

Dynamic DNS

- Client shall be able to configure Dynamic DNS on a device using GetDynamicDNS and SetDynamicDNS operations.

Function	Service	Requirement
GetDynamicDNS	Device	Mandatory
SetDynamicDNS	Device	Mandatory

Zero Configuration

- Client shall be able to configure Zero Configuration on a device using GetZeroConfiguration and SetZeroConfiguration operations.

Function	Service	Requirement
GetZeroConfiguration	Device	Mandatory
SetZeroConfiguration	Device	Mandatory

IP Address Filtering

- Client shall be able to configure, add and remove IP Address Filters on device using GetIPAddressFilter, SetIPAddressFilter, AddIPAddressFilter and RemoveIPAddressFilter operations.

Function	Service	Requirement
GetIPAddressFilter	Device	Mandatory
SetIPAddressFilter	Device	Mandatory
AddIPAddressFilter	Device	Mandatory
RemoveIPAddressFilter	Device	Mandatory

Device Mandatory Features

The Device Mandatory Features section list the features that are mandatory for the device and conditional for Client in order to be conformant.

This means that the requirements are only mandatory if the Client supports this feature.

Discovery

- Client shall be able to discover a device using WS-Discovery as specified in the core specification.
- Client may be able to get and set discovery mode using the operations GetDiscoveryMode and SetDiscoveryMode.
- Client may be able to list, add, modify and remove discovery scopes using the operations GetScopes, AddScopes, SetScopes and RemoveScopes.

Function	Service	Requirement
WS-Discovery	Core	Mandatory
GetDiscoveryMode	Device	Optional
SetDiscoveryMode	Device	Optional
GetScopes	Device	Optional
SetScopes	Device	Optional
AddScopes	Device	Optional
RemoveScopes	Device	Optional

Network Configuration

- Client shall be able to list and configure the device network interface using the GetNetworkInterfaces and SetNetworkInterfaces operations.
- Client shall be able to list and set the default gateway of the device using the GetNetworkDefaultGateway and SetNetworkDefaultGateway operations.
- Client may be able to set the device hostname using the SetHostName operation.
- Client may be able to list and set the DNS using the GetDNS and SetDNS operations.
- Client may be able to list and configure supported network protocols on the device using the GetNetworkProtocols and SetNetworkProtocols operations.

Function	Service	Requirement
GetNetworkInterfaces	Device	Mandatory
SetNetworkInterfaces	Device	Mandatory
GetNetworkDefaultGateway	Device	Mandatory
SetNetworkDefaultGateway	Device	Mandatory
GetHostname	Device	Optional
SetHostname	Device	Optional
GetDNS	Device	Optional
SetDNS	Device	Optional
GetNetworkProtocols	Device	Optional
SetNetworkProtocols	Device	Optional

System

- Client shall be able to get device information such as manufacturer, model and firmware version using the GetDeviceInformation operation.
- Client may be able to get and set time of the device using the GetSystemDateAndTime and SetSystemDateAndTime operations.
- Client may be able to return the device to factory settings using the SetSystemFactoryDefault operation.
- Client may be able to reboot the device using the Reboot operation.

Function	Service	Requirement
GetDeviceInformation	Device	Mandatory
GetSystemDateAndTime	Device	Optional
SetSystemDateAndTime	Device	Optional
SetSystemFactoryDefault	Device	Optional
Reboot	Device	Optional

User handling

- Client shall be able to create, list, modify and delete users from the device using the CreateUsers, GetUsers, SetUsers and DeleteUsers operations.

Function	Service	Requirement
GetUsers	Device	Mandatory
CreateUsers	Device	Mandatory
SetUser	Device	Mandatory
DeleteUsers	Device	Mandatory

Event handling

- Client shall be able to use at least one way to retrieve events out of the following:
 - “Pull” using the SetSynchronizationPoint, CreatePullPointSubscription and PullMessage operations.
 - “Push” using Notify, Subscribe, Renew and Unsubscribe operations from WS-BaseNotification.
- Client may be able to get information about what filter dialects and what topics are supported by the device using the GetEventProperties operation.
- Client may be able to use filters to select what events to retrieve using MessageContentFilter and TopicFilter.

Function	Service	Requirement
Notify	Event	Mandatory*
Subscribe	Event	
Renew	Event	
Unsubscribe	Event	
SetSynchronizationPoint	Event	Mandatory*
CreatePullPointSubscription	Event	
PullMessage	Event	
GetEventProperties	Event	Optional
TopicFilter	Event	Optional
MessageContentFilter	Event	Optional

(* = Client shall implement one or both groups of event functions to be conformant to the event interface)

Media profile configuration

- Client shall be able to list available profiles using GetProfiles or GetProfile operations.
- Client shall be able to create a media profile using the CreateProfile operation.
- Client may be able to delete a media profile using the DeleteProfile operation.

Function	Service	Requirement
GetProfiles	Media	Mandatory
GetProfile	Media	Mandatory
CreateProfile	Media	Mandatory
DeleteProfile	Media	Optional

Video source configuration

- Client shall be able to list the available video source configurations using GetVideoSourceConfigurations operation.
- Client shall be able to modify and add video source configurations using the GetCompatibleVideoSourceConfigurations, GetVideoSourceConfiguration, GetVideoSourceConfigurationOptions and SetVideoSourceConfiguration operations.
- Client may be able to list the available video sources using the GetVideoSources operation.
- Client may be able to remove video source configurations using RemoveVideoSourceConfiguration operation.

Function	Service	Requirement
GetVideoSources	Media	Optional
GetVideoSourceConfiguration	Media	Mandatory
GetVideoSourceConfigurations	Media	Mandatory
AddVideoSourceConfigurations	Media	Mandatory
RemoveVideoSourceConfiguration	Media	Optional
SetVideoSourceConfiguration	Media	Mandatory
GetCompatibleVideoSourceConfigurations	Media	Mandatory
GetVideoSourceConfigurationOptions	Media	Mandatory

Metadata configuration

- Client shall be able to list metadata configurations using GetMetaDataConfigurations operation.
- Client shall be able to modify metadata configurations using the, GetMetaDataConfiguration, GetMetaDataConfigurationOptions and SetMetaDataConfiguration operations.
- Client may be able to add metadata configurations to media profiles using GetCompatibleMetaDataConfigurations and AddMetaDataConfiguration operations.
- Client may be able to remove metadata configurations from media profiles using RemoveMetaDataConfiguration operation.

Function	Service	Requirement
GetMetadataConfiguration	Media	Mandatory
GetMetadataConfigurations	Media	Mandatory
AddMetadataConfiguration	Media	Optional
RemoveMetadataConfiguration	Media	Optional
SetMetadataConfiguration	Media	Mandatory
GetCompatibleMetadataConfigurations	Media	Optional
GetMetadataConfigurationOptions	Media	Mandatory

Reference

- [1] Title: ONVIF Programmer's Guide
Author: ONVIF White Paper
Version: Version 1.0 May 2011
Link: http://www.onvif.org/Portals/0/documents/WhitePapers/ONVIF_WG-APG-Application_Programmer's_Guide.pdf
Date of reference: 19/01/2016

- [2] Title: FAQ about ONVIF
Author: Avigilon
Version: 05/01/2015
Link: <http://avigilon.com/connected/IT/7-faqs-about-the-onvif-standard/>
Date of reference: 20/01/2016

- [3] Title: IP Camera Training: ONVIF explained
Author: MxInstaller
Version: 08/02/2012
Link: https://www.youtube.com/watch?v=wMpXgM_M7Ss
Date of reference: 20/01/2016

- [4] Title: Onvif Milestones
Author: Professional Security Magazine Online
Version: 23/09/2015
Link: <http://www.professionalsecurity.co.uk/products/integrated-systems/onvif-milestones/>
Date of reference: 20/01/2016

- [5] Title: ONVIF online course
Author: Tavcom
Version: unknown, but Tavcom strives to keep it up to date. 2015/12 data is included
Link: <http://www.tavcom.com/distance-learning-courses/onvif/>
Data of reference: 20/01/2016

- [6] Title: ONVIF Profile S Specification
Author: ONVIF
Version: 1.0, December 2011
Link: http://www.onvif.org/Portals/0/documents/op/20984_11763r01ONVIF_Members-Profile_S_Specification.pdf
Data of reference: 25/01/2016

- [7] Title: ONVIF Technical FAQ

Author: -
Version: -
Link: <http://geniusvision.net/doc/ONVIF%20Technical%20FAQ.pdf>
Data of reference: 08/02/2016

- [8] Title: ONVIF Conformance Process Specification
Author: ONVIF
Version: 3.1 / September 2014
Link: http://www.onvif.org/Portals/0/documents/op/ONVIF-Conformance-Process-Specification_v3-1.pdf
Data of reference: 08/02/2016
- [9] Title: ONVIF Interface Guide Specification
Author: ONVIF
Version: 1.0.1 / October, 2013
Link: <http://www.onvif.org/Portals/0/documents/op/ONVIF-Interface-Guide-Specification.pdf>
Data of reference: 08/02/2016
- [10] Title: Specification Set 2.6.1 Release Notes
Author: ONVIF
Version: 1 / June, 2015
Link: http://www.onvif.org/specs/rl/ONVIF_SpecSet_RL_2.6.1.pdf
Data of reference: 11/02/2016

Plan van Aanpak



ONVIF Plug-in

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

Auteur: Tim Kesteloo
Versie: 1.1
Datum: 29/01/2016

1. Inhoudsopgave

1. Inhoudsopgave	...1
2. Inleiding	...2
3. Organisatie	...3
3.1. Algemeen	...3
3.2. Ontwikkelmethode	...3
4. Opdrachtomschrijving	...5
4.1. Probleemstelling	...5
4.2. Doelstellingen	...6
4.2.1. Resultaten	...6
4.3. Randvoorwaarden	...7
4.4. Scope	...7
5. Risicoanalyse	...9
6. Aanpak	...10
6.1. Ontwikkelmethode	...10
6.1.1. Scrumban	...11
6.2. Activiteiten	...12
6.2.1. Onderzoek	...12
6.2.2. Design	...12
6.2.3. Constructie	...12
6.2.4. Testen	...13
6.2.5. Overdracht	...13
6.3. Planning	...13
7. Contact	...15
8. Referenties	...16

2. Inleiding

In onze huidige samenleving zijn camera's bijna niet meer weg te denken: Iedereen heeft er ten allen tijden 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.

Dit Plan van Aanpak beschrijft de aanpak die genomen zal worden om dit probleem tijdens de afstudeerperiode te tackelen, inclusief onder anderen de risico's, werkmethodiek en langetermijnplanning.

3. Organisatie

3.1. Algemeen

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

VDG Security is onderdeel van het technologiebedrijf TKH Group. De TKH Group bestaat uit een groot aantal bedrijven, met in totaal 5.337 werknemers. Zij bestaat uit drie hoofdgroepen: Telecom Solutions, Industrial Solution en Building Solutions. VDG Security valt onder de Building Solutions groep. Binnen deze groep wordt nauw samengewerkt met de zuster bedrijven die ook actief zijn in de security branch. Tijdens deze samenwerking wordt zoveel mogelijk de technologie van de zuster bedrijven gecombineerd met de video technologie van VDG Security.

VDG Security bestaat uit vier afdelingen; Research and Development, Sales, Support en Magazijn. De opdracht zal uitgevoerd worden binnen het team Research and Development.

3.2. Ontwikkelmethode

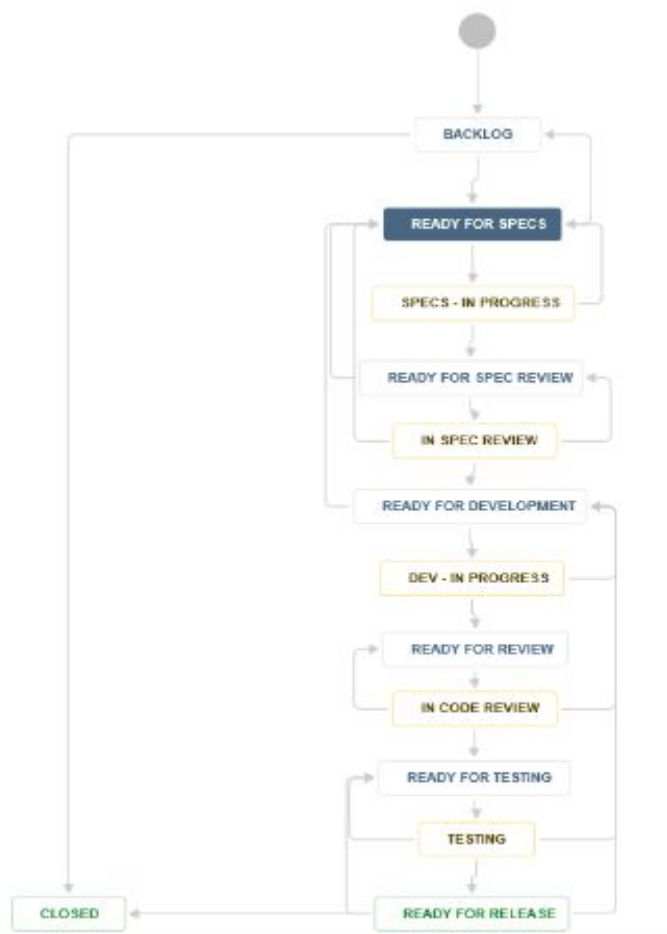
VDG R&D maakt gebruik van Scrumban, een methodiek die van origine was bedacht als overstap van Scrum naar Kanban, maar tegenwoordig als losstaande methode gebruikt kan worden.

Zij past deze toe door bugs en features op te nemen in het management systeem en deze door verschillende fases te laten lopen:

- 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 goedgekeurde specs uit te voeren. Wanneer dit volbracht is, is deze Ready for Review.
- In Code Review: Een andere ontwikkelaar neemt de gemaakt 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 JIRA, de management software die VDG gebruikt, beschreven staat. Hierin geven de pijlen aan wat de mogelijke transities zijn tussen de verschillende fases.



Figuur 1: VDG Workflow

VDG heeft een work-in-progress (WIP) van 1. Dit betekent dat een ontwikkelaar ten allen tijden 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.

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

4.1. Probleemstelling

Regelmatig krijgt de R&D afdeling van VDG Security de opdracht om een nieuwe camera te integreren in de VDG Sense software. Wanneer een nieuwe camera geïntegreerd dient te worden, dan bestaat de ontwikkeling van zo'n driver doorgaans uit de volgende functionaliteiten:

- Camera discovery: Het proces wat er voor nodig is om het IP- en MAC-adres van een camera te achterhalen in een netwerk.
- Video ontvangen: Opbouwen van een videoverbinding met de camera
- Audio ontvangen: Opbouwen van een audioverbinding met de camera
- Alarm inputs detecteren: Detecteren wanneer er input pinnen getriggerd worden
- Alarm outputs zetten: Output pinnen kunnen aansturen
- Pan, Tilt en Zoom (PTZ) commando's sturen: Voor het aansturen van de camera
- Camera specifieke events ontvangen: zoals gezichts- of kentekenherkenning, beweging detectie, etc

Elke camera fabrikant 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. Om dit proces te versnellen zijn de volgende twee initiatieven bedacht:

Pluginarchitectuur

Wanneer een nieuwe camera in de VDG Sense software is toegevoegd, dan moet in de huidige situatie een nieuwe versie van de software uitgebracht worden voordat klanten hiervan gebruik kunnen maken. VDG Security ziet liever dat de verschillende camera drivers in een aparte plugin (DLL) gebouwd kunnen worden met de volgende voordelen:

1. Indien een nieuwe camera-integratie is gedaan of een wijziging heeft plaatsgevonden, dan hoeft alleen de plugin uitgebracht of geüpdatet te worden.
2. Indien de architectuur een duidelijk interface heeft met de VDG Sense software, dan kunnen derde partijen zelf een plugin ontwikkelen die in de VDG Sense software ingeladen kan worden. Dit zal dan als een Software Development Kit (SDK) aangeleverd kunnen worden.

VDG Security heeft een plan en een ontwerp klaar liggen om zo'n dergelijke pluginarchitectuur met bijhorende SDK te ontwikkelen, maar is nog niet aan de ontwikkeling van een plugin begonnen. Door middel van het implementeren van een plugin in de nieuwe pluginarchitectuur kan proefondervindelijk duidelijk worden of de pluginarchitectuur en SDK voldoende generiek zijn.

ONVIF

Ten tweede is VDG Security erg geïnteresseerd in ONVIF. Dit is een protocol/standaard die door een aantal vooraanstaande camerafabrikanten is ontwikkeld, waarmee het voor integrators makkelijker moet worden om met minimale tijd investering meerdere type camera's te ondersteunen. De gedachte hierachter is dat als een integrator het ONVIF protocol in hun VMS heeft geïmplementeerd, deze automatisch alle camera's ondersteunt die ook het ONVIF protocol ondersteunen. Middels dit protocol wordt bijvoorbeeld het opbouwen van een videoverbinding op een uniforme manier afgehandeld en maakt de fabrikant en type van de camera niet meer uit.

4.2. Doelstellingen

De doelstelling van de afstudeeropdracht is om een ONVIF driver te implementeren als plugin in de nieuwe pluginarchitectuur binnen de gestelde [scope](#). Hoewel VDG verantwoordelijk is voor het implementeren van de SDK architectuur, zal veel overleg noodzakelijk zijn zodat de implementatie van de plugin zo optimaal mogelijk zal kunnen plaatsvinden.

Alhoewel VDG Security al een aantal ONVIF camera's ondersteunt, bestaat er twijfel of deze implementatie verschillende ONVIF versies aankan en of alle features (op de juiste manier) ondersteund worden. Daarmee is het tevens een doel om tijdens de afstudeerperiode het ONVIF protocol te onderzoeken binnen de gestelde [scope](#). Hierbij ligt de aandacht op de verschillende features die ONVIF ondersteunt en welke interessant zijn voor VDG.

Omdat dit de eerste plugin zal zijn in de nieuwe pluginarchitectuur zou dit ertoe kunnen leiden dat blijkt dat pluginarchitectuur niet voldoende generiek is opgezet en dus niet alle functionaliteiten kan ondersteunen. In dit geval is het mogelijk om aanbeveling te doen tot het aanpassen of uitbreiden van de pluginarchitectuur.

4.2.1. Resultaten

Het traject wordt binnen VDG beschouwd als een testcase voor de nieuwe pluginarchitectuur. Het einde van dit traject zal dus uitwijzen of de door VDG geïmplementeerde pluginarchitectuur goed genoeg werkt om deze aan haar klanten te kunnen aanbieden.

Als bewijs hiervan zal het aan het eind van de stageperiode mogelijk zijn om in een testomgeving (simplistische versie van de Core / testapplicatie om core-functionaliteiten te imiteren) een ONVIF camera aan te sturen en de geïmplementeerde features te kunnen gebruiken.

Hierbij worden de volgende producten opgeleverd:

- Onderzoek ONVIF, features en implementatie
- Plan van Aanpak (dit document)
- Functional Design
- Technical Design
- Testplan & Testrapport
- Productdocumentatie
- Afstudeerverslag
- ONVIF plug-in

4.3. Randvoorwaarden

Zoals genoemd bij de doelstellingen is VDG verantwoordelijk voor het implementeren van de pluginarchitectuur in VDG Sense. Aangezien de plugin hierop moet aansluiten is er een grote afhankelijkheid van deze implementatie, zowel de voortgang van de ontwikkeling ervan als de tijd beschikbaar om onduidelijkheden of aanpassingen te bespreken.

Daarnaast is gebleken dat bepaalde ONVIF documentatie alleen toegankelijk is bij het bezit van een lidmaatschap. De TKH Group heeft een hoog niveau lidmaatschap, maar het is nog onduidelijk in hoeverre deze gebruikt mag worden. Ook is het nog onduidelijk of deze documentatie noodzakelijk is voor het succesvol uitvoeren van het onderzoek en de opdracht.

4.4. Scope

Uit onderzoek [\[2\]](#) is gebleken dat ONVIF erg groot is: Zij bestaat momenteel uit 5 profielen (S,C,G,A,Q) die elk een specifieke set features implementeert. Deze features zijn vervolgens onder te verdelen in verplichte (mandatory) en optionele (optional) features.

Een feature is een omschrijving van een bepaalde functionaliteit, inclusief welke functies dienen te worden geïmplementeerd of gebruikt. Deze functies zijn ook onderverdeeld in verplichte (mandatory) en optionele (optional) functies.

Er is besloten om tijdens de afstudeerperiode Profiel S te implementeren. Hiermee vallen de overige 4 profielen dus buiten de scope van implementatie.

Profiel S is dusdanig groot dat verwacht wordt deze niet in zijn geheel tijdens de afstudeerperiode geïmplementeerd kan worden.

Vandaar dat hiervoor de volgende prioriteitenlijst is opgezet:

1. Verplichte (mandatory) features met de verplichte (mandatory) functies implementeren, zodat men "Profile S Compliant" is.

Uitzondering: Indien VDG bepaalde optionele functies van de verplichte features expliciet wil implementeren, vallen zij onder prioriteitscategorie #1.

2. Optionele (optional) features die VDG momenteel al ondersteund (voor andere camera's buiten ONVIF) met de verplichte (mandatory) functies implementeren, zodat men "Profile S Compliant" is voor alle features die VDG Sense ondersteund.

3. Optionele (optional) features die VDG momenteel al ondersteund (voor andere camera's buiten ONVIF) met de optionele (optional) functies implementeren indien VDG expliciet van deze functies gebruik wil maken.
4. Optionele (optional) features die VDG graag in de toekomst wil ondersteunen met de verplichte (mandatory) functies. Men is dan ook voor deze features "Profile S Compliant".
5. Optionele (optional) features die VDG graag in de toekomst wil ondersteunen met de optionele (optional) functies indien VDG expliciet van deze functies gebruik wil maken.

Het minimale doel voor de afstudeerperiode is om #3 van de prioriteitenlijst succesvol af te kunnen ronden.

Indien tijdens het implementeren blijkt dat er tijd over zal blijven, zijn er twee mogelijkheden tot uitbreiding:

1. Het implementeren van alle optionele functies voor alle geïmplementeerde features (afhankelijk van hoeveel dit zijn naast de expliciet uitgesproken functies).
2. Het implementeren van overige optionele (optional) features met de verplichte (mandatory) functies.

Tevens valt binnen de beoogde werkzaamheden:

- Het onderzoek naar ONVIF, de huidige implementatie ervan in VDG Sense (voor de beeldvorming van de probleemsituatie. Dit zal inhoudelijk niet uitgebreid worden onderzocht) en de wensen van VDG met ONVIF (wat tevens zal leiden tot de lijst met te implementeren features).
- Het onderzoeken en bespreken van de pluginarchitectuur die gebruikt zal worden.
- Het ontwerpen van de plugin. Afhankelijk van de grote leidt dit tot één of twee ontwerprapporten, het Functional Design en Technical Design.
- Het testen van de plugin. Met behulp van de ONVIF "Client Test Tool" of een andere (zelfgemaakte) applicatie controleren of de plugin werkt zoals beoogd.
- Het bijhouden van het afstudeerrapport. Hier dient voldoende tijd voor over te blijven naast de bovengenoemde activiteiten.

Overige werkzaamheden met betrekking tot de scope (of juist erbuiten):

- Het implementeren van ONVIF profielen C,G,A en Q valt buiten de scope. Globaal onderzoek naar de features en of deze interessant zijn voor VDG kan wel, indien hier interesse in is.
- Het schrijven van een gehele testapplicatie valt buiten de scope. Indien VDG niet in staat is een applicatie aan te bieden waarmee tests verricht kunnen worden, of er geen externe applicaties beschikbaar zijn, dan zal het testplan handmatig uitgevoerd moeten worden. Kleine aanpassingen aan een applicatie zodat hiermee tests gedaan kunnen worden valt wel binnen de scope.

5. Risicoanalyse

Er is een inschatting van de risico's gemaakt, waarbij gekeken is wat de kans is dat dit risico voorkomt en de grootte van de impact indien dit zou gebeuren. De combinatie van deze factoren bepaald de ernst van het risico. De volgende tabel bevat alleen risico's waarbij de ernst "hoog" is. Hierbij is ook een maatregel gegeven die genomen kan worden om te voorkomen dat deze situatie plaatsvindt.

Risico	Kans	Impact	Maatregel
Lastig te definiëren grootte van opdracht en snelheid van ontwikkelen. Hierdoor is het moeilijk de scope te bepalen	5/5	2/5	Proof-of-concept om beter inzicht te krijgen in de moeilijkheid.
Snelheid van reactie TKH Group mbt toegang tot membership-only documenten en programma's	2/5	?	Contact is al gelegd en contactpersoon is toegewezen.
Ziekte of onverwachte omstandigheden die leiden tot langdurig verzuim	1/5	5/5	Geen (behalve gezonde voeding!)
Faillissement VDG of TKH Group	1/5	5/5	Besproken: TKH is financieel gezond. Kans is erg klein.
Beperkte tijd voor ondersteuning door drukte groot project	3/5	2/5	Tweezijdige communicatie kort houden. Tijdig aangeven wanneer men denkt tijd nodig te hebben of geen te kunnen bieden.
Geen ONVIF apparaten beschikbaar voor testdoeleinden	2/5	4/5	Vroegtijdig Contact leggen met Siquira, producent van ONVIF camera's binnen de TKH Group (zelfde werkgroep als VDG)

6. Aanpak

6.1. Ontwikkelmethode

Om te bepalen welke ontwikkelmethode gebruikt zal worden is een afweging gemaakt tussen de volgende mogelijkheden:

Ontwikkelmethode	Ervaring	Voordelen	Nadelen
RUP (Rational Unified Process)	Studie	- Fasering (Inception, Elaboration, Construction, Transition) - Duidelijke indeling van taken en producten	- Veel overhead door documentatie - Outdated
AUP (Agile Unified Process) [3]	Geen	- Alle voordelen van RUP met minder documentatie	- Outdated (laatste versie 2006)
DAD (Disciplined Agile Development) [4]	Geen	- Schaalbaar naar grote projectgroep - Combinatie van best-practices overige ontwikkelmethoden	- Complex om in te stappen zonder begeleiding
Scrum	Stage	- Ingericht op afgebakende sprints	- Voorgedefinieerde tijd aan taken inschatten beperkt flexibiliteit
Kanban [5]	Geen	- Tijdschatting taken optioneel(demand planning) - Flexibele indeling sprints	- Meer gericht op logistiek dan ontwikkel - Continue proces
Scrumban [5]	Geen	- Tijdschatting taken optioneel(demand planning) - Methode VDG - Flexibele indeling sprints	- Minder gericht op sprints (continue proces)

Vanwege de grote overhead en de duur van de afstudeerperiode valt RUP snel af. De Agile variant (AUP) oogt als een interessante oplossing, maar blijkt sinds 2006 niet meer geüpdatet te worden. Dit betekent dat deze ontwikkelmethode 10 jaar achterloopt op de ontwikkelingen. Daarnaast heeft de initiator aangegeven sinds 2009 te helpen met de ontwikkeling van een nieuwe methode; DAD. Deze ziet er erg volledig en volwassen uit, maar brengt hiermee meer complexiteit de opdracht in.

Scrum is al bekend bij de afstudeerder. VDG heeft aangegeven te werken met Scrumban, wat een combinatie is van Scrum en Kanban.

Kanban is een methode die is ontstaan in Japan door Toyota. Dit systeem maakt gebruik van een takenlijst, die erg lijkt op de backlog van Scrum. Echter, waar Scrum de taken toekent aan het begin van de sprint en deze allemaal klaar zijn aan het eind van de sprint, is Kanban flexibeler in het maken van de planning. Een taak wordt opgepakt door het team, waar een ontwikkelaar zelf een taak kan kiezen ('pullen'). De ontwikkelaar werkt aan een maximum aantal taken tegelijkertijd, ookwel de work-in-progress (WIP). Wanneer een nieuwe hogere prioriteit in de takenlijst verschijnt, is het mogelijk een actieve taak terug te plaatsen en aan de nieuwe taak te werken. De 'late-binding' in Kanban ten opzichte van de 'early-binding' bij Scrum laat het verschil in flexibiliteit zien tussen de verschillende methoden.

Scrumban combineert de flexibiliteit van Kanban met de structuur van Scrum. Op deze manier ontstaat een nog flexibeler ontwikkelmethode die is ingericht voor IT projecten. Daarbij maakt VDG al gebruik van Scrumban en heeft hier een workflow voor opgezet die aangetoond heeft succesvol te zijn.

6.1.1. Scrumban

Om de hoogste productiviteit te behalen, is besloten om de ontwikkelmethode van VDG (zoals beschreven in [3.2 Organisatie-Ontwikkelmethode](#)) te zullen toepassen: Scrumban. Deze is gekozen omdat VDG hier veel ervaring mee heeft en de overige alternatieven niet beter aansluiten als methodiek dan Scrumban.

Hier zijn wel enkele aanpassingen op van toepassingen.

Zo kan in de workflow van VDG een taak wisselen van persoon, voornamelijk wanneer deze van fase veranderd (bijvoorbeeld tussen Development en Testing). Voor de afstudeeropdracht zal de afstudeerder verantwoordelijk zijn voor alle fases, behalve de Review fasen; deze gebeuren door de mentor of een specialist.

VDG hanteert een work-in-progress (WIP) van 1. Dit betekent dat een ontwikkelaar niet aan meer dan één taak tegelijk mag werken zonder deze af te ronden of naar de vorige fase terug te zetten. Dit is voornamelijk gekozen om de planner gemakkelijk inzicht te geven in de bezigheden van de ontwikkelaars. Aangezien de afstudeeropdracht hierbuiten valt en door de afstudeerder zelf gepland zal worden, is besloten om deze waarde niet als verplichting te moeten navolgen, maar hier wel naar te streven waar mogelijk.

Daarnaast is Scrumban minder gericht op het inrichten van sprints. Door de korte duur van het traject is besloten om de beoogde taken wel op te delen in sprints en hier vooraf een tijdsverwachting aan te hangen. Op deze manier is het makkelijker om overzicht te houden op de voortgang van het proces.

6.2. Activiteiten

6.2.1. Onderzoek

Om juiste aanbevelingen te kunnen doen aan VDG dienen de volgende aspecten nog te worden uitgezocht:

- VDG heeft al iets met ONVIF gedaan. Wat houdt dit in en wat zijn de verdere wensen van VDG?
- De pluginarchitectuur is in een relatief ver gevorderd stadium; Zo is de designfase zo goed als klaar en zou deze gebouwd kunnen worden. Wat zijn de huidige ideeën en waar denkt men problemen te kunnen tegenkomen?

Deze taken zullen in het begin van het afstudeertraject (+/- 4 weken) duidelijk moeten worden om een juiste inschatting te kunnen maken voor de planning. Naar aanleiding van deze onderzoeken zullen de belangrijkste wensen en requirements duidelijk worden, waarna het Plan van Aanpak af te ronden is.

Daarnaast zal er ook een proof of concept gemaakt worden. Deze heeft als doel om een beter beeld te krijgen van de moeilijkheid van de opdracht en om bekend te raken met de systemen die gebruikt zullen gaan worden. Hierbij ligt de focus op de volgende punten:

- Het inladen van plugins in VDG Sense en de nieuwe architectuur.
- Het geautomatiseerd omzetten van WSDL naar functies, en deze kunnen aanroepen.
- Een kleine testtool maken om de functies van de ingeladen plugin aan te roepen.

6.2.2. Design

Naar aanleiding van het voorgaande onderzoek kan een design worden gemaakt voor de verdere ontwikkeling van de plug-in. Hierbij moet gekeken worden welke ONVIF versies nu geïmplementeerd zullen worden en welke in de toekomst. Tijdens de designfase zal hier in zekere mate rekening mee moeten worden gehouden.

Dit zal leiden tot één of twee designdocumenten, afhankelijk van de grote en complexiteit van de plug-in, namelijk het Functional Design en het Technical Design.

Aangezien aanpassingen aan het design tijdens de constructiefase verwacht worden, zullen deze documenten niet definitief zijn tot het eindproduct naar wens is.

6.2.3. Constructie

Hierin zal de daadwerkelijke plug-in worden geschreven. Nadat de meest belangrijke requirements en technische keuzes vastgelegd zijn, is het mogelijk om hier een proof-of-concept voor te maken. Wanneer deze geslaagd is, zal de code uitgebreid worden en gereviewed, zodat niet alleen de functionaliteit van de plug-in, maar ook de code aansluit bij de wensen van VDG.

6.2.4. Testen

Tijdens het afstudeertraject zal er geen vaste testperiode worden aangewezen. Juist zal ernaar worden gestreefd dat dit een veelvoorkomend aspect is tijdens de constructiefase. Op deze manier is het mogelijk om op korte termijn de verschillen tussen de architectuur en plug-in te bespreken en zo te verbeteren. Aan het eind van het traject zal de gehele plug-in worden getest. Dit in combinatie met de tests tijdens constructie zal leiden tot het Testrapport.

6.2.5. Overdracht

Om zeker ervan te zijn dat VDG gebruik kan gaan maken van alle opgeleverde eindproducten zal er tijd gereserveerd worden om deze bestanden over te dragen. Dit houdt in dat alle producten nogmaals beschreven worden en doorgenomen, zodat deze in de toekomst gebruikt kunnen worden.

6.3. Planning

Alhoewel er Agile zal worden gewerkt, is het erg lastig om de beginfase op deze manier in te delen. Er zal vooraf goed onderzoek moeten worden gedaan naar ONVIF en de requirements van de organisatie. Hierdoor lijken de eerste sprints erg op een waterval-methode.

Wanneer de onderzoeksfase en ontwerpfase afgerond zijn, zal de Agile-ontwikkelmethode goed tot zijn recht komen. Elke sprint kan een set features opgepakt worden om te implementeren, testen, en eventueel in een designverandering door te voeren binnen de plug-in of plug-in architectuur. Aan het eind van elke sprint wordt een werkend product (de geïmplementeerde features in de plug-in) opgeleverd.

Voor het traject van 19 weken is gekozen om een verdeling te maken van sprints die elk 2 weken duren. Hierbij is één sprint gedefinieerd van 3 weken, die gepland staat in de voorlaatste sprint. Deze zal gebruikt worden om de laatste tests uit te voeren en bugs te repareren.

(volgende pagina)

Figuur 2: Strokenplanning afstudeerperiode

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Sprint	1: Oriëntatie		2: Research		3: Design		4: Design		5: Implementatie		6: Implementatie		7: Implementatie		8: Testing		9: Overdracht		
Datum (HHS)	25-1-2016	1-2-2016	8-2-2016	15-2-2016	22-2-2016	29-2-2016	7-3-2016	14-3-2016	21-3-2016	28-3-2016	4-4-2016	11-4-2016	18-4-2016	25-4-2016	2-5-2016	9-5-2016	16-5-2016	23-5-2016	30-5-2016
Taken																			
Kennismaking																			
Onvif Onderzoek																			
Proof of concept																			
Design																			
Implementatie																			
Testing																			
Overdracht																			
Documentatie																			
Tussenproducten																			
Studie deadlines																			
Bezoek Begeleidend examinator (di 16-02)																			
Voortgangsverslag																			
Concept afstudeerdossier (zie aantekeningen)																			
Tussentijds assessment (+2e examinator)																			

Plan van Aanpak

Onvif onderzoeksrapport

Functional Design

Technical Design

Testplan

Testrapport

Afstudeerdossier

25%

45%

60%

100%-3w

100%

7. Contact

Tijdens de afstudeerperiode zijn de volgende mensen actief in de volgende rollen:

Naam	Instantie	Functie	Email
Tim Kesteloo	Student	Afstudeerder	kestelootim@gmail.com
Cobie van der Hoek	Haagse Hogeschool	Eerste begeleider	j.j.vanderhoek@hhs.nl
Tony Andrioli	Haagse Hogeschool	Tweede begeleider	a.andrioli@hhs.nl
Danny van Deutekom	VDG Security	Opdrachtgever	d.vandeutekom@vdgsecurity.com
Robin Hermann	VDG Security	Bedrijfsmentor	r.hermann@vdgsecurity.com

8. Referenties

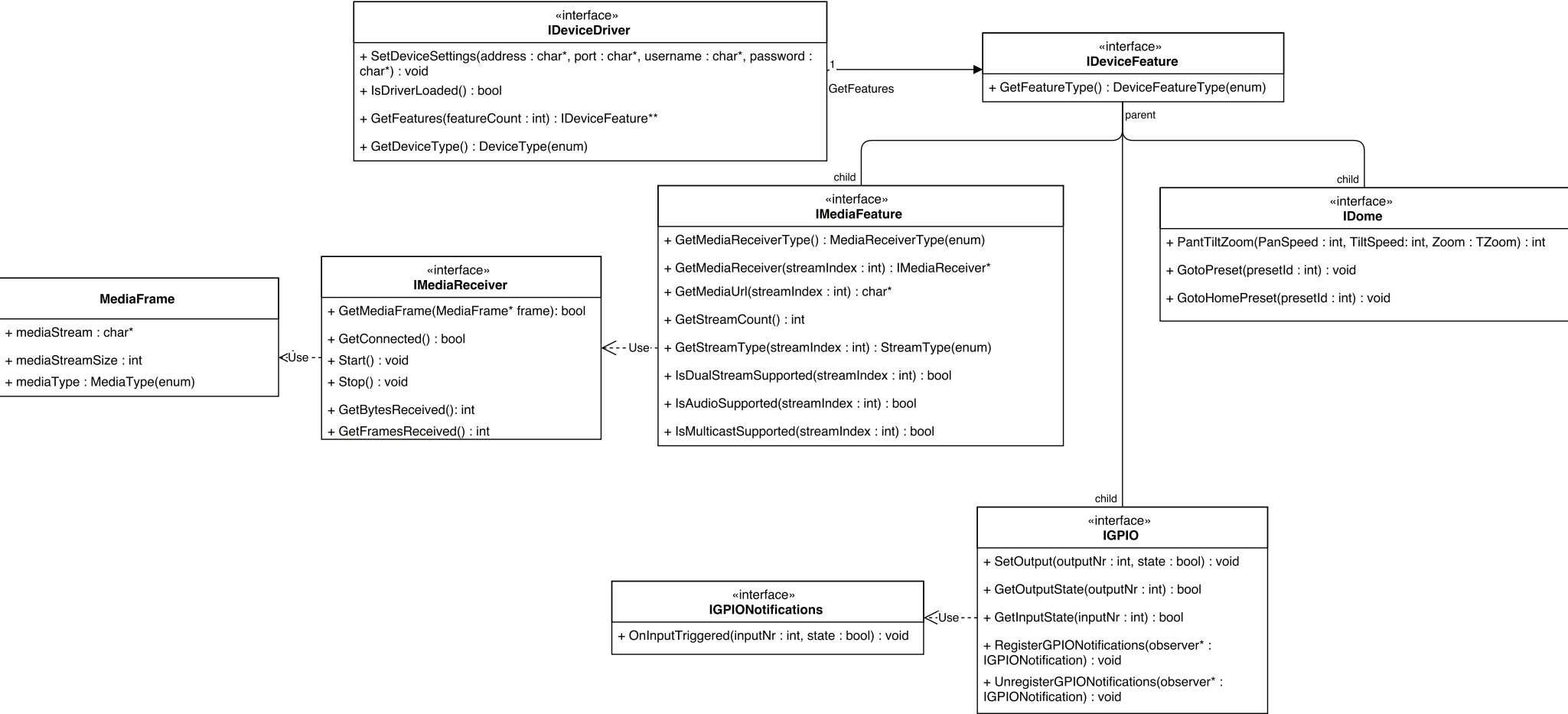
- [1] Titel: Checklist volledigheid Plan van Aanpak
Auteur: Roel Grit, Noordhof Uitgevers
Versie: 2011
Link: http://hoadd.noordhoff.nl/sites/7720/_assets/7720d24.pdf
Datum bezocht: 27/01/2016

- [2] Titel: ONVIF Research
Auteur: Tim Kesteloo
Versie: 26/01/2016
Link: [TO DO]
Datum bezocht: 26/01/2016

- [3] Titel: Agile Unified Process (AUP)
Auteur: Scott Ambler
Versie: 2006
Link: <http://www.ambysoft.com/unifiedprocess/agileUP.html>
Datum bezocht: 26/01/2016

- [4] Titel: Disciplined Agile Delivery (DAD)
Auteur: Scott Ambler
Versie: 2016
Link: www.disciplinedagiledelivery.com
Datum bezocht: 26/01/2016

- [5] Titel: Scrum vs. Kanban vs. Scrumban: Planning, Estimation, and
Performance Metrics
Auteur: Dalia Lasaite
Versie: 02/05/2013
Link: <http://www.eylean.com/blog/2013/05/scrum-vs-kanban-vs-scrumban-planning-estimation-and-performance-metrics/>
Datum bezocht: 26/01/2016



kdwsdl2cpp voorbeeld

13. GetCapabilities

Description:

Any endpoint can ask for the capabilities of a device using the capability exchange request response operation. The device shall indicate all its ONVIF compliant capabilities through the GetCapabilities command. The capability list includes references to the addresses (XAddr) of the service implementing the interface operations in the category. Apart from the addresses, the capabilities only reflect optional functions.

SOAP action:

<http://www.onvif.org/ver10/device/wsdl/GetCapabilities>

Input:

```
[GetCapabilities]
  ■ Category - optional, unbounded; [CapabilityCategory]
    List of categories to retrieve capability information on.
    - enum { 'All', 'Analytics', 'Device', 'Events', 'Imaging', 'Media', 'PTZ' }
```

Output:

```
[GetCapabilitiesResponse]
  ■ Capabilities [Capabilities]
    Capability information.
    ■ Analytics - optional; [AnalyticsCapabilities]
      Analytics capabilities
      ■ XAddr [anyURI]
        Analytics service URI.
      ■ RuleSupport [boolean]
        Indicates whether or not rules are supported.
      ■ AnalyticsModuleSupport [boolean]
        Indicates whether or not modules are supported.
    ■ Device - optional; [DeviceCapabilities]
      Device capabilities
      ■ XAddr [anyURI]
        Device service URI.
      ■ Network - optional; [NetworkCapabilities]
        Network capabilities.
        ○ IPFilter [boolean]
          Indicates support for IP filtering.
        ○ ZeroConfiguration [boolean]
          Indicates support for zeroconf.
        ○ IPVersion6 [boolean]
          Indicates support for IPv6.
        ○ DynDNS [boolean]
          Indicates support for dynamic DNS configuration.
        ○ Dot11Configuration [boolean]
          Indicates support for IEEE 802.11 configuration.
        ○ Dot1XConfigurations [int]
          Indicates the maximum number of Dot1X configurations supported by the device
        ○ HostnameFromDHCP [boolean]
          Indicates support for retrieval of hostname from DHCP.
        ○ NTP [int]
          Maximum number of NTP servers supported by the devices SetNTP command.
        ○ DHCPv6 [boolean]
          Indicates support for Stateful IPv6 DHCP.
      ■ System - optional; [SystemCapabilities]
        System capabilities.
        ○ DiscoveryResolve [boolean]
          Indicates support for WS Discovery resolve requests
```

Visuele weergave van de functie GetCapabilities en (een deel van) de response.


```

class TT__Capabilities
{
public:
    void setAnalytics( const Device::TT__AnalyticsCapabilities& _analytics );
    Device::TT__AnalyticsCapabilities analytics() const;
    void setDevice( const Device::TT__DeviceCapabilities& _device );
    Device::TT__DeviceCapabilities device() const;
    void setEvents( const Device::TT__EventCapabilities& _events );
    Device::TT__EventCapabilities events() const;
    void setImaging( const Device::TT__ImagingCapabilities& _imaging );
    Device::TT__ImagingCapabilities imaging() const;
    void setMedia( const Device::TT__MediaCapabilities& _media );
    Device::TT__MediaCapabilities media() const;
    void setPTZ( const Device::TT__PTZCapabilities& _pTZ );
    Device::TT__PTZCapabilities pTZ() const;
    void setExtension( const Device::TT__CapabilitiesExtension& _extension );
    Device::TT__CapabilitiesExtension extension() const;
    void setAnyAttribute( const KDSOapValue& _anyAttribute );
    KDSOapValue anyAttribute() const;
    KDSOapValue serialize( const QString& valueName ) const;
    void deserialize( const KDSOapValue& mainValue );
    TT__Capabilities();
    ~TT__Capabilities();

public:
    TT__Capabilities( const TT__Capabilities& );
    TT__Capabilities &operator=( const TT__Capabilities& );

private:
    class PrivateDPtr;
    QSharedDataPointer<PrivateDPtr> d_ptr;

```

kdSOAP conversie naar C++ van het Capabilities object, onderdeel van
GetCapabilitiesResponse

```

class TT__NetworkCapabilities
{
public:
    void setIPFilter( bool _iPFilter );
    bool iPFilter() const;
    void setZeroConfiguration( bool _zeroConfiguration );
    bool zeroConfiguration() const;
    void setIPvVersion6( bool _iPVersion6 );
    bool iPVersion6() const;
    void setDynDNS( bool _dynDNS );
    bool dynDNS() const;
    void setExtension( const Device::TT__NetworkCapabilitiesExtension& _extension );
    Device::TT__NetworkCapabilitiesExtension extension() const;
    void setAnyAttribute( const KDSOapValue& _anyAttribute );
    KDSOapValue anyAttribute() const;
    KDSOapValue serialize( const QString& valueName ) const;
    void deserialize( const KDSOapValue& mainValue );
    TT__NetworkCapabilities();
    ~TT__NetworkCapabilities();

```

kdSOAP conversie van het NetworkCapabilities object, onderdeel van het
DeviceCapabilities object in Capabilities.

kdSOAP gebruik van functies (voorbeeld)

```
//GetCapabilities
Device::TDS__GetCapabilities request;
Device::TT__CapabilityCategory getCategory = Device::TT__CapabilityCategory::All;
QList<Device::TT__CapabilityCategory> Categories; Categories << getCategory;
request.setCategory(Categories);
Device::TDS__GetCapabilitiesResponse response = deviceManagement.getCapabilities(request);
```

Het aanroepen van de functie GetCapabilities zoals dit in kdSOAP zou kunnen.

`deviceManagement` is van het type `Device::DeviceBindingService` die in de headerfile is gedefinieerd.

General

VDG currently uses a SOAP service which they find annoying; it's hard to read and debug and in general not pleasant to work with. Since ONVIF will use a SOAP service, we decided to find an alternative which would improve readability.

kdSOAP is an open-source and free to use project provided by KDAB, which is specialized in Qt and C++ development and training.

The goal of kdSOAP was to simplify the use of SOAP services in the Qt environment. This is accomplished by using the Qt-core libraries, and using the kdwsdl2cpp program, which can convert a .wsdl file to a .cpp implementation that will work with kdSOAP.

Review

It has taken quite some time to build the project inside the Sense-Third Party environment. However, we got it to work, as well as the kdwsdl2cpp program. Using the code that these two programs combined produce, it was easy to quickly implement functioncalls that were formatted in a readable manner.

After creating a few simple functions, we moved on to more complex functions, which needed authorization. Here, we stumbled upon a big problem with the library.

kdSOAP **does not** (yet) yield **support** for any WS protocols, which includes the **WS-Security** and **WS-Discovery** protocol that ONVIF demands.

This means that the use of just kdSOAP will currently (while these are not supported) suffice.

In order to continue, a choice must be made concerning the SOAP service that will be used.

1. We can continue to use kdSOAP for easy communication which does not require authorization. When authorization is needed, we can use a different (current) SOAP service.
2. We can write our own implementation in kdSOAP for the WS-protocols that we need. This will probably take a lot of time, which has already been dissuaded by both Robin and Danny.
3. We can try to get KDAB to implement WS-services into kdSOAP. Accepting and completing this request may take a lot of time.
4. We can use the current SOAP service, even though it's not preferred by the other developers.
5. We can try to use a different SOAP implementation of which we know it supports the WS protocols (Security and Discovery at minimum)

Discussion

Talk with Wilco and Robin on 15/02:

Decided that we'd like a generalisation over the communication part. This way, we can be more flexible when something changes in this implementation (e.g. JSON, other SOAP library).

Wilco and Robin think that WS-Discovery should not be a big problem, since sending a multicast message is something that they do often. We could easily make this ourselves. WS-Security might be harder. Wilco and I will try to add our own headers and see how far we get. Simultaneously, we'll ask KDAB for their plans for implementing WS-Security. A workaround could be using the gSOAP library for sending messages that require authorization. This would result in the use of two libraries, which wouldn't exactly improve readability, but could be used as a state in between version, till KDAB finished WS-Security. Another option is using a .NET DLL which is responsible for the SOAP communication. Siquira does this, so we could use their help to accomplish this.

Conclusion

We'll make an extra generalisation over the communication part, so we can easily switch between SOAP or other communication implementations.

KDAB has been asked about their plans for WS-Security, and we found that making our headers inside kdSOAP is impossible.

This leaves us with using gSOAP or try the .NET DLL.

Both will require extra time to research and implement. However, the designphase can continue while this decision is being made.

Design Document



ONVIF Plug-in

Author: Tim Kesteloo

Version: 5.0

Date: 29/04/2016

1. Introduction

This document describes the requirements and design choices made for the ONVIF plug-in. This is divided into two main categories; The Functional Design and the Technical Design. The Functional Design describes the plugin's function and requirements i.e how will it be used. The Technical Design will describe the implementation requirements, choices and diagrams.

2. Changelog

Version#	Notes
1.0-26/02/2016	First release
2.0-11/03/2016	Updated some designs to version 2.0. Added a Conclusion that wraps the designs up.
3.0-25/03/2016	Redesign of the class diagram This was appended to the 5.1 Plugin Design (class diagram (complex)) Updated designs to version 3.0
4.0-08/04/2016	Updated the requirements to include more requirements from Sense-side
4.1-29/04/2016	Updated figure 1 Multiple small typo's

Contents

- [1. Introduction](#)
- [2. Changelog](#)
- [3. Requirements](#)
 - [3.1. Functional requirements ONVIF](#)
 - [3.2. Functional requirements plugin](#)
 - [3.3. Technical \(non-functional\) requirements](#)
- [5. Functional Design](#)
 - [5.1. Plugin architecture](#)
 - [5.2. WSDL generated code](#)
 - [5.3. Profile S Features](#)
- [6. Technical Design](#)
 - [6.1. Plugin Design](#)
 - [6.2. Call from server to plugin](#)
 - [6.3. Feature Extraction](#)
- [7. Conclusion](#)
- [8. References](#)

3. Requirements

Many of the requirements are predetermined by the ONVIF Specification. It describes which applications or protocols shall, shall not, may or may not be implemented or used. This chapter describes the VDG specific requirements that were given to the ONVIF plugin.

3.1. Functional requirements ONVIF

- The goal for the plugin is to become “ONVIF Profile S Compliant”, supporting the features that VDG currently supports with her own driver. This means that the following additional features will be implemented (full explanation in the ONVIF Research [\[1\]](#)):
 - PTZ (all sub-services excluding RelativeMove and AbsoluteMove)
 - Streaming: MPEG4, H264, Audio, Multicast
 - Relay Outputs
 - Events
- The plugin will NOT be responsible for controlling multiple devices at the same time. To achieve this, the server will have to create multiple instances of the plugin, which will each control one ONVIF device.
- The plugin is responsible for authenticating the system to the device. ONVIF demands the usage of the WS-Security protocol. The information will be delivered by the server to the plugin within the first few setup messages.

3.2. Functional requirements plugin

1. The plugin must be able to detect if a camera supports ONVIF
2. The plugin must be able to send authorized SOAP messages to an ONVIF device
3. The plugin can detect ONVIF devices on the network
4. The plugin can detect which features an ONVIF device supports
 - a. If a device supports Media, Sense will be able to show video through the plugin
 - i. If the device Media supports video codec JPEG, H264 and/or MPEG4, Sense will be able to configure and retrieve any stream through the plugin
 - ii. If the device Media supports Multicast streaming, Sense can configure and retrieve a multicast stream through the plugin
 - iii. If the device Media supports audio streaming, Sense can enable this and receive an audio stream through the plugin
 - b. If a device supports PTZ, Sense will be able to control the device through the plugin.
 - i. Moving a camera is done through ‘continuous move’ commands
 - ii. If the device PTZ supports presets, Sense can call the device to move to a preset through the plugin
 - iii. If the device PTZ supports home, Sense can call the device to go to its home position through the plugin

- c. If a camera has a wiper, Sense can (de)activate it through the plugin.
- d. If a camera supports IO inputs, Sense will be able to read the inputs when they are triggered.
- e. If a camera supports IO outputs, Sense will be able to set the output states through the plugin.
- f. If a camera support events, Sense can receive the events through the plugin
 - i. Event Handling is done through the 'Push' method

3.3. Technical (non-functional) requirements

- It was chosen to implement the current, most recent, Specification Function Set, version 2.6.1. This states which versions of services are combined as a full version.
 - DeviceManagement Version 2.5 / December 2014
 - Streaming Version 2.6 / June 2015
 - PTZ Version 2.6.1 / December 2015
 - Core Version 2.6.1 / December 2015
 - Media Version 2.6.1 / December 2015

Because of ONVIF's backwards-compatibility nature, it's not necessary to implement the services below yet. They could be used if they allow for higher usability, but won't be considered until the functionalities of the main services are implemented first.

- Media2 Version 1.0 / December 2015
- DeviceIO Version 2.6 / December 2015

Not all functionalities present in these specifications will be implemented. The functions that will be implemented can be found in ONVIF Research: Profile S [\[1\]](#) or the list "Compliance & Functions: Marked List (normalized to requirements)" [\[2\]](#).

- The plugin will be made in C++(11). Even though it would be possible to link external DLL's written in a different language into the architecture, VDG wants to be able to maintain the plugin from within the current IDE, which is Qt.
- To send the SOAP messages as defined by the ONVIF Specification, we will be using gSOAP, with an option to kdSOAP. kdSOAP has been tested, but problems with the authentication were encountered [\[3\]](#). If these problems are solved before or at the start of the construction phase, it becomes highly favoured over gSOAP because of increased readability. However, without authentication it is useless, which results in the use of gSOAP, which we know works.
- The application will be tested through a combination of manual and automated testing. A test application will be made that calls each implemented function at least once. During this time, a network trace program (e.g. Wireshark) will log the communication. Afterwards, the ONVIF Client Test Tool will parse the network trace log and determine whether all communication is syntactically correct. Manual testing will determine if the functions give the expected results, which validates the content.
- Functions related to the SOAP communication will be implemented as followed whenever possible:

bool functionName(requestType requestName, responseType responseName);

The return value is used as a first level check if the action succeeded. Returning the response as a parameter allows for more flexible implementation and usage.

5. Functional Design

The plugin will never be called directly by a user. This happens through the plugin architecture. The plugin implements all the necessary classes which will ensure that all functions are called when the plugin is being used in the application by the user.

5.1. Plugin architecture

VDG is responsible for creating the architecture where to the plugin will connect. This has been designed to be as generic as possible, so all future plugins will be able to connect to these functionalities, without having to change the architecture.

The first version of the architecture (see [figure 1](#)) depends on a main DeviceDriver interface. This class is responsible for telling the application whether the plugin has been loaded. When this is done, the application can request a list of functionalities that the plugin has implemented; DeviceFeatures. The architecture also defines the DeviceFeatures and the functionality that is expected from them. The DeviceFeatures are Media (streaming video and audio), GPIO (input and output command), Dome (control of the device) and Discovery.

5.2. WSDL generated code

ONVIF provides WSDL definitions with its specifications. Because of this, we're able to automatically generate code which define the functions to communicate with an ONVIF device. These calls already include a SOAP client, so all that needs to be done is set the correct settings for the SOAP client, like the endpoint or authorization.

As concluded in the kdSOAP review [\[3\]](#), it was chosen to start this implementation using the gSOAP service. However, VDG does not want to continue using the gSOAP service, which means that it's already known that these parts will be changed in the code. This has been taken into account for the design, which should result in minimal effort to swap the library that will be used.

It was decided to realize this by building an extra layer between the generated code and where the functions will be called from. This "Communication" generalization will make it easier to possibly change the chosen SOAP library. This also allows for separating the used systems in the project, which allows for higher re-usability.

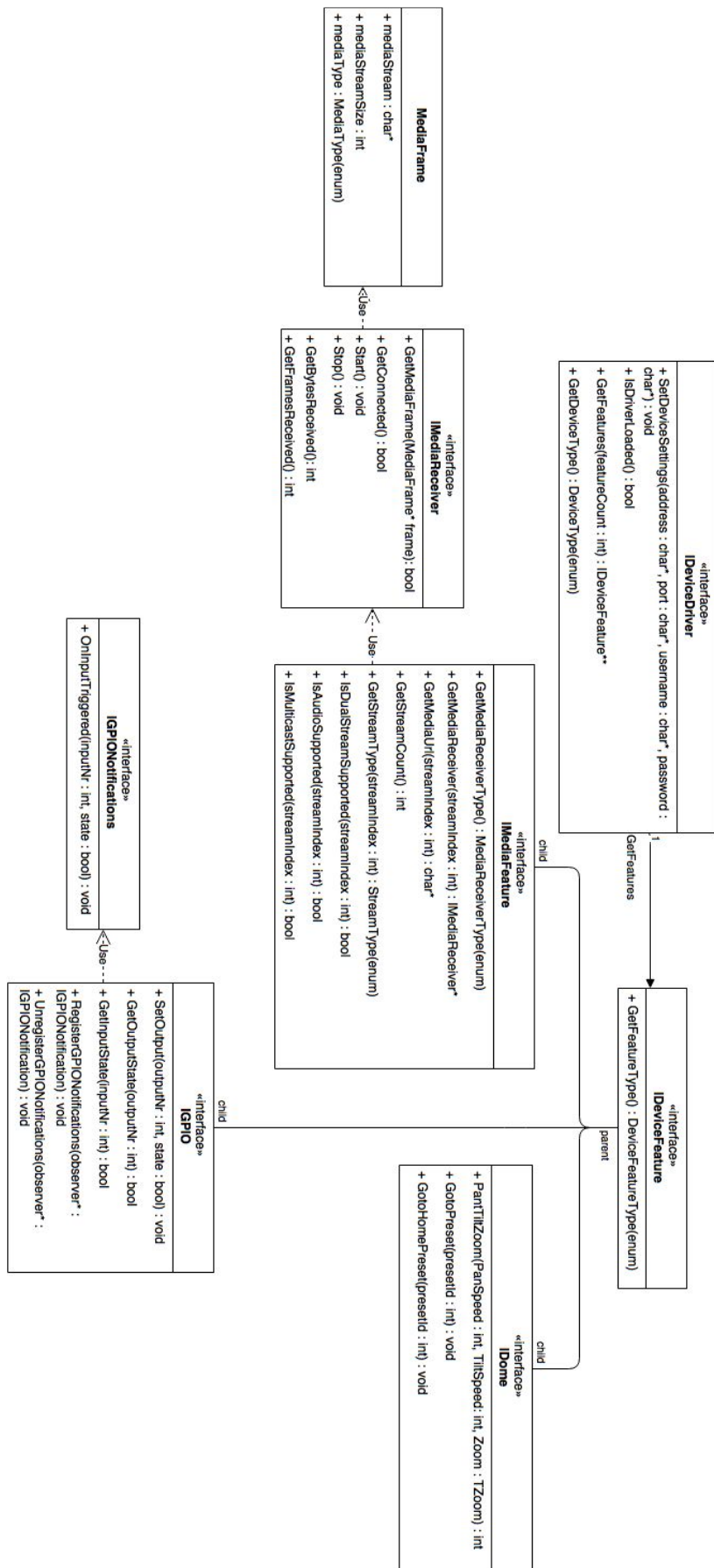


Figure 1: Class diagram plugin architecture

5.3. Profile S Features

Profile S consists of features, divided into mandatory, conditional and optional features to implement. During the ONVIF Research [1] a list was made with all features that will be implemented (see figure 2). Each of these features describe functionalities that need to be implemented, including a set of functions from the corresponding WSDL.

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			
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			
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

Figure 2: Profile S Feature list

6. Technical Design

6.1. Plugin Design

The plugin architecture, designed by VDG, has been described in the previous chapter (see [4.1 Functional Design: Plugin architecture](#)). This chapter will discuss the choices made for the implementation of the plugin.

It is expected that implementing a factory pattern will increase the flexibility when creating the features. DeviceDriver is not responsible for creating the features; it IS responsible for returning them to the server. This means that DeviceDriver should be able to delegate this request to another class. Using a factory pattern makes sense, since we have a set of purely virtual defined objects that have to be made. The next question becomes which factory pattern would be best to achieve this.

After some research [\[4\]](#) [\[5\]](#) and discussion it was decided to not implement this at the time. It might be interesting to implement this idea on server-architecture level, but not solely in the plugin, since this requires some definitions in the interfaces as well. The responsibility for creating the features will still be done in a separate class, but this won't implement a factory pattern.

[Figure 3](#) shows the first draft for the implementation of the plugin.

The white blocks are the pre-defined classes from the server architecture. The grey blocks are classes in the plugin. Many of these classes have close connections to one another, which is why both are shown in the design.

The starting point of the diagram can be interpreted as the inheritance from IDeviceDriver. This specific DeviceDriver will implement the [first calls](#) that will be made to the driver. Though we've stepped away from the idea of implementing a factory pattern, we still use a separate class who will be responsible for deciding which features are present in the specific device. This FeatureFactory will be able create instances of the specific features and return them as a list of DeviceFeatures.

Each feature implements a specific IDeviceFeature parentclass.

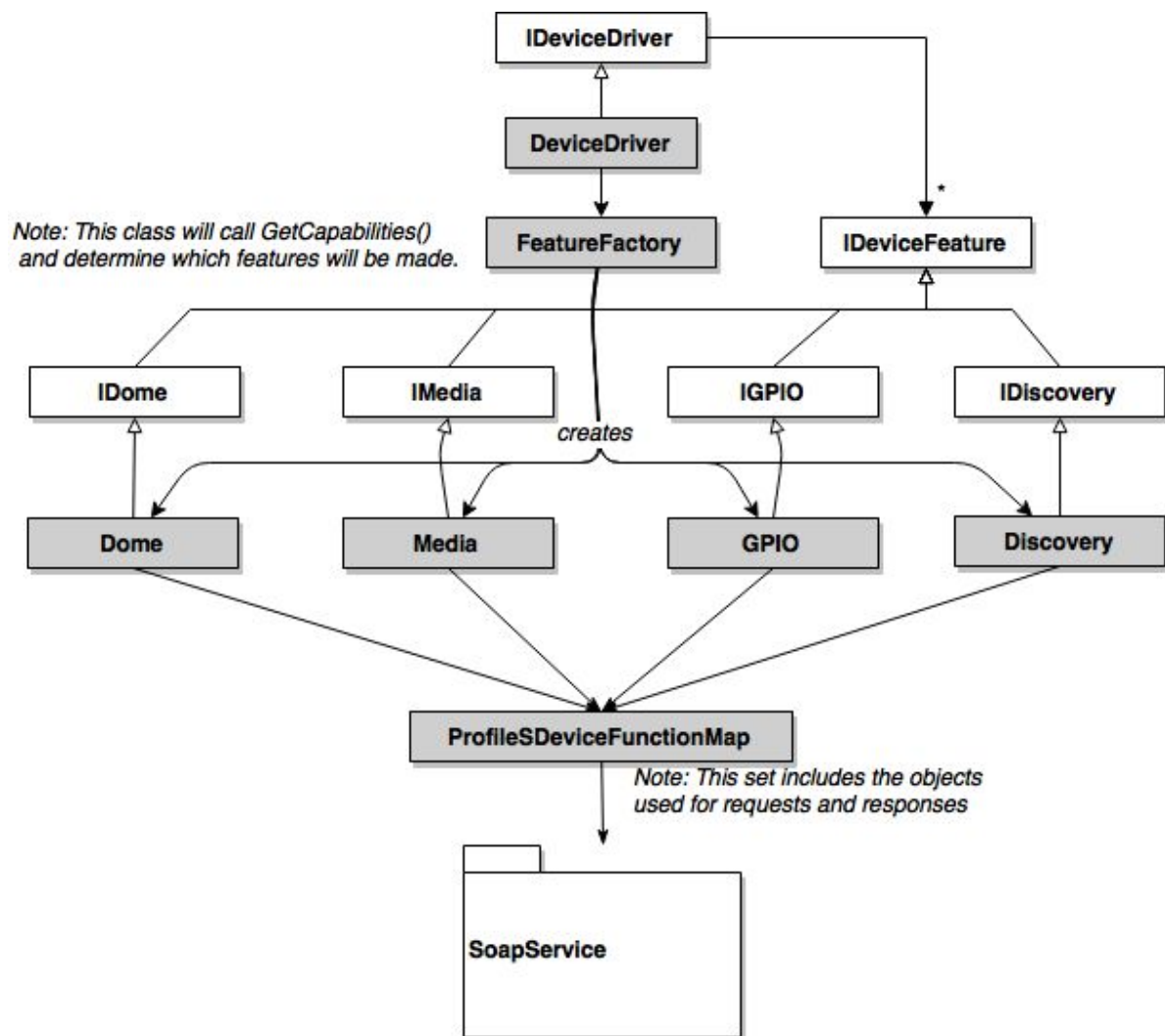


Figure 3: Class diagram (simple) v2.0

This design has now become flexible enough to allow for different soap implementations to be linked to the specific Service interface. As long as the SOAP objects that are returned can be converted to an OnvifType defined by the interface function, the specific feature will be able to use it.

This design has the implications of a factory design pattern, but we've chosen to not implement it as such at this time. A possible use for the pattern would be when a different SOAP implementation was made, but since this won't happen yet (and if it would, they wouldn't both be operative at the same time), it was chosen to not invest time in this at this point.

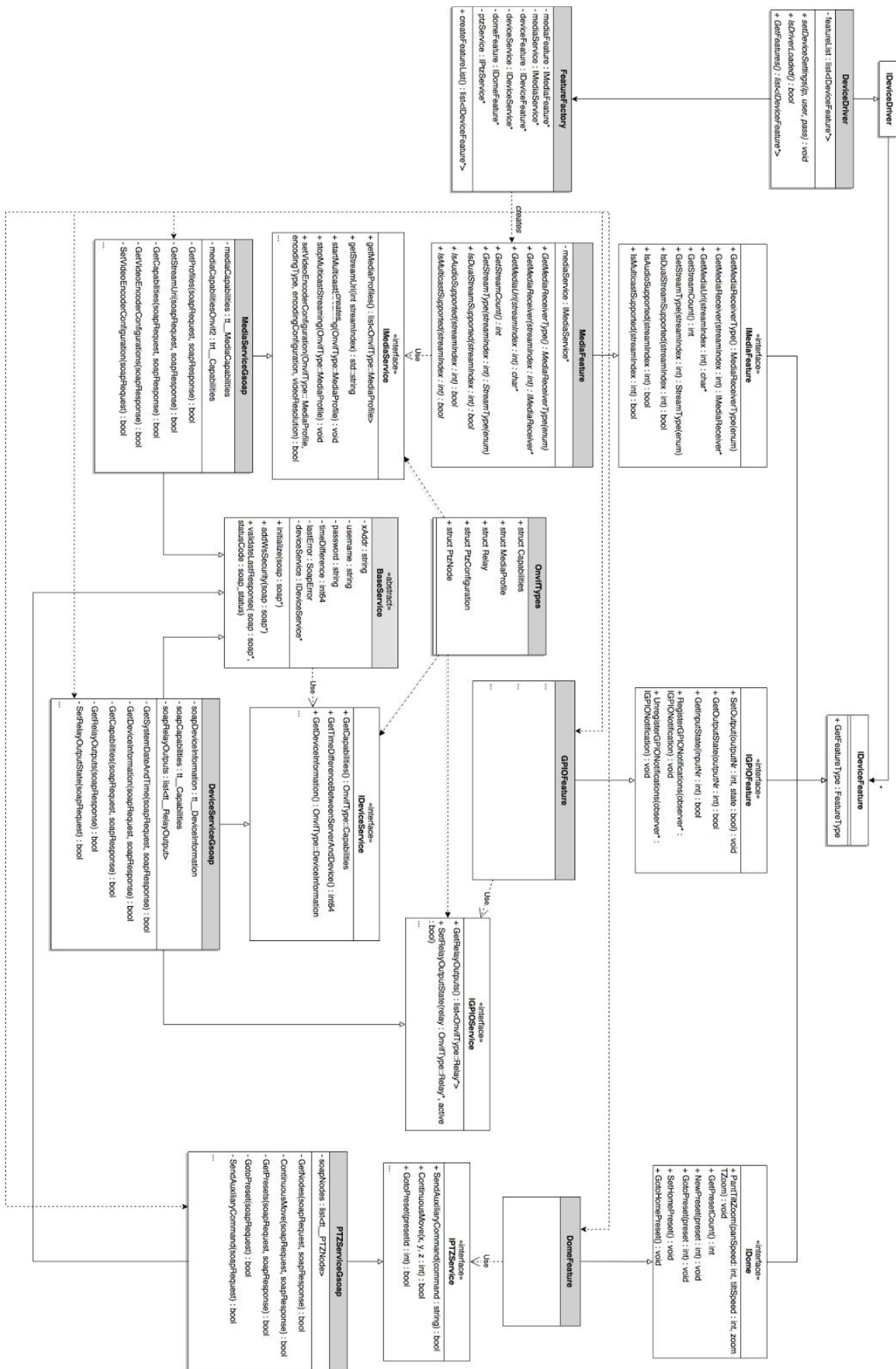


Figure 4: Class diagram (complex) v3.0

6.2. Call from server to plugin

Figure 5 gives a better understanding of what the first messages between the server and plugin will look like. On the server-side, this starts with the IDeviceDriver class. This class is responsible for telling the server whether the driver is actually loaded. The server will then request the featurelist. All further communication will be between the server and feature.

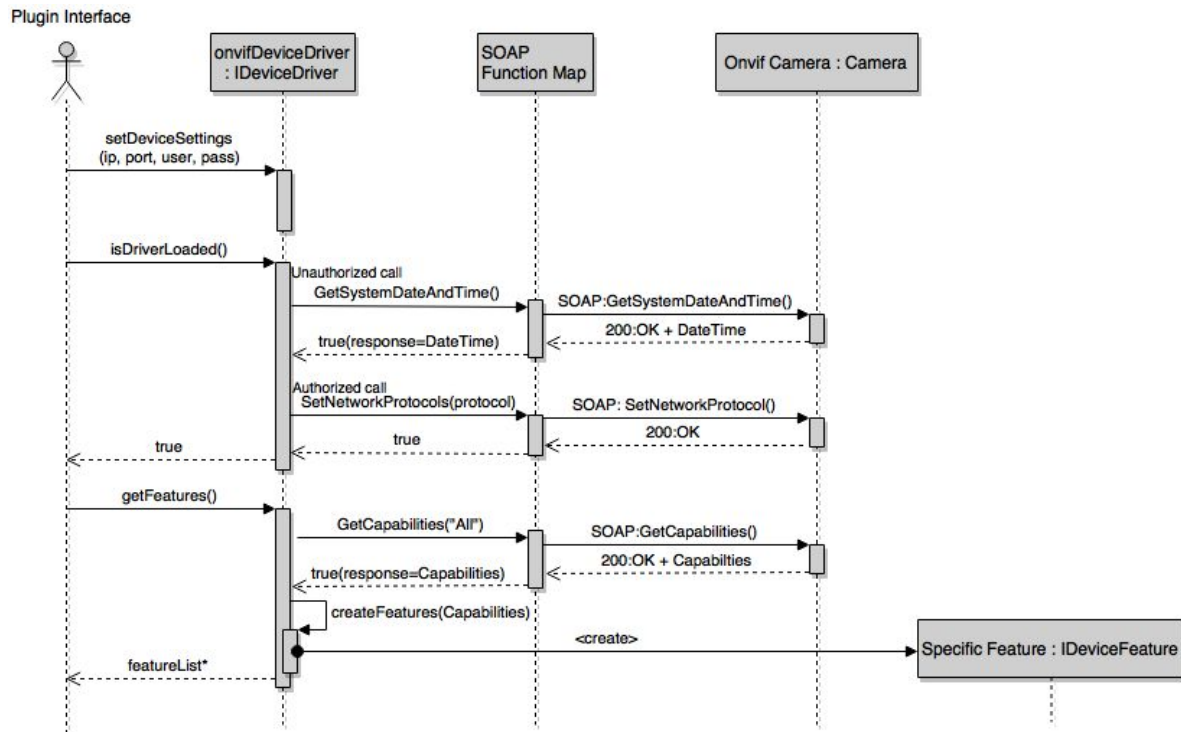


Figure 5: Sequence diagram - server calls to plugin at start

6.3. Feature Extraction

In order to create feature classes as defined by the plugin architecture, we must know what a Device is capable of. We do this by calling the *GetCapabilities()* function defined by ONVIF. Certain parameters in this list will indicate which features are supported by the Device.

The following are already known and defined as deciding parameters:

- IDome : GetCapabilities()->PTZ->XAddr (The address to the PTZ service)
- IMediaFeature : GetCapabilities()->Media ->XAddr (The address to the media service)
- IGPIO : GetCapabilities() -> Device -> IO (InputConnectors/RelayOutputs/Auxiliary)

As defined by the ONVIF S Specification, whenever a Device implements a feature, it's mandatory that it implements all functions for this feature. This means that once we know if a feature exists in the Device, we also know that all its functions are implemented by the Device.

7. Conclusion

[Figure 4](#) shows the complex model that the requirements and choices have resulted into. This structure shows how they layers are build up:

On the top, we have the architectural interfaces defined by Sense. These are divided into features. If someone has to build a driver for a camera, they can choose which features they implement. In case of ONVIF, this is decided by the GetCapabilities command, where we ask the device this very question.

Each feature communicates with the interface of a Service. These define which questions we want to ask a specific part of the camera. The return parameters from these functions are simple types (like string or int) or more complex self-defined OnvifTypes.

At the bottom we find our implementation of this communication layer, which use the gSoap generated proxy classes from the ONVIF WSDL's to establish a connection.

This design allows for multiple scalability changes:

When we want to change the SOAP Service used to communicate with the device, we only have to replace the bottom layer classes and create new implementations of the defined IServices. This also means that if the chosen protocol entirely (e.g. from SOAP to REST/JSON), we can simply create a new class to inherit from the specific IServices.

We also have a very solidly defined line between the responsibility of a Feature and Service; A service will always make a call to device and convert its response to a readable OnvifType. A feature can choose to collect data retrieved in order to reduce the amount of calls needed in the future, or modify the input data before a call is made to the service.

The following design, including the Features and gSoap implementation of the Services, will be implemented and tested over the course of the graduation internship.

8. References

- [1] Title: ONVIF Research
Author: Tim Kesteloo
Version: 1.0 - 12/02/2016
Link: <https://docs.google.com/document/d/1Rmjvfil3ZIXGfTgRskwKloXRgnZ3uzvG8IMWIb6apZc/>
Date of reference: 18/02/2016

- [2] Title: Compliance & Functions spreadsheet
(table 3: "List Marked (normalized to requirements)")
Author: Tim Kesteloo
Version: 24/02/2016
Link: https://docs.google.com/spreadsheets/d/1ncU6iZZDnAqVS9dKZGO7RO6_ahVooKSjTnbSKK_VOjw/#gid=2102960516
Date of reference: 24/02/2016

- [3] Title: kdSOAP Review
Author: Tim Kesteloo
Version: 15/02/2016
Link: <https://docs.google.com/document/d/1MFILjPuBafbiMbDAyxVsNkMURoajxXJu0A3H9qGEFyl>
Date of reference: 24/02/2016

- [4] Title: Simple Factory vs. Factory Method vs. Abstract Factory
Author: Corey Broderick
Version: 05/07/2009
Link: <http://corey.quickshiftconsulting.com/blog/first-post>
Date of reference: 22/02/2016

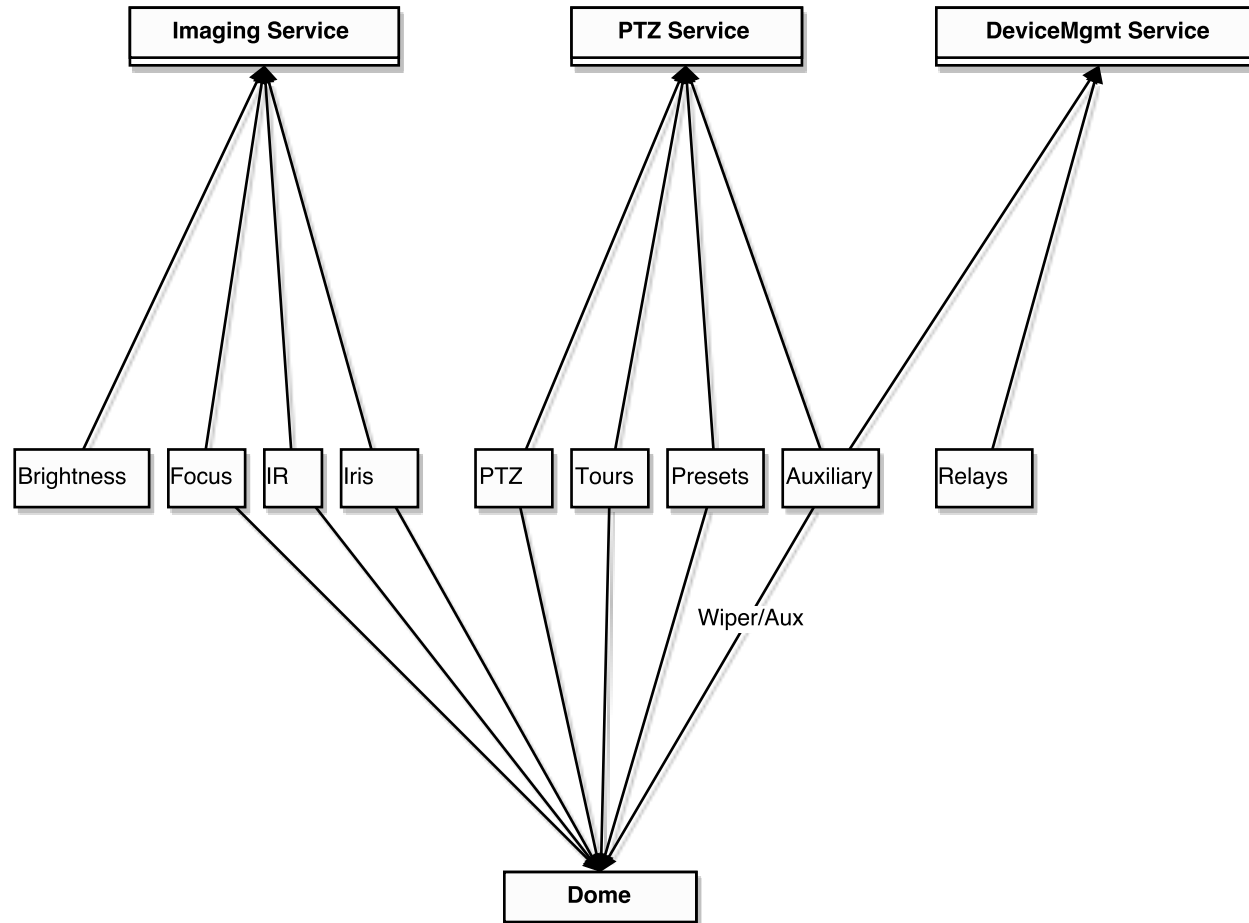
- [5] Title: Abstract Factory vs Factory Method vs Builder
Author: Stack Overflow
Version: 10/09/2010
Link: <http://stackoverflow.com/questions/3687299/abstract-factory-factory-method-builder>
Date of reference: 22/02/2016

	Mandatory	Conditional	Device Mandatory		Total Implemented Mandatory	Difference v1.0-v1.1	Has new definition in other WSDL						
Getter	8	19	16		60								
Setter (Add and Remove commands included)	4	20	14										
Action (neither a real get nor set)	4	11	6										
Profile (S) Mandatory Functions	Service (WSDL)	Requirement	Sub-chapter	List of features that are guaranteed to be supported by the client.									
GetProfiles	Media	Mandatory	Video Streaming				-	Media2.wSDL					
GetStreamUri	Media	Mandatory	Video Streaming				-	Media2.wSDL					
Media Streaming using RTSP	Streaming	Mandatory	Video Streaming				-						
Media Streaming using RTSP - JPEG RTP header extension	Streaming (MJPEG)	Mandatory	Video Streaming				-						
GetVideoEncoderConfiguration	Media	Mandatory	Video Encoder Configuration				Optional	-					
GetVideoEncoderConfigurations	Media	Mandatory	Video Encoder Configuration				Optional	Media2.wSDL					
AddVideoEncoderConfiguration	Media	Optional	Video Encoder Configuration				-	-					
RemoveVideoEncoderConfiguration	Media	Optional	Video Encoder Configuration				-	-				Media2 has RemoveConfiguration, but these are not the same	
SetVideoEncoderConfiguration	Media	Mandatory	Video Encoder Configuration				-	Media2.wSDL				Media2 has SetVideoEncoderConfiguration, but these are not the same	
GetCompatibleVideoEncoderConfigurations	Media	Optional	Video Encoder Configuration				-	-					
GetVideoEncoderConfigurationOptions	Media	Mandatory	Video Encoder Configuration				-	Media2.wSDL					
GetGuaranteedNumberOfVideoEncoderInstances	Media	Optional	Video Encoder Configuration				-	-					
WS-UsernameToken Authentication	Core	Mandatory	User Authentication				-						
HTTP Digest	Core	Mandatory	User Authentication				-						
GetCapabilities	Device	Mandatory	Capabilities				-	-					
GetWsdlUri	Device	Optional	Capabilities				-	-					
Profile (S) Conditional Functions	Service	Requirement	Sub-chapter	The Profile Conditional Features section list the features that shall be implemented if the client supports the feature.									
SetSynchronizationPoint	Media	Optional	Video Streaming (MPEG4)	/			-						
SetSynchronizationPoint	Media	Optional	Video Streaming (H264)	/			-						
AddPTZConfiguration	Media	Mandatory	PTZ				-	-					
RemovePTZConfiguration	Media	Optional	PTZ				-	-					
GetNodes	PTZ	Mandatory	PTZ				-						
GetNode	PTZ	Mandatory	PTZ				-						
GetConfigurations	PTZ	Mandatory	PTZ				-						
GetConfiguration	PTZ	Optional	PTZ				-						
GetConfigurationOptions	PTZ	Optional	PTZ				-						
SetConfiguration	PTZ	Optional	PTZ				-						
ContinuousMove	PTZ	Mandatory	PTZ				-						
Stop	PTZ	Mandatory	PTZ				-						
GetStatus	PTZ	Optional	PTZ				-						
AbsoluteMove	PTZ	Mandatory	PTZ (Absolute Positioning)	P2			-						
RelativeMove	PTZ	Mandatory	PTZ (Relative Positioning)	P2			-						
GetPresets	PTZ	Mandatory	PTZ (Presets)				-						
GotoPreset	PTZ	Mandatory	PTZ (Presets)				-						
SetPreset	PTZ	Optional	PTZ (Presets)				-						
RemovePreset	PTZ	Optional	PTZ (Presets)	/			-						
GotoHomePosition	PTZ	Mandatory	PTZ (Home Position)				-						
SetHomePosition	PTZ	Optional	PTZ (Home Position)				-						
SendAuxiliaryCommand	PTZ	Mandatory	PTZ (Auxiliary Command)				-	-					
AddAudioSourceConfiguration	Media	Mandatory	Audio Streaming				-	-					
GetAudioSourceConfiguration	Media	Optional	Audio Streaming				-	-					
GetAudioSourceConfigurations	Media	Optional	Audio Streaming				-	Media2.wSDL				Media1 doesn't have GetAudioSourceConfigurationOptions for example	
GetAudioSources	Media	Optional	Audio Streaming				-						
RemoveAudioSourceConfiguration	Media	Optional	Audio Streaming				-						
SetAudioSourceConfiguration	Media	Optional	Audio Streaming				-						
GetCompatibleAudioSourceConfigurations	Media	Mandatory	Audio Streaming				-						
GetAudioSourceConfigurationOptions	Media	Optional	Audio Streaming				-						
GetAudioEncoderConfiguration	Media	Optional	Audio Streaming				-						
GetAudioEncoderConfigurations	Media	Optional	Audio Streaming				-						

AddAudioEncoderConfiguration	Media	Mandatory	Audio Streaming				-							
RemoveAudioEncoderConfiguration	Media	Optional	Audio Streaming				-							
SetAudioEncoderConfiguration	Media	Optional	Audio Streaming				-							
GetCompatibleAudioEncoderConfigurations	Media	Mandatory	Audio Streaming				-							
GetAudioEncoderConfigurationOptions	Media	Optional	Audio Streaming				-							
	Streaming	Mandatory	Audio Streaming (G726)											
	Streaming	Mandatory	Audio Streaming (AAC)											
StartMulticastStreaming	Media	Mandatory	Video Streaming (Multicast)				Conditional!	Media2.wsdl						
StopMulticastStreaming	Media	Mandatory	Video Streaming (Multicast)				Conditional!	Media2.wsdl						
GetRelayOutputs	Device	Mandatory	Relay Outputs				-	DeviceIO.wsdl						
SetRelayOutputSettings	Device	Mandatory	Relay Outputs				-	DeviceIO.wsdl						
SetRelayOutputState	Device	Mandatory	Relay Outputs				-	DeviceIO.wsdl	DeviceMgmt doesnt have GetRelayOutputOptions for example					
GetNTP	Device	Mandatory	NTP				-	-						
SetNTP	Device	Mandatory	NTP				-	-						
GetDynamicDNS	Device	Mandatory	Dynamic DNS	x			-							
SetDynamicDNS	Device	Mandatory	Dynamic DNS	x			-							
GetZeroConfiguration	Device	Mandatory	Zero Configuration	x			-							
SetZeroConfiguration	Device	Mandatory	Zero Configuration	x			-							
GetIPAddressFilter	Device	Mandatory	IP Address Filtering	P2			-	-						
SetIPAddressFilter	Device	Mandatory	IP Address Filtering	P2			-	-						
AddIPAddressFilter	Device	Mandatory	IP Address Filtering	P2			-	-						
RemoveIPAddressFilter	Device	Mandatory	IP Address Filtering	P2			-	-						
Device Mandatory Functions	Service	Requirement	Sub-chapter	The Device Mandatory Features section list the features that are mandatory for the device and conditional for Client in order to be conformant.										
WS-Discovery	Core	Mandatory	Discovery	This means that the requirements are only mandatory if the Client supports this feature.										
GetDiscoveryMode	Device	Optional	Discovery				Moved to 1	-						
SetDiscoveryMode	Device	Optional	Discovery				Moved to 1	-						
GetScopes	Device	Optional	Discovery				Moved to 1	-						
SetScopes	Device	Optional	Discovery				Moved to 1	-						
AddScopes	Device	Optional	Discovery				Moved to 1	-						
RemoveScopes	Device	Optional	Discovery				Moved to 1	-						
GetNetworkInterfaces	Device	Mandatory	Network Configuration	x			Moved to 1							
SetNetworkInterfaces	Device	Mandatory	Network Configuration	x			Moved to 1							
GetNetworkDefaultGateway	Device	Mandatory	Network Configuration	x			Moved to 1							
SetNetworkDefaultGateway	Device	Mandatory	Network Configuration	x			Moved to 1							
GetHostname	Device	Optional	Network Configuration	x			Moved to 1							
SetHostname	Device	Optional	Network Configuration	x			Moved to 1							
GetDNS	Device	Optional	Network Configuration	x			Moved to 1							
SetDNS	Device	Optional	Network Configuration	x			Moved to 1							
GetNetworkProtocols	Device	Optional	Network Configuration	x			Moved to 1							
SetNetworkProtocols	Device	Optional	Network Configuration	x			Moved to 1							
GetDeviceInformation	Device	Mandatory	System				Moved to 1	-						
GetSystemDateAndTime	Device	Optional	System	/			Moved to 1	-						
SetSystemDateAndTime	Device	Optional	System				Moved to 1	-						
SetSystemFactoryDefault	Device	Optional	System				Moved to 1	-						
Reboot	Device	Optional	System				Moved to 1	-						
GetUsers	Device	Mandatory	User handling	x			Moved to 1							
CreateUsers	Device	Mandatory	User handling	x			Moved to 1							
SetUser	Device	Mandatory	User handling	x			Moved to 1							
DeleteUsers	Device	Mandatory	User handling	x			Moved to 1							
Notify	Event	Mandatory*	Event handling				Moved to 1							
Subscribe	Event		Event handling				Moved to 1							
Renew	Event		Event handling				Moved to 1							
Unsubscribe	Event		Event handling				Optional							
SetSynchronizationPoint	Event	Mandatory*	Event handling	x			Optional							
CreatePullPointSubscription	Event		Event handling	x			Moved to 1							
PullMessage	Event		Event handling	x			Moved to 1							

GetEventProperties	Event	Optional	Event handling				Moved to 1						
TopicFilter	Event	Optional	Event handling	Filters to select what events to retrieve			Moved to 1						
MessageContentFilter	Event	Optional	Event handling				Moved to 1						
GetProfiles	Media	Mandatory	Media Profile Configuration				Moved to 1	Media2.wsdl					
GetProfile	Media	Mandatory	Media Profile Configuration				Optional	-					
CreateProfile	Media	Mandatory	Media Profile Configuration				Moved to 1	Media2.wsdl					
DeleteProfile	Media	Optional	Media Profile Configuration				Moved to 1	Media2.wsdl					
GetVideoSources	Media	Optional	Video Source Configuration				Moved to 1	-					
GetVideoSourceConfiguration	Media	Mandatory	Video Source Configuration				Optional	-					
GetVideoSourceConfigurations	Media	Mandatory	Video Source Configuration				Optional	Media2.wsdl					
AddVideoSourceConfigurations	Media	Mandatory	Video Source Configuration				Moved to 1	-					
RemoveVideoSourceConfiguration	Media	Optional	Video Source Configuration				Moved to 1	-					
SetVideoSourceConfiguration	Media	Mandatory	Video Source Configuration				Moved to 1	Media2.wsdl					
GetCompatibleVideoSourceConfigurations	Media	Mandatory	Video Source Configuration				Moved to 1	-					
GetVideoSourceConfigurationOptions	Media	Mandatory	Video Source Configuration				Moved to 1	Media2.wsdl					
GetMetadataConfiguration	Media	Mandatory	Metadata configuration	?			Optional	-					
GetMetadataConfigurations	Media	Mandatory	Metadata configuration	?			Optional	Media2.wsdl					
AddMetadataConfiguration	Media	Optional	Metadata configuration	?			Moved to 1	-					
RemoveMetadataConfiguration	Media	Optional	Metadata configuration	?			Moved to 1	-					
SetMetadataConfiguration	Media	Mandatory	Metadata configuration	?			Moved to 1	Media2.wsdl					
GetCompatibleMetadataConfigurations	Media	Optional	Metadata configuration	?			Moved to 1	Media2.wsdl					
GetMetadataConfigurationOptions	Media	Mandatory	Metadata configuration	?			Moved to 1	-					
				Research if metadata contains interesting data (VCA)									

ONVIF Services



Sense classes

Testplan



ONVIF Plug-in

Author: Tim Kesteloo

Version: 1.0

Date: 25/03/2016

1. Introduction

This document describes the process used for testing the Onvif plugin.

To ensure the quality of the product, we will be performing tests both during development as well as after development is done. ...

2. Contents

[1. Introduction](#)

[2. Contents](#)

[3. Changelog](#)

[4. Testing during development](#)

[4.1. Profile S Compliance Tests](#)

[4.2. Sense tests](#)

[4.3. Conclusion](#)

3. Changelog

Version	Changes
v1.0	First release Does not contain a specific planning for the Testing after Development test cases.

4. Testing during development

Multiple different tests will be performed during the development cycle in order to preserve the quality of the code.

Each sprint is completed with a Code Review. During the code review, all changed code will be checked for problems or irregularities. If any of these arise, they can be fixed in the next sprint.

We also want to test the functionality of the plugin. This is divided into two aspects:

- The plugin is ONVIF Profile S Compliant for the chosen set of features for that sprint.
- The plugin can be used by Sense for the chosen set of features for that sprint.

It is important to know that the sets named here differ from each other. For example, Profile S compliance could force us to implement a setter functionality that Sense will never call to.

The current architecture splits the communication layer (currently implemented with gSoap) from the feature layer (the functions that Sense relies on). This gives us the ability to separate the way each of these are tested.

4.1. Profile S Compliance Tests

ONVIF has provided the Onvif Client Test Tool. This program can analyze a network-trace logfile and check which functions have been called to a device, if the function call has been formulated in a legal manner, if the client responded, and if the client responded in a legal manner.

The gSoap implementation that is being used during the graduation internship will implement all the mandatory functions and some conditional that are needed for Onvif Profile S Compliance.

However, since the feature implementation won't have to use all of these functions, a system was created that could call to all these functions. This can be activated, which would result in every device that is configured in Sense to call all its ONVIF Profile S mandatory functions.

An external program can now capture the network packages, which can then be analyzed by the Onvif Client Test Tool. We have chosen to use Wireshark for capturing the network traffic, as it has become a very reliable and complete program for exactly this purpose.

4.2. Sense tests

Testing the plugin and how it's used by Sense has to be approached in a different way. Even though its function will rely on the same principles (whether the soap messages are formatted correctly and get the expected results), we now also have to translate them to information that Sense expects from us. This can be tested in two ways:

- Manual testing
- Unit testing

The most effective test will be manually testing; test if all the buttons in Sense are working and result in functioning camera control. This can be very time consuming when this has to be done after each sprint, but will directly show how the end user will communicate with the plugin.

Another option is writing unit tests. These would be written for the feature implementation, since this layer will connect the plugin-architecture to the soap communication. This would mean that actual communication layer messages are not called and must be mocked. This will take a lot of time to set up, but would result in a automated way to test the functionality of each feature's functions. However, this might not represent the actual behaviour of Sense.

It's possible to do either or both of these methods. Since doing both would result in the optimal test coverage, we have to think about the scope and time constraints of the project.

4.3. Conclusion

For testing Onvif Profile S Compliance, we've made a system that can call all ONVIF Profile S Mandatory functions. These are captured by Wireshark, which will then be fed into the Onvif Client Test Tool. This will show if the functions are being called and responded to in a correct manner.

For testing the plugin usage by Sense, we will be using manual tests. These are the most reliable way to show if a functionality actually works in Sense. Having Unit Tests in addition to the manual tests would be optimal, but since this will take a lot of time to set up, this will not be done during the development phase. Whether it will be done in the final Test Sprint depends on the state of the application at that time and how much time is leftover.

5. Testing after development

After development is done, we have planned a final sprint for testing. At this point, all the separate functionalities should have been implemented and tested individually. The goal of this sprint is to make sure that all functionalities that were developed during the previous sprints are functioning correctly in unison for bigger test cases.

If we run into problems during this period, the goal is also to fix these problems, so the application will function as planned.

Situations that could be tested are:

- Stress Test / Load Test: What happens if we let the plugin control a lot of cameras? Can the plugin handle such a high amount? And what about a very low (0) amount?
- Stability Test: What happens if the plugin runs for a week (relatively small range, but time constraint)? Does everything still work as expected?
 - And what if the camera disconnects but the plugin still runs? Will everything continue normally when it reconnects?

A specific plan for what should be tested is not available at this time. The next version of this document will contain a planning with the particular tests that will be run and how they are performed.

6. Testcases

6.1. ONVIF features by sprint

A sprint-planning was made that defines which ONVIF features will be implemented during each sprint. At the end of each sprint, these functionalities are tested by the Onvif Client Test Tool. The PASS and FAIL conditions for each of these functions can be found in the [ONVIF Profile S Client Test Specification](#).

Onvif Features implemented by Sprint			
Sprint 4 - DeviceService and start MediaService	Sprint 5 - MediaService	Sprint 6 - PTZ	Sprint 7 - Leftovers
Capabilities: - GetCapabilities System: - GetDeviceInformation - GetSystemDateAndTime Relay Outputs: - GetRelayOutputs - SetRelayOutputSettings - SetRelayOutputState Video Streaming: - GetProfiles - GetStreamUri - GetVideoEncoder Configuration - GetVideoEncoder Configurations - SetVideoEncoder Configuration - GetVideoEncoder ConfigurationOptions User Authentication: - WS-UsernameToken Authentication - HTTP Digest	Video Streaming: (functionality on serverside) - 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: (functionality on serverside) - 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 - AddVideoSource Configuration NTP: - GetNTP - SetNTP

6.2. Sense features by sprint

There are multiple milestones that we want to achieve each sprint for the plugin. These are in line with the Onvif features that we want to implement each sprint.

However, because we weren't able to test some functionalities in Sense at the start of the project, some early feature tests have been moved to later sprints.

Sense Features implemented by Sprint			
Sprint 4 - DeviceService and start MediaService	Sprint 5 - MediaService	Sprint 6 - PTZ	Sprint 7 - Leftovers
GPIO: Sense can enable and disable outputs. Device: Sense can request the device information and show it in the UI.	Media: Sense can record (and show) video and audio from a device. Sense can start and stop a multicasts stream. Discovery: Sense can discover ONVIF devices.	PTZ: Sense can move a camera. Sense can tell a camera to go to preset X. Sense can call an auxiliary command.	Event Handling: Sense can receive and interpret an event from the device.

Many of the ONVIF functionalities that are needed for Profile S compliance are not used by Sense (yet). Although many of them are used implicitly, it is possible that Sense wants to be able to directly call the functionalities in the future. However,

Testreport



ONVIF Plug-in

Author: Tim Kesteloo

Version: 1.0

Date: 13/05/2016

Introduction

This document is the testreport for the ONVIF plugin. This shows results of the developmentstages for each sprint and the endproduct that was delivered.

In order to learn more about how these test were performed, you can read the Testplan. This document will not elaborate how the testing was done, only what was supposed to be created during the sprint and explain the results and consequences.

Changelog

Version	Changes
v1.0	First release

Contents

[Introduction](#)

[Changelog](#)

[Contents](#)

[Testplanning](#)

[Sprint 3: Development DeviceService](#)

[Goal](#)

[Results](#)

[Sprint 4: MediaService](#)

[Goal](#)

[Results](#)

[Sprint 5: PTZService](#)

[Goal](#)

[Results](#)

[Sprint 6: Events & Discovery](#)

[Goal](#)

[Results](#)

[Sprint 7: Test Sprint](#)

[Goal](#)

[Results](#)

Testplanning

In the Testplan, we describe what will be created, and thus should be tested.

Each sprint results in the testing of these components through the ONVIF Client Test Tool.

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 - GetVideoEncoder Configurations - 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 - AddVideoSource Configuration NTP: - GetNTP - SetNTP

ONVIF Feature planning by sprint

Sprint 3: Development DeviceService

Goal

Capabilities:

- GetCapabilities

System:

- GetDeviceInformation
- GetSystemDateAndTime

Relay Outputs:

- GetRelayOutputs
- SetRelayOutputSettings
- SetRelayOutputState

Video Streaming:

- GetProfiles
- GetStreamUri
- GetVideoEncoderConfiguration
- GetVideoEncoderConfigurations
- SetVideoEncoderConfiguration
- GetVideoEncoderConfigurationOptions

User Authentication:

- WS-UsernameToken Authentication
- HTTP Digest

Results

HTTP Digest was not completed. However, the test tool seems to approve of this and flags Security as SUPPORTED.

GetServices has never been spoken of in the Profile S Compliance documents. Later research has told us that this is indeed a new feature which is not mandatory for profile S, but is the new way to check for capabilities on a camera. However, this was unknown at this stage. Capabilities shows SUPPORTED.

GetDeviceInformation SUPPORTED

GetSystemDateAndTime SUPPORTED, though not shown in the testreport. However, this information is used to calculate the nonce for the WS-UserNameToken.

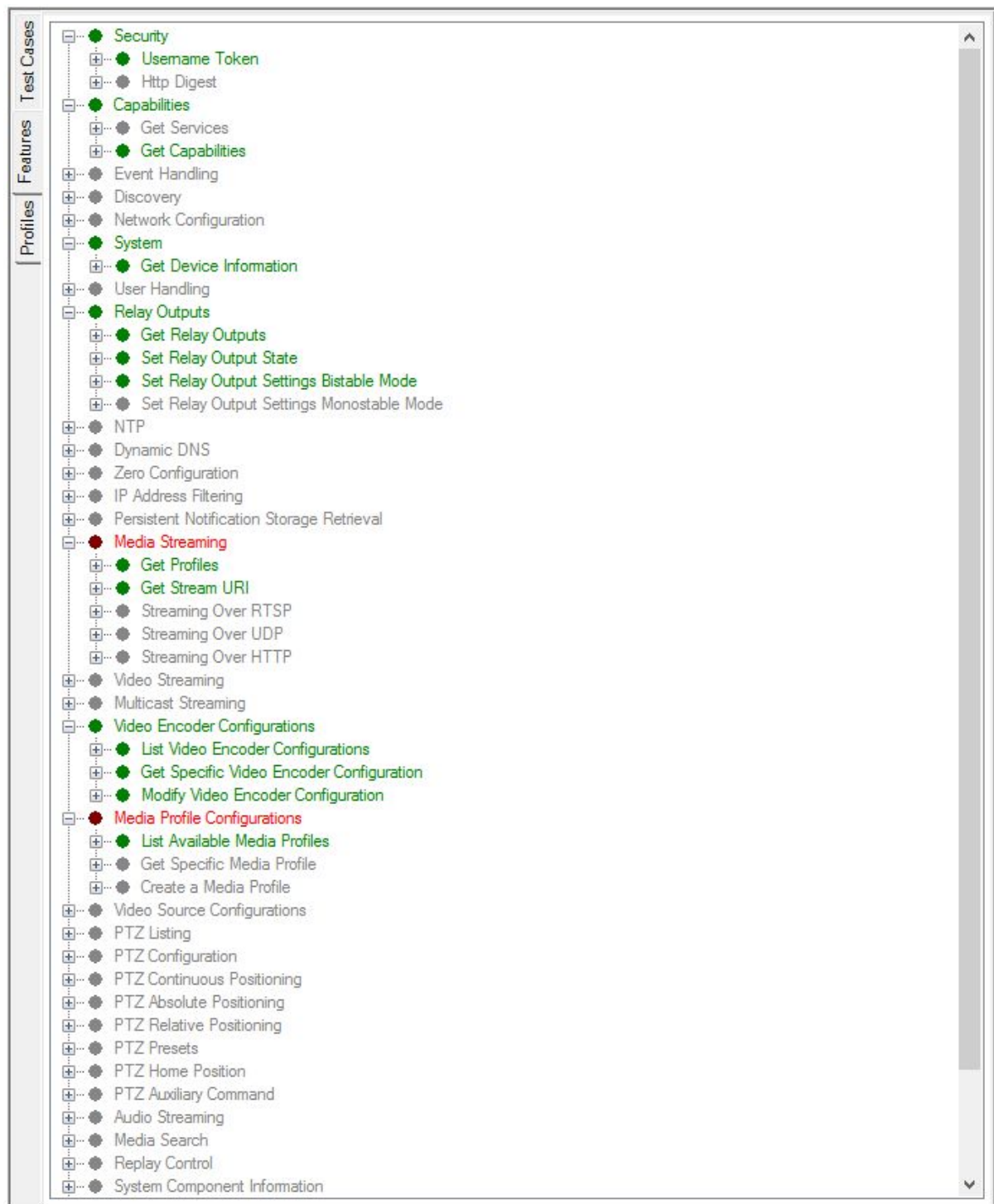
Relay Outputs SUPPORTED

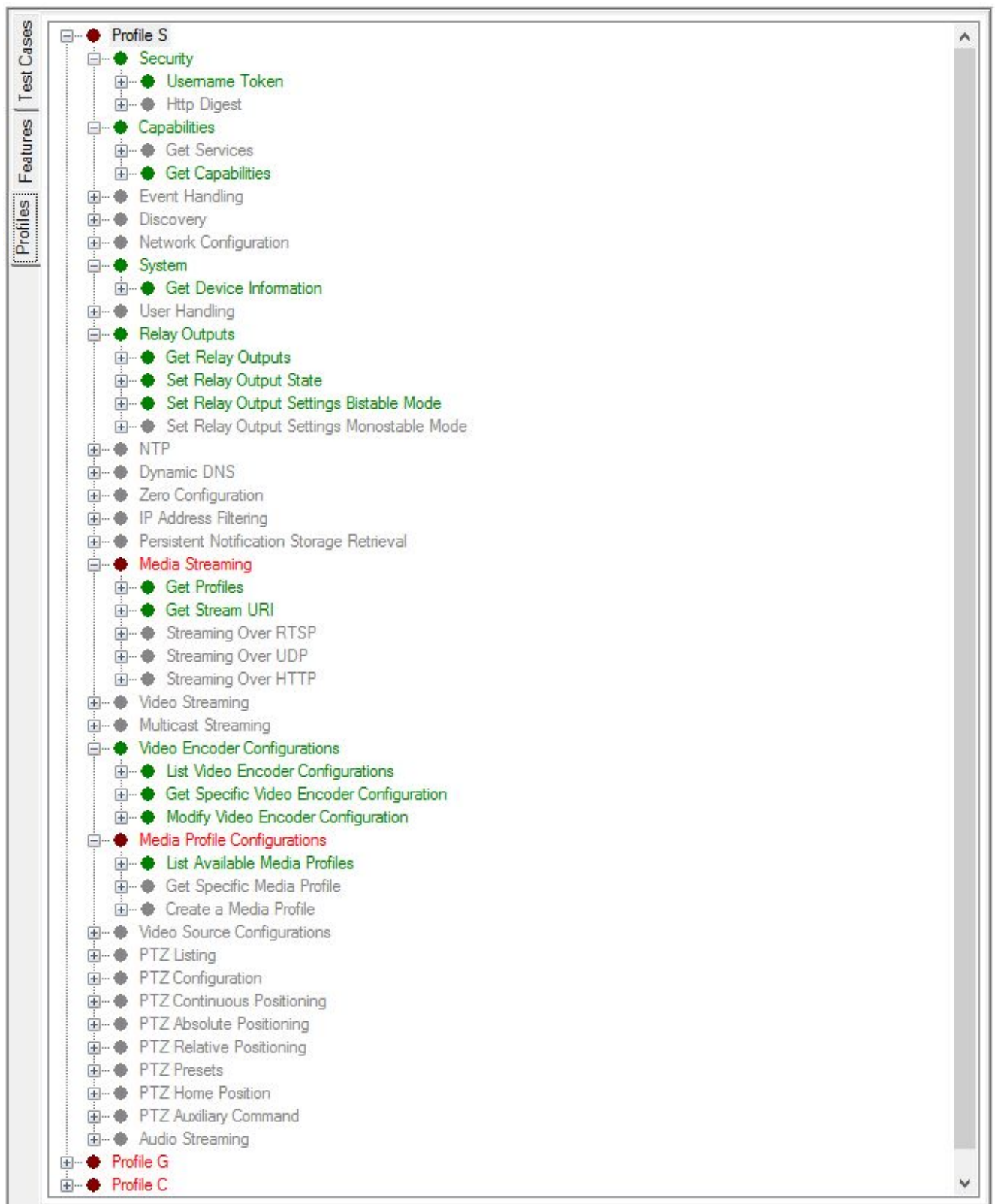
Set Relay Output Settings Monostable Mode could not be tested at this stage with the camera that was being used.

Media Streaming GetProfiles & GetStreamUri SUPPORTED. However, GetStreamUri had a small testcase and is not done yet. This will be done in the next sprint (MediaService)

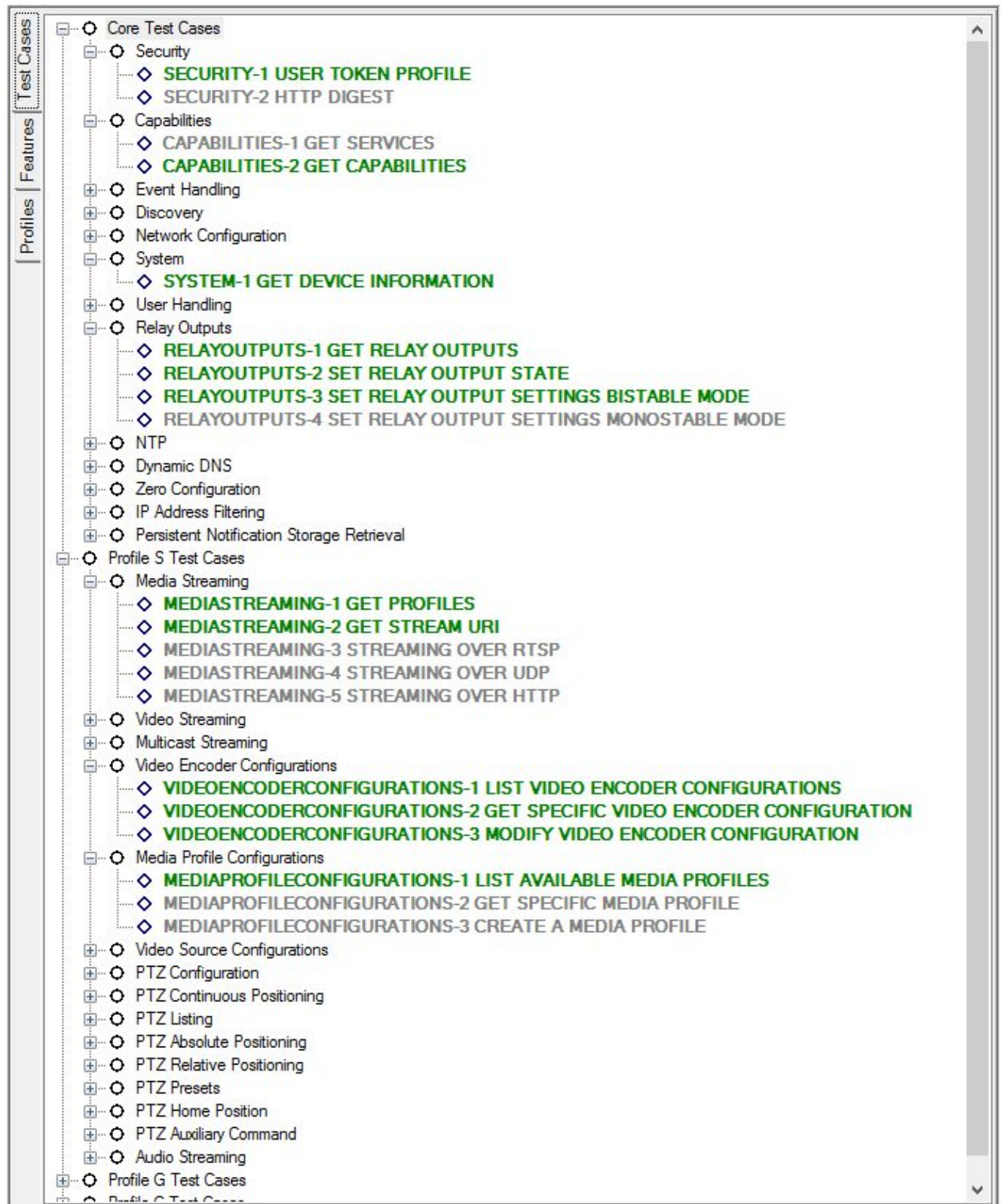
Video Encoder Configurations SUPPORTED. Since we wanted to get a list of the encoders to check statistics, we were forced to also implement the Modify. At this time, we validate this by re-setting the current Encoder Configuration. The next sprint (MediaService) will expand this testcase.

Media Profile Configurations NOT SUPPORTED. We want to request the Media Profiles, but this forces us to implement the Create MediaProfile method. This test must still be made.





Sprint 3 DeviceService - Profiles



Sprint 3 DeviceService - Test Cases

Sprint 4: MediaService

Goal

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

Results

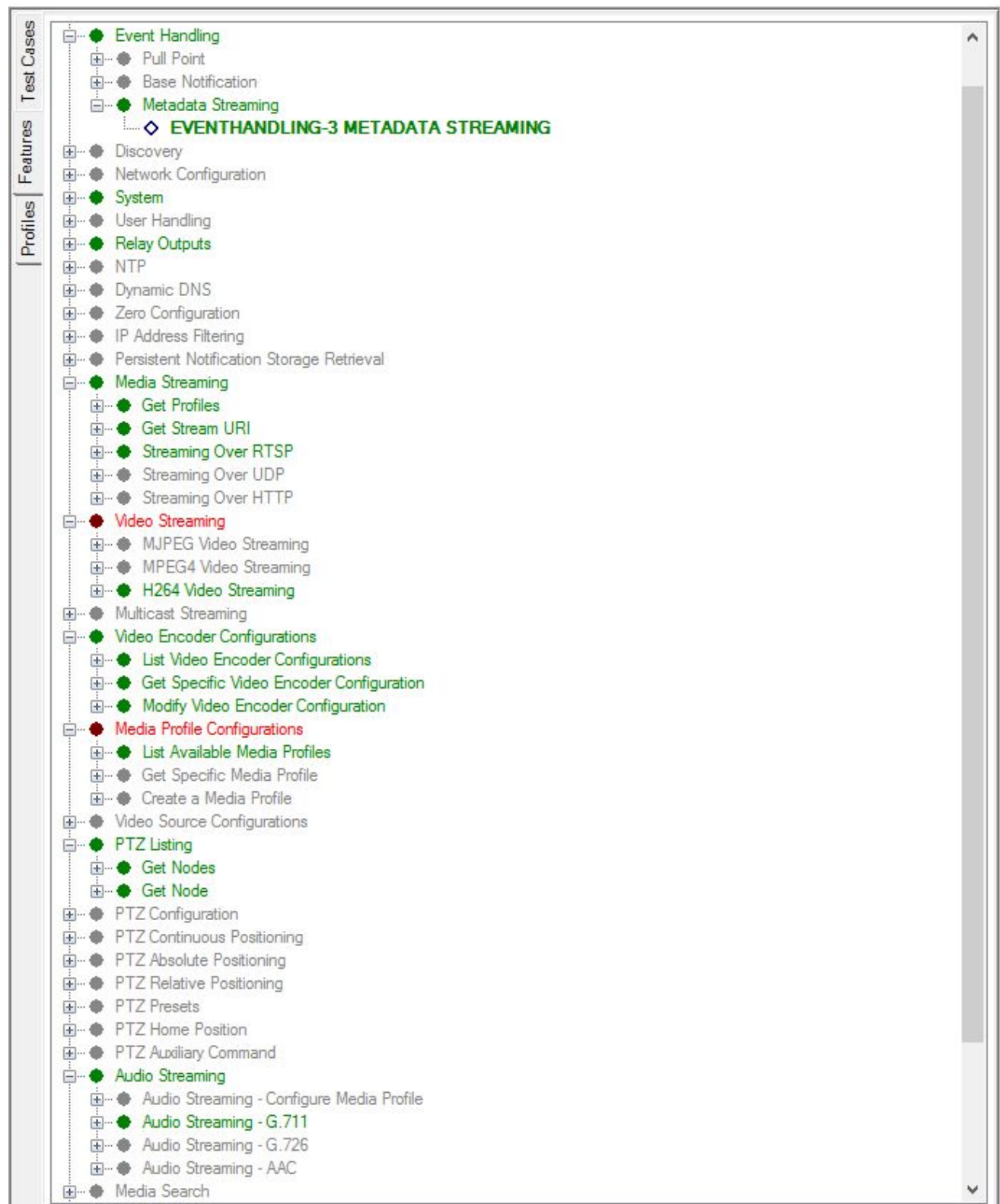
Media Streaming was SUPPORTED. The testcase for GetStreamUri was expanded and still works. An implementation was made in Sense to start an RTSP stream on this Uri, which worked and resulted in camera footage in Sense.

Video Streaming is not fully SUPPORTED yet. We haven't been able to test cameras configured for different video encoders yet. For now, only the H264 stream works, which is not enough to satisfy the entire testcase for Video Streaming since the minimum is MJPEG streaming. When Sense supports streaming an MJPEG stream, it's possible to configure a camera and retrieve the StreamUri for this MJPEG stream.

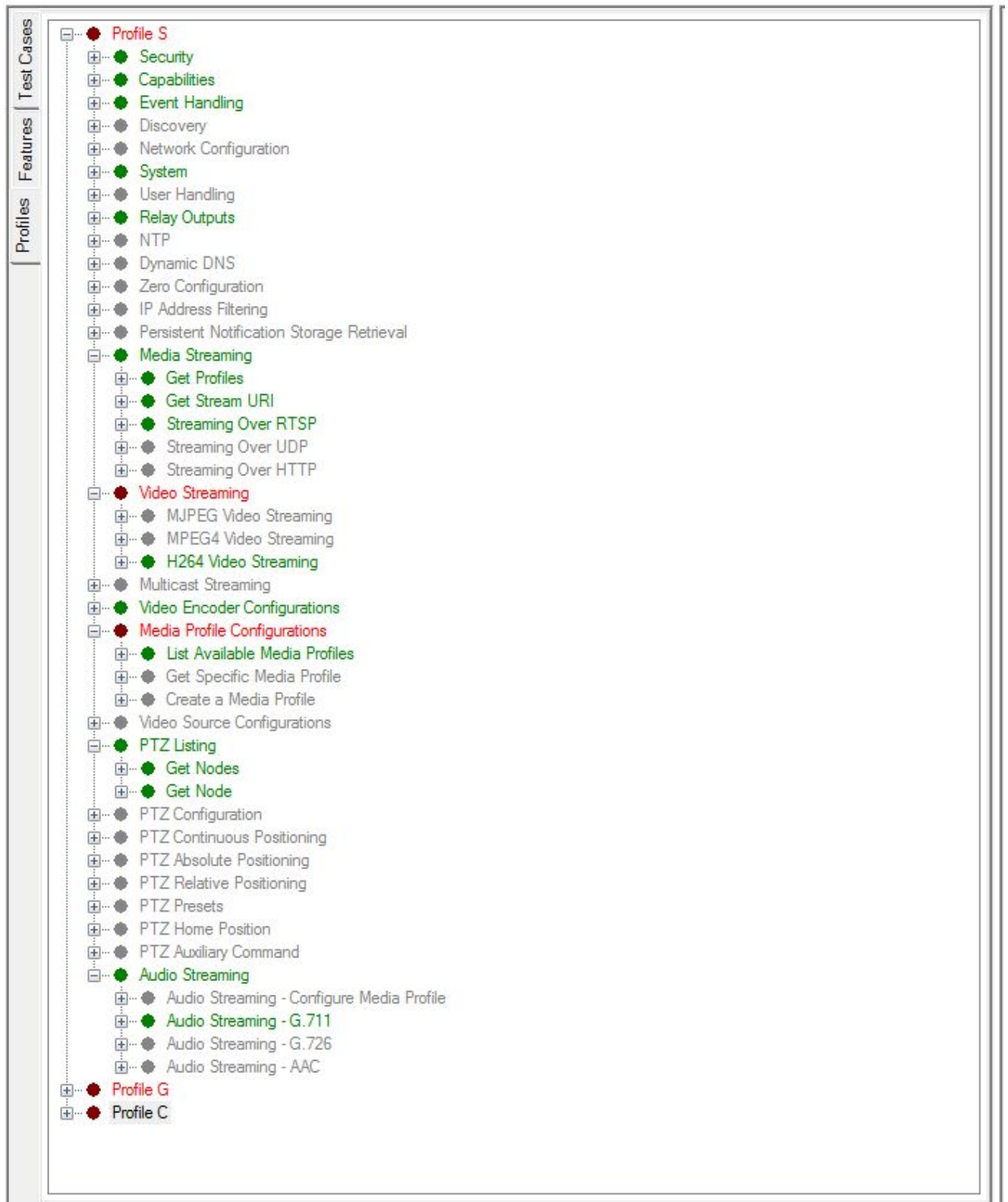
Audio Streaming SUPPORTED. Using the RTSP stream, we can easily receive the G.711 audio as well. This is the minimum and we don't have to support any other protocols.

Discovery was not implemented yet because of planning issues during this sprint. It was moved to the final developmentsprint (6).

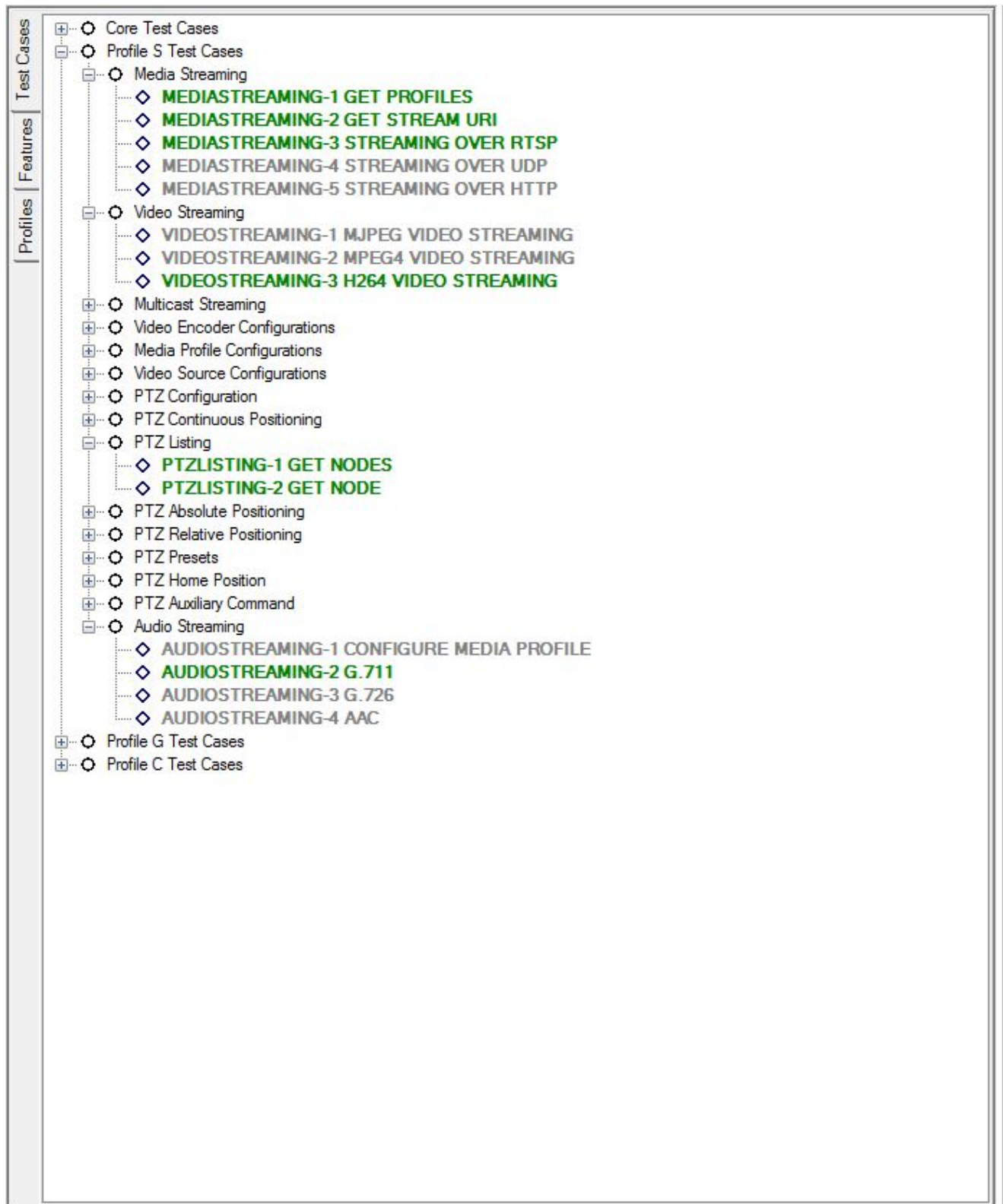
PTZ was started in this sprint, where the first get functions have been implemented. The next sprint (PTZService) will expand on this.



Sprint 4 MediaService - Features



Sprint 4 MediaService - Profiles



Sprint 4 MediaService - Test Cases

Sprint 5: PTZService

Goal

PTZ:

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

Presets:

- GetPresets
- GotoPresets
- RemovePreset

Home:

- GotoHomePosition
- SetHomePosition

Auxiliary:

- SendAuxiliaryCommand

Results

At this stage, it was decided that Sense implementation would be postponed. Because of this, we're unable to properly test any Media- or Video Streaming tests. These will be ignored from now on until VDG resumes development of this.

Relay Outputs suddenly fails on the new PTZ camera. This test will be fixed in the final test sprint when all development is finished.

PTZ ContinuousMove SUPPORTED. Multiple testcases were made for just Pan-Tilt, just Zoom, and combined PTZ. The same goes for Stop Pan-Tilt, Stop Zoom, and Stop PTZ.

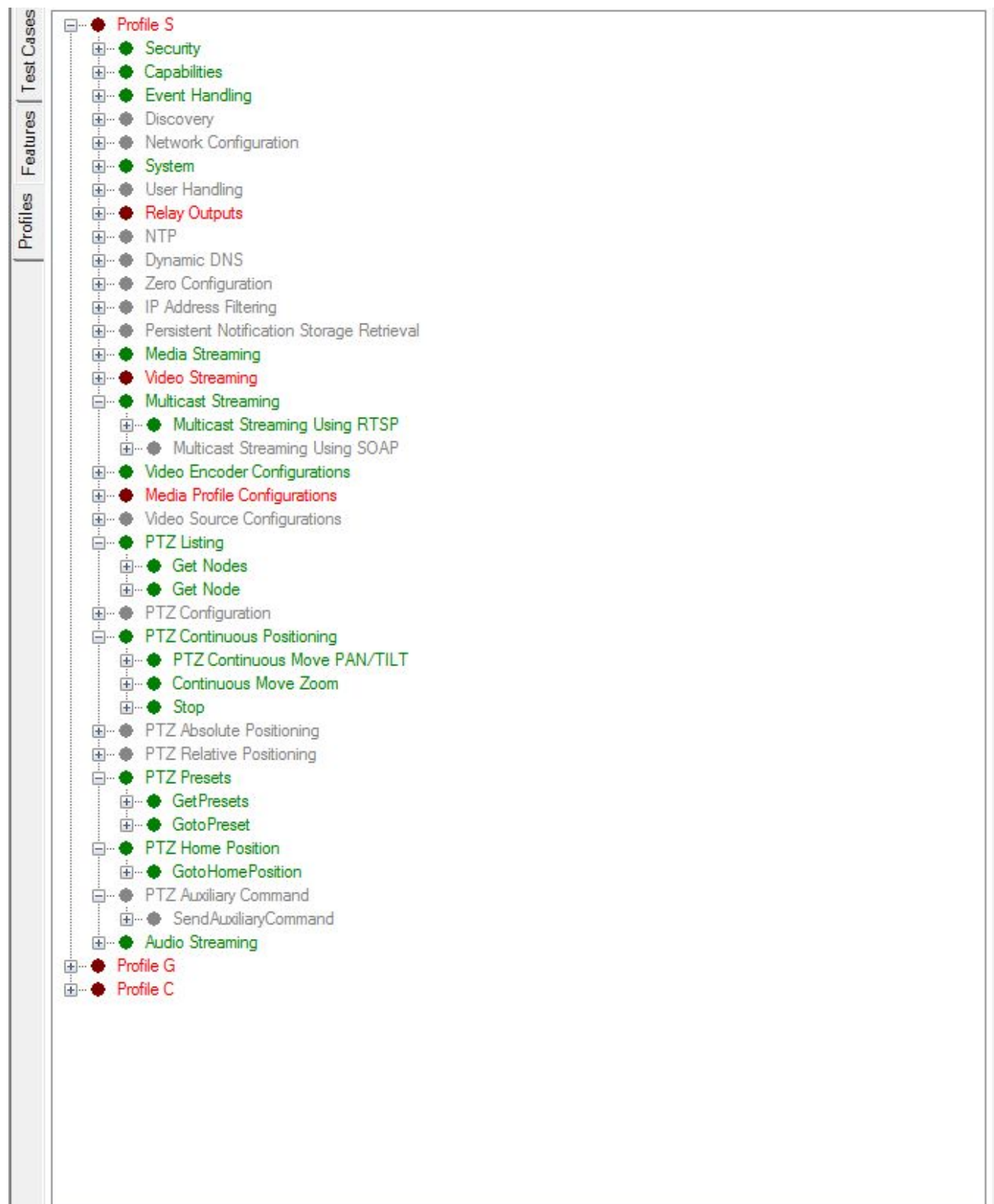
PTZ Presets SUPPORTED. Code for creating a preset was also tested, but is not mandatory for Profile S.

PTZ Home SUPPORTED. Code for setting the Home position was also tested, but not mandatory for Profile S.

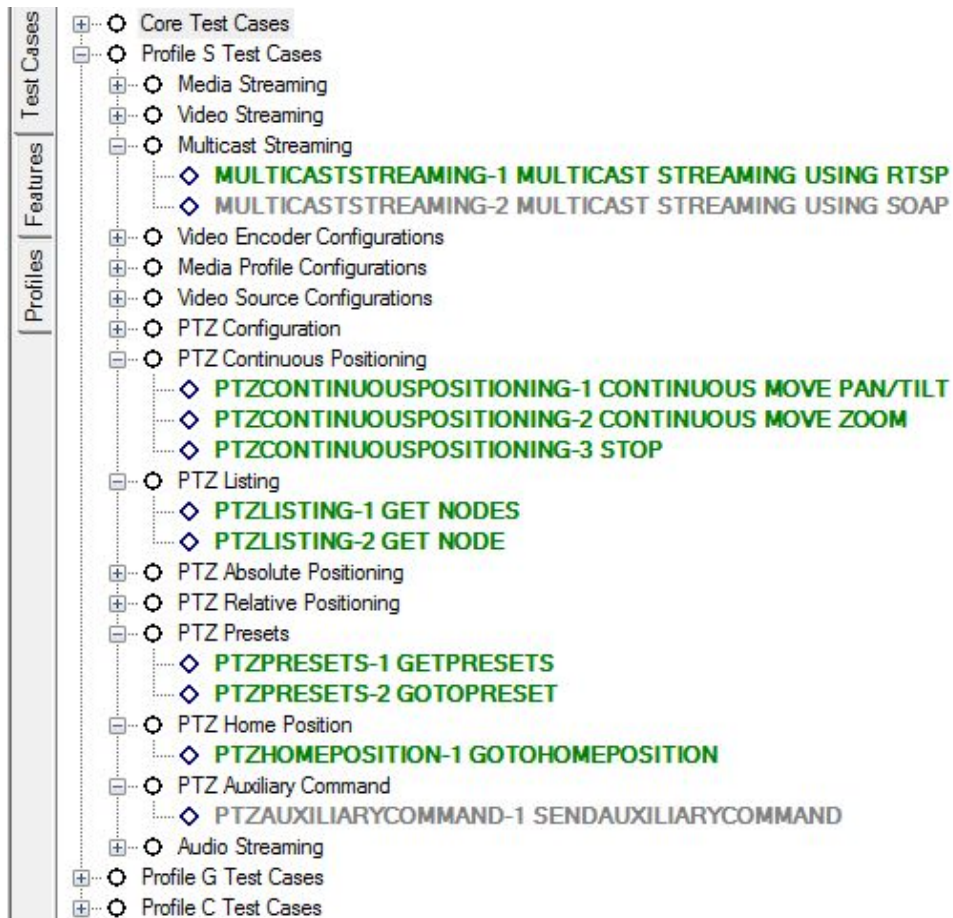
PTZ Auxiliary could not be tested, since the PTZ camera used does not have any commands configured. New cameras will be added to the testset which do have commands configured, so this code can be tested.

Profiles	Features	Test Cases
	- Security - Capabilities - Event Handling - Discovery - Network Configuration - System - User Handling - Relay Outputs - NTP - Dynamic DNS - Zero Configuration - IP Address Filtering - Persistent Notification Storage Retrieval - Media Streaming - Video Streaming - Multicast Streaming - Multicast Streaming Using RTSP - Multicast Streaming Using SOAP - Video Encoder Configurations - Media Profile Configurations - Video Source Configurations - PTZ Listing - Get Nodes - Get Node - PTZ Configuration - PTZ Continuous Positioning - PTZ Continuous Move PAN/TILT - Continuous Move Zoom - Stop - PTZ Absolute Positioning - PTZ Relative Positioning - PTZ Presets - GetPresets - GotoPreset - PTZ Home Position - GotoHomePosition - PTZ Auxiliary Command - SendAuxiliaryCommand - Audio Streaming - Media Search - Replay Control - System Component Information - System Component State - Door Control - Access Points Control - External Authorization	

Sprint 5 PTZService - Features



Sprint 5 PTZService - Profiles



Sprint 5 PTZService - Test Cases

Sprint 6: Events & Discovery

Goal

Event Handling:

- Notify
- Subscribe
- Renew

Media Profile Configuration:

- GetProfiles
- CreateProfile

Video Source Configuration:

- GetCompatibleVideoSourceConfigurations
- GetVideoSourceConfigurationOptions
- SetVideoSourceConfiguration
- AddVideoSourceConfiguration

NTP:

- GetNTP
- SetNTP

Media Profile Configuration and **Video Source Configuration** were scrapped, and thus not tested. **Discovery** is to be implemented with the highest priority. After that, **Event Handling** and **NTP**.

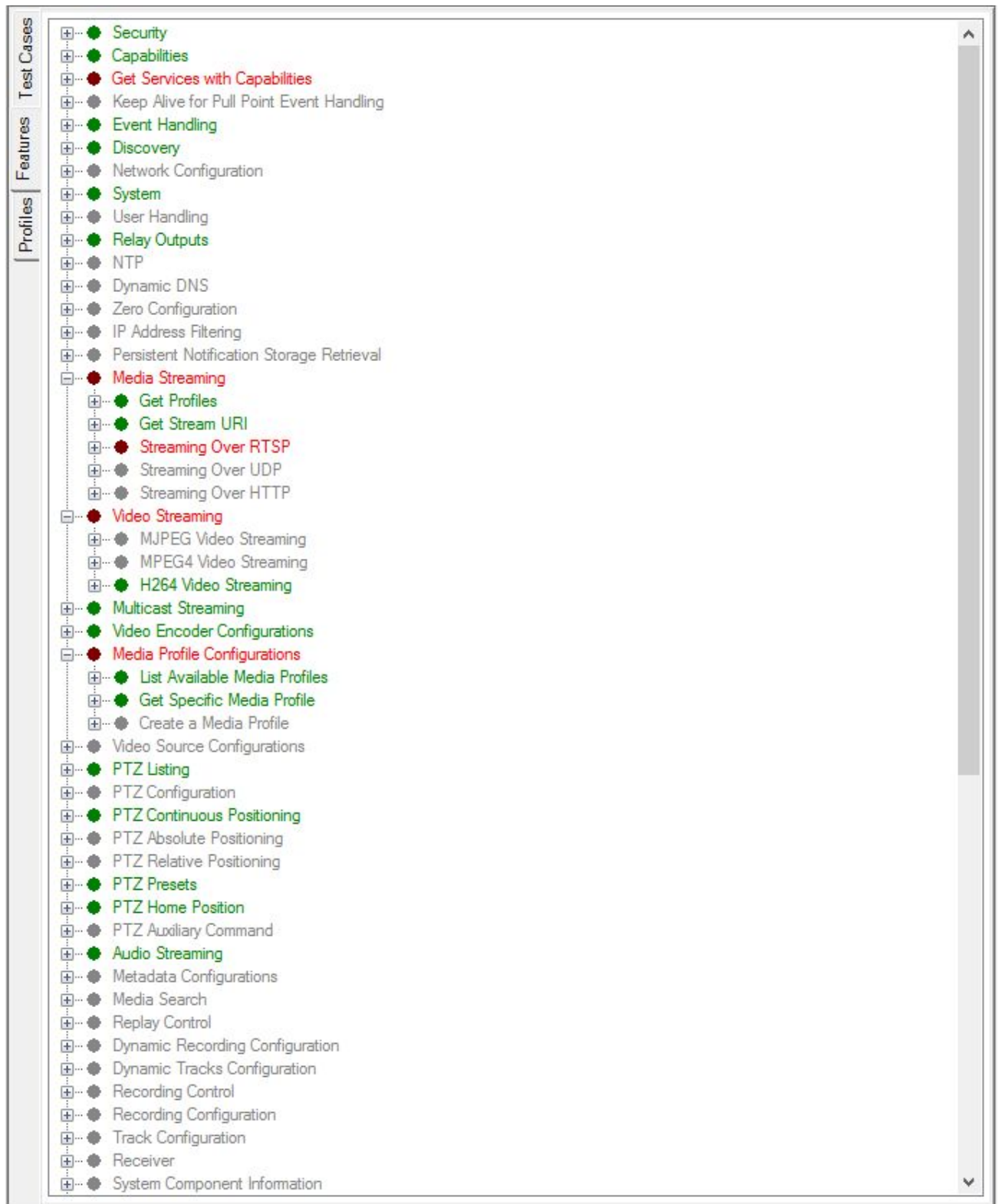
Results

A new Test Tool (ONVIF Client Test Tool v16.01) was released and thus must be used in order to test compliance.

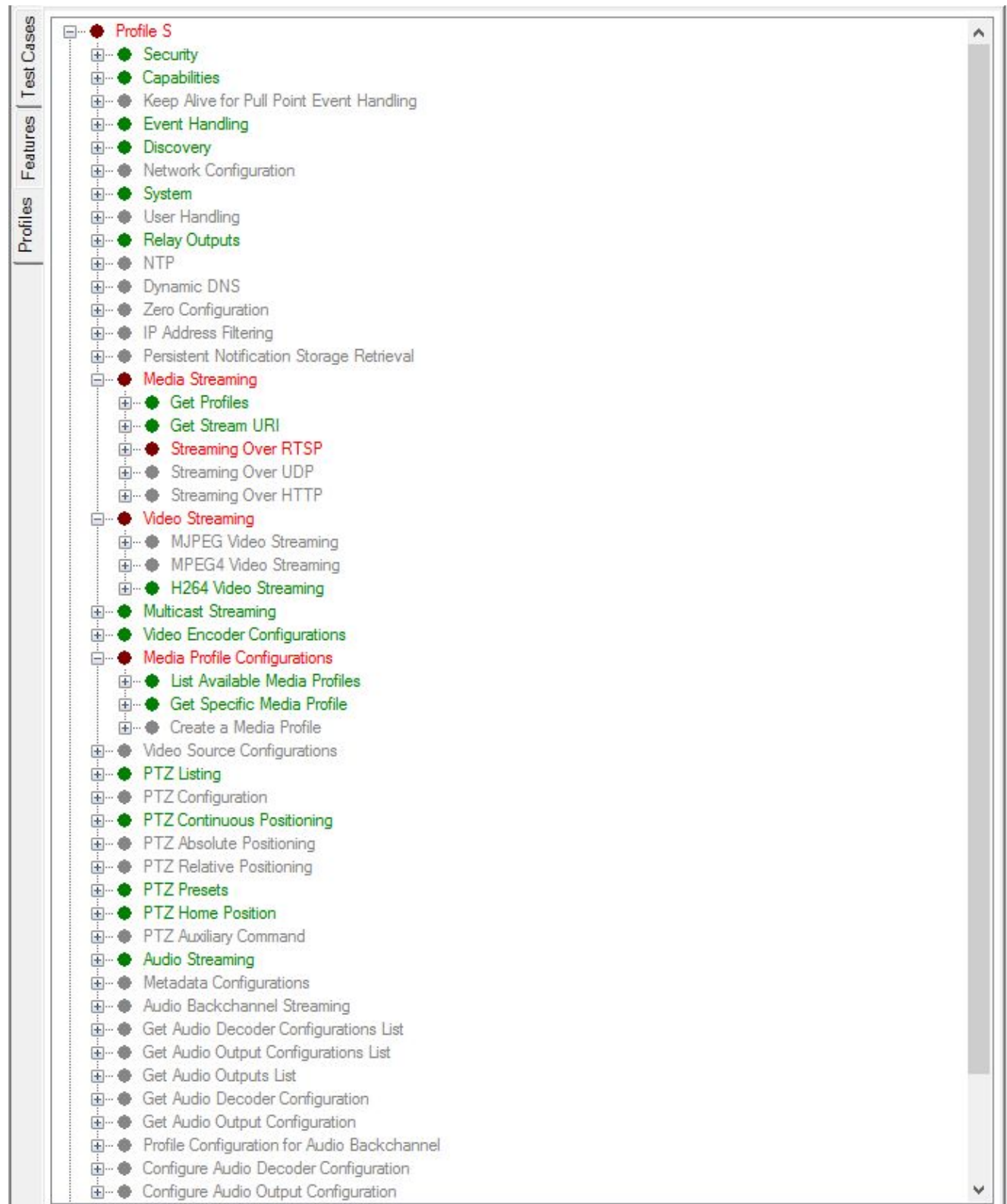
Discovery was implemented through the WS-Discovery protocol. Devices can be found and returned to Sense in realtime.

Events NOT SUPPORTED. Though it is shown as SUPPORTED in the picture, this has to do with **Metadata Streaming**, and does not include the events, as those are not mandatory for Profile S Compliance.

NTP NOT SUPPORTED. Code was created for this, but has not been tested yet as it is still unclear how VDG wants to use this feature. If VDG wants to, the code is ready to be used.



Sprint 6 Events & Discovery - Features



Sprint 6 Events & Discovery - Profiles

Sprint 7: Test Sprint

Goal

The goal of this sprint, as described more accurately in the test plan, was to test the plugin through the use of the VDG Plugin Test Tool. Simultaneously, we would trace the webtraffic and feed this to the ONVIF Client Test Tool to test the compliance while testing the architecture.

However, the VDG Plugin Test Tool was not completed and could not be used.

During this sprint, small bugs were fixed and testcases have been changed to more accurately test the features.

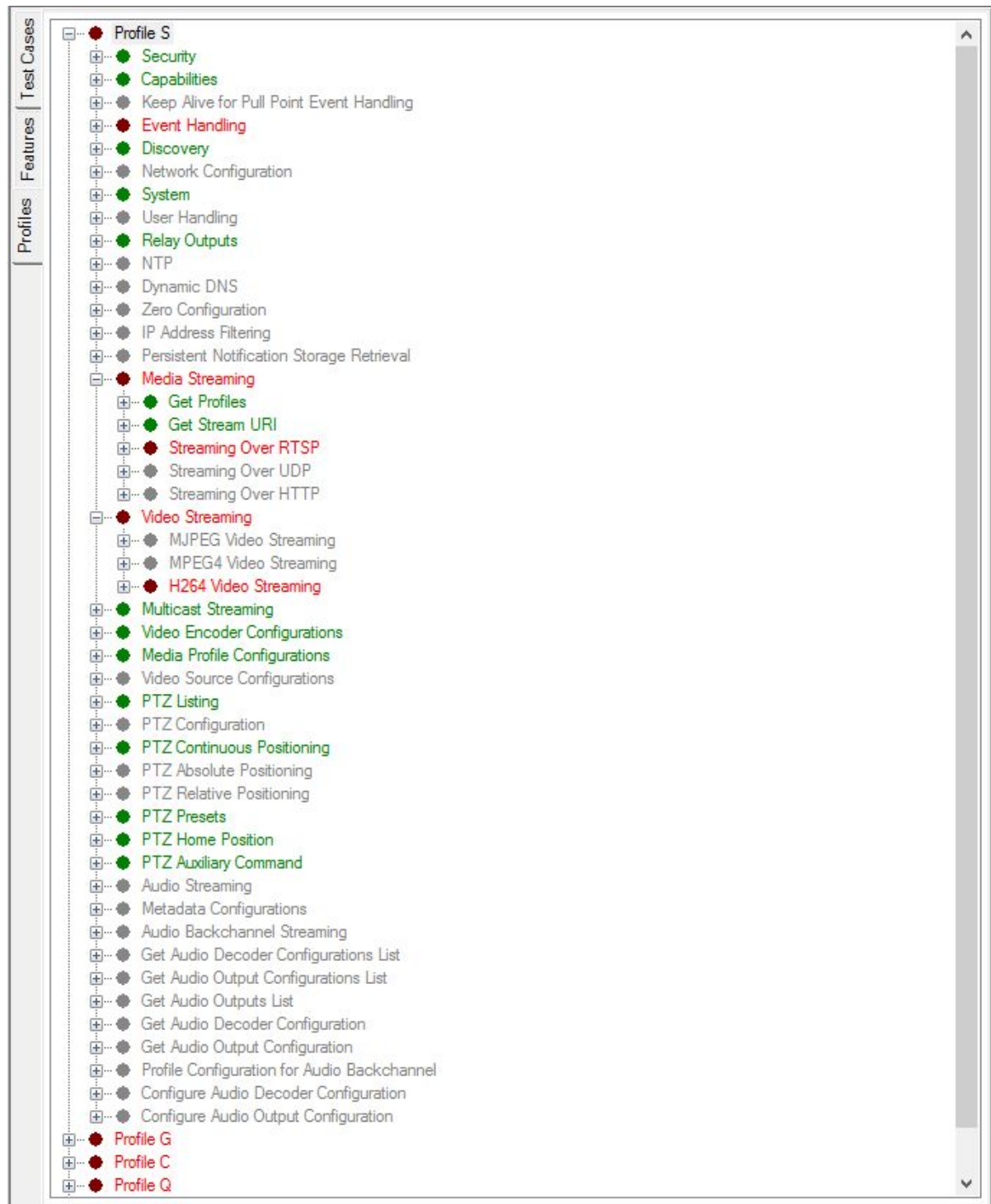
A final test shows the endresult of the Test Sprint.

Results

PTZ Auxiliary Commands SUPPORTED. The new testset contains cameras that do have Auxiliary Commands configured.

The **Metadata**-, **Media**-, and **VideoStreaming** tests broke down, since this would be the responsibility of the VDG Plugin Test Tool.

Because of the expansion of the camera testset, multiple older testcases failed, which either had to do with how the testcase was made or with how the ONVIF call was made. These have all fixed, so all cameras in the testcase are now SUPPORTED for their features, with the exception of streaming.



Sprint 7 Testing - Profiles

Bespreking concept	Tussentijds assessment	Eerste beoordeling
---------------------------	-------------------------------	---------------------------

Formulier bespreking concept afstudeerdossier

Student: Tim Kesteloo

Studentnummer: 11116552

Datum: 7 april 2016

Tijdens de bespreking is het volgende geconstateerd:		ja	nee
a	Het voortgangsverslag is ontvangen	X	
b	Het afstudeerdossier is digitaal beschikbaar	X	
c	Het afstudeerdossier is opgebouwd conform de richtlijnen	X	
d	Het goedgekeurde afstudeerplan is aanwezig		X
e	Het plan van aanpak is aanwezig	X	
f	Reeds geleverd commentaar is aanwezig	Nvt	
g	Het afstudeerdossier geeft voldoende inzicht in de stand van zaken	X	
h	De afstudeeropdracht is tot nu toe naar behoren uitgevoerd	X	

Verbeterpunten:

De opzet van het afstudeerverslag is goed en kan zo verder uitgewerkt worden. Geef meer toelichting op het plan van aanpak en op de resultaten van de sprints. Niet alleen de uiteindelijke resultaten zijn belangrijk, maar ook de tussenresultaten. Vooral de "foute" tussenresultaten die vervolgens tot betere inzichten geleid hebben.

Voeg een duidelijke beschrijving van de requirements toe.

Opmerkingen:

Tta voorlopig gepland op dinsdag 10/5.

Naam begeleidend examiner: J.J. van der Hoek

Datum: 8 april 2016

Dit formulier wordt door de begeleidend examiner digitaal ingevuld en per email naar de student verstuurd met een cc naar de coördinator van ICT & Media @ Work (A.M.Schipper@hhs.nl). Het formulier dient door de student te worden opgenomen in het afstudeerdossier.

Bespreking concept	Tussentijds assessment	Eerste beoordeling
--------------------	-------------------------------	--------------------

Formulier tussentijds assessment

Student: Tim Kesteloo

Studentnummer: 11116552

Datum: 10-5-2016

eerste / TTA: eerste

Tijdens het tussentijds assessment is het volgende geconstateerd:		ja	nee
a	Het voortgangsverslag is ontvangen	X	
b	Het afstudeerdossier is digitaal beschikbaar	X	
c	Het afstudeerdossier is opgebouwd conform de richtlijnen	X	
d	Het goedgekeurde afstudeerplan is aanwezig	X	
e	Het plan van aanpak is aanwezig	X	
f	Reeds geleverd commentaar is aanwezig		X
g	Het afstudeerdossier geeft voldoende inzicht in de stand van zaken	X	
h	De afstudeeropdracht is tot nu toe naar behoren uitgevoerd	x	

Aanpak	O	T	V	G
Passend			x	
Theoretisch verantwoord	x			
Samenhang uitvoering beroepstaken		x		

Beroepstaken op afgesproken niveau uitgevoerd?		O	T	V	G
1	H4 – Problemen analyseren		X		
2	C10 – Ontwerpen van een systeemarchitectuur		X		
3	D17 – Testen van een softwaresysteem		X		
4	J12 – Adviesproces uitvoeren		X		
5					
6					

7					
8					

Producten	O	T	V	G
<i>Tussenproducten</i>			x	
<i>Eindproducten</i>				

Effectief communiceren	O	T	V	G
<i>Binnen afstudeerbedrijf</i>				
<i>Afstudeerdossier</i>		x		

Reflectie	O	T	V	G
<i>Inzicht in eigen functioneren</i>			x	
<i>Inzicht in eigen leerproces</i>			x	

Toelichting per beoordelingscriterium

Aanpak
De aanpak sluit aan bij de standaarden binnen het bedrijf. Tegelijkertijd is de aanpak wel passend gemaakt aan de opdracht. Echter de verantwoording over de aanpassingen en gemaakte keuzes hierin zijn niet, of te summier beschreven, vandaar dat de beoordeling van de onderbouwing onvoldoende is. Er lijkt op basis van de producten een logische samenhang tussen de beroepstaken te zijn, maar ook hier is de onderbouwing onvoldoende

Beroepstaken op afgesproken niveau uitgevoerd?
Voor alle beroepstaken geldt dat het verslag voornamelijk focust op de (technische) resultaten, en dat de totstandkoming van die resultaten te weinig aan bod komt. Doordat dit procesmatige deel te zeer onderbelicht blijft kan er eigenlijk geen oordeel gevormd worden over de beroepstaak als zodanig. Een onvoldoende beoordeling zou dan ook te verantwoorden zijn, maar lijkt gezien de behaalde resultaten ook geen reëel beeld van de stand van zaken te geven.

Producten

Voor zover die in deze tussenversie van het verslag beschreven zijn, geven die de indruk dat er sprake is van doordacht werk.

Effectief communiceren

Er lijkt bij het afstudeerverslag nog niet het juiste beeld van de beoogde doelgroep te zijn. Taalkundig is het verslag goed leesbaar.

Reflectie

Op dit moment nog voornamelijk op basis van het gesprek, is de conclusie gerechtvaardigd dat de student voldoende inzicht heeft in zijn eigen leerproces.

Advies

X	Inleveren (bindend advies)
	Verlengen (vrijblijvend advies)
	Stoppen (vrijblijvend advies)

Besluit student

Aankruisen welke beslissing de student heeft genomen (alleen na vrijblijvend advies)

	Afstudeerdossier wordt op afgesproken datum ingeleverd Inleverdatum:
	Afstudeerperiode wordt verlengd Inleverdatum:
	Student stopt met afstudeeropdracht

Naam begeleidend examiner: J.J. van der Hoek

Naam tweede examiner: A. Andrioli

Datum: 10 – 5 – 2016

Dit formulier wordt door de tweede examiner digitaal ingevuld, waarna de begeleidend examiner het per email verstuurt naar de student met een cc naar de coördinator van ICT & Media @ Work (A.M.Schipper@hhs.nl). Het formulier dient door de student te worden opgenomen in het afstudeerdossier.

Evaluatieformulier afstuderen

In te vullen door opdrachtgever c.q. bedrijfsmentor(en)

Student: Tim Kesteloo

Periode: 25/01/2016 – 03/06/2016

Bedrijf c.q. instelling: VDG Security BV

Bedrijfsmentor: Robin Hermann

Plaats: Zoetermeer

Datum: 26-05-2016

- 1. Heeft de student zich zelf snel en goed ingewerkt in het bedrijf en de uit te voeren afstudeeropdracht?** Ja, de technische documentatie gebruikt tijdens het onderzoek waren snel doorgrond.
- 2. Hoe beoordeelt u de communicatieve vaardigheden van de student (in de samenwerking met collega's, in contacten met de opdrachtgever, bij mondelinge presentaties, schriftelijke rapportages)?** Goed, Tim geeft zelfstandig aan als er behoefte is aan overleg. Zijn Onvif presentatie aan het R&D team is als zeer duidelijk ontvangen.

3. Hoe heeft de student tijdens het uitvoeren van de opdracht gefunctioneerd?

• Qua verantwoordelijkheid	goed / voldoende / matig / onvoldoende
• Qua zelfstandigheid	goed / voldoende / matig / onvoldoende
• Qua planmatig werken	goed / voldoende / matig / onvoldoende
• Qua creativiteit	goed / voldoende / matig / onvoldoende
• Qua productiviteit	goed / voldoende / matig / onvoldoende
• Qua samenwerken met collega's	goed / voldoende / matig / onvoldoende
• Qua draagvlakontwikkeling	goed / voldoende / matig / onvoldoende
• Qua inspelen op bedrijfscultuur	goed / voldoende / matig / onvoldoende
• Qua rekening houden met de specifieke context van het bedrijf	goed / voldoende / matig / onvoldoende
• Qua het op gang brengen van de nodige veranderingen	goed / voldoende / matig / onvoldoende

4. **Hoe beoordeelt u de kennis en kunde van de student in verhouding tot wat u verwacht van een bijna afgestudeerde?** Goed, Tim heeft meer dan genoeg OOP kennis om in elke willekeurige taal te kunnen programmeren. Tim heeft zelfs een advies gegeven om een ontwerp wijziging door te voeren voor bepaalde classes. Hij heeft zich zeer leergierig opgesteld tijdens onze wekelijkse C++ afdeling sessies om ook de laatste C++ mogelijkheden te doorgronden.

5. **Hoe beoordeelt u de kwaliteit van de opgeleverde (tussen)producten?** Goed. De producten zijn ontworpen en geïmplementeerd volgens de VDG standaard..

6. Bent u tevreden over het opgeleverde (eind)product?

- **In hoeverre heeft u gekregen wat is afgesproken?** De opdracht is eigenlijk te groot om alle onderdelen te kunnen implementeren binnen de gestelde tijd. Dit hadden wij al ingeschat. Wij hebben zelf aangegeven welke onderdelen de hoogste prioriteit hebben, en deze onderdelen heeft Tim af kunnen ronden.
- **In hoeverre voldoet het (eind)product aan uw verwachtingen?** Volledig, het bewijs dat de nieuwe plugin architectuur aansluit op de Onvif architectuur stemt ons zeer gelukkig
- **Wat is de bruikbaarheid en onderhoudbaarheid hiervan?** Goed, het geheel is naar VDG standaarden ontworpen en geïmplementeerd.
- **Wat gebeurt er met het opgeleverde (eind)product?** Dit gaat verder afgemaakt worden, en zal dan als een vast onderdeel gebruikt gaan worden in onze video management software.
- **Kunt u direct met het opgeleverde product aan de slag?** Nog niet, er zijn nog onderdelen die afgemaakt moeten worden, voordat het meegeleverd kan worden met onze video management software.

7. Zijn er nog aspecten voor u van belang die nog niet aan de orde zijn geweest? Nee

8. **Bent u bereid een volgende keer weer uw medewerking te verlenen aan het beschikbaar stellen van een afstudeerplaats** (graag met toelichting)? Ja, VDG heeft genoeg uitdagende opdrachten klaar liggen die meer onderzoek vereisen voordat het verder geïmplementeerd kan worden. De focus die een afstudeerder hier aan kan geven pas daar perfect bij.