

Het realiseren van een multifunctioneel videoplatform

Onderzoek op basis van een Proof of Concept



Auteur:	Joost Damen
Datum:	05-06-2012
Versie:	1.0
Plaats:	Tilburg
Opdrachtgever:	Tilburg University
Onderwijsinstelling:	Fontys Hogescholen ICT

Titelpagina

Gegevens document

Document: Afstudeerscriptie
Titel: Het realiseren van een multifunctioneel videoplatform
Ondertitel: Onderzoek op basis van een Proof Of Concept

Gegevens onderwijsinstelling

Onderwijsinstelling: Fontys Hogescholen ICT
Opleiding: Bachelor of ICT, Deeltijd
Richting: Software Engineering
Docentbegeleider: Dhr. Joost Kant

Gegevens student

Naam: Joost M.J. Damen
Studentnummer: 2135822

Gegevens bedrijf

Naam: Tilburg University
Departement: Library and IT Services (LIS)
Afstudeerbegeleider: Dhr. Corno Vromans
Functie: Manager IT

Getekend voor gezien door bedrijfsbegeleider:

Datum:

Plaats:

De bedrijfsbegeleider,

Voorwoord

De afgelopen zes maanden stonden in het teken van mijn onderzoek naar de mogelijkheden voor een nieuw videoplatform voor mijn werkgever Tilburg University (TiU).

Het open source platform MediaMosa biedt diverse functionaliteiten die dit mogelijk zouden kunnen maken. Ik heb de opdracht aangepakt om op basis van een *Proof Of Concept* een testomgeving te bouwen en te onderzoeken hoe dit platform zich in de praktijk gedraagt.

Met MediaMosa als basis van het platform zijn ook de functionaliteiten van een streamingserver en de authenticatie en autorisatie naar het platform onderzocht. Dit alles met het oog op een toekomstvaste oplossing die aansluit op de huidige omgeving binnen TiU.

De werkzaamheden aan deze opdracht hebben plaatsgevonden van december 2011 tot en met mei 2012. Het resultaat is een werkende testomgeving. Aan de hand van mijn onderzoek, bevindingen en keuzes moet dit verslag een leidraad zijn voor de weg naar een nieuw videoplatform voor de toekomst.

Tijdens mijn afstudeerperiode heeft een aantal mensen mij persoonlijk begeleid. Ik wil in het bijzonder Corno Vromans (IT Manager, Tilburg University) en Joost Kant (Afstudeerbegeleider Fontys Hogescholen) bedanken. Zij hebben mij tijdens deze periode ondersteund met hun inzicht en ervaringen.

Daarnaast wil ik twee collega's van de UNIX afdeling bedanken, dit zijn Wessel Dankers en Thijs Kinkhorst. Zij hebben mij met name op weg geholpen bij de initiële installaties van UNIX omgevingen en geholpen op het gebied van de authenticatie oplossing.

Getekend,

Joost Damen

Tilburg, mei 2012

Inhoudsopgave

Titelpagina	- 1 -
Voorwoord	- 2 -
Samenvatting	- 5 -
Summary	- 6 -
Verklarende woordenlijst	- 7 -
1 Inleiding	- 9 -
2 Het bedrijf	- 10 -
2.1 Tilburg University	- 10 -
2.2 Positie van de afstudeerder	- 10 -
3 De afstudeeropdracht	- 11 -
3.1 Beginsituatie	- 11 -
3.2 Probleemstelling.....	- 11 -
3.3 Gewenste eindresultaat.....	- 13 -
3.4 Opdrachtomschrijving	- 14 -
3.5 Onderzoekaspecten	- 15 -
3.6 Projectfasering.....	- 16 -
4 MediaMosa nader verklaard	- 18 -
4.1 Het ontstaan van MediaMosa	- 18 -
4.2 Wat is MediaMosa?	- 18 -
4.3 Architectuur van MediaMosa	- 18 -
4.4 SOA lagen	- 19 -
4.5 Basiscommunicatie via REST	- 20 -
4.6 Objecten	- 20 -
4.6.1 Assets.....	- 20 -
4.6.2 Mediafiles	- 20 -
4.6.3 Stills	- 21 -
4.6.4 Collections	- 21 -
4.6.5 Metadata	- 21 -
4.7 Relaties.....	- 21 -
4.8 Componenten	- 22 -

5	Installatie en configuratie	- 23 -
5.1	Inleiding.....	- 23 -
5.1.1	Hardware specificaties.....	- 23 -
5.1.2	Virtuele machines.....	- 23 -
5.1.3	Netwerkomgeving.....	- 24 -
5.2	Tools.....	- 24 -
6	MediaMosa Servers	- 26 -
6.1	Applicatieserver.....	- 26 -
6.1.1	Extra software	- 26 -
6.1.2	FFmpeg.....	- 26 -
6.1.3	Lua.....	- 27 -
6.1.4	Opbouw.....	- 27 -
6.2	Frontend applicaties.....	- 29 -
6.2.1	MediaMosa White Label Ega (WLE).....	- 29 -
6.2.2	MediaMosa Construction Kit (CK)	- 29 -
6.2.3	MediaMosa Site Builder (SB).....	- 29 -
6.2.4	Keuze.....	- 29 -
6.3	Jobserver	- 30 -
7	Streaming Video.....	- 31 -
7.1	Codecs.....	- 31 -
7.2	Containers.....	- 32 -
7.3	Streaming binnen MediaMosa.....	- 32 -
7.4	Wowza Media Server.....	- 32 -
7.5	Streaming protocollen.....	- 34 -
7.5.1	RTMP: Real-Time Messaging Protocol	- 34 -
7.5.2	Adaptieve Bitrate Streaming	- 34 -
7.6	JWplayer	- 35 -
7.7	Nadelen Wowza / MediaMosa.....	- 35 -
7.8	Resultaat	- 36 -

8	Dilemma's in online video	- 37 -
8.1	FLASH vs HTML5.....	- 37 -
8.2	Compatibiliteit door Fallback.....	- 38 -
8.3	Apache H264 Streaming	- 38 -
8.4	De keuze van de codec.....	- 39 -
8.5	Embedding.....	- 41 -
8.6	Resultaat	- 41 -
9	Authenticatie en autorisatie	- 42 -
9.1	Toelichting SURFfederatie.....	- 42 -
9.2	Werkwijze.....	- 43 -
9.3	Manier van authenticeren.....	- 43 -
9.4	Authenticeren bij het platform.....	- 44 -
9.4.1	Scenario: Gebruiker meldt zich aan bij de SURFfederatie	- 45 -
9.5	Bepalen van de attributen.....	- 46 -
9.6	Autorisatie bij het platform	- 47 -
10	Mogelijkheden voor opschaling	- 48 -
10.1	Opschalingsscenario	- 48 -
11	Conclusies en Aanbevelingen	- 50 -
	Evaluatie.....	- 51 -
	Literatuurlijst	- 52 -

Samenvatting

Dit document beschrijft de afstudeeropdracht die is uitgevoerd voor Fontys Hogelschool ICT te Eindhoven. Het bedrijf waar de afstudeeropdracht heeft plaatsgevonden is Tilburg University (TiU). De opdracht bestaat uit het onderzoek naar de mogelijkheden voor een nieuw multifunctioneel videoplatform.

In de huidige situatie is er een videoserver actief die zijn beperkingen heeft. Deze server kan enkel bestanden verwerken die van het Windows Media formaat zijn (WMV). In de huidige situatie bestaan er geen mogelijkheden tot het authenticeren en autoriseren van gebruikers. Op heel basaal niveau is er een manier van beveiliging van videobestanden mogelijk. Het afspelen van videomateriaal op mobiele apparaten en alle recente webbrowsers wordt op dit moment niet goed ondersteund.

Tijdens de opdracht is een testomgeving gerealiseerd op basis van een Proof of Concept. Het platform wat onderzocht is, is het open source media distributie platform MediaMosa. MediaMosa is al in gebruik bij enkele universiteiten en hogescholen en zou wellicht een toekomstvast oplossing kunnen bieden. MediaMosa is geïnstalleerd op een virtuele machine. De componenten en functionaliteiten van het platform zijn onderzocht, getest en beschreven. Een scheiding tussen functionaliteiten heeft plaats gevonden door een tweede virtuele machine te installeren. Deze neemt de processorbelastende taken voor zijn rekening.

De omgeving met twee virtuele machines is uitgebreid met een streamingserver. Hiervoor is Wowza Media Server gebruikt en deze handelt de streaming verzoeken van MediaMosa af richting de eindgebruikers. Tijdens de opdracht is gebleken dat Wowza niet succesvol met alle verzoeken van MediaMosa omging. Om dit op te lossen is later een speciale module voor Apache webserver toegevoegd. Deze twee opties om video te streamen naar eindgebruikers zijn uitgevoerd in combinatie met JWplayer, een open source video player voor webbrowsers.

Er is onderzoek uitgevoerd naar codecs, containers en andere facetten van online video. De keuze voor een nieuwe bestandsformaat om al het videomateriaal in de toekomst op te slaan is gemaakt, de H.264 codec in een MP4 container.

Om gebruikers te kunnen authenticeren bij het platform is er onderzoek gedaan naar koppelingen met zogeheten Identity Providers. Uiteindelijk is er een koppeling gerealiseerd met de SURFfederatie, dit was een eis van de opdrachtgever. Hiervoor is gebruik gemaakt van simpleSAMLphp en een bijbehorende module voor Drupal. Het autoriseren aan de hand van aangeleverde attributen bij het aanmelden zou ook via de module mogelijk moeten zijn. Helaas is er gebleken dat deze functionaliteit niet naar behoren heeft gewerkt.

Het resultaat is een werkende testomgeving die voldoet aan de gestelde eisen. MediaMosa is een geschikte oplossing om in de toekomst uit te rollen binnen TiU en de huidige videoserver te vervangen. Een andere uitkomst van een onderzoek is dat de documentatie van MediaMosa grotendeels ontbreekt. Informatie moet ingewonnen worden bij andere gebruikers en/of ontwikkelaars. Dit kan ervoor zorgen dat niet alle functionaliteiten in de toekomst gebruikt kunnen worden zonder hulp van buitenaf.

Summary

This document describes the final project that is executed for the ICT undergraduate course at Fontys University of Applied Sciences in Eindhoven and took place at Tilburg University (TiU).

The assessment includes performing research and tests the new possibilities for a new multi functional video platform.

Firstly, the current situation with the video server at Tilburg University is experiencing some difficulties. For example, the server is only capable of streaming Windows Media Format (WMV) files. Besides that, there is no possibility to authorise users. On a very basic level, it was possible to secure videos. Because of the WMV format, playing videos on mobile devices and on current web browsers, was not working properly.

During this assessment, a test environment was realised based on a Proof of Concept. According to an agreement with the University, the open source media distribution platform MediaMosa was researched. The platform MediaMosa is already in use at several universities in The Netherlands. MediaMosa is installed on a Virtual Machine. All the components and functions are researched, tested and documented. A second virtual machine was installed to process tasks that consume lots of processor load.

This environment is expanded with a streaming server. Wowza Media Server deals with the streaming requests from MediaMosa and delivers video to end users. During this project, it came to mind that Wowza is not dealing with all requests correctly. To solve that problem, an Apache web server was added to the platform. Those two options are used to stream video to the users are executed in combination with JWplayer, an open source video player for web browsers.

Research on codecs, containers and other facets of online video can be found. The choice to store all video content in a reliable and future proof format, resulted in the H.264 codec encapsulated in a MP4 container.

In order to authorise the users, some research was done on the so-called Identity Providers. At last, a connection with SURFederation was realised as the University wished to be done so. Therefore, a simpleSAMLphp and an additional module for Drupal was used. The authorisation process should be possible through that module. Unfortunately, the function did not worked as expected.

At last, the result is a successful test environment that will fulfill the needs and wishes. MediaMosa is a great solution, the current videosever can be replaced with a solution containing MediaMosa. Another outcome of the research is that reliable resources of the workings of MediaMosa are missing. Accurate information was collected through other users and/or developers. This could lead to less functionality of the platform. If there is no proper documentation available, the need for knowledge of experts will be needed.

Verklarende woordenlijst

AD	Active Directory, directoryservice implementatie van Microsoft die beheerders in staat stelt rechten en instellingen in een domein toe te passen op gebruikers en gebruikersgroepen.
AV	AV is een veelgebruikte afkorting voor Audio Visueel.
Codec	Codec is software die audio- en video-stromen kan coderen en decoderen.
Container	Formaat dat definieert hoe audio, video en metadata wordt opgeslagen.
CMS	Content Management Systeem, softwaretoepassing die meestal via het web toegankelijk is en waar gegevens eenvoudig en overzichtelijk in worden bijgehouden.
Drupal	Open Source CMS ontwikkeld in PHP, vormt de basis voor MediaMosa.
EGA	Eind Gebruikers Applicatie, frontend van MediaMosa.
Embedding	Het insluiten van objecten binnen webpagina's.
FFmpeg	Open Source Transcodeer-tool.
GUI	Graphical User Interface, interactie met een computer door middel van grafische beelden.
IP	Internet Protocol, laat computernetwerken met elkaar communiceren.
JWplayer	Flexibele player die zowel met Flash als HTML5 overweg kan.
LINUX	Familie van UNIX besturingssystemen.
LIS	Library and IT Services, departement binnen TiU.
MediaMosa	Open Source Media Management Platform, backend applicatie.
Metadata	Gegevens die karaktersistieken over bepaalde gegevens beschrijven.
NAS	Network Attached Storage, harde schijf/opslag benaderbaar via het netwerk.
Open Source	Het vrij beschikbaar stellen van software voor ontwikkeling en herdistributie.
OS	Operating System.
Proof of Concept	Principe waarbij een dienst/software getest en uitvoerig onderzocht wordt om te beoordelen of deze geschikt is voor implementatie.
REST protocol	REpresentational State Transfer, software-architectuur voor gedistribueerde mediasystemen als het Internet.
RTMP	Real Time Messaging Protocol, streaming media protocol om audio, video en data over te brengen tussen een Flash player en een server.
SOA	Service-oriented Architecture, architectuurmodel voor software dat bestaat uit verschillende lagen.
SSO	Single Sign On, manier om gebruikers slechts één maal in te laten loggen en vervolgens kan er automatisch toegang verschaft worden tot applicaties en data die ook SSO gebruiken.
SURF	Stichting SURF, organisatie van samenwerkende hoger onderwijsinstellingen op het gebied van ICT.
SURFnet	Dochterorganisatie van SURF. Zorgt voor internet connectiviteit tussen instellingen en voert innovatieve projecten uit. Founder van MediaMosa.
TiU	Tilburg University (voorheen Universiteit van Tilburg, UvT).
TiU account	Account (username/password) wat iedere student/docent/medewerker

	heeft.
Transcoderen	Het converteren van het ene mediaformaat naar het andere.
UNIX	Uniplexed Operating and Computing System (voorheen UNICS), familie van besturingssystemen.
URL	Uniform Resource Locator, internetadres.
Videoserver	Server die specifieke mediaservices draait en zo op afroep bepaalde mediabestanden kan afspelen naar gebruikers.
VLAN	Virtual LAN, virtuele scheiding van netwerken die zich fysiek in hetzelfde netwerk bevinden.
VPN	Virtual Private Network, netwerk wat door een ander netwerk getunneld wordt.
WAYF	Afkorting die staat voor “Where Are You From”, wordt bij de SURFfederatie gebruikt op de keuzepagina voor de Identity Provider.
Wowza	Software voor Streaming Media Services.
WMV	Windows Media Video Formaat, videocompressie formaat ontwikkeld door Microsoft.
XML	Standaard om gestructureerde gegevens vast te leggen in de vorm van tekst.

1 Inleiding

Tilburg University (TiU) is een gerenommeerde universiteit die haar studenten en docenten wil bedienen met de nieuwste technische voorzieningen. Video is een belangrijk onderdeel in het hedendaags onderwijs.

Binnen TiU is er een voorziening aanwezig om videobestanden te ontsluiten naar eindgebruikers. Deze voorziening bestaat uit een videoserver die enkel bestanden kan *streamen* in het Windows Media formaat (WMV). Omdat de huidige voorziening enkel om kan gaan met dit bestandsformaat levert dit vandaag de dag compatibiliteitsproblemen op. Er is vrijwel geen beheersbaarheid op de bestanden omdat aanvullen informatie, zoals titel, eigenaar en onderwerp ontbreken. Deze aanvullende data wordt metadata genoemd.

De afscherming van het videomateriaal dat beschikbaar wordt gesteld is van een laag niveau. Met afschermen bedoelen we het beveiligen van de bestanden zodat ze niet voor iedereen toegankelijk zijn. Docenten en wetenschappers willen videomateriaal veelal voor een kleine groep gebruikers beschikbaar stellen.

Vanuit TiU is er al geruime tijd de wens om een open source platform te onderzoeken wat een oplossing zou kunnen bieden voor de huidige probleemsituatie. Dit platform is het media distributie platform MediaMosa. Het platform is al in gebruik bij andere universiteiten en hogescholen en zou wellicht ook een passende oplossing voor TiU kunnen bieden.

In hoofdstuk 2 wordt het bedrijf beschreven waar de afstudeeropdracht is uitgevoerd. In hoofdstuk 3 wordt de opdracht beschreven. Van beginsituatie tot gewenst eindresultaat wordt in beeld gebracht en wordt verantwoord waarom deze opdracht van belang is en wat het probleem is in de huidige situatie. Een introductie van het platform MediaMosa is weergegeven in hoofdstuk 4, de objecten, componenten en functionaliteiten van het platform behoeven een uitleg. Tijdens de opdracht is er een testomgeving gebouwd op basis van een Proof of Concept. Hoofdstuk 5 geeft de details weer over de hard- en software die zijn gebruikt voor de beheersing en uitvoer van een testomgeving. Hoofdstuk 6 geeft weer hoe dit is uitgevoerd en waarom bepaalde keuzes gemaakt zijn.

Nadat de basis van het platform met MediaMosa gerealiseerd is, vindt het onderzoek en de realisatie plaats van een streaming videoserver. Deze server dient als uitbreiding op MediaMosa. De details van deze server en het gebruik van streaming video staan beschreven in hoofdstuk 7. De wereld van streaming video is complex en moeilijk te doorgronden, in hoofdstuk 8 worden de dilemma's toegelicht. Tevens is de keuze voor een nieuwe codec verantwoord.

De facetten van autorisatie en authenticatie die de toegang tot het platform mogelijk maken en bepalen wat een gebruiker mag wanneer er toegang is staan beschreven in hoofdstuk 9. Tijdens de opdracht is er een omgeving gerealiseerd op basis van een Proof of Concept. Mogelijkheden voor uitrol en opschaling van het platform op basis van de resultaten en kennis die zijn opgedaan staan beschreven in hoofdstuk 10.

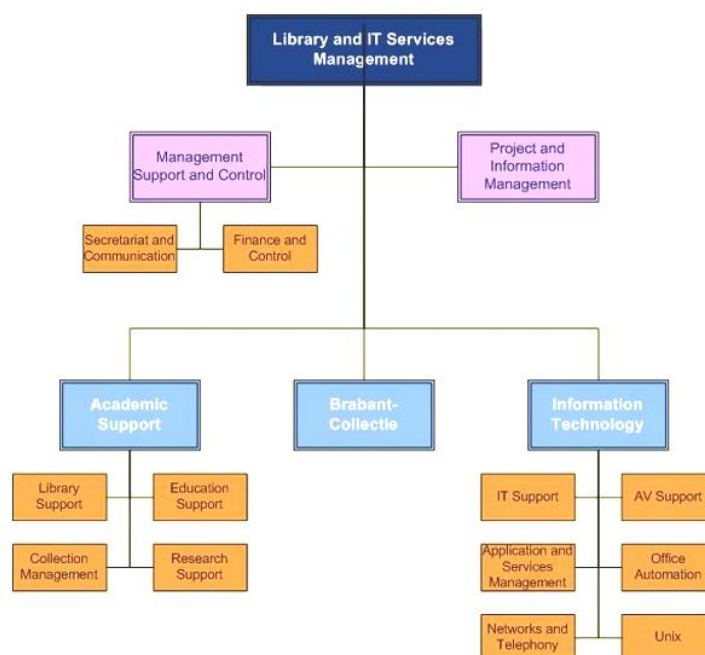
2 Het bedrijf

2.1 Tilburg University

Tilburg University (TiU) is een gerenommeerde universiteit die is gevestigd op een centrale campus aan de rand van Tilburg. De universiteit is gespecialiseerd in alfa- en gammawetenschappen en verzorgt al meer dan tachtig jaar onderwijs. Er wordt onderzoek gedaan naar maatschappelijke vraagstukken. Voorbeelden hiervan zijn de economische crisis, klimaatbeheersing, de multiculturele samenleving en privacybescherming. Met toponderzoek en excellent onderwijs wil TiU een bijdrage leveren aan een betere samenleving. Nieuwe ideeën en inzichten worden verkregen door te onderzoeken, te leren en te begrijpen. Iedere dag weer wordt er gezocht naar meer begrip van het maatschappelijk functioneren in deze wereld. TiU bestaat uit vijf faculteiten met daaraan verbonden tal van gerelateerde onderzoeksgroepen, instellingen en partners. De TiU populatie bestaat uit zo'n 2000 medewerkers en 13000 studenten.

2.2 Positie van de afstudeerder

De afstudeerder is binnen TiU werkzaam bij de dienst Library and IT Services (LIS). De dienst bestaat uit twee hoofdclusters, een *supportstaff* en project- en informatiemanagement. Het centrale IT cluster van TiU is ondergebracht bij LIS, er zijn circa 175 medewerkers werkzaam. Binnen LIS bestaat de afdeling LIS-AV waar AV staat voor AudioVisueel. De afstudeerder is werkzaam op deze afdeling waar verschillende expertises aanwezig zijn. Kerntaken van de afstudeerder op deze afdeling zijn het in stand houden en vernieuwen van de gecombineerde ICT/AV infrastructuur. Ook houdt hij zich hier onder andere ook bezig met *streaming* media en *videoconferencing*. De verdieping van de afstudeeropdracht vindt plaats binnen meerdere disciplines binnen het departement LIS. Er is een afdeling die gespecialiseerd is in UNIX systeembeheer, deze collega's kunnen ondersteuning bieden bij installaties en specifieke configuraties. Verdere toelichting op het departement LIS met bijbehorende afdelingen wordt weergegeven in onderstaand organigram.



Figuur 1 Organigram Library and IT Services

3 De afstudeeropdracht

3.1 Beginsituatie

Het gebruik van video in het onderwijs is de laatste jaren enorm toegenomen. Colleges worden opgenomen en als video online aangeboden. Docenten refereren aan videofragmenten binnen leeromgevingen en vaardigheden worden beoordeeld aan de hand van opgenomen sessies. De laatste jaren is ook de groei naar nieuwe toepassingen van video in het onderwijs gegroeid zoals voor- en nabesprekingen van tentamens. Om aan deze behoefte te voldoen heeft TiU rond 2002 een Windows Media Server ingericht. Al snel werden er grote hoeveelheden video ontsloten en tot de dag van vandaag gebeurt dit nog steeds via deze server. Op de server staan circa 3500 mediabestanden. Er is niet goed vastgelegd hoe lang bestanden ontsloten worden en wie de eigenaar is van een bestand. Bestanden worden zonder enige toegevoegde informatie op de server geplaatst. Deze oplossing die nu nog in gebruik is voldoet niet meer aan de vraag en is vrijwel onbeheersbaar geworden. Het uitgangspunt was dat anno 2012 deze oplossing zo sterk verouderd is en de huidige “YouTube-generatie” andere verwachtingen heeft bij online video.

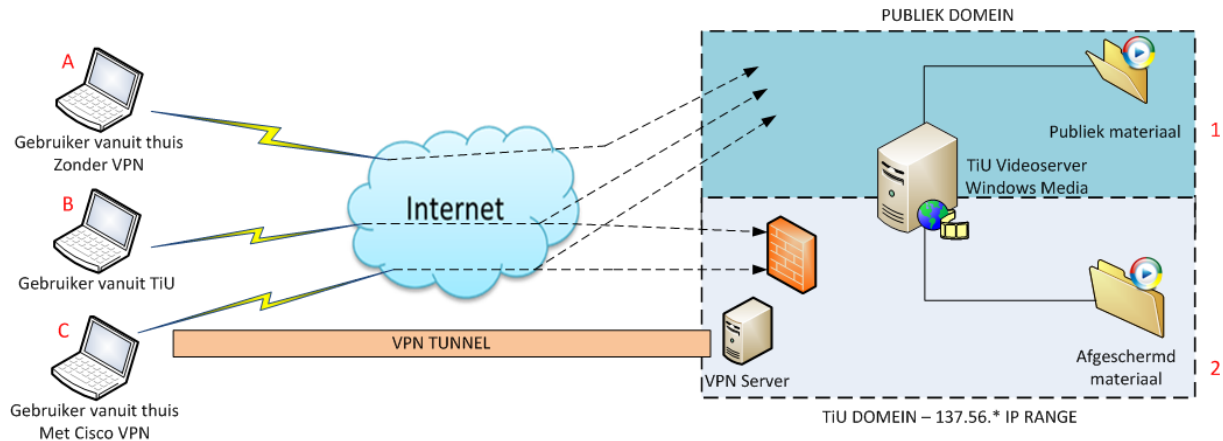
3.2 Probleemstelling

De problemen met huidige situatie zijn:

- het ontbreken van correcte metadatering, dit resulteert in onbeheersbaarheid van materiaal;
- compatibiliteitsproblemen met het uploaden en afspelen van videobestanden, er kunnen enkel WMV bestanden aangeboden en afgespeeld kunnen worden;
- het zelfstandig plaatsen van bestanden door eindgebruikers is niet mogelijk, dit moet via de dienst LIS-AV;
- het afschermen of beveiligen van het materiaal voor een selecte groep eindgebruikers is niet mogelijk.
- het is niet mogelijk om het materiaal als *download* beschikbaar te stellen voor eindgebruikers.

Het dilemma in deze kwestie is, hoe los je deze problemen op? Dat dient tijdens deze opdracht duidelijk te worden. In de tijd dat de Windows Media Server aangeschaft werd, was dit een voor de hand liggende keuze. Het bood voldoende functionaliteit en vergeleken met formaten als *Flash* (en destijds) *RealPlayer* was dit waarschijnlijk de beste optie. Vandaag de dag is er een geweldige hoeveelheid van bestandsformaten die alles behalve compatibel zijn met elkaar en ook niet op elk gewenst apparaat afgespeeld kunnen worden. Het uploaden van bestanden op de Windows Media Server kan enkel gebeuren door mensen die hier rechten voor hebben. Dit zorgt voor een inefficiënte werkwijze om bestanden op de server te plaatsen. Bijkomstigheid is dat alle bestanden die geupload worden in het Windows Media formaat moeten zijn. Als dit niet het geval is, vindt er bij de dienst LIS-AV een handmatige transcodeer slag plaats van het oorspronkelijke formaat naar het juiste Windows Media formaat.

Wanneer de bestanden geplaatst zijn bestaan er twee manieren qua afscherming. De ene is publiek, iedereen kan de URL openen en het materiaal afspelen, enkel downloaden is niet mogelijk. De tweede manier is, het materiaal enkel beschikbaar maken voor gebruikers binnen het TiU domein, ook hier geldt enkel afspelen en niet downloaden.



Figuur 2 Visualisatie huidige situatie, links de clients en rechts de videoservert.

In de figuur hierboven zijn de scenario's te zien waarin we drie soorten gebruikers kennen:

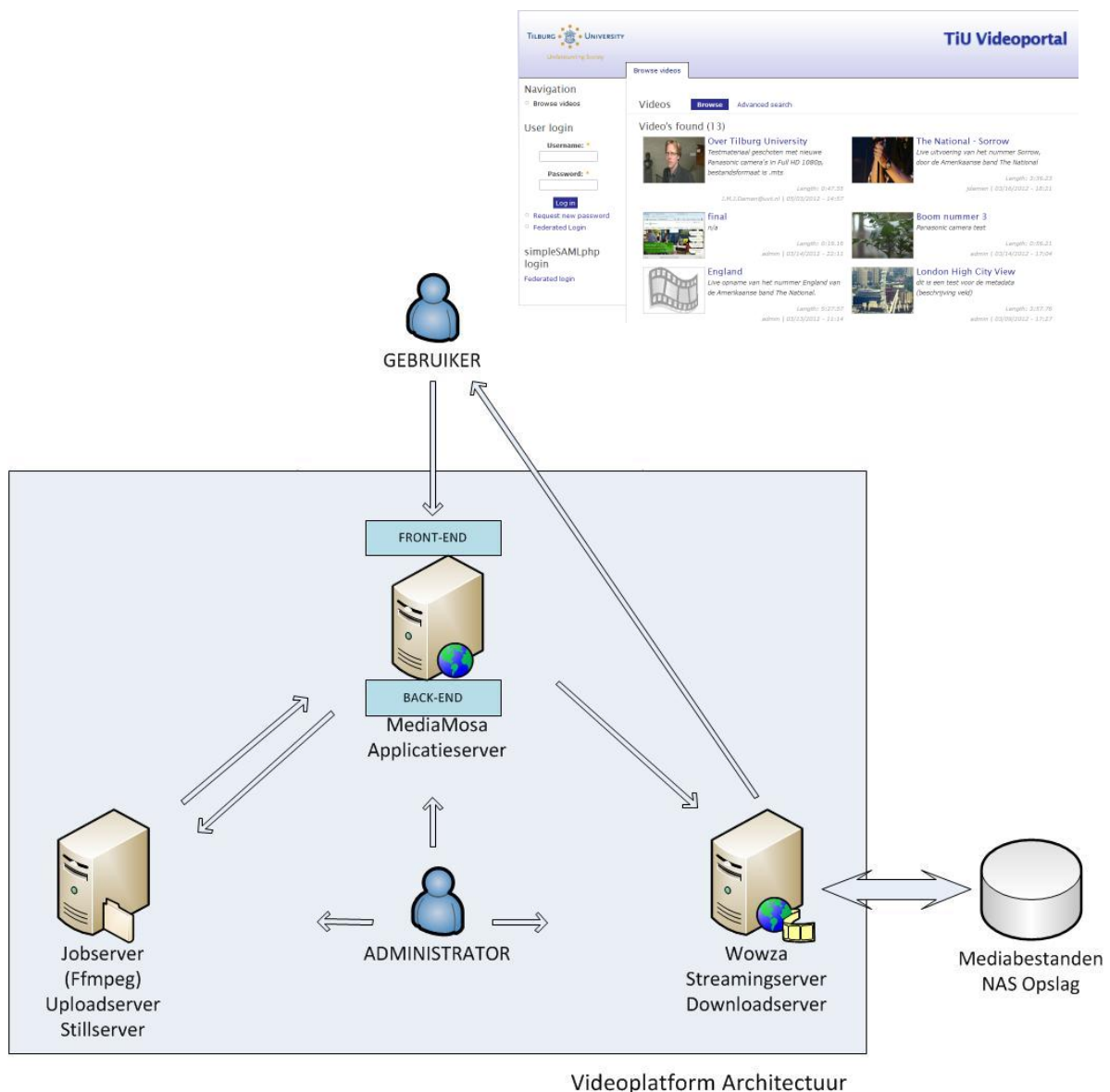
- A: een gebruiker die zich buiten TiU bevindt en geen VPN connectie heeft met TiU;
- B: een gebruiker die binnen TiU bevindt en geen VPN connectie (nodig) heeft;
- C: een gebruiker die zich buiten TiU bevindt en wel een VPN connectie met TiU heeft.¹

De enige manier van afscherming is in de figuur zichtbaar: een gebruiker buiten TiU kan het afgeschermd materiaal niet afspelen (A). Het TiU domein bevindt zich in de 137.56.* IP-range. Iedereen die zich in deze range bevindt (B/C), al dan niet met een VPN connectie, kan al het materiaal afspelen (1/2). Zij hebben toegang tot het publieke(1) en het afgeschermd gedeelte(2). Voor beide gevallen geldt dat men de URL moet weten, bij het laatste geval(2) is dan enkel de 'veiligheid' ingebouwd dat niemand het materiaal kan zien buiten de betreffende IP-range. Deze manier van afschermen wordt ook wel *security through obscurity* genoemd. Op deze manier de bestanden afschermen is niet meer van deze tijd, mediabestanden bevatten soms vertrouwelijke of privacy gevoelige informatie die niet altijd voor iedereen geschikt is.

¹ Website over VPN, TiU: <http://www.tilburguniversity.edu/nl/studenten/ict/vpn/>

3.3 Gewenste eindresultaat

Het doel van de afstudeeropdracht is om door onderzoek en realisatie een duidelijk beeld te krijgen van de mogelijkheden en beperkingen van een nieuw videoplatform. De realisatie vindt plaats op basis van een Proof of Concept, de kern van het videoplatform is MediaMosa. MediaMosa is een *open source* media management platform en zou een goede basis kunnen bieden voor de toekomst en zou de huidige problemen uit de weg kunnen ruimen met zijn multifunctionele en flexibele opbouw. Door de mogelijkheden te onderzoeken en in een testomgeving een proefopstelling te realiseren moet er duidelijkheid komen in de facetten die voor TiU van belang zijn. In onderstaande figuur is het gewenste eindresultaat weergegeven.



Figuur 3 Architectuur van de testomgeving als gewenst eindresultaat, GUI als weergave voor gebruiker.

3.4 Opdrachtomschrijving

De afstudeeropdracht is om de mogelijkheden te onderzoeken die het Media Management Platform MediaMosa biedt, in de vorm van een *proof-of-concept*.

De onderzoeksvraag is:

Kunnen de bezwaren die in de huidige omgeving bestaan opgelost worden met de functionaliteiten die MediaMosa biedt?

De onderdelen en functionaliteiten die onderzocht en beschreven worden zijn:

- opbouw en werking van het platform met bijbehorende componenten en functionaliteiten;
- architectuur en beargumentering testomgeving behorende bij het project;
- eindgebruikers moeten mediabestanden zelfstandig kunnen plaatsen en hebben zelf de controle over de toegankelijkheid (afscherming of publiek toegankelijk);
- de eigenaar van het materiaal kan zelf aangeven of men het materiaal beschikbaar wil stellen als download door de eindgebruiker;
- onderzoek naar de mogelijkheden om het platform te gebruiken als hulpmiddel voor batch uploads, dit houdt in het uploaden van meerdere videobestanden;
- onderzoek naar het insluiten van materiaal in webpagina's (embedding) en de mogelijkheden om materiaal te herdistribueren via Social Media of potentiële gebruikers te notificeren door middel van email.
- mogelijkheden om de gebruikersinterface van de Eind Gebruikers Applicatie (EGA) conform de huisstijl van TiU in te richten;
- onderzoek naar gangbare bestandsformaten (codecs/containers) van mediabestanden, moet resulteren in een definitief bestandsformaat voor de toekomst;
- onderzoek naar authenticatie middels *Single Sign On* (SSO) via de SURFfederatie.

Wat vooraf vast staat:

- Er wordt geen ander media management platform dan MediaMosa onderzocht;
- Als streamingserver wordt enkel Wowza Media Server onderzocht;
- De UNIX omgeving die gebruikt wordt is Debian 64-bit.

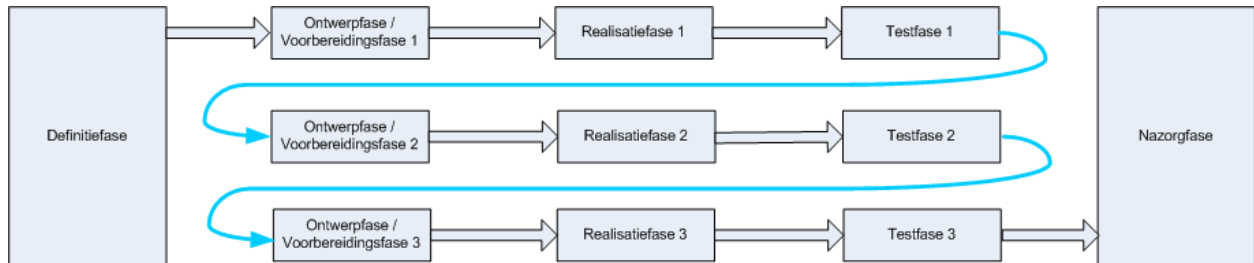
3.5 Onderzoeksaspecten

Om tot de antwoorden op de onderzoeksvraag te komen word er hoofdzakelijk literatuuronderzoek gedaan om informatie te vergaren over de verschillende onderwerpen. Over veel aspecten van het project had ik vooraf weinig kennis, hieronder staat een opsomming van de onderdelen waar het literatuuronderzoek is toegepast.

- UNIX (Debian), Apache, MySQL, PHP;
- MediaMosa met bijbehorende componenten en functionaliteiten;
- Authenticatie via SURFfederatie, SAML 2.0;
- Streaming met Wowza, Adaptieve en RTMP streaming protocollen;
- JWplayer mogelijkheden met Flash & HTML5;
- Codecs en Containers.

3.6 Projectfasering

De fasering heb ik in verschillende onderdelen ondergebracht. Tijdens de voorbereidingsfase kwam ik erachter dat het voor de beheersing van het project beter zou zijn om de uitvoering van de opdracht onder te verdelen in drie deelfases.



Definitiefase

Dit is de eerste fase van het project waarin het Plan van Aanpak is opgesteld. In dit document zijn onder andere de opdracht, doelstellingen, activiteiten, risico's, organisatie en planning beschreven. Het Plan van Aanpak is bijgevoegd in de bijlage (Bijlage I).

Ontwerpfase / Voorbereidingsfase 1

Deze fase is gestart nadat het Plan van Aanpak is goedgekeurd en er een akkoord was om het project te beginnen. De architectuur en opbouw van MediaMosa is hier verder onderzocht. De testomgeving om de virtuele machines op te installeren is klaargemaakt.

Realisatiefase 1

De eerste stap was de installatie en configuratie van de eerste omgeving die enkel bestond uit de applicatieserver met bijbehorende componenten. Deze eerste omgeving is op een virtuele machine gerealiseerd en MediaMosa is geïnstalleerd en geconfigureerd in combinatie met een *frontend*. Hierna vond de installatie & configuratie van de tweede omgeving plaats waar een tweede virtuele machine is toegevoegd. De functionaliteiten zijn verdeeld over de twee virtuele machines en er is gebruik gemaakt van gedeelde opslag. Hoe en waarom de functionaliteiten verdeeld zijn is verderop in dit document beschreven in Hoofdstuk 6. In de realisatiefase zijn de essentiële onderdelen in de praktijk onderzocht en zijn er tal van testen uitgevoerd. Aan de MediaMosa *backend* zijn verschillende frontends gekoppeld en met de opgedane kennis uit de vorige fases is hier het onderzoek in de praktijk voortgezet.

Testfase 1

De koppelingen met de frontends zijn uitvoerig getest, het gaat dan met name om de acties (REST calls) die een frontend doet naar MediaMosa toe. Wat er gebeurt in de *backend* en hoe het transcoderen van mediabestanden met FFmpeg in zijn werk gaat.

Ontwerpfase / Voorbereidingsfase 2

Het platform is in de eerste fase onderzocht, gerealiseerd en getest. Na deze eerste fase is het tijd om de extra functionaliteiten die aan MediaMosa gekoppeld moeten worden te onderzoeken. Het gaat hier om Wowza Media Server die de streamingverzoeken van MediaMosa moet gaan afhandelen.

Realisatiefase 2

In deze fase is de derde virtuele machine geïnstalleerd en geconfigureerd. Deze omgeving gaat fungeren als Streaming Server op basis van Wowza Media Server. Het koppelen met de MediaMosa servers en de centrale opslag zodat Wowza gebruik kan maken van de bestanden en verzoeken.

Testfase 2

De mogelijkheden van Wowza zijn uitgebreid onderzocht en hoe Wowza aangeroepen kan worden vanuit MediaMosa. Streamingprotocollen, *players* en codecs zijn onderzocht en vervolgens beschreven. Uit het testen met Wowza is gebleken dat de combinatie met MediaMosa niet in alle gevallen een werkende oplossing biedt. In dit document word dit nader toegelicht.

Ontwerpfase / Voorbereidingsfase 3

Na het functioneel maken van het platform is onderzocht wat de mogelijkheden zijn om authenticatie via de gewenste Single Sign On omgeving mogelijk te maken. Er is overleg gevoerd over de mogelijkheden van implementatie met collega's van de UNIX afdeling.

Realisatiefase 3

Het installeren, configureren en koppelen van Service Providers en Identity Providers speelde in deze fase een rol. Stapsgewijs is het platform gekoppeld, eerst aan een testomgeving, daarna aan een productieomgeving en uiteindelijk aan de testomgeving van de SURFfederatie.

Testfase 3

Testen van de mogelijkheden van authenticatie en instellingen van de simpleSAMLphp module. Getest met verschillende scenario's zodat er uiteindelijk gekoppeld kon worden met de testomgeving van de SURFfederatie. Verder is de autorisatie binnen Drupal op basis van het aanmelden via simpleSAMLphp getest en onderzocht.

Nazorgfase

De laatste fase is de nazorgfase, deze bestaat uit het evalueren en documenteren van de bevindingen. Dit document met zijn bijlages worden afgerond in deze fase. Na de oplevering van dit document vinden er twee maal een presentatie plaats. De eerste wordt gegeven op Tilburg University voor collega's en betrokkenen. De tweede is de afstudeervoordracht op Fontys Hogescholen.

4 MediaMosa nader verklaard

4.1 Het ontstaan van MediaMosa

In 2007 zijn SURFnet en Kennisnet gestart met de ontwikkeling van een videodistributieplatform, dit werd gedaan onder de naam ‘VP-core’. Vanuit ‘VP-core’ is onder andere het platform SURFmedia ontstaan waar (hogere) educatieve instellingen videomateriaal kunnen uploaden en ontsluiten. SURFmedia is een dienst die door SURFnet geleverd wordt. SURFnet heeft aangekondigd deze dienst te zullen stoppen en wel vanaf 1 januari 2013. ‘VP-core’ is de basis voor wat nu MediaMosa is, een opensource mediadistributie platform gebaseerd op het Drupal Content Management Systeem (CMS). MediaMosa heeft een flinke ontwikkeling doorgemaakt door het SURFnet/Kennisnet Innovatieprogramma. De subsidie voor de doorontwikkeling van MediaMosa is gestopt, maar omdat er veel vraag is vanuit universiteiten en hogescholen wordt er gezocht naar een toekomstvast oplossing. Hoewel TiU geen grootverbruiker was van SURFmedia, is de wens om een videoplatform in te richten op basis van MediaMosa wel aanwezig, voornamelijk als vervanging van de huidige videoserver. Dit document moet verduidelijking brengen in de werkwijze van MediaMosa en alle facetten hier omheen door het in de praktijk als een Proof Of Concept neer te zetten.

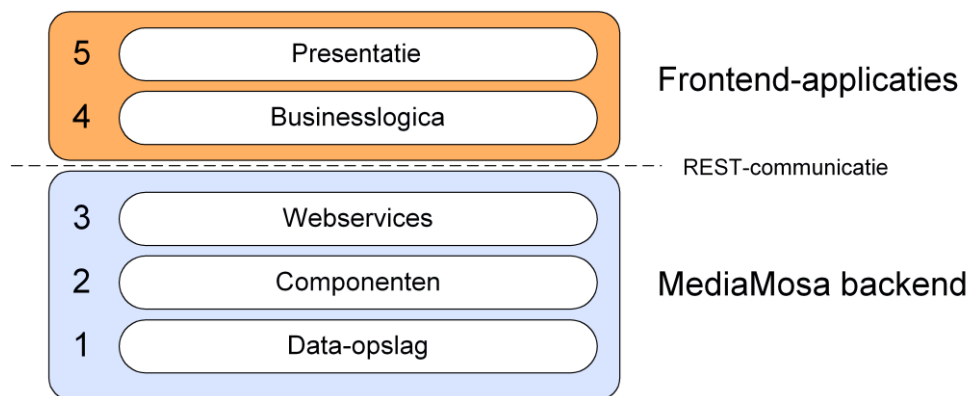
4.2 Wat is MediaMosa?

MediaMosa is een mediadistributie platform. Het woord media wordt hier gebruikt in plaats van video omdat het platform met tal van bestanden en formaten overweg kan. In het Engels is deze term wellicht beter bekend, namelijk als *Digital Asset Management*². Vanuit deze visie is het platform ook ontwikkeld. MediaMosa beheert *assets* en aan een asset hangen één of meer mediabestanden. Een asset is een *entry* in een database die een uniek ID bevat. Aan een asset worden mediabestanden en metadata gekoppeld. Deze metadata is nodig om assets terug te kunnen vinden. Hoe een asset zich verhoudt tot de daadwerkelijke mediabestanden en metadata wordt toegelicht in het volgende onderdeel.

4.3 Architectuur van MediaMosa

MediaMosa is gebouwd volgens een *Service Oriented Architecture* (SOA). Dit houdt onder andere in dat MediaMosa is verdeeld in een backend en een frontend. De backend bestaat uit een aantal webservices, componenten en dataopslag. Hier worden alle taken uitgevoerd zoals transcoderen, uploaden en afspelen. De frontend wordt gevormd door applicaties die eindgebruikers en beheerders gebruiken om met de backend te communiceren. Het moet mede door onderstaande figuur duidelijk zijn dat MediaMosa enkel in een backend voorziet en dat het dus geen ‘out of the box’ oplossing is. MediaMosa kan niet werken zonder dat het wordt aangesproken door frontend applicaties. Deze communicatie verloopt via een standaard protocol, REpresentational State Transfer (REST).

² Toelichting op Digital Asset Management: <http://www.contentmanager.eu.com/dam.htm>



Figuur 4 SOA lagen model

SOA bestaat uit 5 lagen, de scheiding tussen de lagen is in de figuur duidelijk zichtbaar. De lagen 1 t/m 3 worden gevormd door MediaMosa (backend) en de lagen 4 en 5 (dienen te) worden gevormd door een frontend applicatie.

4.4 SOA lagen

Laag 5: Presentatie

Eindgebruikers communiceren met de laag *Presentatie*. Deze laag bestaat vooral uit presentatie in de vorm van een grafische interface; hij bevat vrijwel geen eigen functionaliteiten maar roept functies uit de laag *Businesslogica* aan.

Laag 4: Businesslogica

Deze laag bevat alle gebruikersfuncties van de eindgebruikersapplicatie (EGA), die worden aangeroepen vanuit de laag *Presentatie*. De laag *Businesslogica* roept vervolgens services aan uit de laag *Webservices* volgens het REST-protocol.

Laag 3: Webservices

Via deze laag communiceert MediaMosa met de verschillende applicaties. De laag *Webservices* bevat bouwblokken die een snelle realisatie van de functies uit de laag *Businesslogica* mogelijk maken. Deze bouwblokken, de webservices, spreken een of meer componenten aan uit de laag *Componenten*.

Laag 2: Componenten

Deze laag bevat de componenten waarvan de webservices gebruik maken om verzoeken van de applicaties af te handelen. Voorbeelden hiervan zijn het uploaden, transcoderen en streamen van een mediabestand.

Laag 1: Data-opslag

Deze laag zorgt voor de opslag van de data in een dataopslag, waarbij onderscheid gemaakt wordt tussen de mediafiles, de bijbehorende metadata en de autorisatiegegevens.

4.5 Basiscommunicatie via REST

Er zijn circa 180 webservices beschikbaar die een frontend applicatie kan gebruiken om de MediaMosa backend taken uit te laten voeren. Al deze services maken deel uit van de *REST interface* van MediaMosa. De *REST* definitie bestaat uit een URL aanroep op basis van het HTTP protocol.³

Voorbeelden van zulke aanroepen zijn:

- Het aanmaken van een nieuwe *mediafile*: *mediafile/create [POST]*
- Details opvragen van een *mediafile*: */mediafile/{mediafile_id} [GET]*

In de configuratie van MediaMosa is er een *REST Call Developer* aanwezig, dit hulpmiddel is ideaal om *REST calls* uit voeren en zo het antwoord in XML leesbaar te zien. Tijdens het uitvoeren van de opdracht is er veelvuldig gebruik gemaakt van deze *tool* om zo de werking van de communicatie, de aanroepen en de antwoorden te begrijpen.

4.6 Objecten

MediaMosa maakt gebruik van objecten, de belangrijkste zijn:

- *Assets*;
- *Mediafiles*;
- *Stills*;
- *Collections*;
- *Metadata + Technical Metadata*.

4.6.1 Assets

Wanneer een eindgebruiker via een Eind Gebruikers Applicatie (EGA) een mediabestand wil uploaden, dient er eerst een asset te worden aangemaakt door MediaMosa. Een asset vormt dan een *entry* in de database van MediaMosa. Met enkel een asset heeft men nog geen functionaliteit in de vorm van een mediabestand. Een asset bevat mediabestanden met aanvullende informatie, metadata. Door een asset, met zijn unieke asset ID als referentieobject te beschouwen maakt dit het geheel schaalbaar en flexibel. Een asset kan namelijk meerdere mediabestanden bevatten. Mediabestanden hebben een uniek ID, deze ID's worden bijgehouden in de bijbehorende asset. Later kunnen er bijvoorbeeld videobestanden in een ander bestandsformaat toegevoegd worden, het asset ID blijft gelijk en de asset is dus consistent benaderbaar.

4.6.2 Mediafiles

Het belangrijkste object voor de eindgebruiker is een mediafile, een mediabestand wat is opgeslagen in een eerder gedefinieerd bestandsformaat. Er kan worden vastgelegd welke formaten er beschikbaar moet komen van mediabestanden die geüpload worden. Deze mediabestanden worden toegekend aan een asset.

³ RESTful Web Services: <http://www.ibm.com/developerworks/webservices/library/ws-restful/>

4.6.3 Stills

Een *still* is een momentopname van een frame uit een mediabestand. Stills worden gebruikt om een mediabestand visueel te identificeren. In een EGA kan zo'n still fungeren als *thumbnail* en zo spreekt een video een gebruiker sneller aan dan een hyperlink met enkel tekst.

4.6.4 Collections

Een *collection* is een verzameling van assets, er kunnen dus meerdere assets in een collection zitten. Een asset kan deel uit maken van meerdere collections.

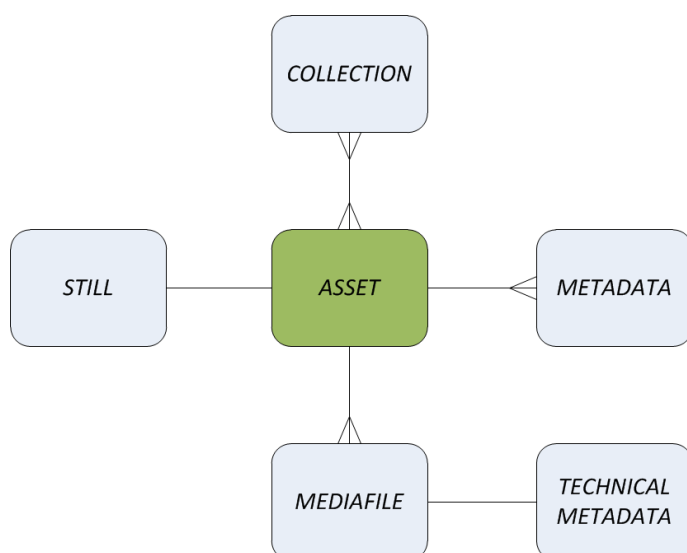
4.6.5 Metadata

Een onderdeel dat vaak onderbelicht blijft is metadata. Metadata zorgt ervoor dat bestanden benoemd kunnen worden met tal van attributen. Voorbeelden hiervan zijn titel, onderwerp, eigenaar en taal. Metadata wordt gebruikt om bestanden te beschrijven en vervolgens zoekbaar en vindbaar te maken middels MediaMosa. Men kan over honderden mediabestanden beschikken maar zonder de juiste informatie zijn deze bestanden vrijwel onbruikbaar. Het is een zeer tijdrovende en inefficiënte manier om bestanden op te slaan en proberen te beheren zonder metadata. Het invoeren van metadata is een handmatige actie en neemt tijd in beslag om het goed en zorgvuldig te doen. Er zijn enkele metadata standaarden waarin de te gebruiken attributen zijn vastgelegd. De standaarden die door MediaMosa worden ondersteund zijn: Dublin Core (DC) en Qualified Dublin Core (QDC). De DC standaard wordt nader toegelicht in de technische bijlage (Bijlage II, paragraaf 2.1).

Welke velden er beschikbaar worden gesteld ter invoer van metadata wordt gespecificeerd in de EGA, hoe dit in de praktijk in zijn werk gaat komt later aan bod bij de toelichting op Frontend Applicaties in hoofdstuk 6. Technische metadata wordt door MediaMosa gedurende de analyse van mediabestanden uitgelezen, verwerkt en opgeslagen. Voorbeelden hiervan zijn tijdsduur, bestandsformaat, *container* en resolutie, het begrip container wordt later toegelicht in Hoofdstuk 7.

4.7 Relaties

De objecten zijn met elkaar verbonden volgens de relaties die weergegeven zijn in onderstaande figuur.



Figuur 5 Contentorganisatie MediaMosa, relaties tussen objecten.

4.8 Componenten

MediaMosa bestaat uit een aantal componenten die de functionaliteit van een media distributie platform bieden. De onderstaande componenten maken deel uit van MediaMosa, de dikgedrukte componenten worden in de technische bijlage (Bijlage II, Hoofdstuk 1) toegelicht. Dit omdat zij verduidelijking brengen in de essentiële werking en opbouw van het platform welke relevant zijn voor deze opdracht.

- **Registratie**
Het registreren van frontend applicaties bij MediaMosa.
- **Authenticatie**
Toegang verlenen van frontend applicaties bij MediaMosa.
- **Autorisatie**
Het autoriseren van gebruikers en of groepen voor de juiste mediabestanden.
- **Play Proxy**
Geeft response op afspeelverzoeken van frontend applicaties.
- Zoeken
Behandelt zoekopdrachten van frontend applicaties.
- Job
Component die taken uitvoert die tijd in beslag nemen: analyseren, transcoderen, stills.
- Transcoding
Het analyseren en converteren van videobestanden door FFmpeg.
- Upload
Zorgt voor afhandeling van individuele bestand uploads en FTP Batch Uploads.
- **OAI-PMH**
Speciale *harvesters* kunnen metadata van mediabestanden hierdoor ophalen.
- Media Management
Interpreteert data van andere componenten voor verdere werking op assets/bestanden.
- Logging
Biedt functies voor het wegschrijven van logmeldingen, die is configureerbaar.
- Statistieken
Handelt statistiekverzoeken af die kunnen worden geraadpleegd via de backend.

5 Installatie en configuratie

5.1 Inleiding

MediaMosa is enkel te installeren binnen een UNIX omgeving. In overleg met de UNIX afdeling is er gekozen om voor de Linuxdistributie van Debian. Voornamelijk omdat Debian goed wordt ondersteund voor tal van *packages* en *add-ons*. De UNIX afdeling is voornamelijk gespecialiseerd in Debian en dat heeft meegewogen in het kader van ondersteuning. Specifiek is er gekozen voor de Debian 64-bit installatie zonder Graphical User Interface (GUI). Dit omdat de installaties en configuraties toch via de *commandline* oftewel *terminal* plaats zullen vinden, de GUI zou enkel een extra belasting zijn.

5.1.1 Hardware specificaties

Op de afdeling waar ik werkzaam ben beschikken we over een machine die krachtig genoeg is om tijdens de opdracht te gebruiken. De machine die gebruikt wordt is een DELL Precision R5400 met twee Xeon E5430 CPU's met een kloksnelheid van ieder 2,67 Ghz. Verder is hij uitgerust met 8GB intern geheugen. Het besturingssysteem is Windows 7 Professional 64-bit.

5.1.2 Virtuele machines

Zoals in de inleiding aangegeven wordt er gebruik gemaakt van Debian 64-bit, dit gebeurt wel op basis van Virtuele Machines (VM's). VM's kunnen met speciale software worden gemaakt en zijn complete op zichzelf staande systemen die door het fysieke systeem geëmuleerd worden. Deze keuze is gemaakt omdat virtuele machines uitermate geschikt zijn voor testdoeleinden zoals deze opdracht. Een belangrijk voordeel is dat uitbreiding met meerdere VM's zeer eenvoudig is en meerdere VM's de fysieke systeemcapaciteiten delen.

Het gewenste eindresultaat is een testomgeving waarin drie VM's aanwezig zijn, deze machines worden geëmuleerd met VirtualBox. VirtualBox is software waarmee VM's beheerd worden.

In de eerste fase is er één VM die alle onderdelen van het systeem voor zijn rekening zal nemen. Later wordt er onderzocht hoe de verschillende onderdelen verdeeld kunnen worden over verschillende VM's. Alle machines hebben toegang nodig tot dezelfde *storage*, hoe dat in de praktijk in zijn werk gaat wordt later toegelicht. De architectuur die is beoogd bij het eindresultaat (zie 3.3) is als volgt onderverdeeld:

- Applicatieserver (6.1);
- Jobserver (6.3);
- Streamingserver (7.4).

De redenen voor deze verdeling zijn beschreven in Hoofdstuk 6, mogelijkheden voor opschaling van het systeem op andere manieren wordt beschreven in Hoofdstuk 10.

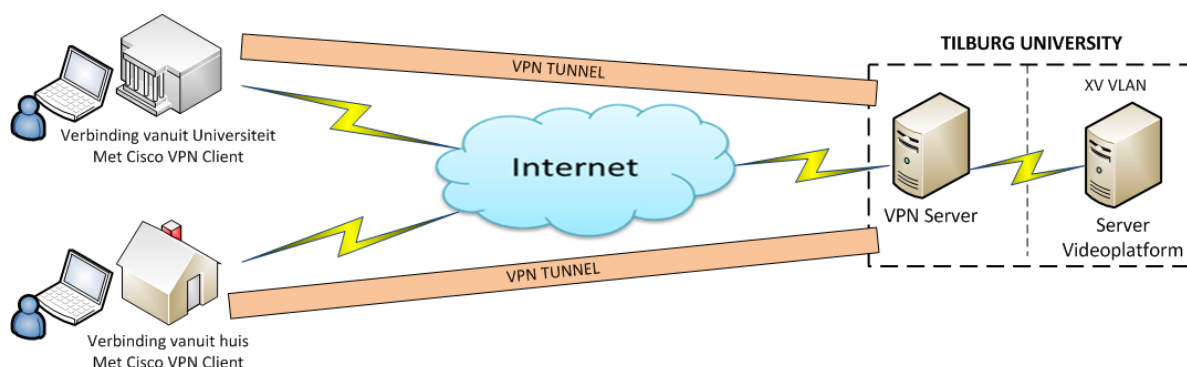
5.1.3 Netwerkomgeving

Vanwege veiligheidsoverwegingen en omdat het om een testopstelling gaat waarin niet alle facetten voor de buitenwereld per definitie al behandeld worden is er gekozen alle virtuele machines in een Virtueel Netwerk te plaatsen (VLAN). Een VLAN is een afgebakend netwerk binnen een bestaand fysiek netwerk. Het zorgt ervoor dat niet iedereen toegang heeft tot de machines binnen zo'n netwerk en dat maakt het uitermate geschikt voor testdoeleinden. Het VLAN netwerk is een schil die op basis van MAC adressen machines authenticceert. Er bestond al een 'Video VLAN' dat binnen TiU wordt aangeduid als het 'XV VLAN', dit VLAN is gebruikt tijdens de uitvoering van de opdracht.

5.2 Tools

Om de opdracht uit kunnen voeren heb ik enkele *tools* gebruikt, hieronder volgt een opsomming met korte toelichting:

Cisco VPN Client: gebruikt om verbinding te maken met het TiU netwerk. Wegens een speciaal *profile* is er toegang mogelijk tot het juiste XV VLAN binnen het TiU netwerk. Overal waar een internetverbinding aanwezig is kan er verbinding gelegd worden met de machines in dit VLAN. Dit zorgt ervoor dat er vrijwel overal aan de opdracht gewerkt kon worden. In onderstaande figuur zijn ter verduidelijking de scenario's weergegeven bij een verbinding vanuit huis en vanaf de werkplek. De verbinding met deze VPN Client dient opgezet te zijn vóór de andere *tools* gebruikt kunnen worden.

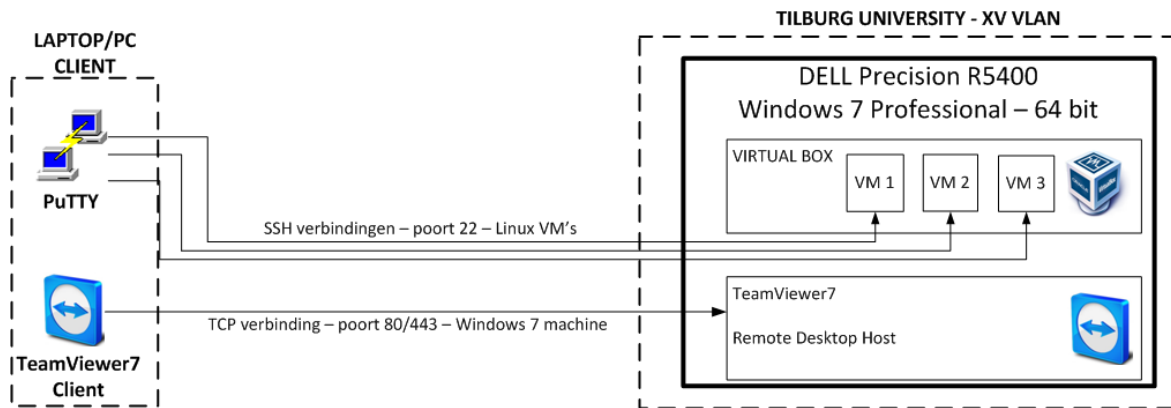


Figuur 6 Toegang tot de testomgeving via VPN client.

PuTTY: gebruikt om direct verbinding te leggen met de verschillende VM's. PuTTY maakt verbinding op basis van een IP adres via poort 22, het verkeer op poort 22 vindt plaats via het SSH protocol. SSH staat voor *Secure Shell* wat het mogelijk maakt om op een versleutelde manier in te loggen op een andere computer.⁴ De *client* maakt gebruik van een *public key* welke vooraf op elke Linux server in het *authorized_keys* bestand toegevoegd wordt.

⁴ Informatie over Secure Shell(SSH): http://en.wikipedia.org/wiki/Secure_Shell

TeamViewer 7 / Windows: gebruikt om verbindingen van buitenaf mogelijk te maken naar de fysieke machine. Op de fysieke machine waar de VM's op draaien is TeamViewer geïnstalleerd. Er kan met behulp van een gedeeld ID en wachtwoord verbinding gemaakt worden. Vooral nuttig om *snapshots* binnen VirtualBox te maken op afstand. Een *snapshot* is een vastgelegde status van een virtuele machine op een bepaald moment. Wanneer er iets misgaat, kan op deze manier snel systeemherstel plaatsvinden, ideaal voor prototypes en testdoeleinden als dit project.



Figuur 7 Tools om toegang te krijgen tot de server

6 MediaMosa Servers

In dit hoofdstuk wordt beschreven hoe de testomgeving tot stand is gekomen. In de eerste paragraaf (6.1) staat beschreven hoe de omgeving is opgebouwd na installatie van de eerste VM. In de tweede paragraaf (6.2) wordt aandacht besteed aan de verschillende frontend applicaties die beschikbaar zijn, en de keuze voor de frontend applicatie wordt toegelicht. In de derde paragraaf (6.3) staat de uitbreiding met een tweede VM beschreven. Deze zal de eerste machine, de applicatieserver, ontlasten van processorbelastende taken. In hoofdstuk 7 staat de *streamingserver* beschreven. Dit is de derde en laatste VM zoals die staat weergegeven bij het gewenste eindresultaat paragraaf 3.3.

6.1 Applicatieserver

De applicatieserver is de eerste virtuele machine die geïnstalleerd is:

- Aantal CPU's: 2;
- Intern geheugen: 1GB;
- Harddisk: 25GB (fixed),
- Netwerkadaptor: bridge naar fysieke adaptor machine.

Van de Debian website kan zeer eenvoudig en uiteraard gratis het meest recente *image* in de vorm van een ISO-bestand gedownload worden.

6.1.1 Extra software

Tijdens de Debian installatie kan er op een bepaald moment software worden geselecteerd die optioneel mee geïnstalleerd kan worden. Apache 2.2 webserver is nodig voor MediaMosa dus deze installeert meteen mee. Om toegang met PuTTY mogelijk te maken via SSH is de SSH module in Debian nodig, ook deze wordt meteen tijdens de installatie mee geïnstalleerd.

De volgende additionele componenten zijn nodig voor MediaMosa:

- PHP 5.2.x;
- MySQL 5.x;
- FFmpeg;
- Lua.

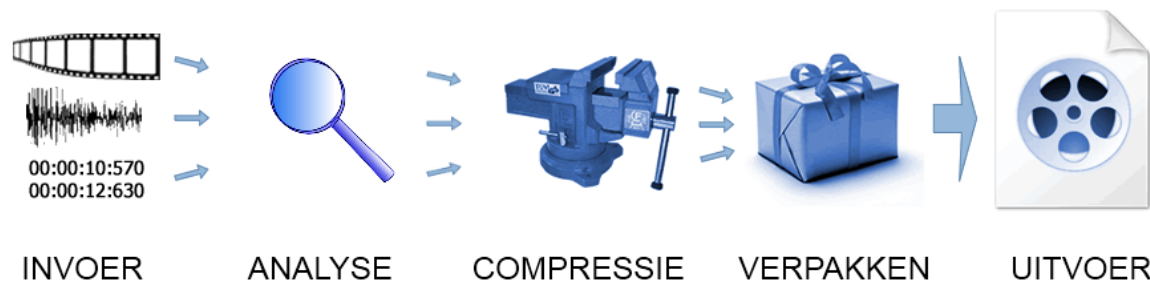
De eerste twee componenten zijn relatief bekend, de laatste twee wat minder deze worden hier extra toegelicht.

6.1.2 FFmpeg

FFmpeg is een belangrijke component. Het voegt een cruciale functionaliteit toe aan MediaMosa, namelijk het analyseren en transcoderen van mediabestanden. De code van MediaMosa roept via Lua (zie volgende paragraaf) FFmpeg aan met parameters die in de backend via de *web interface* kunnen worden ingesteld. Het transcoderen dat FFmpeg voor zijn rekening neemt werkt met een invoer en een uitvoerbestand, FFmpeg is zeer uitgebreid en er zijn tal van parameters mogelijk om een gewenst uitvoerbestand te krijgen.⁵

⁵ FFmpeg documentatie: <http://ffmpeg.org/ffmpeg.html>

FFmpeg maakt gebruik van codecs om bestanden van en naar een bepaald bestandsformaat te transcoderen. In de figuur hieronder is de werking van FFmpeg en het principe van transcoderen weergegeven. Informatie over codecs wordt toegelicht in het volgende hoofdstuk.



Figuur 8 Basisprincipe transcoderen (FFmpeg)

6.1.3 Lua

Lua wordt omschreven als een *multi-paradigm* programmeertaal.⁶ Een aantal scripts binnen de MediaMosa omgeving zijn geschreven in Lua. Code die geschreven is in Lua dient te worden uitgevoerd door de virtuele machine van Lua. Dit houdt in dat de code uit een script niet direct wordt uitgevoerd maar eerst wordt gecompileerd naar bytecode en vervolgens wordt uitgevoerd door de virtuele machine. Voor de scripts in MediaMosa maakt Lua gebruik van een extra *library*, de *Parsing Expression Grammars* (LPeg).⁷ Lua wordt gebruikt met de aanvullende LPeg om FFmpeg aan te spreken en gegevens van FFmpeg te destilleren. De scripts worden gebruikt om videobestanden te analyseren te transcoderen via FFmpeg. De gegevens die tijdens de analysefase worden gedestilleerd uit FFmpeg door het Lua script worden gebruikt om bij het betreffende mediabestand de technische metadata in te vullen.

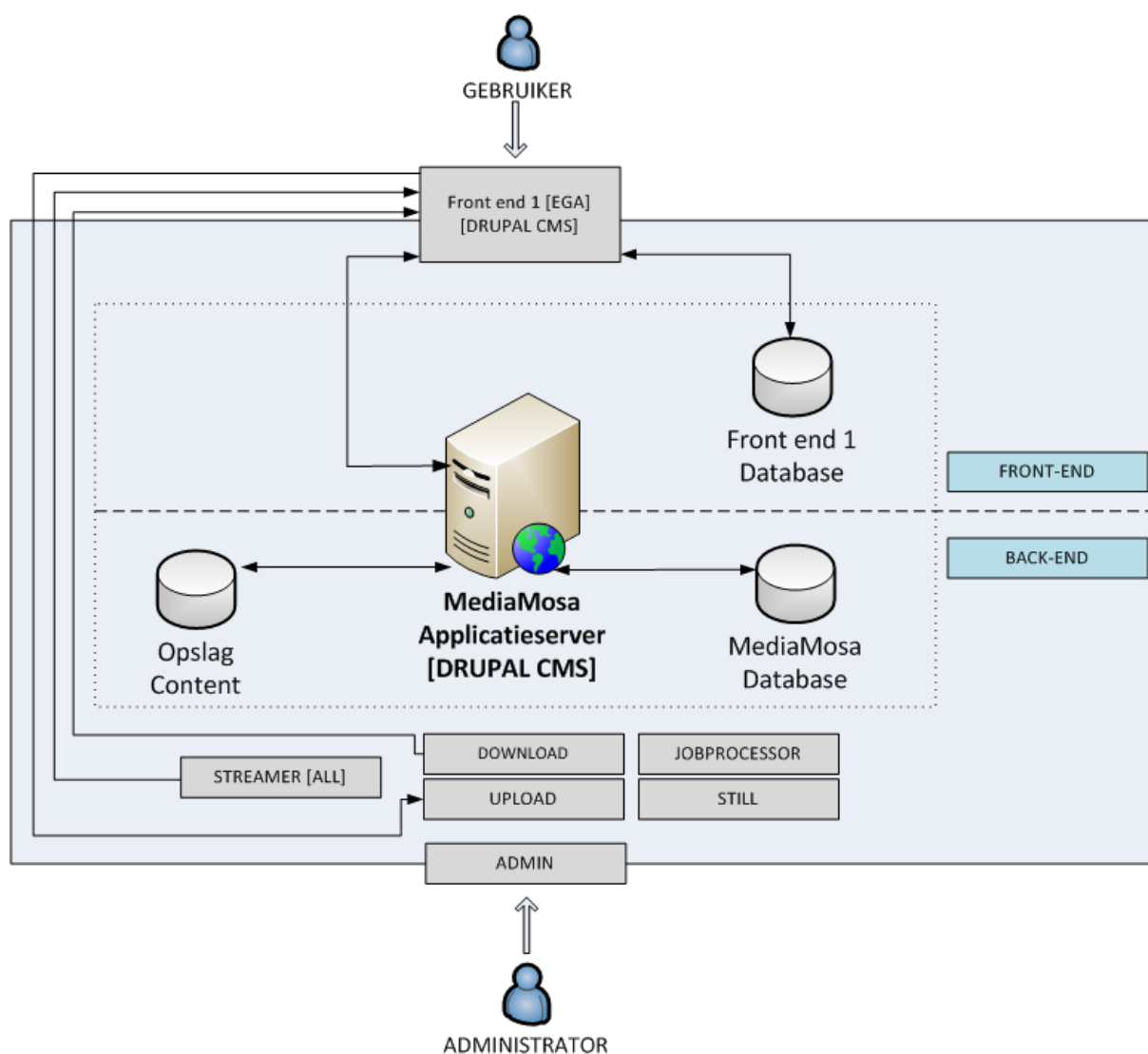
6.1.4 Opbouw

De functionaliteiten in MediaMosa kunnen gescheiden en verspreid worden over meerdere servers. Dit is aan te raden voor productieomgevingen en naar mate het gebruik kan men capaciteit flexibel opschalen. De volgende componenten zijn aan verschillende servers toe te wijzen.

- **Download**
Zorgt voor de afhandeling van downloads van MediaMosa naar eindgebruiker.
- **Jobprocessor**
Zorgt voor het analyseren en transcoderen van mediabestanden en kan stills genereren uit mediabestanden.
- **Still**
Het genereren van stills kan ook gescheiden worden van de jobprocessor.
- **Streaming**
Streaming servers zorgen voor het streamen van mediabestanden naar eindgebruikers, per codec, container en/of protocol kan naar wens een streaming server worden aangemaakt.
- **Upload**
Zorgt voor de afhandeling van uploads van eindgebruiker naar MediaMosa.

⁶ Toelichting Lua, Wikipedia: [http://en.wikipedia.org/wiki/Lua_\(programming_language\)](http://en.wikipedia.org/wiki/Lua_(programming_language))

⁷ Toelichting LPeg library: <http://www.inf.puc-rio.br/~roberto/lpeg/>



Figuur 9 Testomgeving na installatie eerste virtuele machine.

In de afbeelding is de architectuur te zien na installatie van de eerste VM. Er is nu één server in gebruik die in alle componenten voorziet, inclusief de opslag van de mediabestanden. In de architectuur die is weergegeven bij de gewenste eindsituatie is te zien wat de uiteindelijke opbouw van het prototype is. Daar is een verdeling gemaakt in de componenten voor de uiteindelijke architectuur bij oplevering van het project. In dit project moet op deze manier van opbouwen en scheiden van functionaliteiten duidelijk worden wat de mogelijkheden en werkwijzen zijn voor dergelijke scenario's. Zowel MediaMosa (backend) als de EGA (frontend) zijn geïnstalleerd op deze server. Beiden zijn Drupal installaties met ieder zijn eigen database.

In Hoofdstuk 10 word een mogelijk opschalingsscenario weergegeven waarin de eerder genoemde componenten verder worden verdeeld over seperate servers.

6.2 Frontend applicaties

Op de MediaMosa website worden enkele *templates* aangeboden waarmee relatief snel een werkende frontend applicatie gerealiseerd kan worden. Tijdens de opdracht zijn er drie beschikbare frontends getest welke hieronder kort toegelicht worden. De frontends die beschikbaar worden gesteld zijn allen volledige Drupal installaties die gebruik maken van een speciale *connector module*. Deze module zorgt ervoor dat de frontend met MediaMosa (backend) kan communiceren via het REST-protocol. De componenten die dit mogelijk maken zijn *Registratie* en *Authenticatie* welke beiden worden toegelicht in de technische bijlage (Bijlage II).

6.2.1 MediaMosa White Label EGA (WLE)

De WLE is de eerste template die is uitgebracht voor MediaMosa maar inmiddels wat gedateerd. Toch zeer overzichtelijk in gebruik en daardoor efficiënt om opbouw mee te onderzoeken en te testen. Deze frontend is gebruikt om de opties van metadateren en de mogelijkheid om bestanden naar wens als *download* beschikbaar te stellen, te onderzoeken en uit te voeren. In de technische bijlage (Bijlage II, Paragraaf 2.2) staat weergegeven hoe en waarom de implementatie van deze downloadfunctionaliteit heeft plaatsgevonden.

6.2.2 MediaMosa Construction Kit (CK)

Uitbreiding van de eerdere WLE en is beschikbaar voor zowel Drupal 6 als 7. Na wat onderzoek bleek dat de versie voor Drupal 7 nooit is uitgebracht, althans niet stabiel en daarom werd deze afgeraden voor gebruik. De versie voor Drupal 6 is wel getest en hier bleken wat onduidelijkheden in te zitten. Zo bleef er bij het aanmaken van een nieuwe asset altijd een lege onbekende asset over. De interface en de configuratieopties waren te onduidelijk, zodoende bood deze frontend tijdens deze opdracht weinig meerwaarde ten opzichte van de eerdere WLE.

6.2.3 MediaMosa Site Builder (SB)

De Site Builder is de lang verwachte opvolger van de Construction Kit die uiteindelijk pas in mei van dit jaar(2012) als stabiele versie beschikbaar is gesteld. De Site Builder is een Drupal 7 installatie en biedt daardoor enkele nieuwe grafische functionaliteiten. Via de ontwikkelaars is een eerdere versie verkregen en deze is geïnstalleerd en getest. Echter waren er enkele functionaliteiten nog niet werkend waaronder de downloadfunctionaliteit die juist expliciet tijdens deze opdracht naar voren moest komen. Waarschijnlijk zal deze frontend de standaard worden en zullen komende *releases* van MediaMosa voorbereid gaan worden voor gebruik met deze Site Builder.

6.2.4 Keuze

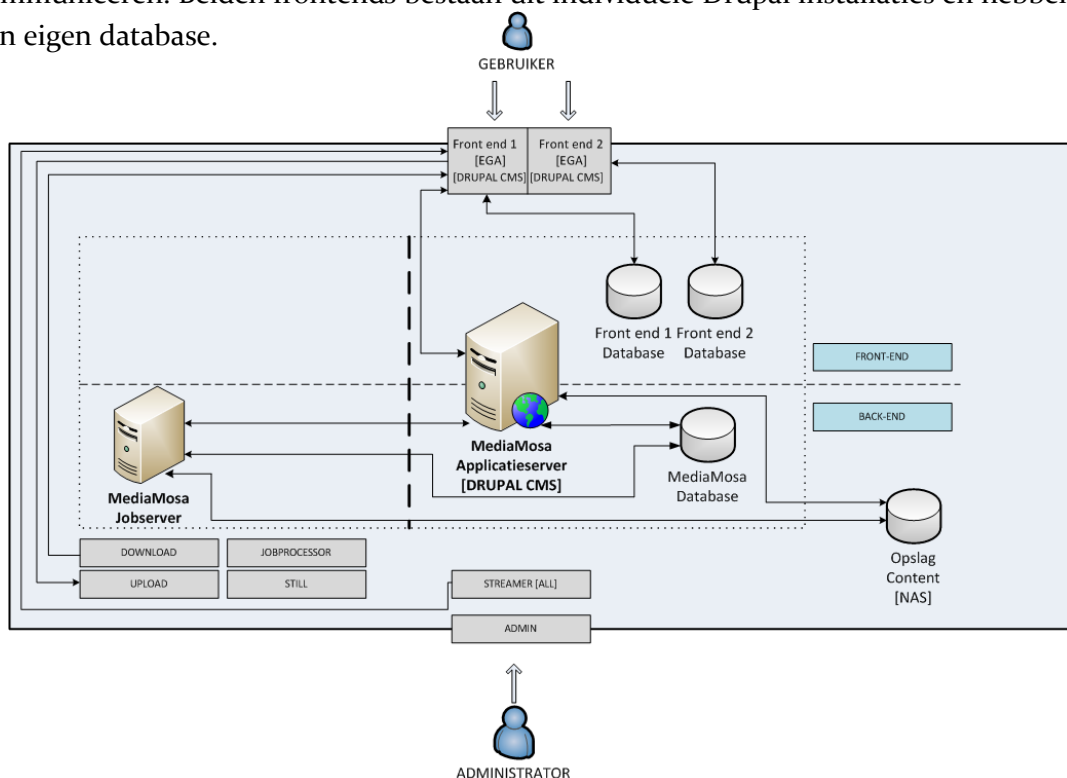
Omdat tijdens het uitvoeren van de opdracht de *release* van de Site Builder een nog onbekende vertraging heeft opgelopen, is er gekozen om de WLE te gebruiken om functionaliteiten uitgebreid te testen. De verschillen tussen de verschillende versies zit hem met name in de grafische interface en het gebruik van verschillende modules die kenmerkend zijn voor Drupal versie 7.

6.3 Jobserver

Om de applicatieserver niet te belasten met transcodeertaken van FFmpeg is er besloten deze functionaliteit door een separate server uit te laten voeren. MediaMosa werkt met een *jobmechanisme* dat de *jobs* zoals transcoderen en het genereren van *stills* aanmaakt en verdeelt over beschikbare *Job Processors*. De jobserver bestaat uit een complete MediaMosa installatie die moet worden aangesproken door de applicatieserver, de specificaties zijn gelijk aan die van de applicatieserver. Een belangrijke voorwaarde van het spreiden van functionaliteiten over meerdere servers is dat alle servers de beschikking moeten hebben over de mediabestanden. Ook moet de data uit de centrale MediaMosa database beschikbaar zijn voor de jobserver.

Om bestanden toegankelijk te maken voor meerdere servers is er gekozen voor een centrale externe opslag in de vorm van een Netwerk Harde Schijf (NAS). Deze NAS was aanwezig binnen de afdeling en wordt gebruikt voor backup doeleinden. Op deze NAS met een apart aangemaakte *share* kunnen de servers bestanden opslaan en raadplegen. Omdat er een aparte *share* wordt gebruikt, komen de bestanden die hier staan als backup niet in gevaar. Op beide servers is een *mount point* aangemaakt. Een *mount point* in Linux is een verwijzing naar een (netwerk)locatie. Door een netwerklocatie zoals de gebruikte NAS te *mounten* lijkt het voor het systeem dat deze bestanden op het systeem aanwezig zijn. Het *mountpoint* doet zich voor als een lokaal volume. Voor de databasekoppeling heeft de jobserver toegang gekregen tot de database op de applicatieserver.

Onderstaande figuur geeft de architectuur van de testomgeving in de tweede fase weer. In de figuur is te zien dat er ten opzichte van de eerste fase er een scheiding tussen de twee servers is. Rechtsonder is de NAS weergegeven die dienst doet als externe opslag. De toegevoegde server neemt nu alle taken, die in de paragraaf 6.1.4 genoemd zijn, voor zijn rekening en ontlast hiermee de applicatieserver. De applicatieserver geeft de jobserver opdrachten tot uitvoering. Er zijn twee frontend applicaties actief die beiden met één MediaMosa backend communiceren. Beiden frontends bestaan uit individuele Drupal installaties en hebben ieder een eigen database.



Figuur 10 Testomgeving na installatie ze VM, scheiding tussen applicatieserver, jobserver en opslag.

7 Streaming Video

In dit hoofdstuk worden de verschillende onderdelen die betrekking hebben op *streaming* video nader toegelicht. Het *streamen* van een videobestand is het verzorgen van de weergave van het opgevraagde videobestand naar de eindgebruiker. Dit kan echter op een aantal verschillende manieren. Videobestanden kunnen gecodeerd/gedecodeerd worden volgens verschillende codecs. Containerformaten kunnen verschillende codecs bevatten. Hoe dit in zijn werk gaat wordt nader toegelicht in paragraaf 7.1 en 7.2. Verder vindt er in dit hoofdstuk toelichting plaats op het streamen van videobestanden. In paragraaf 7.3 wordt uitgelegd hoe MediaMosa standaard omgaat met streaming video. Om het streamen van videobestanden uit te besteden was het vooraf bekend dat Wowza gebruikt ging worden. Toelichting op Wowza en de bevindingen van Wowza in combinatie met MediaMosa worden in dit hoofdstuk toegelicht vanaf paragraaf 7.4.

7.1 Codecs

Het woord codec komt van coderen en decoderen. Een codec is een bestand dat gegevens bevat over hoe een video opgeslagen (gecodeerd) moet worden en hoe een bestand afgespeeld (gedecodeerd) moet worden. Een codec moet aanwezig zijn op een computer om materiaal te kunnen verwerken. Tijdens het coderen en decoderen vindt er compressie en decompressie plaats, er wordt namelijk gecodeerd omdat men een bestand wil verkleinen in bestandsgrootte en zo min mogelijk kwaliteit daarbij wil verliezen. Zowel audio als video wordt via een bepaalde codec gecodeerd, bij het coderen zijn er tal van parameters van toepassing. De balans die gevonden moet worden bij het coderen van een bestand is kwaliteit vs. bestandsgrootte. Hoe hoger men de kwaliteit wenst hoe groter het bestand zal worden. De factoren die bepalend zijn voor de kwaliteit van een videobestand zijn de resolutie en de *bitrate* van een bestand. De resolutie wordt uitgedrukt in een formaat van horizontale pixels bij verticale pixels (bijvoorbeeld 1280 x 720). De bitrate wordt uitgedrukt in kbps of Mbps, dit wil zeggen hoeveel kilobits of Megabits er per seconde worden opgeslagen. Een hogere waarde resulteert in een groter bestand van een hogere kwaliteit. Omdat er in de huidige situatie gebruik wordt gemaakt van de WMV codec en deze niet compatibel genoeg is voor moderne apparaten en ook qua compressietechniek verouderd is dient er een nieuwe codec voor de toekomst gevonden te worden. In hoofdstuk 8 wordt er verder ingegaan op de keuze voor een nieuwe codec.

Hedendaagse populaire codecs zijn :

Video

- H.264;
- Theora;
- VP8.

Audio

- Advanced Audio Coding (AAC);
- MPEG-1, Audio Layer 3 (MP3);
- Vorbis.

7.2 Containers

Een container staat voor het bestandsformaat van een videobestand, een container bevat veelal verschillende codecs. Zo is aan de extensie van een bestand niet te zien met welke codec het bestand gecodeerd is. De termen containers en codecs worden vaak te pas en te onpas door elkaar heen gebruikt. Een container definieert hoe data opgeslagen moet worden binnen het betreffende containerformaat. Meerdere codecs kunnen binnen één containerformaat opgeslagen worden. Verder is er één codec aanwezig voor de *videostream* en één codec voor de *audiostream*. In een container wordt ook gedefinieerd hoe de metadata wordt weggeschreven. In onderstaande tabel is weergegeven hoe hedendaagse, populaire codec en containers zicht tot elkaar verhouden, ook zijn de bestandsextensies weergegeven.

	Container	Video codec	Audio codec	Extensie(s)
1	MP4	H.264 / MPEG-4	AAC	.mp4 / .m4v
2	Ogg	Theora	Vorbis	.ogv
3	WebM	VP8	Vorbis	.vp8

Figuur 11 Overzicht populaire containers en bijbehorende codecs en extensies.

7.3 Streaming binnen MediaMosa

In MediaMosa staan standaard servers gedefinieerd die het mogelijk maken om videobestanden te streamen. Hierbij wordt gebruik gemaakt van *MIME types*, deze types geven aan wat voor materiaal er aangeroepen wordt. MediaMosa verstuurt deze bestanden via HTTP requests naar de client. De players die gebruikt worden voor het afspelen dienen aanwezig te zijn op het apparaat van de gebruiker, dus aan de clientzijde. Aan de hand van het gedetecteerde MIME type (bijv. mp4 of flv) weet MediaMosa welke player er aangeroepen moet worden aan de clientzijde. Het gebruik van deze streaming functionaliteiten zijn slechts aan te raden op kleine schaal en/of voor testdoeleinden. Het is zaak dat MediaMosa het streamen van mediabestanden uit gaat besteden aan streaming servers (ook wel streamers genoemd). Streamers zijn geschikt om grote hoeveelheden eindgebruikers te bedienen met streaming video. Bekende videostreamingsoftware is Wowza Media Server. Gegeven is dat er al meerdere configuraties elders zijn gerealiseerd met MediaMosa icm Wowza (SURFnet, Universiteit van Amsterdam).

7.4 Wowza Media Server

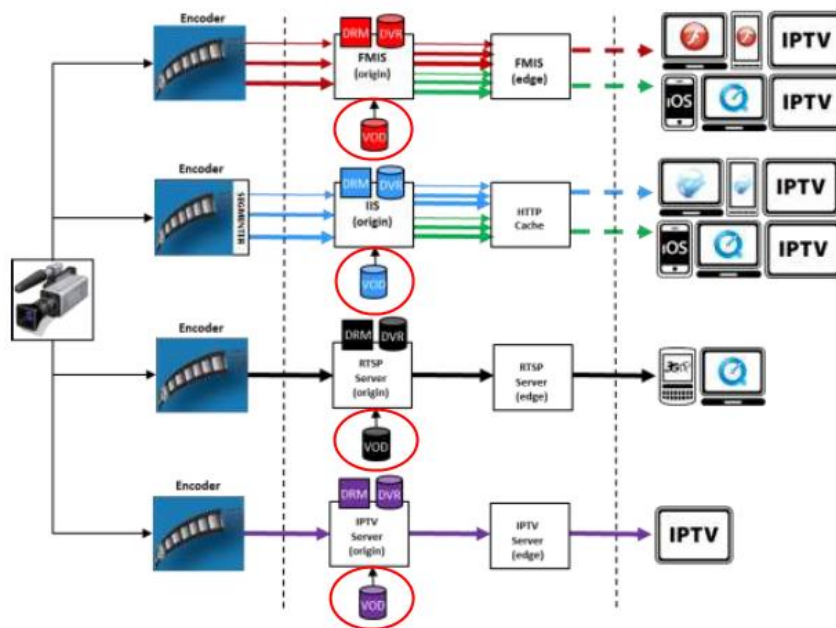
Wowza is een belangrijke speler in de wereld van online video. Eind 2011 lanceerden ze versie 3 van hun product Wowza Media Server 3 (WMS3).⁸ Het is één van de meest geavanceerde producten op het gebied van videodistributie. On-line video kan grofweg verdeeld worden in twee varianten: *On Demand Video* en *Live Video*. Tijdens de opdracht zullen enkel de mogelijkheden van *On Demand Video* onderzocht worden.

De mogelijkheden van live videostreaming worden achterwege gelaten, omdat het niet is zinvol om dat voor het toekomstige platform te onderzoeken.

⁸ Website Wowza: <http://www.wowza.com/>

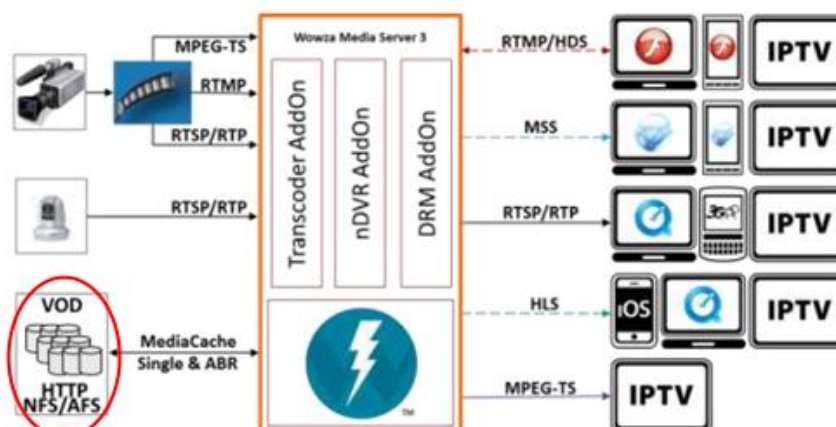
Het gaat hier om het on demand aanbieden van videomateriaal wat op elke gewenst moment geraadpleegd moet kunnen worden. Live video streaming wordt op dit moment op andere manieren gedaan binnen TiU en deze worden voorlopig gehandhaafd.

Wowza werkt volgens een nieuw principe wat ze het “Any Screen Done Right” principe noemen. Het komt erop neer dat er naar (vrijwel) elk apparaat *gestreamd* kan worden vanuit één enkel bestand. Wowza zorgt voor de afhandeling naar de verschillende apparaten/browsers en is in staat simultaan één bestand naar meerdere clients te sturen die allen een ander protocol kunnen eisen.



Figuur 7 Traditionele manier van streaming servers met seperate bestanden(opslag is rood omcirkeld)

Een goede vergelijking is te zien als men bovenstaande figuur bekijkt, dit is hoe traditioneel videobestanden door *streamers* werden aangeboden. Aan de rechterzijde is te zien dat er op deze manier ook tal van apparaten te bedienen zijn. Echter de roodomcirkelde figuren geven de opslag aan, elk protocol vereist een eigen encoder en dit resulteert in een videobestand. Te zien is dat het in dit geval resulteert in vier videobestanden. Uitermate inefficiënt qua storage en beheer ten opzichte van onderstaande figuur. Daar is linksonder de opslag weergegeven van één enkel videobestand.



Figuur 8 Streamen met WMS 3, multiclient distributie met één videobestand(opslag is rood omcirkeld)

Volgens de aanbevolen specificaties is Wowza is geïnstalleerd op een virtuele machine met de volgende specificaties:

- Aantal CPU's: 2;
- Intern geheugen: 2GB;
- Harddisk: 25GB (fixed),
- Netwerkadapter: bridge naar fysieke adapter machine.

7.5 Streaming protocollen

Het sturen van een bestand naar een eindgebruiker wordt streaming genoemd. Een streaming server (of *streamer*) dient via een protocol een bestand af te spelen naar het apparaat van de eindgebruiker. Een protocol is gerelateerd aan een player, oftewel een protocol zorgt ervoor dat de (media)server en de player met elkaar kunnen praten, elkaar begrijpen en zo dus functionaliteiten kunnen uitwisselen.

7.5.1 RTMP: Real-Time Messaging Protocol

RTMP is een protocol gebaseerd op het TCP protocol (Transport Controlled Protocol) het zorgt voor een verbinding tussen client en server en kan bidirectioneel met data omgaan. De server deelt het videobestand op in verschillende blokken en zendt deze gecontroleerd achter elkaar door. Controle vindt plaats door TCP, RTMP zorgt voor de afhandeling van de videostream. De meeste bekende toepassing die gebruik maakt van RTMP is de Adobe Flash Player. RTMP connecties vinden standaard plaats via poort 1935.

7.5.2 Adaptieve Bitrate Streaming

Het principe van Adaptieve Bitrate Streaming is dat de prestaties van de clientapplicatie worden geanalyseerd en de server daar met een passende videostream op reageert. Bij de client wordt er met name gekeken naar beschikbare CPU capaciteit en beschikbare bandbreedte. De client opent een indexbestand (XML of anders) en hier zijn de URL's met bijbehorende bitrates beschreven. Dit bestand wordt ook wel een *manifest* genoemd en de aanroep verschilt per protocol. De client roept dus niet direct een videobestand aan maar werkt via een indexbestand dat informatie heeft over de beschikbare versies van de videobestanden. Het resultaat is dat de aangeroepen video vrijwel direct begint, zonder het bufferen van een grote videostream. Er wordt herhaaldelijk gekeken naar de prestaties van de client en zo kan er aan de hand van die prestaties de bitrate naar gelang van de vertoning wijzigen. Dit gebeurt naadloos zodat de gebruiker er geen last van ondervindt. Het belangrijk voordeel van dit principe is dat het verkeer plaats vindt op basis van het HTTP protocol (poort 80). Dit zorgt ervoor dat dit protocol eventuele firewall problematiek omzeilt, omdat er gebruik wordt gemaakt van de standaardpoort 80 waar al het reguliere internetverkeer over geschied.

De adaptieve protocollen die Wowza ondersteunt zijn:

- Flash HTTP Streaming (HDS);
- Apple HTTP Live Streaming (HLS), ook wel bekend als Cupertino Streaming;
- Smooth Streaming (Microsoft Silverlight).

7.6 JWplayer

Wowza wordt gebruikt in combinatie met een player, dit is een stuk software wat ervoor zorgt dat videobestanden afgespeeld kunnen worden bij de eindgebruiker. Een belangrijk verschil is dat de player niet aanwezig hoeft te zijn op het apparaat van de client maar aan de serverzijde geïnstalleerd is.

Tijdens het onderzoek naar de aspecten van streaming video is het duidelijk geworden dat de materie vrij complex is. Bij de Universiteit van Amsterdam en SURFnet is er navraag gedaan met betrekking tot hun ervaringen en expertises. Een goede optie die zij hebben aangedragen is de JWplayer⁹, de player biedt namelijk sinds versie 5 ondersteuning voor HTML5 en de ondersteuning in combinatie met Wowza is ook bekend. Echter de combinatie met Wowza en de Flash modus van JWplayer is bekend, wanneer de video's overgebracht worden via het eerder genoemde RTMP protocol. Meer toelichting over HTML5 en streaming video staat beschreven in het volgende hoofdstuk.

7.7 Nadelen Wowza / MediaMosa

Op basis van één MP4 bestand zorgt Wowza voor de afhandeling naar de verschillende clients. In combinatie met MediaMosa is gebleken dat dit een aantal nadelen met zich meebrengt. Bij het gebruik van HTTP Adaptieve streaming moet de Wowzaserver vanuit MediaMosa aangeroepen worden vanuit een script. Dit script is de object code die de player met bijbehorende attributen en parameters aan zal roepen. De gebruikte player is zoals eerder vermeld JWplayer, JWplayer roept de streaming server aan op basis van een streaming protocol. Onderstaande tabel laat deze drie adaptieve protocollen zien met de bijbehorende aanroep, inclusief het RTMP protocol en de aanroep die werkend is op basis van Flash.

Protocol	Voorbeeld URL MediaMosa
Flash RTMP (traditioneel en werkend)	rtmp://137.56.72.176:1935/mediamosa/mp4:{TICKET_URI}
Flash HTTP streaming (Flash player)	http://137.56.72.176:1935/mediamosa/mp4:{TICKET_URI}/manifest.f4m
Apple HLS (Cupertino)	http://137.56.72.176:1935/mediamosa/mp4:{TICKET_URI}/playlist.m3u8
Smooth Streaming (Silverlight)	http://137.56.72.176:1935/mediamosa/mp4:{TICKET_URI}/Manifest

Het verschil in aanroepen is duidelijk zichtbaar, ieder HTTP protocol heeft zijn eigen suffix achter de reguliere URL. Door zo een aanroep uit te voeren zou Wowza moeten reageren met de bijbehorende manier van streamen.

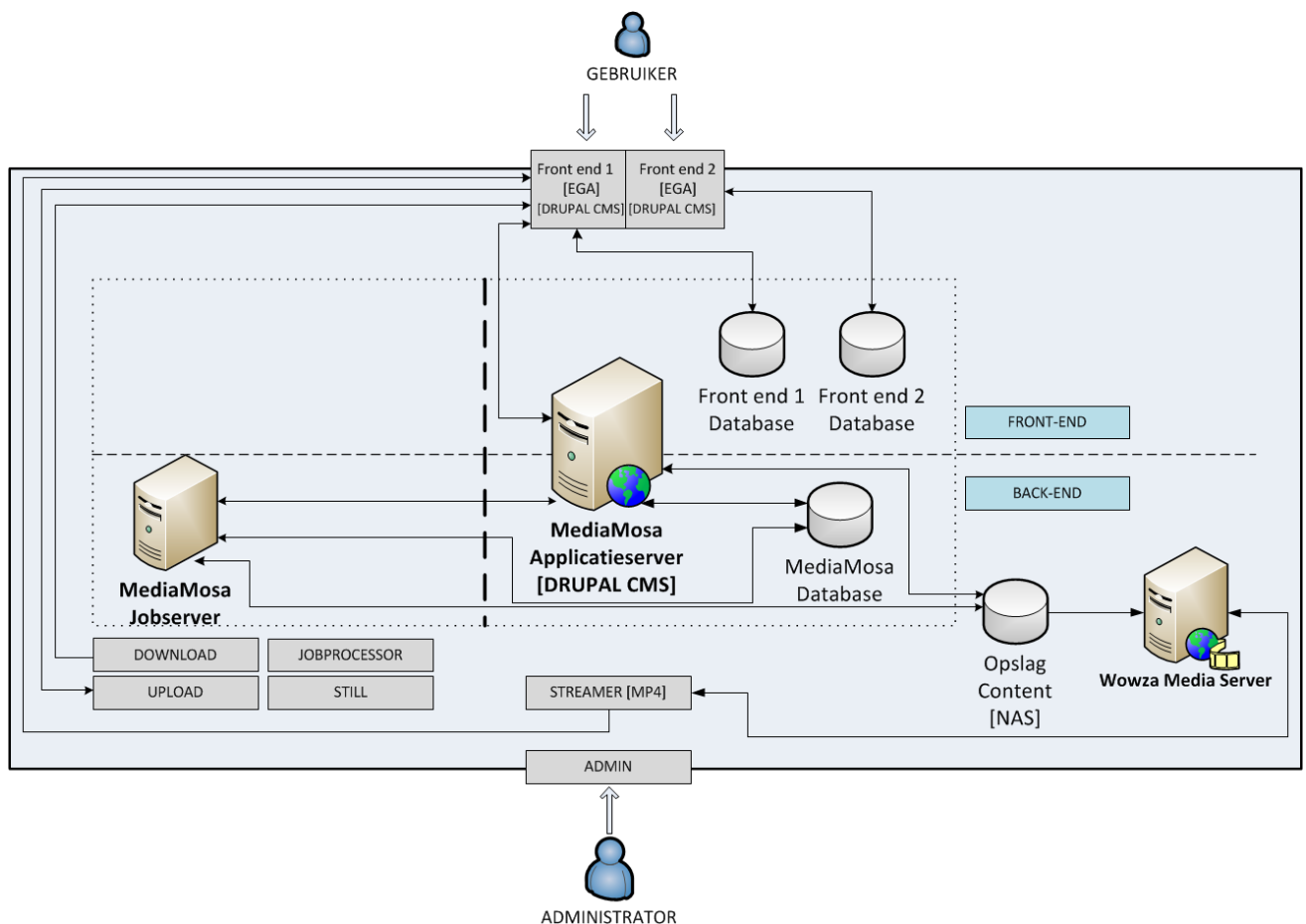
Tijdens het testen is gebleken dat Wowza niet goed omgaat met de aanroep vanuit MediaMosa. Dit lijkt niet zozeer een gebrek in Wowza maar een tekortkoming of een onduidelijkheid in MediaMosa. In de laatste versie van MediaMosa is er al wel rekening gehouden met Adaptieve Streaming maar niemand heeft me kunnen helpen met specifieke vragen over deze implementatie. Het vermoeden is er dat het mis gaat in de bestandsopslag van MediaMosa. Een test met HTTP Flash streaming met een bestand wat fysiek aanwezig is op de Wowza server werkt wel met de betreffende aanroep.

⁹ Website JWplayer: <http://www.longtailvideo.com/players/>

Zodra er een URL uit bovenstaande tabel gebruikt wordt, wordt er geen bestand afgespeeld en verschijnt de foutmelding dat het bestand niet gevonden kan worden. In hoofdstuk 8 wordt verder ingegaan op de dilemma's van online video en wordt deze tekortkoming nog belicht.

7.8 Resultaat

De combinatie Wowza met MediaMosa is werkend opgeleverd, streaming vindt plaats via het RTMP protocol naar de JWplayer in Flash modus. Echter zijn de nadelen die in de vorige paragraaf worden weergegeven nog wel zaken die met de ontwikkelaars van MediaMosa bekeken moeten worden. Tijdens de opdracht is er veelvuldig contact geweest met Tom Kuipers, ICT ontwikkelaar aan de Universiteit van Amsterdam (UvA). Ook hij ondervond problemen met deze implementatie en het gebruik van adaptieve streaming. Tijdens het testen van de mogelijkheden van Wowza was het helaas onmogelijk om te testen met mobiele apparaten. Omdat er gebruik gemaakt wordt van een Cisco VPN client om toegang te kunnen krijgen tot het juiste VLAN en de servers was dit voor mobiele apparaten ook nodig. Echter om mobiele apparaten binnen TiU toegang te geven tot een VLAN middels VPN is een *Mobile VPN Client* nodig. Binnen TiU heeft de netwerkdienst hier geen licenties voor beschikbaar wat testen met apparaten als tablets en smartphones onmogelijk maakte. Hieronder is de architectuur te zien na implementatie van de Wowza Media Server.



Figuur 9 Architectuur omgeving 3e fase, Wowza is toegevoegd en handelt de streamingverzoeken af.

8 Dilemma's in online video

Het belangrijkste aspect van streaming videodistributie is wel dat het mogelijk maakt om videobestanden af te spelen op een apparaat. Dit klinkt logisch en vanzelfsprekend maar dit is net het cruciale punt waarom sommige players, codecs of containers beter wel of niet gebruikt kunnen worden. Elk apparaat en elke webbrowser ondersteunt een beperkt aantal codecs. Waarom? Omdat er zwaarwegende belangen zijn voor elke producent, fabrikant en ontwikkelaar om een specifieke codec, container of player wel of niet te ondersteunen.

8.1 FLASH vs HTML5



Een van de meest bekende mogelijkheden om video te streamen is via Flash players, dit gaat via het eerder beschreven RTMP protocol waar Wowza goed geschikt voor is. Maar de laatste jaren zijn er grote spelers in de markt die het afspelen/weergeven van Flash materiaal niet ondersteunen. De meeste bekende en grote speler is Apple, Apple ondersteunt met zijn iOS apparaten (iPad, iPhone, iPod touch) waarop de Safari browser aanwezig is geen Flash. Dit resulteert dus in zwarte pagina's, foutmeldingen en verzoeken om Flash op het apparaat te installeren (wat niet mogelijk is). Omdat een toekomst vaste oplossing een belangrijk aspect is van het nieuwe videoplatform is er onderzoek gedaan wat de beste keuze is, en of wellicht beide opties mogelijk zijn.

Om meer compatibiliteit te kunnen bieden tussen browsers en videoformaten en het insluiten van video's op webpagina's eenvoudiger te maken is er een nieuwe ontwikkeling gaande binnen de opkomst van HTML5. HTML5 biedt namelijk de mogelijkheid om video's direct af te spelen in een webbrowser zonder bijkomstige plugins, zoals met de Flash Players wel het geval is. HTML5 maakt gebruik van een `<video>` tag, wat een insluiten eenvoudig mogelijk maakt zoals bij een figuur kan door middel van de `` tag.¹⁰ In oudere versies van HTML konden video's enkel worden ingesloten met behulp van de `<object>` en de `<embed>` tags wat veel code met parameters oplevert.

¹⁰ Toelichting Video in HTML5: <http://www.html5rocks.com/en/tutorials/video/basics/>

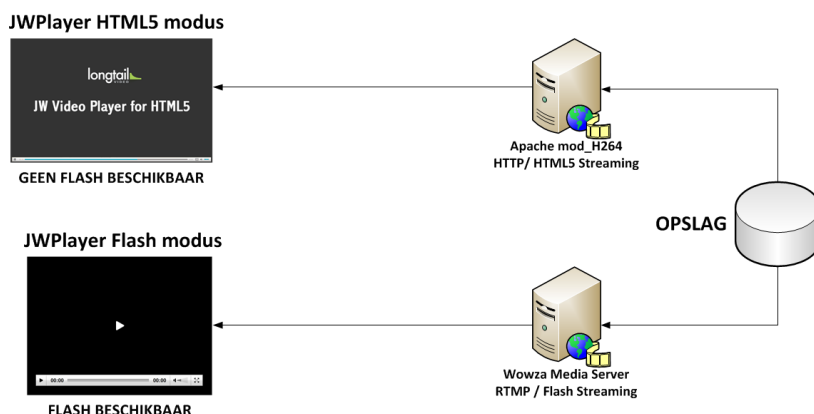
8.2 Compatibiliteit door Fallback

De browsers die HTML5 ondersteunen zijn continu aan verandering onderhevig. Met name gebruikers met verouderde browsers (zoals IE6/IE7/IE8) hebben niet de mogelijkheid om HTML5 materiaal weer te geven, zoals een HTML5 player die *embed* is in een webpagina en gebruikt maakt van de *<video>* tag. Om een scenario te realiseren dat ten alle tijden alle browsers ondersteund worden kan men gebruik maken van een *fallback* scenario. Dit houdt in dat, wanneer een browser Flash ondersteunt de Flash player wordt gebruikt. Wanneer er een browser niet in staat is Flash af te spelen wordt er overgeschakeld naar de HTML5 modus van JWplayer. Na veel tijd te hebben gestoken naar onderzoek om een dergelijke oplossing te realiseren bleek dat JWplayer goede mogelijkheden biedt voor HTML5, maar niet in combinatie met Wowza.

8.3 Apache H264 Streaming

Uit het onderzoek is gebleken dat op dit moment streaming naar HTML5 players niet mogelijk is op basis van de eerder genoemde protocollen met Wowza.¹¹ Dat zou concreet inhouden dat het hierboven beschreven Fallback scenario niet zou gaan werken. Resultaat zou zijn dat er enkel afgespeeld zou kunnen worden met de Flash player. Maar er is een andere optie gebleken, de enige mogelijkheid om dit op dit moment te doen is op basis van HTTP Pseudo Streaming. Hier kan de Apache Webserver voor gebruikt worden met aanvullende modules.¹² Pseudo Streaming houdt in dat het bestand gedownload wordt maar al wel direct afspeelbaar is. De rest van de video wordt gedownload terwijl de video afspelt.

Wanneer de Apache Extension Tool (apxs2) wordt gebruikt en de H264 Streaming Module(mod_h264) voor Apache wordt geïnstalleerd op een webserver blijkt dit een werkende oplossing op te leveren. Tijdens de opdracht is deze geïnstalleerd op de applicatieserver omdat hier Apache al als webserver draaide. De objectcode voor JWplayer die gebruikt is om dit scenario werkend te krijgen is te vinden in de technische bijlage (Bijlage II, paragraaf 4). In de figuur hieronder is het scenario met de twee streaming varianten gevisualiseerd.



Figuur 12 Detail streamingvarianten naar JWplayer.

¹¹ Informatie over de manieren van Streamen: / JWplayer:
<http://www.longtailvideo.com/blog/19578/what-is-video-streaming>

¹² Informatie over mod_H264: voor Apache:
<http://h264.code-shop.com/trac/wiki/Mod-H264-Streaming-Apache-Version2>

8.4 De keuze van de codec

Omdat HTML5 belangrijk zal zijn voor de toekomst van online video dient er een codec gekozen te worden die wordt ondersteund door HTML5. Momenteel worden er drie formaten ondersteund in HTML5. Deze formaten zijn eerder toegelicht in paragraaf 7.1 en 7.2

Omdat Wowza volledig gebruik maakt van bestanden die een MP4-container formaat hebben is dit de meest voor de hand liggende keuze, voor dit moment. Wowza biedt namelijk geen enkele voorziening om bestanden met een Ogg of WebM container te streamen. FFmpeg biedt goede transcodeerfaciliteiten voor de H.264 codec en maakt hiervoor gebruik van de open source x264 encoder. Voor enkele browsers zijn er *plugins* nodig om H.264 bestanden af te spelen, in onderstaande tabel is een overzicht te zien.

Codecs vs. Browser & Platform	H.264	WebM	Ogg/Theora
Mobile: iOS	✓	✗	✗
Mobile: Android	✓	(no HW)*	✗
Mobile: Other	✓	✗	✗
Desktop: Safari	✓	(future?)	✗
Desktop: Chrome	(via Flash)	✓	✓
Desktop: Opera	(via Flash)	✓	✓
Desktop: Firefox	(via Flash)	(future?)	✓
Desktop: IE9	✓	✓**	✗
Desktop: IE8 and below	(via Flash)	(future?)	✗

* Android supports WebM as of 2.3, but only via software rendering.

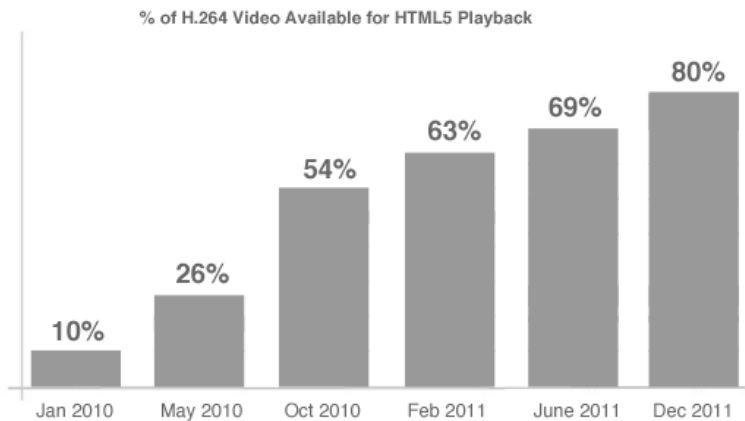
** IE9 supports WebM if installed separately.

Figuur 13 Ondersteuning van (HTML5) codecs in webbrowsers¹³

Een duidelijke reden om dus te kiezen voor enkel de H.264 codec voor de toekomst. Maar dat is helaas te kort te bocht, de 'codec-war' is in volle gang en sterk aan verandering onderhevig. De richting van deze strijd wordt bepaald door grote bedrijven als Google en Apple maar ook door populaire webbrowsers als Mozilla Firefox. Google heeft in 2011 aangekondigd in Chrome de ondersteuning voor H.264 te laten vallen om haar eigen WebM container en de Ogg container sterker in de markt te zetten. Het argument van Google was vooral dat zij vinden dat het internet niet afhankelijk mag zijn van een codec op licentiebasis, want dat is de H.264 codec. De andere codec, de VP8 codec (WebM) en Theora codec (Ogg) zijn dat niet, deze zijn open source.

¹³ Bron: <http://farukat.es/journal/2011/01/488-google-h264-and-video-web>

Echter tot op heden ondersteunt Google's Chrome nog steeds H.264 videomateriaal. Daarbij komt nog dat in 2010 de stap is gezet om H.264 door de sterke opkomst voor de komende vijf jaar (dus tot 2015) *royalty-free* aan te bieden. Deze keuze lijkt doorslaggevend geweest voor de invoer van H.264 materiaal op het internet, onderstaande figuur laat de trend zien die het laatst gemeten is in december 2011.



Figuur 14 Opkomende trend H.264 gebruik op het web.

De populaire webbrowser Mozilla Firefox heeft naar aanleiding van dit succes besloten om de ondersteuning weer terug te bieden, na deze eerder in 2010 te hebben laten vallen.¹⁴ Apple blijft met alle iOS apparaten enkel H.264 ondersteunen en Microsoft blijft met Internet Explorer ook standaard ondersteuning aanbieden. Wat er na 2015 gaat gebeuren is nog onduidelijk, het lijkt erop dat de trend van het gebruik van H.264 nog door zal zetten. De kans bestaat wel dat wanneer er licentiekosten worden geïntroduceerd er na die tijd er een omslag plaatsvindt naar een andere codec. En niet te vergeten Google is eigenaar van het grootste videoplatform YouTube en zal hier in de toekomst overgaan naar HTML5 players en daarbij gebruiken van, de WebM container met bijbehorende VP8 codec.

Het advies voor de codec/container en bijbehorende instellingen zijn:

- H.264/AVC videocodec;
- AAC audiocodec;
- MP4-container;
- Baseline profile (ivm iOS apparaten¹⁵);
- Resolutie van 640 x 360 voor standaard kwaliteit (SD);
- Resolutie van 1280 x 720 voor hoge kwaliteit (HD).

¹⁴ Nieuwsbericht code war, cnet.com:

http://news.cnet.com/8301-30685_3-57397031-264/mozilla-execs-capitulate-in-h.264-web-video-war/

¹⁵ Toelichting op gebruik van Baseline profile ivm iOS apparaten:

<http://www.onlinevideo.net/2012/03/streaming-to-apple-devices-understanding-profiles/>

8.5 Embedding

Het embedden of wel het insluiten van mediabestanden in bijvoorbeeld webpagina's is mogelijk, maar dan dient wel de player ook ingesloten te worden. Dit komt omdat MediaMosa niet direct afspeel URL's genereert wanneer er gebruik gemaakt wordt van externe streaming servers. De mogelijkheid die ook de player mee insluit werkt op basis van HTML object code.

MediaMosa kan deze HTML objectcode teruggeven na een aanroep met het responsetype object, een voorbeeld hiervan is:

```
https://wle.mediamosa.uvt.nl/play/pETPfwNTMRIQUc7rLRQvNgEq?response=object&filename=00030.mp4
```

Wanneer deze URL wordt uitgevoerd in een webbrowser wordt direct de player met de bijbehorende video getoond. Een interessante bijkomstigheid is dat het *play proxy* component hier dienst doet met behulp van *ticketing*, dit component is beschreven in de technische bijlage (Bijlage II, paragraaf 1.4) samen met een voorbeeld van de object code.

8.6 Resultaat

De wereld van online video is alles behalve helder en eenvoudig. Het noodzakelijke gebruik van de Apache mod_H264 module is misschien niet wenselijk maar op dit moment niet te vermijden. De oplossing die nu gerealiseerd is op basis van fallback is gekozen omdat Wowza geschikt is om grote hoeveelheden eindgebruikers te bedienen.

Apache en Wowza maken voor het streamen gebruik van één en hetzelfde MP4 videobestand, Wowza doet dit op basis van het RMTP protocol via de Flash modus van de JWplayer. Apache doet dit met de mod_H264 module via HTTP (poort 80) en wordt gebruikt wanneer de HTML5 player wordt aangesproken. Met de optie van fallback wordt er dus enkel voor de gebruikers die niet met Flash om kunnen gaan de HTML5 player gebruikt.

JWplayer is een player met tal van mogelijkheden en er is zeer goede documentatie beschikbaar op de website.¹⁶ De gekozen H.264 codec is uitermate geschikt voor Wowza en Apache mod_H264 in combinatie met JWplayer. De codec heeft een zeer goede compressieverhouding. Dit houdt in dat er een goede kwaliteit behaald kan worden (hoge bitrate) wat resulteert in een relatief klein MP4 bestand.

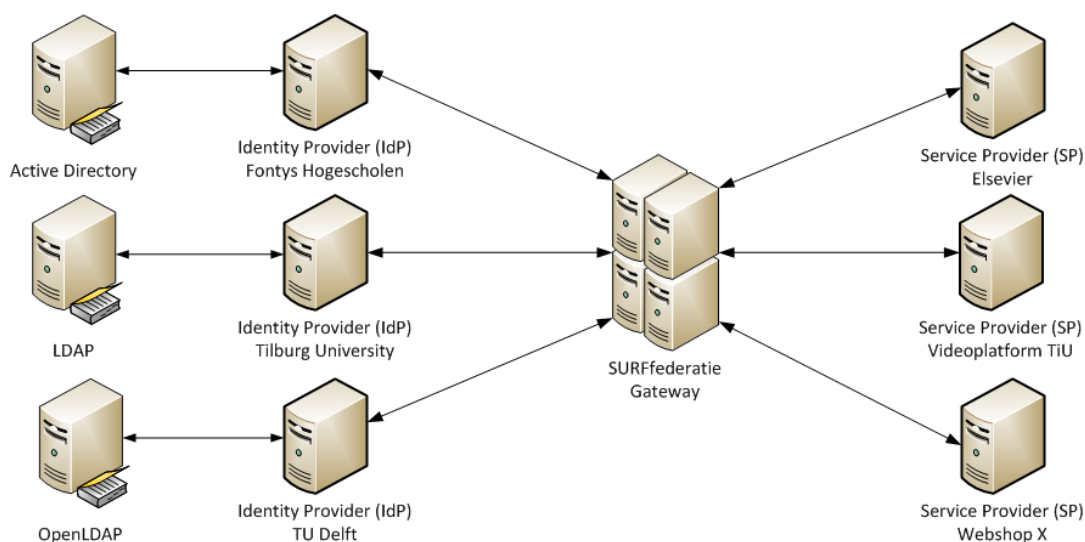
¹⁶ Website JWplayer: <http://www.longtailvideo.com/players/>

9 Authenticatie en autorisatie

Om toegang te verlenen aan medewerkers en studenten binnen TiU is het noodzakelijk om een koppeling te leggen met een bestaande *directory* waar de gebruikers met bijbehorende attributen in bekend zijn. Gebruikersbeheer binnen een frontend applicatie is niet wenselijk, het wordt al snel onbeheersbaar omdat er dubbele gegevens ontstaan in verschillende systemen. De gewenste situatie is dat er binnen het platform gebruik kan worden gemaakt van gebruikers die aanwezig zijn in een centrale *directory* en deze toegang te kunnen verlenen aan het platform. Als deze functionaliteit niet gerealiseerd kan worden moeten alle gebruikers van het platform handmatig worden aangemaakt, dit brengt ontzettend veel handelingen en administratie met zich mee. De wens van de opdrachtgever is om te onderzoeken of een koppeling met de SURFfederatie mogelijk is. Om dit te kunnen realiseren is er onderzoek gedaan naar de mogelijkheden en zijn er eerst testopstellingen gemaakt en getest voor er een daadwerkelijke koppeling met de SURFfederatie kon worden gelegd. Toelichting op deze werkwijze is beschreven in paragraaf 9.2.

9.1 Toelichting SURFfederatie

SURFnet heeft de SURFfederatie in het leven geroepen om samenwerking tussen organisaties en dienstenleveranciers eenvoudiger te maken.¹⁷ Het doel is dat studenten en medewerkers met hun eigen instellingsaccount toegang krijgen tot informatie en diensten van andere aanbieders. Elke instelling heeft zijn eigen *directory* met gebruikers en kan deze koppelen met de SURFfederatie. TiU is met haar *directory* al gekoppeld met de federatie. SURFfederatie zorgt ervoor dat de gebruikers hun identiteit kunnen bewijzen door gebruik te maken van gegevens die de instelling in dit geval TiU daartoe uitgeeft en beheert. TiU is voor de SURFfederatie een *Identity Provider* (IdP). De Identity Provider heeft als taak de identiteit van een gebruiker vast te stellen en de vooraf opgegeven gebruikerskenmerken vrij te geven aan de federatie. Het videoplatform wil in deze opzet niets meer dan gebruik maken van gegevens die zijn opgeslagen bij de verschillende Identity Providers die zijn aangesloten bij de SURFfederatie. In eerste instantie is dit alleen de Identity Provider van TiU.



Figuur 15 Architectuur voorbeeldopstelling met SURFfederatie.

¹⁷ Website SURFfederatie: <http://www.surfnet.nl/nl/Thema/SURFfederatie/over/Pages/Default.aspx>

Voor de toekomst kan er eenvoudig toegang verleend worden aan gebruikers van andere instellingen althans voor diegene die met Identity Providers zijn aangesloten bij de SURFfederatie. Bijvoorbeeld in het geval van samenwerkingsverbanden zoals gedeelde cursussen tussen universiteiten. In de figuur op de vorige pagina is een voorbeeld te zien hoe Identity Providers van verschillende instellingen zijn gekoppeld aan de SURFfederatie, deze Identity Providers halen hun gegevens uit een *directory*. De *Service Providers* (SP) willen gebruik maken van deze gegevens en kunnen via de SURFfederatie gegevens opvragen bij Identity Providers waar zij toegang toe hebben.

9.2 Werkwijze

In dit onderdeel wordt beschreven wat de bevindingen zijn van de gevonden authenticatiemogelijkheden en op wat voor manier deze geïmplementeerd kunnen worden in een frontend applicatie van het videoplatform.

De volgende stappen zijn genomen om tot het resultaat te komen:

1. Onderzoek naar authenticatie binnen TiU omgeving en protocollen;
2. Onderzoek naar authenticatie bij het platform en de mogelijkheden;
3. Onderzoeken van simpleSAMLphp en bijbehorende Drupal module;
4. Onderzoek en koppelen met de testomgeving binnen TiU;
5. Onderzoek en koppelen met de productieomgeving binnen TiU;
6. Onderzoek en koppelen met de SURFfederatie.

9.3 Manier van authenticeren

Binnen TiU is er een SSO omgeving actief, deze omgeving zorgt ervoor dat gebruikers zich slechts eenmaal hoeven aan te melden bij een applicatie, dienst of omgeving. Wanneer men tijdens dezelfde browsersessie naar een andere applicatie, dienst of omgeving gaat die is aangesloten binnen de SSO omgeving dan hoeft men zich niet nogmaals aan te melden. De gebruikersgegevens zijn dan al een keer geauthenticeerd en er is op dat moment een geldige SSO sessie actief. De SSO omgeving maakt gebruik van het *Security Assertion Markup Language 2.0* protocol (SAML 2.0). SAML is een standaard die is ontwikkeld door het *Security Services Technical Committee of the Organization for the Advancement of Structured Information Standards* (SSTC / OASIS).¹⁸ SAML maakt gebruik van een op XML gebaseerd *framework* wat de mogelijkheid biedt om gebruikersauthenticatie, rechteninformatie en attribuutinformatie te communiceren. SAML zorgt ervoor dat SSO mogelijk is door gebruikers te verifiëren bij een Identity Provider en zo toegang te kunnen verlenen aan Service Providers zonder extra authenticatie handeling door de gebruiker. Dit zorgt voor meer gebruikersgemak omdat men zich niet telkens opnieuw hoeft aan te melden bij andere Service Providers die zijn aangesloten.

¹⁸ Figuur en Informatie deels afkomstig van: SAML V2.0 Executive Overview (12 april, 2005)
<http://www.oasis-open.org/committees/download.php/13525/sstc-saml-exec-overview-2.0-cd-01-2col.pdf>

De Identity Provider heeft toegang tot een bestaande *directory* waar gebruikersinformatie is opgeslagen zoals gebruikersnaam, wachtwoord, groep, emailadres en tal van andere attributen. Binnen TiU is er zowel een *Active Directory* (AD) als een *Lightweight Directory Access Protocol* (LDAP) aanwezig, voor de toepassing van SSO wordt er gebruik gemaakt van de LDAP server. De Service Provider wil een applicatie dienst of omgeving beschikbaar stellen aan gebruikers. Een Service Provider heeft geen directe weet van de gebruikers en gebruikt de Identity Provider hiervoor, de Identity Provider geeft een login-pagina weer voor de gebruiker. Wanneer een gebruiker zich succesvol aanmeldt bij de Identity Provider, stuurt de Identity Provider een bericht naar de Service Provider waarin deze meldt dat de betreffende gebruiker een geautoriseerde gebruiker is.

9.4 Authenticeren bij het platform

Eerder in dit document is al benadrukt dat het videoplatform, wat voornamelijk bestaat uit MediaMosa, is gescheiden tussen een frontend en een backend. De authenticatie van gebruikers vindt enkel plaats bij een frontend applicatie. Bij authenticatie wordt er gekeken of een gebruiker of applicatie daadwerkelijk is wie hij beweerd te zijn. Authenticatie tussen frontend en backend vindt plaats met een gedeelde sleutel tussen deze twee, dit deel wordt dus afgedekt binnen het platform (zie Frontend Applicaties). Of een gebruiker daadwerkelijk toegang verleend wordt, wordt dus bepaald in de frontend van het platform. De frontends die zijn gebruikt tijdens het project zijn allen Drupal installaties. Het onderzoek geeft antwoord op de vraag welke mogelijkheden er zijn om binnen een Drupal installatie de gewenste manier van authenticatie mogelijk te maken. Drupal is een populair CMS en daardoor zijn er een groot aantal modules beschikbaar die gebruikt kunnen worden in Drupal CMS omgevingen. Een module die na onderzoek gevonden is, is de simpleSAMLphp module voor Drupal. De module maakt gebruik van simpleSAMLphp software die op de server aanwezig moet zijn.

Om het proces te verduidelijken zijn hieronder de acties beschreven die plaatsvinden in het uiteindelijk laatste scenario, de koppeling met de (test) SURFfederatie. In de technische bijlage (Bijlage II, paragraaf 3.3 en 3.4) zijn de eerste twee scenario's beschreven. In de figuur is duidelijk de architectuur gevisualiseerd om zo de communicatie tussen de verschillende entiteiten duidelijk te maken. De beschreven condities geven aan wat de begin en eindsituatie zijn.¹⁹

In de bijlage (Bijlage II, paragraaf 3.5) zijn de bijbehorende screenshots van de loginprocedure weergegeven. Deze verduidelijken het beeld bij de acties die in het scenario plaatsvinden.

¹⁹ Voorbeelden/toelichtingen op proces en acties:

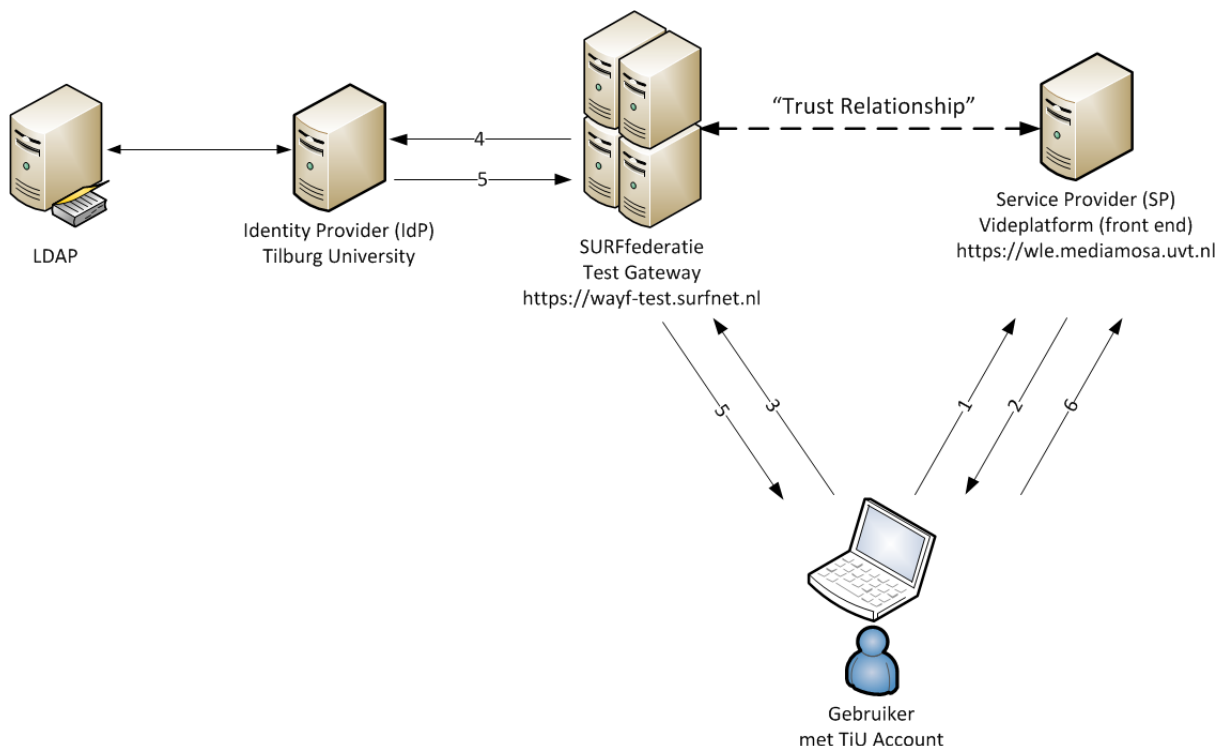
http://wiki.developerforce.com/page/Single_Sign-On_for_Desktop_and_Mobile_Applications_using_SAML_and_OAuth

9.4.1 Scenario: Gebruiker meldt zich aan bij de SURFfederatie

Condities

Pre: Gebruiker heeft een TiU account en is zo bekend in de TiU LDAP. Zowel de Identity Provider van TiU als de Service Provider (Videoplatform Frontend) zijn bekend bij de SURFfederatie en hebben weet van elkaars metadata en hebben zo een *Trust Relationship*.

Post: Gebruiker is succesvol aangemeld bij het videoplatform en beschikt over een SSO sessie die actief is via de SURFfederatie, dit zorgt er nu voor dat men zich bij zowel interne applicaties als bij andere Service Providers aangesloten bij de SURFfederatie niet nogmaals hoeft aan te melden.



Acties

1. Gebruiker wil zich aanmelden bij het videoplatform via webbrowser.
2. *Redirect* vindt plaats naar simpleSAMLphp pagina in frontend, Identity Provider (SURFfederatie) wordt gekozen of is onthouden van eerder verzoek.
3. Gebruiker komt terecht op de WAYF-pagina waar alle beschikbare Identity Providers weergegeven zijn en kiest "Universiteit van Tilburg".
4. Gebruiker voert gebruikersnaam en wachtwoord in op pagina van de gekozen Identity Provider "Universiteit van Tilburg".
5. Authenticatie is voltooid door Identity Provider, Identity Provider zendt antwoord terug in de vorm van een *SAML response* via de SURFfederatie.
6. Response wordt doorgegeven aan Service Provider, sessie wordt toegewezen en er vindt een redirect plaats naar het platform en de gebruiker is succesvol ingelogd.

9.5 Bepalen van de attributen

Een Identity Provider levert een aantal attributen aan een Service Provider, welke attributen er worden overdragen wordt vastgesteld in instellingen op de Identity Provider. Via de simpleSAMLphp Drupal module is er in de *web interface* een testfunctionaliteit aangebracht waarmee de aangeleverde attributen zichtbaar kunnen worden gemaakt. Na een *login* bij de SURFfederatie worden de volgende beschikbare attributen weergegeven.

Uw attributen

urn:mace:terena.org:attribute-def:schacHomeOrganization	uvt.nl
urn:mace:dir:attribute-def:eduPersonEntitlement	<ul style="list-style-type: none">urn:mace:dir:entitlement:common-lib-termsurn:mace:terena.org:tcs:escience-userurn:mace:terena.org:tcs:personal-user
urn:mace:dir:attribute-def:anr	376055
urn:mace:dir:attribute-def:givenName	Joost
urn:mace:dir:attribute-def:uid	jdamen
urn:mace:dir:attribute-def:eduPersonPrincipalName	376055@uvt.nl
urn:mace:dir:attribute-def:cn	J.M.J. Damen
urn:mace:dir:attribute-def:sn	Damen
urn:mace:dir:attribute-def:mail	J.M.J.Damen@uvt.nl
WindowsLiveID	jdamen@tst-cloud.uvt.nl
groups	<ul style="list-style-type: none">usersmembers

Figuur 16 Overzicht van de attributen die worden doorgegeven via SURFfederatie

Uit de beschikbare attributen moeten de bruikbare attributen worden gehaald die uiteindelijk bepalend zijn voor de gebruikers die zich aanmelden bij het platform. De manier waarop gebruikers geïdentificeerd worden moet uniek zijn. De optie om hiervoor het attribuut 'uid' te kiezen is niet in alle gevallen geschikt. Bij koppeling met de SURFfederatie bestaat de kans dat dit attribuut niet uniek is. Hierdoor zou binnen het platform data corrupt kunnen raken en rechten zouden aan de verkeerde gebruikers toegekend kunnen worden. Er bestaat namelijk een kans dat iemand van een andere instelling die gekoppeld is met de SURFfederatie ook *uid*: jdamen heeft.

In de simpleSAMLphp Drupal module is er ruimte voor drie attributen:

- Attribuut wat gebruikt gaat worden als gebruikersnaam;
- Attribuut wat gebruikt gaat worden als unieke identificatie;
- Attribuut wat gebruikt gaat worden als het emailadres van de gebruiker.

Als attribuut voor de te gebruiken gebruikersnaam is gekozen voor het 'mail' attribuut. Dit is een goede representatie van de gebruiker en het heeft als voordeel dat dit direct een contactgegeven is van de gebruiker. Tevens is het zeker dat dit attribuut uniek is omdat het ook het mailadres van de gebruiker is welke toch al uniek is.

Het attribuut wat als unieke identificatie wordt gebruikt is het 'eduPersonPrincipalName'. In het geval van TiU is dit het unieke nummer wat iedere student en medewerker heeft verkregen bij aanstelling, gevolgd door het domein. In mijn geval is dat: 376055@uvt.nl, in veel gevallen wordt een dergelijk attribuut gebruikt voor de weergave van de gebruikersnaam. Maar in dit geval zegt dit attribuut niet veel als gebruikersnaam, vandaar de keus om dit als unieke *identifier* te houden en het mailadres te gebruiken als zichtbare gebruikersnaam.

Voor het mailadres wordt hetzelfde attribuut als gebruikt voor gebruikersnaam, namelijk het 'mail' attribuut.

9.6 Autorisatie bij het platform

Wanneer een gebruiker zich voor de eerste keer aanmeldt bij het platform wordt voor deze gebruiker automatisch een account aangemaakt binnen de Drupal frontend. Vanaf dat moment kan een gebruiker bestanden uploaden en zo ook eigenaar worden van bestanden. Een gebruiker is dan dus bekend binnen de frontend en is lokaal opgeslagen in de bijbehorende Drupal database. Echter enkel de gebruikersnaam is opgeslagen, de Drupal frontend heeft geen weet van wachtwoorden maar luistert enkel naar de berichten van de Identity Provider.

In de configuratie van de Drupal frontend hebben gebruikers een bepaalde rol. Aan een rol hangen de rechten die een gebruiker toegekend krijgt wanneer hij een rol krijgt toebedeeld. De simpleSAMLphp module heeft de mogelijkheid om een rol automatisch toe te kennen op basis van een opgegeven attribuut. Het meest voorkomend onderscheid dat gewenst is, is het onderscheid tussen studenten en medewerkers. Na overleg met collega's die betrokken zijn bij het bepalen van de attributen binnen TiU is gebleken dat dit attribuut op dit moment nog niet gebruikt kan worden. Het gaat om het *eduPersonAffiliation* attribuut waarin straks bijgehouden gaat worden of iemand *student* of *employee* is. Het attribuut zal *multivalued* worden, in het geval van studenten die ook werkzaam zijn bij TiU bijvoorbeeld, deze personen hebben dan beide waardes bij dat attribuut.

Aan de hand van dat attribuut kan via de simpleSAMLphp module automatisch een rol worden toegekend. Dit gaat op basis van een regel, bijvoorbeeld wanneer *mail* attribuut achter het @ teken *uvt.nl* bevat dan *role number 2* toekennen. Na uitvoerig te hebben getest met verschillende attributen blijkt dit in de huidige testopstelling niet te werken. Er wordt geen andere rol toegekend wanneer er zo'n dergelijke regel actief is. Tijdens de opdracht is geen contact verkregen met de makers van de module om hier informatie over in te winnen. Het is wel noodzaak dat bij het gebruik van deze module dit verder word onderzocht en er desnoods via SURFnet contact wordt gelegd met de ontwikkelaars. De module is ontwikkeld door Uninett²⁰. Dit is een soortgelijke organisatie als SURFnet maar dan voor Noorse instellingen.

²⁰ Website Uninett: <https://www.uninett.no/>

10 Mogelijkheden voor opschaling

De opbouw van MediaMosa bevat de volgende onderdelen die door de schaalbare opbouw van MediaMosa toebedeeld kunnen worden aan separate servers:

Job, Download, Upload, Still, Streaming

Op de volgende pagina staat een mogelijke architectuur ten behoeve van een uitrol van het systeem weergegeven. Na onderzoek van de verschillende componenten van MediaMosa en navraag bij andere instellingen is dit een reële mogelijkheid voor een toekomst vaste en schaalbare oplossing.

10.1 Opschalingsscenario

In dit scenario zijn er twee *jobserver*s weergegeven, deze voeren in dit geval de analyse, transcodeer en still generatie taken uit. De verdeling over deze *jobserver*s gebeurt door de *Job Scheduler* die in MediaMosa is ingebouwd en actief is op de applicatieserver.

De applicatieserver (MediaMosa backend) en de frontend applicatie(s) zijn beiden op een separate server geïnstalleerd.

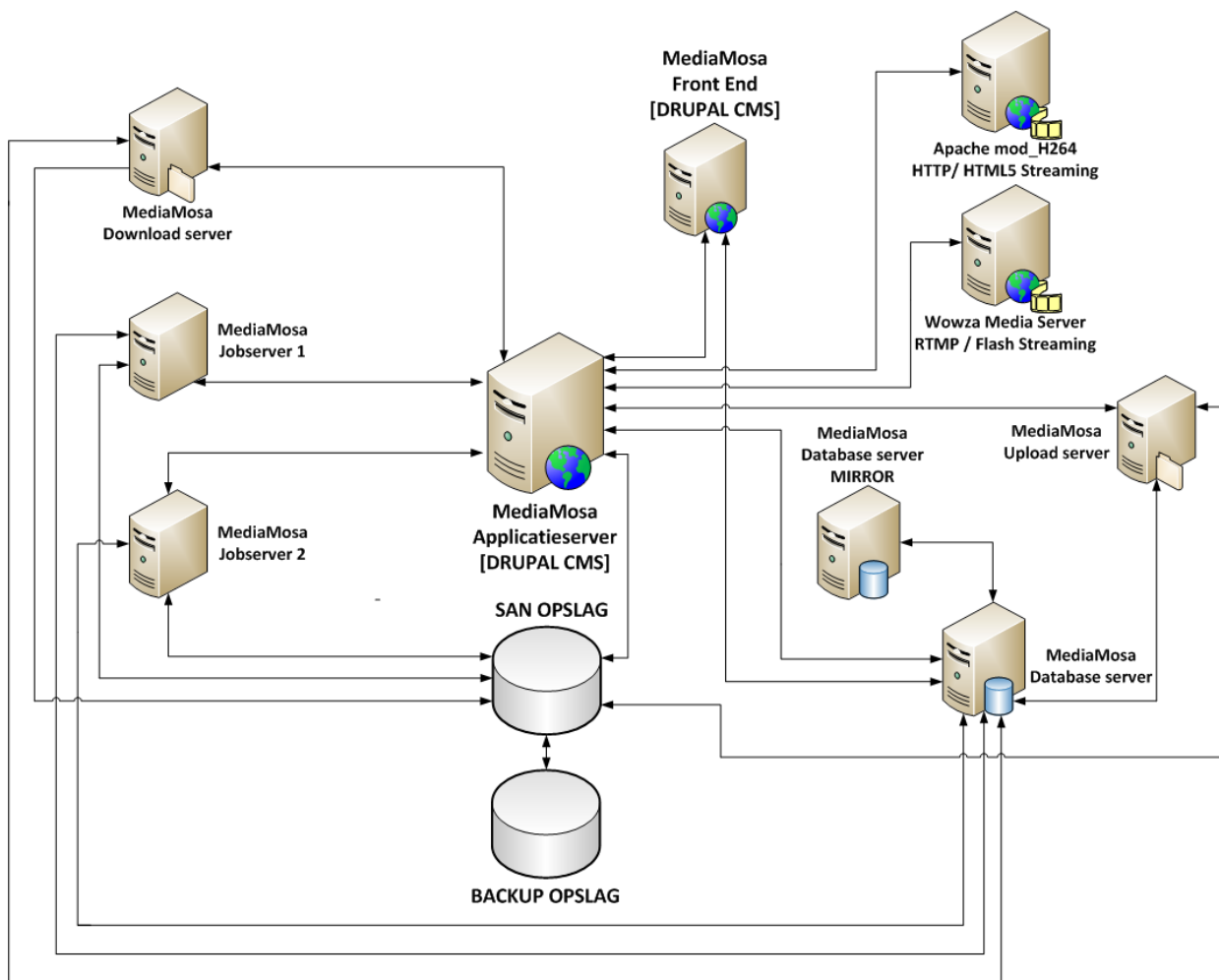
Voor de bestanden die beschikbaar zijn als *download* zal een aparte Download server deze verzoeken voor zijn rekening nemen.

Voor het verwerken van de uploads is een uploadserver voorzien, deze plaatst bestanden van eindgebruikers op het opslagmedium.

Het afhandelen van de streaming verzoeken gebeurt door twee afzonderlijke servers. Dit zijn de servers die nodig zijn voor de afhandeling van videostreams naar zowel een Flash player als een HTML5 player. Het streamen via beide servers gebeurt op basis van één MP4-bestand.

Zoals in de architectuur van de testomgeving ook te zien was hebben alle separate servers toegang nodig tot dezelfde opslag, hier staan de mediabestanden. Als opslagmedium is het Storage Area Network (SAN) van TiU weergegeven. Een backup opslag wordt binnen TiU vrijwel overal gebruikt dus is deze ook hier weergegeven.

De MediaMosa database en de database van de frontend applicatie zijn de belangrijkste onderdelen binnen de architectuur. Beide maken zij gebruik van een centrale databaseserver. Als een database verloren zou gaan door een crash zijn alle relaties en koppelingen tussen assets, bestanden en metadata verloren of beschadigd. MediaMosa zou dan onbruikbaar worden, ook al zijn de bestanden nog intact. De belangrijk aanvullende gegevens zouden verloren zijn, daarom is voor deze oplossing gekozen. Zowel de centrale MediaMosa database als de database van de frontend applicatie moet gedeeld worden met alle separate servers. Ze moeten allen toegang hebben en moeten kunnen lezen en schrijven. Een tweede database server is opgenomen als *mirror* van de centrale databaseserver, dit zorgt voor een *failover* optie. Op deze manier kan er snel overgeschakeld worden naar de tweede database server in geval van calamiteiten, continuïteit wordt zo gewaarborgd.



Figuur 17 Mogelijk scenario voor schaalbare uitrol TiU Videoplatform

11 Conclusies

Het doel van het onderzoek was het duidelijk krijgen of een dergelijke oplossing geschikt is voor gebruik binnen TiU. Daarnaast werden de volgende vragen beantwoord: Is de oplossing een duidelijke vooruitgang ten opzichte van de huidige situatie en zijn de bezwaren bij de beschreven aanpak opgelost?

Het antwoord op beide vragen is ja. Want het is gebleken dat de oplossing met als basis MediaMosa goede uitkomsten biedt voor een toekomst vaste oplossing. De omgeving is flexibel en schaalbaar en is goed te implementeren binnen de TiU omgeving. Het is duidelijk dat MediaMosa veel meer is dan een platform dat enkel videobestanden kan opslaan en afspelen.

De functionaliteiten van MediaMosa zijn zeer uitgebreid, maar uiteindelijk bepaalt de frontend applicatie de functionaliteiten die aan de eindgebruiker worden aangeboden. Het is een afweging om een beschikbare *template* te gebruiken of zelf een frontend applicatie te ontwikkelen. In eerste instantie zou het advies zijn, maak gebruik van een beschikbare template. Deze kan grotendeels naar wens kan worden aangepast aan de huisstijl van TiU. Wanneer er een MediaMosa omgeving actief is kan men, indien gewenst, met een eigen omgeving die ontwikkeld wordt testen zonder dat de productieomgeving hierdoor in gevaar komt.

De gewenste koppeling met de SURFfederatie is gerealiseerd en zo kan gebruik worden gemaakt van een bestaande directory met gebruikersgegevens. Ook het gebrek aan metadata is met het platform opgelost; het is bekend wie een bestand heeft geüpload en de overige metadata kan door de gebruiker zelf worden toegevoegd.

De afspeelmogelijkheden zijn gerealiseerd met het oog op compatibiliteit. De keuze van de H.264 codec op basis van MP4 bestanden in combinatie met Wowza en Apache mod_H264 is geschikt gebleken voor de toekomst. Het gebruik van JWplayer en de toegepaste fallback modus zorgt voor zowel een Flash player als een HTML5 player en daarmee voor compatibiliteit.

De mogelijkheden die gerealiseerd zijn met Wowza en Apache mod_H264 zijn werkend gebleken in de testomgeving, maar men weet niet hoe deze zich houdt bij opschaling en of Wowza dit in de toekomst wellicht af kan handelen.

Tijdens de uitvoering van de opdracht zijn de gevolgen van het stopzetten van de doorontwikkeling merkbaar. De doorontwikkeling en ondersteuning liggen nagenoeg stil en bij SURFnet is het merkbaar dat men MediaMosa grotendeels heeft losgelaten. Documentatie is er nauwelijks of het is gedateerd en de hulp van de MediaMosa Community is nodig bij enkele zaken die specifiek zijn voor het platform. Een aantal vragen over met name nieuwe functionaliteiten zijn onbeantwoord gebleven. Het succes voor de toekomst van MediaMosa is grotendeels afhankelijk van de doorontwikkeling en ondersteuning die wordt aangeboden. Als die achterwege blijft kan het ervoor zorgen dat niet alle functionaliteiten in de toekomst gebruikt kunnen worden zonder hulp van buitenaf.

Evaluatie

Het realiseren van een Proof of Concept en het onderzoeken van de verschillende onderdelen heeft veel resultaten opgeleverd. Nu bekend is hoe dat de oplossing met MediaMosa zich in de praktijk gedraagt binnen de TiU omgeving, kunnen er concrete stappen gezet worden om de huidige videosever te vervangen. De architectuur die als scenario voor opschaling is geschetst kan hier houvast voor bieden.

Het is achteraf een hele klus gebleken die zowel voor TiU als opdrachtgever als wel voor mij als student meer werk en complexiteit in zich had dan we bij aanvang dachten. MediaMosa is geen software die je download en met een goede gebruikershandleiding in productie brengt. Tijdens het project was er regelmatig contact nodig met ontwikkelaars en beheerders van andere instellingen die hun ervaringen, bevindingen en problemen met mij wilden delen. Door het gebrek aan documentatie is er in het begin wat tijd verloren gegaan om zaken zelf uit te zoeken. Het hele onderzoek is samen met de realisatie één groot leermoment geweest. Ik heb een brede kennis opgedaan op tal van onderdelen die in dit document beschreven zijn.

Ik heb me op veel gebieden moeten oriënteren om het project tot een succesvol einde te brengen en een werkende testomgeving op te kunnen leveren. Vooral het onderdeel streaming video wat betrekking had op het brengen van videobestand naar eindgebruiker. Het uitzoeken van bijbehorende codecs, containers en protocollen was een zoektocht op het internet. Deze wereld is constant aan verandering onderhevig, alle onderdelen die met elkaar samenwerken moeten voor elkaar toegepast kunnen worden.

Het koppelen met de SURFfederatie was voor mij compleet nieuw, het koppelen tussen Identity Providers en Service Providers zijn voor mij nu helder en duidelijk. Dit is kennis die ik in de toekomst goed kan overbrengen naar collega's van mijn eigen afdeling.

De doorontwikkeling van met name MediaMosa, Wowza en JWplayer zullen er hopelijk in de toekomst voor zorgen dat het streamen wat eenvoudiger en consistentere kan worden opgelost.

Ik heb veel kennis opgedaan over de werking en configuraties van MediaMosa binnen een UNIX omgeving. De kennis die ik heb opgedaan kan door middel van dit document en mijn ervaringen worden overgebracht naar collega's. De kennis en ervaring op het gebied van streaming video met hedendaagse oplossingen is vrijwel niet vertegenwoordigd binnen TiU.

Ik ben blij dat ik deze opdracht met succes heb uitgevoerd. Ik heb er vertrouwen in dat deze omgeving de studenten en docenten van TiU zal stimuleren om nog meer video in onderwijs en onderzoek te gaan gebruiken. En belangrijk voor het departement LIS, alles met het oog op een beheersbare omgeving: gemetadateerde bestanden, fijnmazige afscherming met behulp van autorisatie, authenticatie via de SURFfederatie en afspeelmogelijkheden op elk apparaat!

Literatuurlijst

Literatuur onderzocht voor orientatie en uitvoering, als genoemd in paragraaf 3.5.

UNIX (Debian), Apache, MySQL, PHP

Apache HTTP Server 2.0 Documentation: <http://httpd.apache.org/docs/2.0/>

Debian Wiki: <http://wiki.debian.org/>

MySQL 5.0 Reference Manual: <http://dev.mysql.com/doc/refman/5.0/en/index.html>

PHP Manual: <http://www.php.net/manual/en/index.php>

MediaMosa met bijbehorende componenten en functionaliteiten

MediaMosa documentatie: <http://mediamosa.org/node/71>

Authenticatie via SURFfederatie, SAML 2.0

simpleSAMLphp voor Identity en Service Providers, SURFfederatie (PDF):

http://www.surfn.nl/Documents/handleiding_201002_simpleSAMLphp.pdf

Streaming met Wowza, Adaptieve en RTMP streaming protocollen

Wowza Quick Start / Tutorials:

<http://www.wowza.com/forums/content.php?3-quick-start-guide>

Wowza Media Server 3, User Guide (PDF):

http://www.wowza.com/resources/WowzaMediaServer_UsersGuide.pdf

JWplayer mogelijkheden met Flash & HTML5

JWplayer versie 5, documentatie:

<http://www.longtailvideo.com/support/jw-player/jw-player-for-flash-v5>

Codecs en Containers

MOV, WMV, AVI, MPEG, MP3, OGG, WMA, ... Media Formats Explained, Dr.Lex:

<http://www.dr-lex.be/info-stuff/mediaformats.html>

Net Tuts, The state of video on the web:

<http://net.tutsplus.com/articles/the-state-of-video-on-the-web/>

De bronnen waar in het document naar wordt verwezen staan vermeld op de volgende pagina.

Bronnen vermeld in het document op volgorde van verschijnen

1. Website over VPN, TiU: <http://www.tilburguniversity.edu/nl/studenten/ict/vpn/>
2. Toelichting op Digital Asset Management:
<http://www.contentmanager.eu.com/dam.htm>
3. RESTful Web Services, The Basics, Website IBM:
<http://www.ibm.com/developerworks/webservices/library/ws-restful/>
4. Informatie over Secure Shell(SSH): http://en.wikipedia.org/wiki/Secure_Shell
5. FFmpeg documentatie: <http://ffmpeg.org/ffmpeg.html>
6. Toelichting Lua, Wikipedia:
[http://en.wikipedia.org/wiki/Lua_\(programming_language\)](http://en.wikipedia.org/wiki/Lua_(programming_language))
7. Toelichting LPeg library: <http://www.inf.puc-rio.br/~roberto/lpeg/>
8. Website Wowza: <http://www.wowza.com/>
9. Website JWplayer: <http://www.longtailvideo.com/players/>
10. Toelichting Video in HTML5: <http://www.html5rocks.com/en/tutorials/video/basics/>
11. Informatie over de manieren van Streamen: / JWplayer:
<http://www.longtailvideo.com/blog/19578/what-is-video-streaming>
12. Informatie over mod_H264: voor Apache:
<http://h264.code-shop.com/trac/wiki/Mod-H264-Streaming-Apache-Version2>
13. Blog Faruk Ateş : <http://farukat.es/journal/2011/01/488-google-h264-and-video-web>
14. Nieuwsbericht codec war, cnet.com:
http://news.cnet.com/8301-30685_3-57397031-264/mozilla-execs-capitulate-in-h.264-web-video-war/
15. Toelichting op gebruik van Baseline profile ivm iOS apparaten:
<http://www.onlinevideo.net/2012/03/streaming-to-apple-devices-understanding-profiles/>
16. Website JWplayer: <http://www.longtailvideo.com/players/>
17. Website SURFfederatie:
<http://www.surfnet.nl/nl/Thema/SURFfederatie/over/Pages/Default.aspx>
18. PDF Document SAML V2.0 Executive Overview:
<http://www.oasis-open.org/committees/download.php/13525/sstc-saml-exec-overview-2.0-cd-01-2col.pdf>

19. Voorbeelden/toelichtingen op SAML authenticatie proces en acties:
http://wiki.developerforce.com/page/Single_Sign-On_for_Desktop_and_Mobile_Applications_using_SAML_and_OAuth
20. Website Uninett: <https://www.uninett.no/>

BIJLAGE I

Plan van Aanpak

Realiseren van een multifunctioneel videoplatform

Binnen Tilburg University

Document: Plan van Aanpak
Auteur: Joost Damen
Datum: 20-01-2012
Plaats: Tilburg

Versie: 1.0

Opdrachtnemer: Joost Damen
Opdrachtgever: Tilburg University

Inhoudsopgave

Verklarende woordenlijst	3
Inleiding.....	4
1 Achtergrond	5
1.1. Organisatie	5
1.1.1 Library and IT Services	5
1.1.2 AV Support	5
2 Probleemstelling en Doelstelling.....	6
2.1 Probleemstelling	6
2.2 Doelstelling	6
2.2.1 Reden van uitvoering	6
2.2.2 Doelstellingen omschreven.....	7
3 De Projectopdracht	8
3.1 Opdrachtomschrijving.....	8
4 Projectactiviteiten	9
4.1 Initiatieffase: Plan van Aanpak.....	9
4.2 Initiële installaties	9
4.3 Installatie van de losse onderdelen	10
4.4 Het configureren van de onderdelen en componenten	10
4.5 Het ontwikkelen en koppelen van de EGA/GUI	10
4.6 Het plaatsen van materiaal voor testdoeleinden	10
4.7 Koppelen middels een Single Sign On oplossing.....	11
4.8 Platform beschikbaar maken voor een selecte groep	11
4.9 Oplevering en evaluatie van het project.....	11
5 Projectgrenzen / Randvoorwaarden	12
5.1 Projectgrenzen	12
5.1.1 Binnen de scope	12
5.1.2 Buiten de scope.....	12
5.1.3 Overige zaken.....	12
5.2 Welslagen van het project	13
5.3 Risico's.....	13

6	Producten.....	14
6.1	Documenten	14
6.2	Functionaliteiten	14
6.3	Eindproduct.....	14
7	Kwaliteit.....	15
7.1	Tools.....	15
7.2	Extern advies.....	15
8	De projectorganisatie	16
9	Planning.....	17
10	Kosten/baten-overzicht	19
	Bijlage I: Communicatieplan	20

Verklarende woordenlijst

AD	Active Directory, directoryservice implementatie van Microsoft die beheerders in staat stelt rechten en instellingen in een domein toe te passen op gebruikers en gebruikersgroepen.
AV	AV is een veelgebruikte afkorting voor Audio Visueel.
Codec	Codec is software die audio- en video-stromen kan coderen en decoderen.
CMS	Content Management Systeem, softwaretoepassing die meestal via het web toegankelijk is en waar gegevens eenvoudig en overzichtelijk in worden bijgehouden.
Drupal	Open Source CMS ontwikkelt in PHP, vormt de basis voor MediaMosa.
EGA	Eind Gebruikers Applicatie, 'front end' kant van MediaMosa.
Embedding	Het insluiten van objecten binnen webpagina's.
FFmpeg	Open Source Transcodeer-tool.
GUI	Graphical User Interface.
IP	Internet Protocol, laat computernetwerken met elkaar communiceren.
iTunesU	Service opgericht door Apple, binnen het programma iTunes is educatief materiaal te bekijken van hoger onderwijsinstellingen over de hele wereld.
LINUX	Familie van UNIX besturingssystemen.
LIS	Library and IT Services.
MediaMosa	Open Source Media Management Platform, 'back end' applicatie.
NAS	Network Attached Storage, harde schijf/opslag benaderbaar via het netwerk.
Open Source	Het vrij beschikbaar stellen van software voor ontwikkeling en herdistributie.
OS	Operating System.
Proof of Concept	Principe waarbij een dienst/software getest en uitvoerig onderzocht wordt om te zien of de oplossing geschikt is voor implementatie.
SAN	Storage Area Network, architectuur tussen koppeling van servers en opslag.
SSO	Single Sign On, manier om gebruikers slechts één maal in te laten loggen en vervolgens kan er automatisch toegang verschaft worden tot applicaties die ook SSO gebruiken.
SURF	Stichting SURF, organisatie van samenwerkende hoger onderwijsinstellingen op het gebied van ICT.
SURFnet	Dochterorganisatie van SURF. Zorgt voor internet connectiviteit tussen instellingen en voert innovatieve projecten uit. Founder van MediaMosa.
TiU	Tilburg University (voorheen Universiteit van Tilburg, UvT).
TiU account	Account (username/password) wat iedere student/docent/medewerker heeft.
Transcoderen	Het converteren van het ene mediaformaat naar het andere.
UNIX	Uniplexed Operating and Computing System (voorheen UNICS), familie van besturingssystemen.
URL	Uniform Resource Locator, internetadres.
Videoserver	Server die specifieke mediaservices draait en zo op afroep bepaalde mediabestanden kan afspelen naar gebruikers.
VLAN	Virtual LAN, virtuele scheiding van netwerken die zich fysiek in hetzelfde netwerk bevinden.
Wowza	Software voor Streaming Media Services.
WMV	Windows Media Video Formaat, videocompressie formaat ontwikkelt door Microsoft.

Inleiding

Aanleiding van dit document is het plan tot uitvoeren van het project: “Realiseren van een multifunctioneel videoplatform, binnen Tilburg University”. In dit Plan van Aanpak (PvA) wordt in detail beschreven wat de projectactiviteiten zullen zijn, wat het doel is van het project, wie er betrokken zijn en wat er wel en niet binnen de scope van het project valt. De opdracht vanuit de opdrachtgever staat beschreven in dit document. Wanneer opdrachtgever akkoord gaat met dit PvA kan er met het project gestart worden en zijn opdrachtnemer en opdrachtgever de uitvoering van het project overeengekomen.

Binnen Tilburg University (TiU) is er al geruime tijd het plan om een vervangende oplossing te zoeken voor de huidige manier waarin videobestanden opgeslagen en ontsloten worden. Dit gebeurt nu namelijk via een verouderde videosever die slechts één bestandsformaat kan afhandelen. Deze manier is verouderd en voldoet niet meer aan de eisen die de huidige student en docent verwacht. Omdat TiU nauw betrokken is bij SURF (en de producten en diensten die daar ontwikkeld worden) is de te onderzoeken oplossing al voor handen in de “markt”. Dit is namelijk de open source oplossing MediaMosa, ontwikkeld door SURFnet en uitermate geschikt om een videoplatform op maat te realiseren. Opdracht is dan ook om te realiseren en te onderzoeken of een videoplatform op basis van MediaMosa interessant is als toekomstige dienst voor TiU. Het platform wordt neergezet als een “proof of concept” en is bij oplevering van het project nog niet gereed voor uitrol. Technische en functionele specificaties worden in dit project belicht en zullen uiteindelijk gedetailleerd in een afstudeerverslag worden beschreven.

1 Achtergrond

1.1. Organisatie

De Universiteit van Tilburg, tegenwoordig Tilburg University (TiU) is een gerenommeerde universiteit die al meer dan 80 jaar onderwijs verzorgt en onderzoek doet naar complexe maatschappelijke vraagstukken zoals de economische crisis, klimaatbeheersing, de multiculturele samenleving en privacybescherming. Met toponderzoek en excellent onderwijs wil Tilburg University (TiU) een bijdrage leveren aan een betere samenleving. Nieuwe ideeën en inzichten krijgen we door te onderzoeken, te leren en te begrijpen. Iedere dag weer zoeken we naar meer begrip van het menselijk functioneren in deze wereld. TiU bestaat uit zes faculteiten met daaraan weer tal van gerelateerde onderzoeksgroepen, instellingen en partners. Met zo'n 2500 medewerkers en 12000 studenten dragen zij bij ter ondersteuning van de maatschappij met onderzoek en onderwijs.

1.1.1 Library and IT Services

Binnen TiU zijn er vier ondersteunende diensten die medewerker, student en docent ondersteunen. Het IT cluster als onderdeel van de dienst Library and IT Services (LIS) is daar een voorbeeld van. De dienst bestaat uit circa 160 medewerkers waarvan ongeveer de helft zich direct met techniek bezighoudt. Binnen het IT-cluster lopen de expertises van de verschillende afdelingen uiteen van Audio Visuele Support (AVS) tot UNIX en van Office Automation (OA) tot Netwerk/telefoniebeheer (NOC). Het doel van LIS is partner te zijn voor het primaire proces. Service en klantgerichtheid zijn belangrijke pijlers die mede het bestaansrecht vormen voor de dienst.

1.1.2 AV Support

Een afdeling die zijn oorsprong vond binnen het Facilitair bedrijf en sinds 2007 deel uitmaakt van LIS is Audio Visuele Support. Deze dienst zorgt voor ondersteuning op het gebied van multimedia, registraties, AV-infrastructuur en andere diensten. Onderwijs en onderzoek ondersteunen met beeld en geluid in brede zin is kortweg wat deze dienst doet.

2 Probleemstelling en Doelstelling

2.1 Probleemstelling

Video binnen het onderwijs is al jaren lang onmisbaar, mede omdat de huidige generatie die studeert ook wel de bijnaam “YouTube-generatie” heeft. Studenten zijn gewend om video’s en ander materiaal te bekijken wanneer ze willen en op welk apparaat en locatie ze dat willen. Binnen Tilburg University worden er op allerlei manieren video opgenomen en ontsloten, vaak met oude achterhaalde en slecht afgeschermdde technieken en middelen. Ook is er geen ondersteuning voor gangbare mobiele apparaten van deze tijd. Video’s kunnen enkel in Windows Media formaat (WMV) door middel van een Windows Media Streaming Server worden afgespeeld. Dit zorgt voor een groot compatibiliteitsprobleem voor o.a. mobiele apparaten. Video’s die aangeleverd worden in een ander formaat dan het WMV-formaat moeten handmatig getranscodeerd worden naar WMV en vervolgens handmatig gekopieerd worden naar een map op de videoserver.

De URL (Uniform Resource Locator) die vervolgens gecommuniceerd wordt naar de “klant” (student/docent/medewerker) wordt op basis van de bestandslocatie op de server gemaild. Heeft een klant dus meerdere video’s krijgt deze dus verschillende URL’s aangeboden, heeft de klant al eerdere video’s aangeboden dan zijn deze niet allemaal in één overzicht zichtbaar. Ook de afscherming van het materiaal is op dit moment niet goed geregeld, er bestaat slechts één manier om het materiaal enigszins af te schermen. Dit door het materiaal onder een hoofdmap te zetten die enkel bereikbaar is vanuit de IP-range binnen Tilburg University. De andere optie is het ‘publiek’ zetten van het materiaal en in beide opties geldt dat men moet de URL weten om videobestanden af te spelen. Dit fenomeen noemen we ook wel “Security through Obscurity” en garandeert in geen geval een correcte afscherming van het materiaal.

2.2 Doelstelling

2.2.1 Reden van uitvoering

De situatie die geschetst is in de eerder genoemde probleemstelling is voor de opdrachtgever voldoende reden om onderzoek te doen naar een nieuwe oplossing. Een mogelijke oplossing ligt relatief dichtbij, namelijk de open source oplossing MediaMosa die ontwikkeld is door SURFnet. Met deze oplossing als hart van het te realiseren platform en de uitbreidingen en implementatiemogelijkheden die mogelijk zijn, is er een principe keuze gemaakt dit videoplatform te gaan gebruiken. Het doel is om het eenvoudig mogelijk te maken om mediabestanden (zowel audio als video) te plaatsen, ontsluiten en af te spelen door middel van een oplossing op maat op basis van MediaMosa. Een proof-of-concept moet aantonen dat de bezwaren die in de huidige omgeving bestaan, met MediaMosa zijn opgelost.

2.2.2 Doelstellingen omschreven

Hoofddoelstelling:

Opleveren van een werkend videoplatform in de vorm van een proof-of-concept met bijbehorend afstudeerverslag.

Gespecificeerd houdt dit het volgende in;

- Opleveren van een toegankelijke en begrijpelijke Eind Gebruikers Applicatie (EGA);
- Toegang mogelijk maken voor studenten, docenten en medewerkers (op aanvraag);
- Het kunnen uploaden van (vrijwel) elk mediaformaat (audio & video);
- Het toevoegen van metadatering om zo bestanden te kunnen zoeken en vinden;
- De mogelijkheid bieden tot afscherming van materiaal voor het gehele domein, groepen gebruikers of individuele gebruikers.
- Het afspelen van materiaal op gangbare moderne apparaten;
- De mogelijkheid bieden om materiaal beschikbaar te stellen als “download”;
- Mogelijkheden om notificaties naar e-mailontvangers en social media te versturen;
- Onderzoek naar koppelingsmogelijkheden met andere platformen zoals Youtube en iTunesU.

3 De Projectopdracht

De opdrachtgever (Tilburg University / LIS) heeft aan de opdrachtnemende persoon de volgende opdracht gegeven:

3.1 Opdrachtomschrijving

Realiseer op basis van de open source software MediaMosa een 'Proof of Concept' om aan te tonen dat met Mediamosa de beschreven problemen zijn opgelost.

Hierbij moet expliciet aandacht gegeven worden aan:

- Inloggen middels Single Sign On(via de SURFfederatie);
- Eindgebruikers moeten mediabestanden zelfstandig kunnen plaatsen en hebben zelf de controle over de toegankelijkheid (afscherming of publiek toegankelijk);
- De eigenaar van het materiaal kan zelf aangeven of men het materiaal beschikbaar wil stellen voor lokale opslag (download);
- Er dienen verschillende gangbare formaten (codecs) van deze mediabestanden ondersteund te worden;
- Het materiaal dat geplaatst is moet in te sluiten zijn in webpagina's (embedding) en het moet mogelijk zijn materiaal te herdistribueren via Social Media of mensen te notificeren door middel van email. De gebruikersinterface (EGA/GUI) dient conform de huisstijl van TiU te worden opgemaakt en moet intuïtief zijn voor eindgebruikers;
- Voor de medewerkers van de dienst LIS-AV is het mogelijk het platform te gebruiken als hulpmiddel voor "transcodeer-batches", dit houdt in het omzetten van videomateriaal in grote hoeveelheden naar een ander videoformaat.

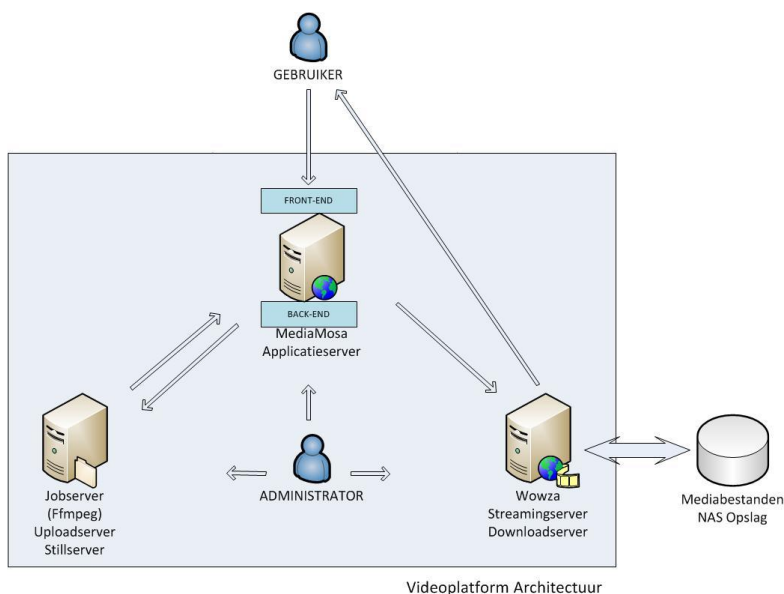
4 Projectactiviteiten

4.1 Initiatieffase: Plan van Aanpak

De opdrachtgever heeft expliciet aangegeven de opdracht uit te voeren met MediaMosa als basis. Alternatieven zullen niet worden behandeld in dit project, er zal geen vorm van pakketselectie plaats vinden. In dit document zijn de kaders afgebakend en is er zo goed mogelijk vastgelegd wat de uiteindelijk op te leveren producten zijn binnen het project. Vanuit SURFnet en andere universiteiten is er kennis en ervaring opgedaan met MediaMosa als oplossing. Met deze kennis en eerder opgedane ervaring is deze opdrachtomschrijving tot stand gekomen. Gevolgd door dit Plan van Aanpak waarin de doelstellingen, activiteiten, randvoorwaarden, producten en aanverwante informatie beschreven wordt.

4.2 Initiële installaties

De initiële installatie van de (virtuele) machines die als drietal gaan samenwerken binnen het project. Er wordt onderzoek gedaan naar de eisen van deze machines, welke besturingssystemen er vereist zijn of aanbevolen worden. Deze zullen binnen de netwerkomgeving gekoppeld worden. Het betreffende VLAN (Virtueel LAN) dient bepaald te worden evenals regels van firewalls, toegang op afstand, etc. MediaMosa is gebouwd om te kunnen werken met een uitgebreide architectuur. De architectuur die er nu voor ogen is staat hieronder weergegeven.



MediaMosa maakt onderscheid in verschillende servertypen, deze zijn nu ondergebracht en verdeeld bij de drie virtuele machines. Er kan gedurende het project besloten worden om functionaliteiten uit te splitsen en het aantal virtuele machines te vergroten.

Servertypen MediaMosa: download, upload, still, job, streaming, applicatie.

De 'front end' en 'back end' worden beide geïnstalleerd op de applicatieserver, er is de mogelijkheid om deze ook fysiek te scheiden omdat beide componenten afzonderlijk van elkaar werken. De virtuele machines zijn weergegeven binnen een lichtblauw kader, dit kader geeft weer dat de virtuele machines allen aanwezig zijn op één en dezelfde fysieke machine.

4.3 Installatie van de losse onderdelen

Verder op in dit document is aangegeven hoe de drie machines worden ingericht. De applicatieserver vormt hier het hart van het systeem met MediaMosa als kerncomponent. De Jobserver ontvangt en verwerkt taken die een bepaalde tijd in beslag nemen zoals transcodeertaken, uploadtaken, etc. De jobserver wordt aangestuurd door MediaMosa. De streamingserver is verantwoordelijk voor het afspelen van mediafiles en deze roept MediaMosa hiervoor aan. Alle installaties vinden plaats op virtuele machines, het besturingssysteem wat nu voor ogen is, is Linux: Debian 64-bit. Binnen TiU wordt er gebruikt gemaakt van Debian, voornamelijk qua compatibiliteit met modules en 'packages'.

4.4 Het configureren van de onderdelen en componenten

Als de installaties zijn voltooid dient er veel geconfigureerd te worden op alles zo optimaal mogelijk te laten verlopen. Binnen MediaMosa zijn er tal van opties die onderzocht en correct ingesteld dienen te worden. De installatie is gebaseerd op Drupal en de uitbreidingen van MediaMosa met de modules en 'building blocks' zijn volop beschikbaar. De transcodeerserver die als software de open source tool FFmpeg gebruikt dient op maat geconfigureerd te worden. FFmpeg wordt aangeroepen op basis van 'command-line' commando's en parameters, er is geen GUI. De streamingserver zal Wowza Media Server (3) gaan draaien, Wowza is een zeer geavanceerde toepassing die de mediabestanden op een juiste, efficiënte en snelle manier moet afspelen. Het onderzoeken en uitvoeren van deze configuraties valt binnen deze activiteit.

4.5 Het ontwikkelen en koppelen van de EGA/GUI

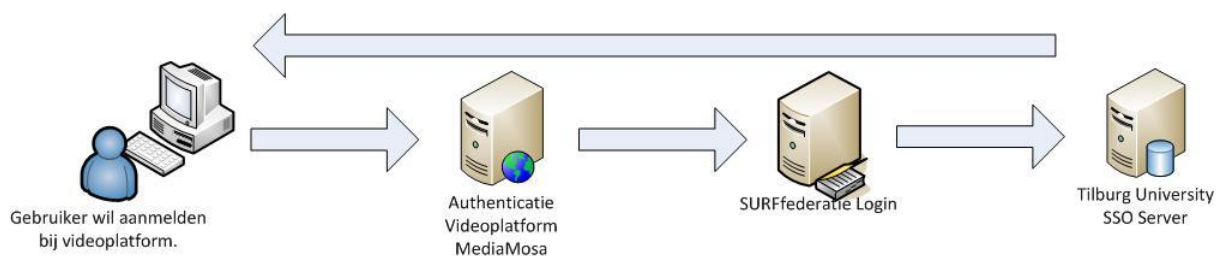
MediaMosa zorgt enkel voor de 'back end' op de applicatieserver, de 'front end' ook wel Eind Gebruikers Applicatie (EGA) genoemd in MediaMosa begrippen) dient aangepast te worden op basis van een template die door SURFnet beschikbaar wordt gesteld. De template die al enige tijd geleden is uitgegeven is de 'WLE template'. Deze template biedt de basis voor de 'front end' maar biedt niet de mogelijkheid om een aantal recente functionaliteiten van de 'back end' te benaderen. Het is bekend dat er tijdens het schrijven van dit Plan van Aanpak een template in ontwikkeling is waarvan de opleverdatum nog niet bekend is. Deze template is een nieuwe 'Construction Kit' en is geënt op Drupal 7, de CMS die wordt gebruikt bij de huidige MediaMosa releases (v.3.x). De 'Construction Kit' is al wel beschikbaar als ontwikkelaars-release maar er is nog twijfel aan de stabiliteit. Er zal onderzoek gedaan worden naar de beschikbare templates en modules, de EGA presenteert de GUI (Graphical User Interface) aan de gebruikers. De uiteindelijke GUI dient zoveel mogelijk conform de huisstijl van TiU opgeleverd te worden.

4.6 Het plaatsen van materiaal voor testdoeleinden

Als eenmaal alles geïnstalleerd en geconfigureerd is en de portal te bereiken is via een URL, dient er testmateriaal geplaatst te worden om te zien hoe alles functioneert en of de beoogde processen werken. Een belangrijk aspect hierbij is het definiëren van de metadata velden die men in dient te vullen bij het plaatsen van een bestand. Een bestand in een grote vergaarbak zonder enige informatie over het bestand is vrijwel nutteloos. Het afschermen van geplaatst materiaal wordt ook in deze activiteit meegenomen.

4.7 Koppelen middels een Single Sign On oplossing

Na een definitieve keuze voor Mediamosa bestaat de mogelijkheid dat dit platform als 'clouddienst' zal worden afgenomen. Eén van de securityrichtlijnen van TiU is dat accountgegevens (username/password) niet aan derden worden afgestaan. Dit is mogelijk door middel van een zgn. federatieve koppeling die door SURFnet wordt aangeboden. Dit houdt in dat wanneer een gebruiker zich aan wil melden bij het extern gehoste platform het inlogverzoek gestuurd wordt via de SURFfederatie naar de inlogserver van TiU. De kracht van deze methode is dan het eenvoud voor de gebruiker met zich meebrengt omdat er wordt gewerkt met Single Sign On (SSO). Wanneer een gebruiker al is aangemeld in een browsersessie bij een andere applicatie, pagina of portal waarbij ook deze koppeling is gemaakt met de SURFfederatie hoeft men niet nogmaals in te loggen. Omdat externe partijen als Google en Microsoft ook gekoppeld zijn en/of gaan koppelen met de SURFfederatie is de aanmeldmethode voorbereid op de toekomst. Hieronder is in een schema te zien hoe de koppeling gerealiseerd moet gaan worden.



4.8 Platform beschikbaar maken voor een selecte groep

Door het platform in eerste instantie beschikbaar te maken voor een selecte groep gebruikers kan er door interviews met gebruikers achterhaald worden of het systeem gebruiksvriendelijk en intuïtief werkt. Het is belangrijk om dit te doen voordat de portal aan een grotere groep gebruikers beschikbaar wordt gesteld. Facetten van 'Usability' en 'Understandability' zullen hier in terugkomen.

4.9 Oplevering en evaluatie van het project

Nadat alles in bedrijf is gesteld en er getest is met (potentiële) gebruikers is het platform gereed voor oplevering als 'proof of concept'. De uitgevoerde werkzaamheden en opgeleverde producten worden geëvalueerd met betrokkenen. Hierbij dient wel benadrukt te worden dat als het product succesvol wordt opgeleverd en ook het project succesvol wordt afgesloten, het aan het management is om het gerealiseerde platform verdere invulling te geven qua beleid en eventuele opschaling.

5 Projectgrenzen / Randvoorwaarden

5.1 Projectgrenzen

In dit Plan van Aanpak beschrijven we wat er WEL en NIET in het project uitgevoerd zal worden, de grenzen van het project bakenen we af.

5.1.1 Binnen de scope

Mogelijkheden die tijdens het project **WEL** worden onderzocht/gerealiseerd:

- Het is mogelijk elk mediabestand met een gangbaar formaat te uploaden;
- Het is mogelijk elk mediabestand van de juiste metadata te voorzien;
- Het is mogelijk elk geupload mediabestand naar een gangbaar formaat te transcoderen;
- Het is mogelijk elk mediabestand met een gangbaar apparaat/software af te spelen;
- Het is mogelijk elk mediabestand op de juiste manier af te schermen naar wens;
- De werking van het platform en bijbehorende componenten wordt WEL beschreven;
- Er wordt WEL onderzocht welke bestandformaten en resoluties de voorkeur hebben;
- De communicatie tussen 'back end' en 'front end' zal WEL worden beschreven;
- De initiële installaties van de machines zal slechts beperkt worden beschreven;
- De architectuur van de componenten van MediaMosa zal worden beschreven;
- Mogelijk opschalingsscenario's zullen WEL worden beschreven.

5.1.2 Buiten de scope

Mogelijkheden die tijdens het project **NIET** worden onderzocht/gerealiseerd:

- Er worden GEEN andere platformen dan MediaMosa onderzocht;
- Er worden GEEN andere transcodeertools dan FFmpeg onderzocht;
- Er worden GEEN andere streaming-services dan Wowza Media Server onderzocht;
- Er worden GEEN ander besturingssystemen dan Linux: Debian 64-bit onderzocht;
- Het platform zal GEEN gebruik maken van centrale dataopslag van TiU;
- Het platform is NIET per definitie voor iedereen binnen de instelling toegankelijk.

5.1.3 Overige zaken

- Alle benodigde servers worden gevirtualiseerd op één fysieke machine;
- Er wordt een omgeving ontwikkeld die een beperkt aantal gebruikers simultaan kan bedienen;
- Het beheer van de server vindt plaats door leden uit de projectorganisatie en dus NIET op centraal niveau;
- De fysieke machine hangt NIET in de centrale serverruimte van LIS en zodoende worden services, spanningsuitval en netwerkconnectiviteit NIET bewaakt.
- De projectorganisatie is NIET verantwoordelijk voor beleidsmatige taken;
- Binnen het project wordt niet besloten en/of beschreven hoe om te gaan met technisch en functioneel beheertaken bij uitrol;

5.2 Welslagen van het project

Het project is geslaagd wanneer er aan de volgende punten word voldaan.

- Het platform voor (een groep) gebruikers werkend beschikbaar is;
- Wanneer er een duidelijk verslag is van de uitgevoerde werkzaamheden;
- Uit onderzoek blijkt wat de meerwaarde is vergeleken met de huidige (probleem)situatie;
- Er is duidelijk wat de opbouw van het systeem is en waarom bepaalde afwegingen zijn gemaakt;
- In het eindverslag scenario's worden beschreven die opschaling en uitrol van het platform weergeven.

5.3 Risico's

R / G / M	Omschrijving
Risico	Het uitvallen van projectleider/opdrachtnemer tijdens de looptijd van het project wegens ziekte.
Gevolg	Slagingskans project in gevaar.
Maatregel	Overleg projectorganisatie, taken bekijken, slagingskans project bepalen.
Risico	Wegens de werkdruk op het bedrijf komen de uren gepland voor de afstudeeropdracht in het gedrang.
Gevolg	Planning niet haalbaar.
Maatregel	Overleg met opdrachtgever en teamleider om tot een oplossing te komen.
Risico	Beoogde componenten en/of onderdelen blijken niet met elkaar te kunnen werken.
Gevolg	Elke functionaliteiten niet haalbaar.
Maatregel	Melden bij projectorganisatie, gevolgen afwegen.
Risico	Single Sign On (SSO) koppeling met SURFfederatie is niet mogelijk.
Gevolg	Gevraagde aanmeldmethode kan niet gerealiseerd worden.
Maatregel	Melden bij opdrachtgever en tot oplossing zien te komen.

6 Producten

De op te leveren producten in dit project zijn op te delen in documenten en functionaliteiten:

6.1 Documenten

- Opdrachtschrijving;
- Plan van Aanpak;
- Communicatieplan (als bijlage bij PvA);
- Afstudeerverslag;
- Urenverantwoording.

6.2 Functionaliteiten

- Applicatieserver met MediaMosa en verwante databases/componenten/'building blocks';
- Jobserver met transcodeer-functionaliteiten op basis van FFmpeg;
- Streamingserver met Streaming Software op basis van Wowza;
- Een EGA met een GUI die passend is binnen de TiU omgeving qua huisstijl;
- Toegang tot het platform middels TiU account en bij voorkeur via de SURFfederatie en SSO.

6.3 Eindproduct

Het eindproduct is een werkend multifunctioneel videoplatform waar de beoogde gebruikers zelfstandig via een EGA hun opgenomen en/of verworven mediabestanden kunnen uploaden, beheren en beschikbaar stellen. Het platform is beschikbaar via een gepaste URL binnen het TiU domein en men meldt zich (bij voorkeur) aan met het TiU account via de SURFfederatie (zie paragraaf 4.7). De mediabestanden kunnen op een goede manier van metadata worden voorzien waardoor ze vindbaar zijn. De mediabestanden die beschikbaar zijn kunnen vervolgens worden geraadpleegd door alle gangbare apparaten met verschillende besturingssystemen. Het transcoderen van mediabestanden vindt plaats door het platform naar gangbare formaten. De mogelijkheid om mediabestanden als 'download' beschikbaar te stellen is aanwezig. In het afstudeerverslag is uitvoerig en duidelijk beschreven hoe het platform werkt en hoe de verschillende componenten gebruikt worden.

7 Kwaliteit

De kwaliteit van het project wordt gewaarborgd door dat er tussentijds korte bijeenkomsten plaatsvinden tussen opdrachtnemer en opdrachtgever (afstudeerbegeleider), dit zullen voornamelijk 1-op-1 bijeenkomsten zijn. De hardware en software wordt op cruciale punten ingesteld met medewerking van zeer ervaren collega's zodat vooral de kwaliteit van de beveiliging kan worden gegarandeerd. De gehele projectorganisatie is beschreven in het volgende hoofdstuk (hoofdstuk 8), een uitgebreid communicatieplan is toegevoegd als bijlage (Bijlage I).

7.1 Tools

Tijdens de installaties en configuraties zullen er logbestanden worden bijgehouden zodat er later terug te vinden is welke handelingen er zijn uitgevoerd. Dit zijn vooral binnen een UNIX omgeving belangrijke zaken omdat we gebruik zullen maken van een Debian omgeving zonder GUI. De machines zijn allen virtueel en worden beheerd door 'Oracle Virtualbox', hierin kunnen we 'snapshots' opslaan van de machines in bepaalde toestanden. Als er onverhoopt problemen optreden kan men terugvallen op zo'n 'snapshot'. Vóór en na de installaties van nieuwe componenten worden er 'snapshots' gemaakt om de kwaliteit en continuïteit van het project te garanderen. Tijdens de installatie van de componenten en bij het wijzigen van instellingen wordt er een logbestand bijgehouden met de uitgevoerde acties.

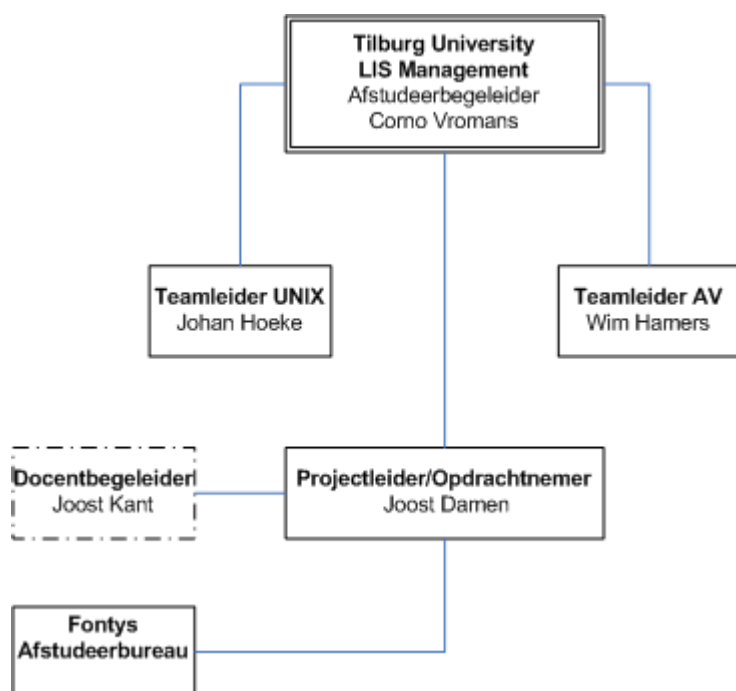
7.2 Extern advies

Op de MediaMosa website is een forum aanwezig waar mensen uit de 'Community' samenkomen met vragen, problemen en meldingen. Via dit forum kan er bij mensen met specifieke kennis over MediaMosa en bijbehorende componenten navraag worden gedaan over zaken. Beschrijvingen over de webservices van MediaMosa en achtergronddocumenten over de samenstelling zijn ook te vinden als naslagwerk op de MediaMosa website.

8 De projectorganisatie

De projectorganisatie bestaat uit de volgende personen:

Resource	Functie	Taken
Joost Damen	Opdrachtnemer	Projectleider
Corno Vromans	IT Manager	Afstudeerbegeleider / IT Manager
Johan Hoeke	Teamleider UNIX	Ondersteuning bij installatie/componenten
Wim Hamers	Teamleider LIS-AV	Leiding geven eigen afdeling
Joost Kant	Docentbegeleider	Advies geven, informeren, besluitvorming



Bovenstaande personen zijn deels bij het project begonnen, het project met bijbehorende werkzaamheden vindt voor circa 90% plaats door de opdrachtnemer, projectleider ofwel student. Andere personen zijn ondersteunend en hebben allen verschillende bevoegdheden en mandaten. De afstudeerbegeleider die tevens Manager IT is, heeft de grootste bevoegdheid binnen het departement en kan zo ook op cruciale punten met betrekking tot organisatorische problemen in actie komen. De 'lijnen' zijn intern bewust kort gehouden om het project succesvol te laten slagen.
































Als bijlage (Bijlage I) bij dit document is een communicatieplan toegevoegd waarin extra informatie is opgenomen met betrekking tot de communicatie tussen de belanghebbenden bij dit project. Hier zijn enkel de belanghebbenden in opgenomen die een verplichte en besluitvormende rol in dit project vervullen.

9 Planning

De planning is opgesteld in MS Project en wordt dus elektronisch bijgehouden. Het project is van start gegaan op 1 december en de afsluiting is gepland in de laatste week van mei 2012. In juni zal de presentatie van het project plaatsvinden. De projectactiviteiten die beschreven staan in hoofdstuk 4 zijn vastgelegd in de planning. De activiteiten worden ingedeeld in de volgende fases.

- Initiatieffase/Definitiefase: Plan van Aanpak;
- Ontwerpfase/Vorbereidingsfase: architectuur onderzoeken, initiële installaties;
- Realisatiefase: Componenten realiseren, EGA koppelen/ontwerp;
- Testfase: beschikbaar maken platform voor gebruikers, interviews;
- Nazorgfase: Evaluatie met gebruikers en opdrachtgever, beschrijven in verslag.

De planning is opgesteld met MS Project, op de volgende pagina is de planning te zien en is per activiteit (uit Hoofdstuk 4) beknopt beschreven wat er wanneer uitgevoerd gaat worden. Diverse activiteiten vinden parallel plaats omdat de ontwikkeling van het platform een proces is waarbij de onderdelen nauw met elkaar verbonden zijn.

Id		Taakmodus	Taaknaam	Duur	Begindatum	Einddatum
1			Initiatieffase en domeinanalyse	44 dagen	maa 21-11-11	don 19-1-12
2			Opdrachtschrijving	6 dagen	maa 21-11-11	maa 28-11-11
3			Plan van Aanpak	38 dagen	din 29-11-11	don 19-1-12
4			Initiele installaties	30 dagen	maa 16-1-12	vri 24-2-12
5			Installatie Eerste VM	11 dagen	maa 16-1-12	zat 28-1-12
6			Installatie Tweede VM	10 dagen	maa 30-1-12	vri 10-2-12
7			Installatie Derde VM	10 dagen	maa 13-2-12	vri 24-2-12
8			Installatie + configuratie van de losse onderdelen	30 dagen	maa 16-1-12	vri 24-2-12
9			Installatie + Configuratie componenten VM1	11 dagen	maa 16-1-12	zat 28-1-12
10			Installatie + Configuratie componenten VM2	10 dagen	maa 30-1-12	vri 10-2-12
11			Installatie + Configuratie componenten VM3	10 dagen	maa 13-2-12	vri 24-2-12
12			Het ontwikkelen en koppelen van de EGA/GUI	53 dagen	maa 9-1-12	woe 21-3-12
13			Onderzoek naar EGA	21 dagen	maa 9-1-12	maa 6-2-12
14			Koppelen EGA met backend	10 dagen	din 7-2-12	maa 20-2-12
15			EGA op maat aanpassen	22 dagen	din 21-2-12	woe 21-3-12
16			Werken aan afstudeerverslag	68 dagen	don 1-3-12	maa 4-6-12
17			Het plaatsen van materiaal voor testdoeleinden	12 dagen	maa 19-3-12	din 3-4-12
18			Materiaal selecteren en uploaden	3 dagen	maa 19-3-12	woe 21-3-12
19			Formaten onderzoeken + definiëren	10 dagen	woe 21-3-12	din 3-4-12
20			Het koppelen met SSO	19 dagen	woe 4-4-12	maa 30-4-12
21			Onderzoek naar SSO mogelijkheden	8 dagen	woe 4-4-12	vri 13-4-12
22			SSO koppeling realiseren	11 dagen	maa 16-4-12	maa 30-4-12
23			Platform beschikbaar maken voor een selecte groep	14 dagen	din 1-5-12	zon 20-5-12
24			Gebruikers selecteren + informeren	3 dagen	din 1-5-12	don 3-5-12
25			Platform beschikbaar maken voor LIS-AV	4 dagen	vri 4-5-12	woe 9-5-12
26			Testfase met gebruikers	8 dagen	don 10-5-12	zon 20-5-12
27			Oplevering en evaluatie van het project	11 dagen	maa 21-5-12	maa 4-6-12
28			Evaluatie met gebruikers	6 dagen	maa 21-5-12	maa 28-5-12
29			Evaluatie met opdrachtgever	4 dagen	maa 28-5-12	don 31-5-12
30			Inleveren afstudeerverslag	2 dagen	zat 2-6-12	maa 4-6-12

10 Kosten/baten-overzicht

De kosten die in dit project aan de orde komen zijn voornamelijk uit te drukken in werkuren. De hardware die vereist is voor het project is al aangeschaft en in bezit bij de organisatie. De kosten die dit project voor de opdrachtgever met zich mee brengt zijn de uren die de werknemer binnen werktijden aan de opdracht werkzaam is. De werkgever is hiervan bewust maar ziet dit niet als nadeel, omdat de kennisvergaring en het expertiseniveau op de te onderzoeken en te realiseren componenten als zeer belangrijk worden beschouwd.

Hieronder is het kostenoverzicht weergegeven

Resource	Functie	Taken	Uren*	Uurloon***	Kosten
Joost Damen	Opdrachtnemer	Projectleider	400**	€ 30,00	€ 12.000,00
Corno Vromans	IT Manager	Afstudeerbegeleider / IT Manager	12	€ 50,00	€ 600,00
Johan Hoeke	Teamleider UNIX	Ondersteuning bij installatie/componenten	20	€ 40,00	€ 800,00
Wim Hamers	Teamleider LIS-AV	Leiding geven eigen afdeling	20	€ 35,00	€ 700,00
Overige	Algemene uren	Navraag andere collega's/afdelingen/etc.	15	€ 40,00	€ 600,00
Totaal			467		€ 14.700,00

* Uren zijn de te verwachten uren gemaakt op basis van inschatting.

** Deze 400u wordt gemaakt onder werktijd de resterende 200u in eigen tijd.

*** Uurloon is op basis van inschatting en herleid van salarissen + overhead.

Ter verduidelijking wordt er nog aangegeven wat er qua componenten wordt gebruikt en nagenoeg "kosteloos" is. Er zijn althans geen aanschafkosten, de stroomkosten en opslagkosten zijn nihil voor deze kostenramingen.

Component	Functie	Kosten (?)
Virtuele machines	Beschikbaar stellen systemen	Nee, er wordt gebruik gemaakt van machines die al in huis zijn.
Opslag	Opslag van mediafiles (SAN/NAS)	Nee, er wordt gebruik gemaakt van "Storage" die in huis is.
MediaMosa	Back end core platform	Nee, MediaMosa is Open Source.
FFmpeg	Transcodeer tool	Nee, Ffmpeg is Open Source.
Wowza Media Server	Streaming Server Software	Nee, Wowza biedt een "Free Developers License"
Teamviewer	Beheer/bediening op afstand	Teamviewer is gratis te gebruiken.

Bijlage I: Communicatieplan

Belanghebbenden bij het project

Wie	Namens	Belang	Communicatievorm
Joost Damen	Opdrachtnemer voor Tilburg University, Student namens Fontys Hogeschool ICT	Projectleider	Uitvoeren, informeren, besluiten, overleggen, presenteren.
Corno Vromans	Tilburg University	Afstudeerbegeleider	Begeleiden, adviseren, informeren, overleggen, besluiten.
Joost Kant	Fontys Hogeschool ICT	Docentbegeleider	Begeleiden, adviseren, informeren, overleggen, besluiten.
Afstudeerbureau	Fontys Hogeschool ICT	Administratie	Informeren, begeleiden

Communicatiekanalen

Van	Naar	Informatie	Medium	Frequentie of data
Projectleider	Afstudeerbegeleider	Overleg, informeren, rapporteren, documenten	Mondeling, E-mail, telefonisch	Maandelijks en/of eerder wanneer nodig.
Projectleider	Docentbegeleider	Vragen, documenten, rapportage	E-mail, mondeling	Maandelijks en/of eerder wanneer nodig.
Projectleider	Afstudeerbureau	Vragen, Documenten	E-mail	Wanneer nodig
Afstudeerbegeleider	Projectleider	Ondersteuning, advies, besluit	Mondeling, e-mail	Wanneer nodig
Docentbegeleider	Projectleider	Ondersteuning, informatie, protocol	E-mail, telefoon, mondeling	Wanneer nodig/vastgestelde data
Afstudeerbureau	Projectleider	Ondersteuning, informatie	E-mail	Wanneer nodig/vastgestelde data

BIJLAGE II

Technische Bijlage

Toelichting en verdieping op relevante zaken

Inhoudsopgave

1	Technische verdieping MediaMosa	2
1.1	Front end applicaties	2
1.2	Registratie	3
1.3	Authenticatie	4
1.3.1	Autorisatie.....	4
1.4	Play Proxy.....	6
1.4.1	Voorbeeld.....	6
1.4.2	Toelichting.....	7
1.5	FTP Batch Upload	7
1.5.1	Probleemstelling	7
1.5.2	Bestanden	7
1.5.3	Software	8
1.5.4	Bevindingen.....	8
1.5.5	Voorbeeld VUF bestand	9
1.6	OAI-PMH	10
2	Aanpassingen aan Front end applicaties	10
2.1	Metadata.....	10
2.2	Downloadfunctionaliteit	11
2.2.1	Noodzaak van de downloadfunctionaliteit.....	11
2.2.2	Toevoegen van de downloadfunctie.....	11
2.2.3	Toevoegen van de downloadfunctie.....	12
2.2.4	Resultaat	14
3	simpleSAMLphp	15
3.1	Inleiding simpleSAMLphp.....	15
3.2	Configuratie simpleSAMLphp	17
3.3	Scenario: Gebruiker meldt zich aan bij test SSO server	19
3.4	Scenario: Gebruiker meldt zich aan bij SSO server	20
3.5	Screens Loginproces.....	21
3.5.1	Extra: scenario met gebruiker van andere Identity Provider.....	23
4	Object Code tbv Streaming	24

1 Technische verdieping MediaMosa

1.1 Front end applicaties

MediaMosa een *back end* die niet direct door een eindgebruiker benaderd kan worden. In de architectuur is een duidelijke scheiding zichtbaar tussen de *front end handling* en *back end handling*. Een *front end* applicatie wordt binnen MediaMosa ook wel omschreven als een Eind Gebruikers Applicatie (EGA). De ontwikkelaars van MediaMosa hebben enkele EGA *templates* ontwikkeld in de vorm van Drupal modules of beschikbaar gesteld als een complete Drupal installatie. Met deze installaties kan men redelijk snel een verbinding realiseren met een MediaMosa *back end*. Communicatie tussen deze EGA's vindt plaats via het REST-protocol (REpresentational State Transfer). REST is een variant op het *Simple Object Access Protocol* (SOAP), waarbij verzoeken aan de server niet in XML maar in URL vorm gedaan worden. De antwoorden van de server zijn altijd in XML vorm. Elke applicatie die die 'REST' kan praten zou met een MediaMosa *back end* gekoppeld kunnen worden. Dit kan dus ook een website of andere applicatie zijn die geen gebruikt maakt van de *templates* die worden aangeboden.

Voorbeeld van een verzoek van een EGA aan MediaMosa

EGA vraagt afspel-URL op van een mediabestand van een *asset*.

```
/asset/ncbXQIKiCTaWSLFWiuMEz6Zl/play?user_id=admin&mediafile_id=12MUUUqTkHuTaVXERk3a6kJo&response=plain&autostart=true&mediamosa_version=3.0.0
```

MediaMosa antwoord in dit geval met een ticket wat is aangemaakt voor het afspelen van het mediabestand, dit is een verwijzing naar het daadwerkelijke mediabestand op de server. (*hieronder is slechts een deel van het antwoord in XML weergegeven*).

```
<items>
  <item>
<output>http://app.mediamosa.uvt.nl/ticket/99/OjbTHTnRbZdfeTpBLoaR70dj</output>
    <content_type/>
    <ticket_id>OjbTHTnRbZdfeTpBLoaR70dj</ticket_id>
  </item>
</items>
```


Back end

MediaMosa bestaat uit componenten die samen de functionaliteit van een media management platform bieden. In dit hoofdstuk worden de verschillende componenten beschreven en aan de hand van voorbeelden wordt de werking van componenten en hun eigenschappen duidelijk gemaakt.

Dit hoofdstuk beschrijft de volgende componenten:

- Registratie
- Authenticatie
- Autorisatie
- Play Proxy
- OAI-PMH

1.2 Registratie

In de MediaMosa *back end* dienen *front end* applicaties te worden aangemaakt alvorens ze toegang kunnen krijgen tot MediaMosa. In de *back end* staan deze applicaties te boek als *Client Applications*. Een client heeft een uniek numeriek ID, een naam en een wachtwoord. Dit wachtwoord is een gedeelde sleutel welke aan de clientzijde en serverzijde opgeslagen (en uiteraard gelijk) moet zijn. Aan de clientzijde wordt opgegeven tegen welke MediaMosa *back end* deze applicatie moet communiceren. Om communicatie mogelijk te maken is er een *MediaMosa Connector Module* voor Drupal beschikbaar deze handelt de verzoeken en antwoorden af. Het aanmaken van een *client application* en de uitwisseling en het vastleggen van de gedeelde sleutel vindt één maal plaats in de configuratie van beide applicaties. De gedeelde sleutel wordt later gebruikt om de *front end* te authenticeren tegen de *back end*, dit staat beschreven in de volgende paragraaf.

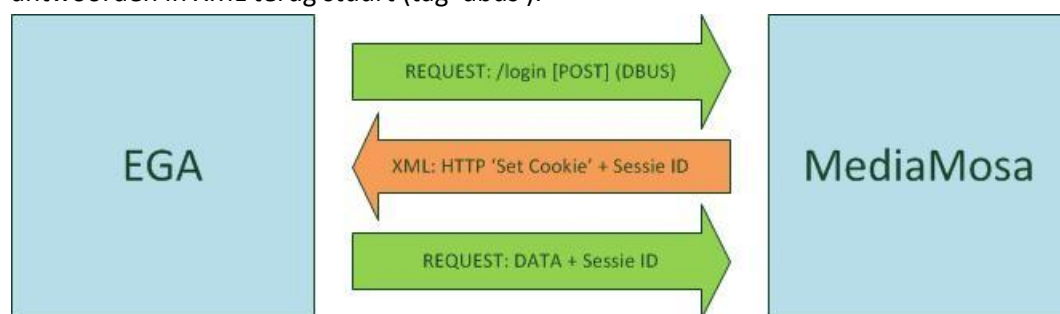
The screenshot displays the 'MediaMosa connector' interface, divided into two main sections. The left section, titled 'Connection settings to MediaMosa REST interface', contains a success message 'Normal call test successful, settings saved.' and a form for entering connection details. The form includes fields for 'URL' (http://app.mediamosa.uvt.nl), 'Username' (mmsb_client), and 'Password' (FTYXsWHYkTYTiEjKAPgTLsja), each with a descriptive label and a 'Test connection' button. The right section, titled 'Client application', shows the 'Settings for client application.' form. It includes fields for 'CLIENT APPLICATION NAME' (mmsb_client), 'APP ID' (66), and 'SHARED KEY' (FTYXsWHYkTYTiEjKAPgTLsja). A 'Paste generated key' button is located below the shared key field. A green callout box on the right explains: 'The shared key is the password that the application must use during the application verification.'

Figuur 1 (links) MediaMosa Connector Module, front end applicatie, (rechts) Registratie van front end bij MediaMosa back end.

1.3 Authenticatie

Voordat een EGA gebruik kan maken van MediaMosa, moet de EGA zich authenticeren bij MediaMosa door middel van een REST-call. Bij de eerste *request* van het authenticatieproces wordt een cookie teruggegeven door de MediaMosa-server (via de 'Set-Cookie' HTTP header string). Deze cookie bevat een sessie-id, die bij iedere volgende request ook weer als cookie moet worden teruggegeven (via de 'Cookie' HTTP header string). De EGA krijgt namelijk binnen de bijbehorende sessie toegang tot MediaMosa. Authenticatie van de EGA gebeurt volgens een challenge-response-protocol dat gebaseerd is op het D-Bus-protocol.¹

De client stuurt de D-Bus-berichten als HTTP POST data (variabele 'dbus'), waarna de server de antwoorden in XML terug stuurt (tag 'dbus').



Figuur 2 Visualisatie van het authenticatieproces tussen front end en back end.

Toelichting op drie stappen volgens bovenstaande figuur:

1. EGA doet een 'LOGIN request' en stuurt "AUTH DBUS_COOKIE_SHA1 <gebruikersnaam EGA>";
2. MediaMosa antwoordt met een Sessie ID en info wanneer de sessie verloopt;
3. EGA verzint een 'random' tekst en berekent de response string, deze heeft de volgende syntax: sha1(<challenge>:<random>:<wachtwoord>) (DBUS protocol).

1.3.1 Autorisatie

Bij autorisatie worden via afschermingsregels de rechten vastgelegd voor het raadplegen van mediabestanden. Het doel van afschermen is het verlenen van toegang aan gewenste groep gebruikers en het weigeren van de toegang aan de rest van de gebruikers. Via een EGA kunnen er afschermingsregels worden vastgelegd voor mediabestanden die via die EGA zijn vastgelegd.

Rechten worden op *mediafile* niveau toegekend, wat de mogelijkheid biedt om bijvoorbeeld een lage kwaliteit video aan een grote groep gebruikers aan te bieden. Een hoge kwaliteits versie van het bestand zou aangeboden kunnen worden aan een selecte groep gebruikers. De autorisatie hiervan vindt plaats bij het verzoek tot afspelen van een *mediafile*.

¹ Beschrijving DBUS protocol: <http://dbus.freedesktop.org/doc/dbus-specification.html>

Wanneer een mediafile wordt geüpload is deze standaard niet afgeschermd, het bestand is dan ‘publiek’ beschikbaar. Na het uploaden kan men indien gewenst de volgende afschermingsregels toepassen:

Domein

Een internet domeinnaam, bijvoorbeeld geef alleen “*uvt.nl*” toegang, dit geldt ook voor alle subdomeinen van de opgegeven domeinnaam. Ook is er de mogelijkheid om een groep domeinnamen te maken waarin meerdere domeinen zijn opgenomen. Dit kan in het geval van TiU interessant zijn voor aanverwante instellingen en instituten. Een voorbeeld is “*NEXUS.group*”, een domeingroep dient te eindigen met “*.group*”.

Realm

Voorbeelden zijn “*Joost@uvt.nl*” of “*@asset-economics.nl*”. Het eerste voorbeeld is slechts voor één gebruiker, het tweede voorbeeld geeft toegang aan alle gebruikers die voldoen aan “*@asset-economics.nl*” en eventuele subdomeinen daarvan. Ook bij deze optie zijn groepen mogelijk een groep van realms kan bijvoorbeeld zijn: “*@NEXUS.group*”, deze groepen dienen ook te eindigen op “*.group*” en te starten met een “*@*”.

Gebruikers

Dit zijn gebruikers binnen *front end* applicatie, er wordt gekeken naar het *user_id* dat wordt meegegeven bij een opdracht tot het afspelen van een mediabestand. Het gaat hier om een exacte match van een *user_id*, dus zoals een gebruiker bekend is in de *front end* applicatie. Voorbeeld van het betreffende attribuut is: `ACL_USER[] = J.M.J.Damen@uvt.nl`, in dit voorbeeld heeft enkel deze gebruiker toegang tot het betreffende mediabestand.

Groepen

Een gebruiker kan in een groep zitten en een groep kan geautoriseerd worden op mediabestanden. Dit is een interessante optie wanneer een selecte groep gebruikers toegang moet krijgen tot mediabestanden.

EGA's

Een speciale optie is om aan een andere EGA afspeelrechten toe te kennen op een bestand wat door een EGA wordt beheerd. Hier voor moet het *applicatie-id* bekend zijn bij de andere EGA. Binnen deze EGA is de mediafile dan publiek beschikbaar, mits er geen realm- of domeinrechten zijn gezet. Als deze wel gezet zijn, moet de andere EGA ook voldoen aan de realm- of domeinrechten om de mediafile te kunnen bekijken.

1.4 Play Proxy

De component die afspeelverzoeken verwerkt is de *Play Proxy* component. Een afspeelverzoek vindt plaats op basis van een *asset_id* en een *mediafile_id*. Een afspeelverzoek is afkomstig van een EGA, het antwoord is afhankelijk van het verzoek. Als response wordt een URI naar de streaming videosever teruggegeven (plain) al dan niet in een gewenst formaat (bv metafile of object). De volgende stappen en acties vinden plaats in MediaMosa bij een play proxy verzoek:

1. Haal informatie van de betreffende asset op;
2. Controleer de autorisatie op basis van afschermingsregels;
3. Creeeren van een afspeelticket en een symbolic link (symlink) in de dataopslag;
4. Genereer de betreffende response (object, url, etc) en haal de afspeelgegevens op uit de objectcode van de betreffende codec/container.

1.4.1 Voorbeeld

Aanroep:

<https://wle.mediamosa.uvt.nl/play/pETPfwNTMRIQUc7rLRQvNgEq?response=object&filename=00030.mp4>

Response object code:

```
jwplayer("pETPfwNTMRIQUc7rLRQvNgEq").setup({
  height: skin_height,
  width: 320,
  image: "",
  modes: [
    { type: "flash",
      src: "http://job1.mediamosa.uvt.nl/mmplayer/jwplayer/player.swf",
      config: {
        file: "99/S1aYifWCILkRgYOZUJymrEOo.mp4",
        streamer: "rtmp://137.56.72.176:1935/mediamosa",
        provider: "rtmp"
      }
    },
    { type: "html5",
      "controlbar.position": "bottom",
      config: {
        file: "http://app.mediamosa.uvt.nl/ticket/99/S1aYifWCILkRgYOZUJymrEOo.mp4",
        'provider': 'http',
        'x-webkit-airplay': 'allow'
      }
    }
  ]
});
```

1.4.2 Toelichting

Het originele Mediafile ID (lichtblauw gearceerd) enkel te zien is voor de *setup* van de player. Voor het daadwerkelijk afspelen van het bestand maakt MediaMosa gebruik van een *ticketing* constructie. Een *ticket* is 20 minuten geldig, wanneer een niet geautoriseerde gebruiker het bestand wil afspelen is dat op deze manier vrijwel onmogelijk.

1.5 FTP Batch Upload

1.5.1 Probleemstelling

Voor de afdeling LIS-AV is het zeer nuttig om meerdere mediabestanden in een batch te uploaden. Klanten leveren vaak grote hoeveelheden video aan en deze zouden via de 'normale' weg stuk voor stuk via een front end toegevoegd worden. Dit is zeer arbeidsintensief omdat men alle stappen telkens weer moet doorlopen. Het doel van FTP Batch Upload is dat de stappen die een reguliere eindgebruiker moet doorlopen overgeslagen kunnen worden. FTP upload gaat buiten een front end applicatie om. Hierdoor zou men in een korte tijd meerdere bestanden kunnen uploaden via FTP. Het is echter niet mogelijk om enkel mediabestanden via FTP te uploaden. Samen met deze mediabestanden dienen aanvullende bestanden die de metadata bevatten meegestuurd te worden. De metadata zou anders per bestand nog achteraf toegevoegd moet worden of bestanden zouden geen metadata bevatten. Dat is niet wenselijk omdat dit een vereiste is om bestanden te zoeken en vindbaar te maken.

1.5.2 Bestanden

De drie benodigde bestanden bij FTP batch uploads zijn:

Videobestand

Het daadwerkelijk videobestand dat beschikbaar moet worden via MediaMosa.

.qdc bestand (qdc = Qualified Dublin Core)

Bevat de metadata in XML formaat, moet voldoen aan Dublin Core en/of Qualified Dublin Core standaard.

.vuf bestand (vuf = Video Upload File)

Bevat de bestandsnaam van het bijgevoegde videobestand, metadatabestand en variabelen als downloadable en available kunnen hier direct meegegeven worden. Ook dit bestand is in XML formaat.

1.5.3 Software

Om FTP batch uploads mogelijk te maken is de volgende software op de server benodigd:

FTP server software

Maakt het mogelijk dat een server ook beschikbaar is voor FTP toegang. Software waarmee dit onder andere mogelijk is, is ProFTPd.

XML-parser

Deze parser is nodig om de gegevens uit een Video Upload File (.vuf bestand) te vertalen naar REST-calls die de back end kan verwerken.

VUF Watcher software

Voor MediaMosa is watcher software gemaakt die een opgegeven map monitored op de aanwezigheid van VUF bestanden.

1.5.4 Bevindingen

De twee aanvullende bestanden moeten in een teksteditor of iets dergelijks gemaakt worden. Dit is voor de beoogde gebruikers niet echt gebruiksvriendelijk. Om dit eenvoudiger en werkbaar te maken zou het zinvol zijn als er een tool gebruikt kan worden waar men tekstvelden en variabelen eenvoudig kan invullen en aanvinken. Het ontwikkelen van een tool valt niet binnen de scope van het project. Aan de hand van de mogelijkheden en om een beter inzicht te geven hoe deze tool eruit zou zien is er een mockup gemaakt.

The mockup shows a web form titled "MediaMosa - FTP BATCH UPLOAD TOOL". It contains several input fields for metadata: "Titel*", "Onderwerp*", "Eigenaar*", "Identifier", "Taal", "Collectie", "Maker", and "Email notificatie". There are also dropdown menus for "Afscherming" (with "Groep" selected) and "Transcodeerprofiel" (with "H264 SD" selected). A "Mediabestand" field is followed by an "Open" button. At the bottom, there are two checkboxes, "Downloadable" and "Beschikbaar", both of which are currently unchecked. A large "Genereer bestanden" button is positioned at the very bottom of the form.

Figuur 3 Mock up van een mogelijke tool, geschikt voor medewerkers LIS-AV

1.5.5 Voorbeeld VUF bestand

<FTPUpload> the root element

<email>[EMAIL ADDRESS]</email> the email address where upload status messages will be sent to

<videos> the list of assets that are to be uploaded

<video> an asset

<action>[save|update|delete]</action> the action that has to be executed on the media file(s) and/or metadata file

<referenceId>[STRING]</referenceId> a unique identifier of the asset, created by the uploader

<metadataFile>[FILENAME]</metadataFile> a reference to the metadata file

<formats> the list of media files that are to be uploaded

<format> a media file

<action>[save|update|delete]</action> the action that has to be executed on the media file

<external>[URL]</external> a reference to the media file if it is stored outside MediaMosa

<mediaFile>[FILENAME]</mediaFile> a reference to the media file (if it has to be uploaded to MediaMosa)

<available>[yes|no]</available> indicates whether the media file has to be visible in the interface or not

<downloadable>[yes|no]</downloadable> indicates whether the media file has to be downloadable or not

<createStill>[yes|no]</createStill> indicates whether a video still has to be created or not

<frameTime>[INTEGER]</frameTime> the number of seconds after the beginning of the video, indicating the frame of which the still is created

</format>

</formats>

</video>

</videos>

<collections> a list of collections in which all of the uploaded media files have to be included

<collectionId>[STRING]</collectionId> a unique identifier of a collection

</collections>

<categories> a list of categories to which all of the uploaded media files have to be related

<categoryId>[STRING]</categoryId> a unique identifier of a category

</categories>

<accessRules> a list of rules that set down which end users are allowed to playback the media files in this upload

<accessRule>

<ruleType>[DOMAIN|REALM|GROUP|APP]</ruleType> one of the four types of user sets

<ruleValue>[STRING]</ruleValue> the value of the domain (e.g. surfnet.nl), realm (e.g. peter@surfnet.nl), group (e.g. academia.group)

</accessRule>

</accessRules>

<transcodings> a list of transcodings which have to be executed on all of the uploaded media files

<transcodingName>[TRANSCODING PROFILE]</transcodingName> the name of a transcoding profile laid down before

</transcodings>

<recentList>[yes|no]</recentList> indicates whether or not all of the uploaded media files have to be included in a variety of lists on the website

</FTPUpload>

1.6 OAI-PMH

OAI-PMH staat voor Open Archive Initiative – Protocol Metadata Harvesting. Dit protocol is een internationale afspraak om metadata uit een repository op te halen. MediaMosa zou hier functionaliteiten voor aanbieden. Na onderzoek is gebleken dat deze functionaliteit nooit is uitgebracht. Het idee was om een applicatie uit te brengen die op een aparte machine zou kunnen draaien. Dit zou een losstaande module zijn die verder ook geen gebruik zou maken van andere componenten binnen MediaMosa. De applicatie zou de metadata van de assets in MediaMosa beschikbaar kunnen stellen zodat ze door een Harvester opgehaald kunnen worden. Harvesters halen metadata op van providerapplicaties om deze metadata in bijvoorbeeld een zoekmachine beschikbaar te stellen. Omdat de doorontwikkeling van MediaMosa op het moment van schrijven nog onduidelijk is, is er nog niet bekend of deze functionaliteit nog wel wordt geïmplementeerd.

2 Aanpassingen aan Front end applicaties

Er zijn diverse *front end* applicaties beschikbaar gesteld in de vorm van een *template*. Deze zijn deels vindbaar via de MediaMosa website maar het is nogal onduidelijk welke versie de meest complete en recente is. Na wat uitzoekwerk kwam ik er achter dat er twee *githubs* online staan die beide versies bevatten die werkbaar waren.² De doorontwikkeling van nieuwe *front ends* voor Drupal 7 hebben flinke vertraging opgelopen en werden tijdens de looptijd van de opdracht nog steeds als niet stabiel aangemerkt. In een laat stadium van de opdracht heb ik de beschikking gekregen over de MediaMosa SiteBuilder (SB) *front end* de nieuwste *front end* die wordt aangeboden via de MediaMosa website. In het verslag is aangegeven dat de WLE is gebruikt om de invoervelden voor metadatering en de downloadfunctionaliteit te onderzoeken en toe te voegen.

2.1 Metadata

De metadata die vóór een *asset* aangemaakt moet worden ingevuld is in de bestaande code aangepast tot de velden die voor ons relevant waren. Hierbij ben ik er uitgegaan van de Dublin Core Standaard. De relevante velden zijn hieronder weergegeven met de Dublin Core benaming en de beschrijving:

- dc:title (de titel);
- dc:subject (een onderwerp in de vorm van sleutelwoorden);
- dc:description (een beschrijving);
- dc:type (type: audio of video);
- dc:language (de taal van de resource);
- dc:identifier (de unieke identifier, voor TiU interessant voor bijv. vakcode);
- dc:rights (de rechten die op het materiaal berusten, toelichting voor gebruik);
- dc:creator (de maker van het materiaal, persoon kan het uploaden uitbesteden);
- dc:contributor (andere deelnemers, zoals, presentatoren, acteurs, etc)
- dc:date (de datum van opname, verwerving of de aanmaak datum);
(wanneer datum niet wordt ingevoerd wordt automatisch de datum van dat moment toegevoegd)

² MediaMosa github: <https://github.com/mediamosa/> en <https://github.com/Driesvanl/>

2.2 Downloadfunctionaliteit

2.2.1 Noodzaak van de downloadfunctionaliteit

Het beschikbaar stellen van bestanden als *download* is een belangrijke functie. Hiermee wordt bedoeld het *downloaden* van bestanden zodat men ze lokaal op een computer kan gebruiken. Dit is praktisch als de beoogde gebruikers het materiaal verder moeten verwerken en ze hebben de mogelijkheid het materiaal te bekijken wanneer ze niet over een internetverbinding beschikken. Dergelijke situaties komen bijvoorbeeld voor bij gebruikers die een presentatie op locatie moeten geven en waar geen internetverbinding voor handen is. Ook komt het voor dat studenten die voor een periode naar het buitenland vertrekken toegang nodig hebben tot mediabestanden maar waarvan men vooraf al weet dat er geen (goede) internetverbinding aanwezig is. De afdeling LIS-AV wordt ook regelmatig benaderd met verzoeken om materiaal wat opgenomen is verder te bewerken tot een fatsoenlijk eindproduct. In de huidige situatie gebeurt dit door het kopiëren van deze bestanden op de 'TEMP drive' door een medewerker van de LIS Service Desk. Dit is een schijf die door iedere medewerker van LIS beschikbaar is en dus niet bepaald geschikt qua privacy en veiligheid. Medewerkers van de dienst LIS-AV kopiëren dit bestand vanaf de 'TEMP drive' voor verdere verwerking. Omdat deze wens bij gebruikers steeds meer toeneemt, is het noodzakelijk dat deze functionaliteit geautomatiseerd kan worden en een betere beveiliging van de bestanden moet garanderen.

2.2.2 Toevoegen van de downloadfunctie

Omdat de optie om mediabestanden als *downloadable* beschikbaar te stellen niet beschikbaar was in de *front end* heb ik besloten deze zelf in de bestaande code te implementeren. Er was bekend dat via een REST *call* naar de MediaMosa *backend* de mogelijkheid bestond om dit werkend te krijgen. Hiervoor is eerst de mogelijkheid onderzocht met behulp van de *MediaMosa REST Call Developer*, op deze manier was te controleren of er voor een bestaand mediabestand het attribuut '*is_downloadable*' van *FALSE* (*default*) naar *TRUE* gezet kan worden. Het resultaat was dat dit inderdaad mogelijk was, door de aanroep te simuleren naar de *back end*. In de technische bijlage is de code met toelichting te vinden die de implementatie van de downloadfunctionaliteit binnen de WLE *front end* beschrijft.

Zodra alle metadata velden zijn ingevuld en de gewenste *downloadable* optie gekozen is, wordt er een *asset* met een bijbehorende (nog lege) mediafile aangemaakt.

2.2.3 Toevoegen van de downloadfunctie

Bron: `/opt/wle/sites/all/modules/wle_upload/wle_upload.module`

Toelichting: Dropdown-box op formulier plaatsen, keuze Ja/Nee die respectievelijk *true* of *false* meegeeft voor het attribuut. Standaardwaarde is Nee (*false*).

```
$form['is_downloadable'] = array(
  '#type' => 'select',
  '#title' => t('Maak bestand downloadbaar'),
  '#default_value' => 'Nee',
  '#options' => array(
    'false' => t('Nee'),
    'true' => t('Ja'),
  ),
);
```

Toelichting: creëert een *asset* in MediaMosa, een *asset* wordt aangemaakt met een willekeurig (uniek) ID. Wanneer MediaMosa een *asset* ID heeft gegenereerd, aangemaakt en heeft teruggegeven word het *asset* ID bekend bij de *front end* applicatie.

```
/**
 * Helper function to create a new asset
 */
function _wle_upload_create_asset(&$vpx) {
  $response = $vpx->request('POST', '/asset/create');

  if ($vpx->check_result($response)) {
    return (string)$response->items->item->asset_id;
  }
  return FALSE;
}
```

Toelichting: creëert een lege mediafile die gerelateerd wordt aan de eerder aangemaakt *asset* (*asset_id*), het attribuut '*is_downloadable*' wordt hier als parameter meegegeven. Dit attribuut bepaald of het (nog te uploaden) mediabestand als download beschikbaar word gesteld of niet. Wanneer de mediafile is aangemaakt door MediaMosa word het '*mediafile_id*' voor verdere *settings* als referentie gebruikt binnen de *front end* applicatie.

```

/**
 * Helper function to create a new mediafile
 */
function _wle_upload_create_mediafile(&$vpx, $asset_id, $is_downloadable = 'false') {
  $response = $vpx->request('POST', '/mediafile/create', array('asset_id' => $asset_id, 'is_downloadable'
=> $is_downloadable));
  if ($vpx->check_result($response)) {
    return (string)$response->items->item->mediafile_id;
  }
  return FALSE;
}

```

Bron: **/opt/wle/sites/all/modules/wle_asset/wle_asset.module**

Toelichting: Deze code zorgt ervoor dat de *download URL* wordt opgevraagd wanneer de *asset* wordt aangeroepen binnen de portal door de eindgebruiker.

```

// Fetch the mediafile's HTML play object
$size = explode('x', variable_get('wle_player_object_size', '320x240'));
$player = $vpx->request('GET', '/asset/'. $asset_id .'/play?autostart=false&mediafile_id=' .
$mediafile_id . '&response=object&width=' . $size[0] . '&height=' . $size[1] . $start);
$download = $vpx->request('GET', '/asset/'. $asset_id .'/play?autostart=false&mediafile_id=' .
$mediafile_id . '&response=download' . $start);

return theme('wle_asset_details', $metadata, $mediafiles, $player, $mediafile_id, $asset_id,
$download);

```

Bron: **/opt/wle/sites/all/modules/wle_shared/wle_shared_theme.inc**

Toelichting: Deze code zorgt ervoor dat de tekst “download” wordt weergegeven wanneer de waarde aanwezig is (*true*). De onderste twee regels laten zien dat de *download URL* als *hyperlink* wordt neergezet in de EGA.

```

function theme_wle_asset_details($metadata, $mediafiles, $player, $active_mediafile_id, $asset_id,
$download) {
  drupal_add_css(drupal_get_path('module', 'wle_shared') .'/wle_shared_theme.css');
  $title = wle_shared_trim_string($metadata['title'], 60);
  drupal_set_title($title);
  $edit = wle_shared_acl_check('edit assets', 'asset', $asset_id);

  if (isset($download->items->item->output)) {
    $add .= ' | <a href="' . (string) $download->items->item->output . '">download</a>';
  }
}

```

2.2.4 Resultaat

Men heeft de keuze om tijdens het uploaden via het *drop-down menu* de optie kan kiezen of het mediabestand als *download* beschikbaar moet zijn voor de geautoriseerde gebruikers (kan in het geval van publiek ook iedereen zijn) (nieuwe zin, verwarring. Men heeft de keuze...optie kan kiezen, is vrijwel hetzelfde). Wanneer de uploader er voor kiest het bestand niet als *download* beschikbaar te stellen, heeft de uploader zelf wel de mogelijkheid zijn eigen bestand te downloaden. Dit is een legitieme optie omdat men het bestand ook zelf beschikbaar heeft gesteld. Er is voor een download geen tussenkomst meer nodig van een medewerker.

3 simpleSAMLphp

3.1 Inleiding simpleSAMLphp

simpleSAMLphp is een applicatie die in PHP geschreven is en mogelijkheden biedt om te fungeren als *Service Provider* of *Identity Provider*. Er is voor Drupal (6 & 7) een module beschikbaar om simpleSAMLphp te gebruiken binnen een Drupal installatie. Bij deze module dient eerst bekeken te worden op welke Drupal versie hij gebruikt gaat worden. In het geval van het gerealiseerde *Proof of Concept* hebben we twee Drupal *frontends* beschikbaar namelijk de MediaMosa WLE en de MediaMosa SiteBuilder. De WLE *front end* is een Drupal 6 installatie dus is de simpleSAMLphp 6.x-2.7 module nodig³. Een aandachtspunt is wel dat deze module gebruik maakt van simpleSAMLphp, de module zelf is dat dus niet.

In het metadata bestand worden er gegevens vastgelegd over de *Identity Provider*, in dit geval *test.saml.uvt.nl*.

Dit is (deels) de inhoud van het bestand:

```
$metadata['https://test.saml.uvt.nl/saml2/idp/metadata.php'] = array (
  'metadata-set' => 'saml20-idp-remote',
  'entityid' => 'https://test.saml.uvt.nl/saml2/idp/metadata.php',
  'SingleSignOnService' => 'https://test.saml.uvt.nl/saml2/idp/SSOService.php',
  'SingleLogoutService' => 'https://test.saml.uvt.nl/saml2/idp/SingleLogoutService.php',
  'certData' => 'Certificaat'
```

Het 'entityID' wordt gebruikt als unieke identificatie voor een SAML entiteit. Bij dit bestand is er gekozen voor de complete URL van de locatie waar het metadata bestand te vinden is, dit is namelijk ook een uniek adres. De 'SingleSignOnService' wordt omschreven als een *endpoint* binnen SAML, het is een URL waar een *redirect* naar plaatsvindt. SAML 2.0 maakt gebruik van *HTTP-Redirect binding*⁴ om verbinding te maken met zo'n *endpoint*. In dit geval is het de pagina waar doorverwezen word bij het inloggen bij de *Identity Provider*. Voor de 'SingleLogoutService' geldt hetzelfde: ook dit is *endpoint* waar naar *geredirect* wordt, in dit geval gaat het om de pagina die wordt weergegeven wanneer een gebruiker uitlogt.

Binnen het metadata bestand is dit zichtbaar aan de manier waarop de *binding* is gedefinieerd:

```
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect."
```

Op de positie waar 'Certificaat' is ingevuld staat een certificaat wat bestaat uit meer dan 1600 karakters. Deze karakters zijn geëncodeerd via het Base64 *encoding scheme*⁵, deze manier van encodieren van data wordt gebruikt voor gegevens die worden opgeslagen en overgebracht over netwerken. Base64 wordt ook toegepast bij het coderen van bijlages in emailberichten, in dit geval gaat het om en encodieren en later decoderen van een grote hoeveelheid data in XML, namelijk het certificaat van de *Identity Provider*.

³Module pagina simpleSAMLphp: http://drupal.org/project/simplesamlphp_auth

⁴<http://simplesamlphp.org/docs/1.6/simplesamlphp-reference-idp-remote>

⁵<http://en.wikipedia.org/wiki/Base64>

Onze Service Provider nog bekend worden bij de Identity Provider. De Identity Provider heeft de URL nodig die verwijst naar het metadata bestand (XML) van de Service Provider. Binnen de simpleSAMLphp *web interface* is deze eenvoudig op te vragen, onderstaande afbeelding geeft een idee hoe dit aan de Service Provider kant weergegeven word. In de Identity Provider wordt de Service Provider vastgelegd aan de hand van de metadata, de Identity Provider en Service Provider hebben nu “weet” van elkaar en kunnen met elkaars verzoeken omgaan.

SAML 2.0 SP Metadata

English | Bokmål | Nynorsk | Sámi | Suomeksi | Dansk | Svenska | Deutsch | Español | Français | Nederlands | Luxembourgish | Hrvatski | Magyar | Język polski | Slovenščina | Português | Português brasileiro | Türkçe

SAML 2.0 SP Metadata

Dit is de metadata die automatisch is gegenereerd door simpleSAMLphp. U kunt deze metadata uitwisselen met uw federatiepartners.

U kunt deze directe URL gebruiken om de metadata XML op te vragen:

<http://wle.mediamosa.uvt.nl/simplesamlphp/module.php/saml/sp/metadata.php/default-sp>

Metadata

In SAML 2.0 Metadata XML formaat:

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" entityID="http://wle.mediamosa.uvt.nl/simplesamlphp/module.php/saml/sp/metadata.php/default-sp">
  <md:SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:1.1:protocol urn:oasis:names:tc:SAML:2.0:protocol">
    <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Location="http://wle.mediamosa.uvt.nl/module.php/saml/sp/LogoutService.php/default-sp">
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="http://wle.mediamosa.uvt.nl/module.php/saml/sp/AssertionConsumerService.php/default-sp">
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post" Location="http://wle.mediamosa.uvt.nl/module.php/saml/sp/AssertionConsumerService.php/default-sp">
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact" Location="http://wle.mediamosa.uvt.nl/module.php/saml/sp/AssertionConsumerService.php/default-sp">
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01" Location="http://wle.mediamosa.uvt.nl/module.php/saml/sp/AssertionConsumerService.php/default-sp">
  </md:SPSSODescriptor>
  <md:ContactPerson contactType="technical">
    <md:GivenName>Administrator</md:GivenName>
    <md:EmailAddress>na@example.org</md:EmailAddress>
  </md:ContactPerson>
</md:EntityDescriptor>
```

In simpleSAMLphp flat file formaat - gebruik dit wanneer uw federatiepartner ook simpleSAMLphp gebruikt

```
$metadata['http://wle.mediamosa.uvt.nl/simplesamlphp/module.php/saml/sp/metadata.php/default-sp'] = array (
  'AssertionConsumerService' => 'http://wle.mediamosa.uvt.nl/simplesamlphp/module.php/saml/sp/saml2-acs.php/default-sp',
  'SingleLogoutService' => 'http://wle.mediamosa.uvt.nl/simplesamlphp/module.php/saml/sp/saml2-logout.php/default-sp',
);
```

Copyright © 2007-2010 Feide RnD

Figuur 4 Screenshot Metadata weergave Service Provider via Drupal module

3.2 Configuratie simpleSAMLphp

Tijdens de reguliere installatieprocedure van simpleSAMLphp dienen er wat zaken handmatig gewijzigd te worden. Wanneer men gebruik maakt van het beschikbare package worden er verschillende handelingen uitgevoerd bij installatie.

De software gaat straks fungeren als webpagina binnen een bestaande website, het gebruikte front end is bereikbaar als website en in de Apache configuratie is een alias aangemaakt voor simpleSAMLphp:

```
Alias /simplesamlphp /usr/share/simplesamlphp/www
<Directory /usr/share/simplesamlphp/www>
    Order allow,deny
    Allow from all
</Directory>
```

De WLE *front end* is bereikbaar via wle.mediamosa.uvt.nl, de alias `/simplesamlphp` zorgt ervoor dat simpleSAMLphp *web interface* toegankelijk is via een browser. Om de wijzigingen plaats te laten vinden dient de Apache webserver herstart te worden. In de map `/usr/share/simplesamlphp/` op de applicatieserver is simpleSAMLphp geïnstalleerd, dit path is gebruikt door het gebruik van het *package*. In de `www` map staan de betreffende bestanden die ervoor zorgen dat de *web interface* benaderbaar is.

Het belangrijkste configuratiebestand is *config.php* wat door het package is geplaatst in de map `/etc/simplesamlphp/`. In dit bestand worden normaliter het 'auth.adminpassword' en de zogenaamde 'secretsalt' opgeslagen. In het geval van het gebruikte package zijn deze belangrijke en vertrouwelijke gegevens op een andere plaats opgeslagen en maakt het configuratiebestand door middel van een *include* gebruik van een extra bestand: `/var/lib/simplesamlphp/secrets.inc.php`.

Het 'auth.adminpassword' is nodig om toegang te krijgen tot sommige pagina's van simpleSAMLphp via de *web interface*, tijdens de installatie is een willekeurige string gegenereerd van 8 karakters. De 'secretsalt' wordt gebruikt door onderdelen van de software om hiermee *hashes* te genereren, het is een willekeurige string die ook tijdens de installatie is gegenereerd en is 32 karakters lang.

In het bestand *authsources.php* wordt de naam opgegeven van de Service Provider, de standaardwaarde gebruik ik voor deze test en is 'default-sp'. Het bestand bestaat niet bij installatie en kan verkregen worden door een beschikbare *template* te kopiëren, bestanden zijn ingepakt en zijn hier te vinden `/usr/share/doc/simplesamlphp/examples/config-templates/authsources.php.gz`.

De installatie en configuratie die is gerealiseerd heeft betrekking op feit dat het platform wil acteren als Service Provider. De Service Provider moet vooraf weten met welke Identity Provider(s) hij moet gaan verbinden, dit is het meest cruciale stuk omdat hier de kern van de toekomstige authenticatiestructuur geconfigureerd word. De communicatie vindt plaats op basis van metadata bestanden in XML formaat.

Er is een metadata bestand gebruikt dat aangeleverd is, er is namelijk binnen TiU een SSO testomgeving beschikbaar (test.saml.uvt.nl). Deze treedt op als Identity Provider en maakt ook gebruik van simpleSAMLphp. Binnen de simpleSAMLphp installatie zijn ook hier templates voor beschikbaar maar dit is in dit geval niet nodig om het juiste bestand wat nodig is al bestaat. Er zijn drie verschillende bestanden voor handen namelijk: saml20-idp-hosted.php, saml20-idp-remote.php en saml20-sp-remote.php. In mijn geval is er een voorbereid bestand verkregen en dat is de saml20-idp-remote.php variant, het scenario is namelijk om verbinding te maken met een Identity Provider op afstand. Bestandslocatie: */etc/simplesamlphp/metadata/saml20-idp-remote.php*

De scenarios die zijn gebruikt om de module in combinatie met simpleSAMLphp en de TiU omgeving te testen zijn hieronder weergegeven. Het eerste scenario maakt gebruik van een standaard (onbeveiligde) http verbinding naar de test SSO-server van TiU. Voor het tweede scenario is er door Teun Nijssen een certificaat aangemaakt voor de *front end* (wle.mediamosa.uvt.nl) en zo kan het verkeer plaatsvinden via https. De koppeling is nu niet gemaakt met de testomgeving maar met de SSO productieomgeving van TiU.

3.3 Scenario: Gebruiker meldt zich aan bij test SSO server

Opmerkingen: de verbindingen vinden plaats via *http*, verkeer is dus niet beveiligd

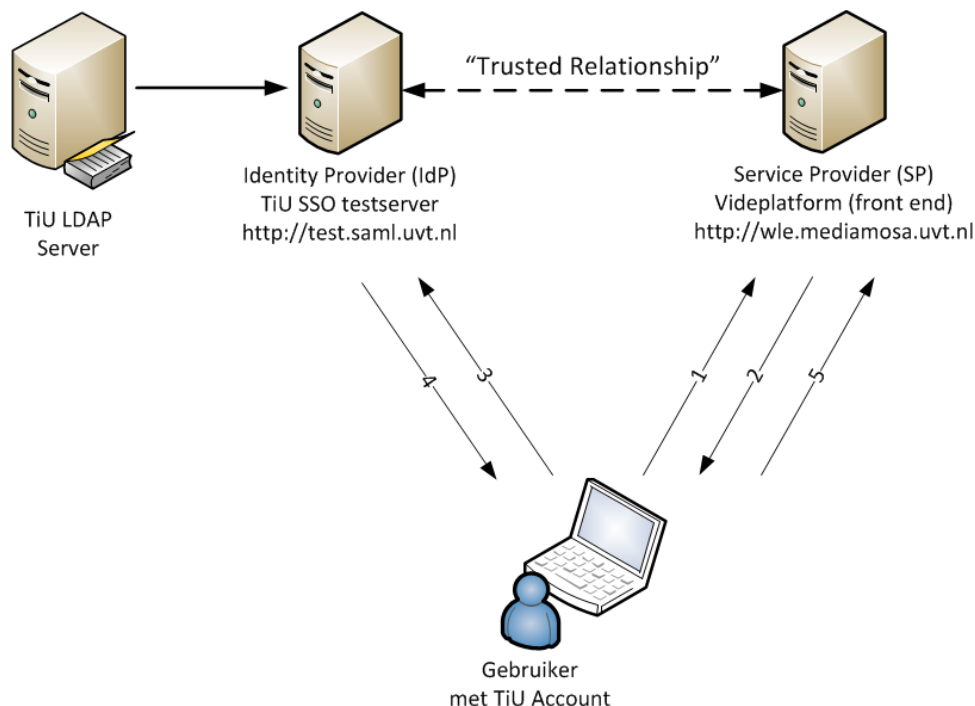
Condities

Pre: Gebruiker is bekend in de TiU LDAP en dus bij de Identity Provider.

Post: Gebruiker is succesvol aangemeld bij het videoplatform en beschikt over een SSO sessie.

Acties

1. Gebruiker wil zich aanmelden (via *federated login*) bij platform via webbrowser;
2. *Redirect* vindt plaats naar simplSAMLphp pagina in *front end*, Identity Provider wordt gekozen of is onthouden van eerder verzoek;
3. Gebruiker voert gebruikersnaam en wachtwoord in op pagina van Identity Provider;
4. Authenticatie is voltooid door Identity Provider, Identity Provider zendt antwoord terug in de vorm van een *SAML response*;
5. Response wordt doorgegeven aan Service Provider, sessie wordt toegewezen en er vindt een *redirect* plaats naar het platform.



Figuur 5 Eerste testscenario, verbinding met SSO testserver TiU.

3.4 Scenario: Gebruiker meldt zich aan bij SSO server

Opmerkingen: de verbindingen vinden plaats via *https*, er is een certificaat aangemaakt en het verkeer tussen het platform en de Identity Provider is beveiligd.

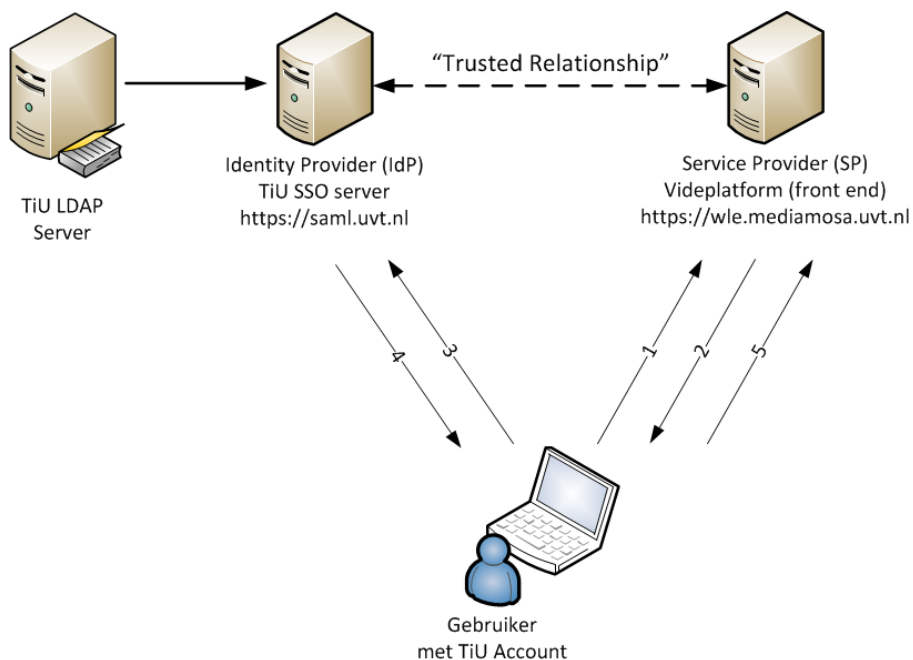
Condities

Pre: Gebruiker is bekend in de TiU LDAP en dus bij de Identity Provider.

Post: Gebruiker is succesvol aangemeld bij het videoplatform en beschikt over een SSO sessie.

Acties

1. Gebruiker wil zich aanmelden (via *federated login*) bij platform via webbrowser;
2. *Redirect* vindt plaats naar simplSAMLphp pagina in *front end*, Identity Provider wordt gekozen of is onthouden van eerder verzoek;
3. Gebruiker voert gebruikersnaam en wachtwoord in op pagina van Identity Provider;
4. Authenticatie is voltooid door Identity Provider, Identity Provider zendt antwoord terug in de vorm van een *SAML response*;
5. Response wordt doorgegeven aan Service Provider, sessie wordt toegewezen en er vindt een *redirect* plaats naar het platform.



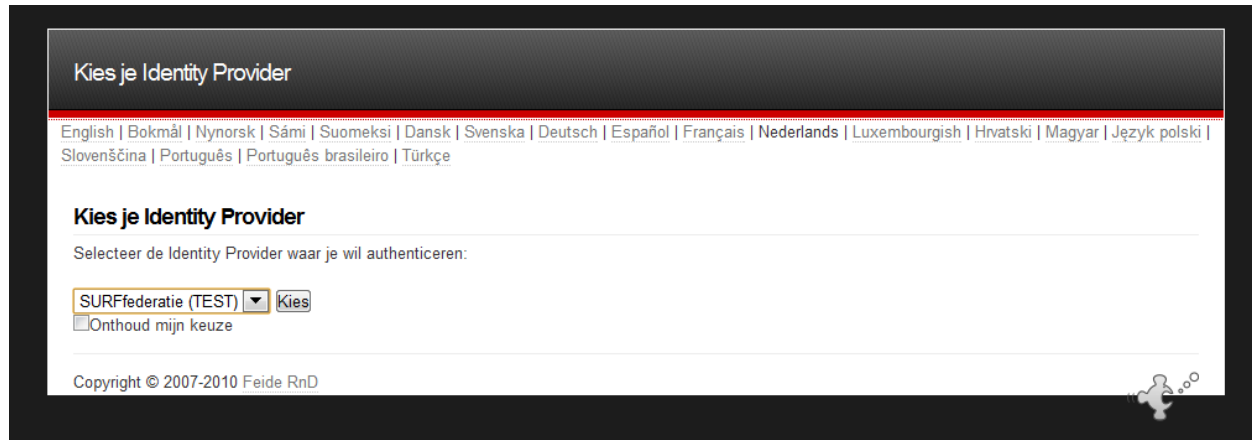
Figuur 6 Tweede testscenario, verbinding met SSO productieserver via *https*.

3.5 Screens Loginproces

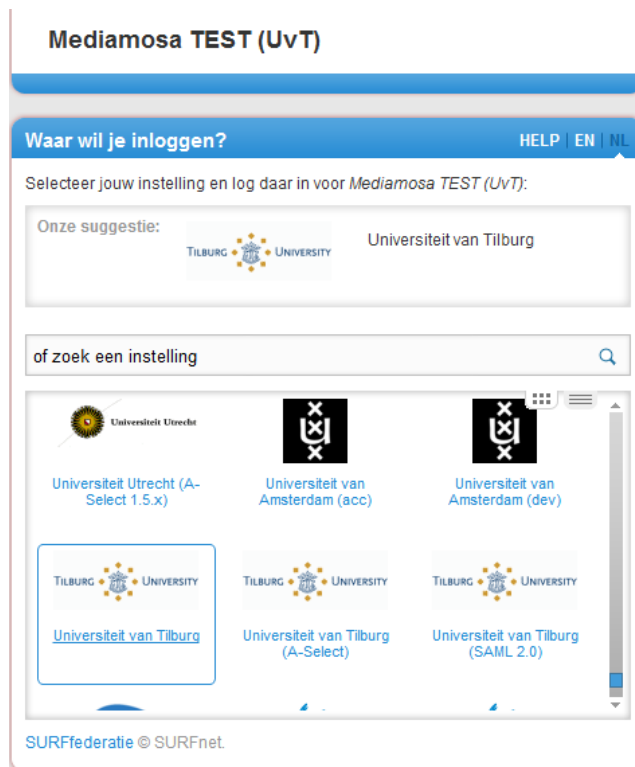
simpleSAMLphp
login

Federated login

Figuur 7 Login knop in front end



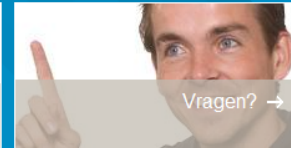
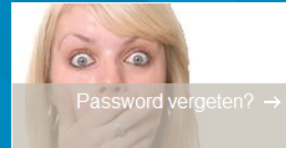
Figuur 8 simpleSAMLphp pagina, keuze voor Identity Provider (optie kan overgeslagen worden indien dit altijd dezelfde is)



Figuur 9 Overzicht van de beschikbare Identity Providers bij de SURFfederatie

Single Sign-On authenticatie (SSO)

Gemakkelijke toegang tot webapplicaties van Tilburg University



↓ Tilburg University

↓ Single Sign-On

Inloggen

→ Statusoverzicht

→ Voorkeuren

→ Attributen

→ Uitloggen

Inloggen

Vul je Tilburg University username en password in. Klik op 'Log in' of druk op 'Enter' om verder te gaan.

Username	<input type="text" value="jdamen"/>
Password	<input type="password" value="....."/>
<input checked="" type="checkbox"/> Single Sign-On	
<input type="button" value="Inloggen"/> <input type="button" value="Annuleren"/>	

Meer informatie

Na authenticatie start een SSO-sessie die je inloggegevens onthoudt. Je krijgt dan automatisch toegang tot alle webapplicaties die de SSO-authenticatie ondersteunen. Deze sessie verloopt na 10 uur. Je kan de sessie handmatig beëindigen door uit te loggen bij de SSO-authenticatieservice of deze browser te sluiten.

Als ook anderen deze computer gebruiken (bijvoorbeeld in een internetcafé) dan is het sluiten van deze browser strikt noodzakelijk om je persoonlijke informatie te beschermen.

Figuur 10 Inlogpagina voor Single Sign On van Tilburg University

simpleSAMLphp login

Logged in as:

J.M.J.Damen@uvt.nl

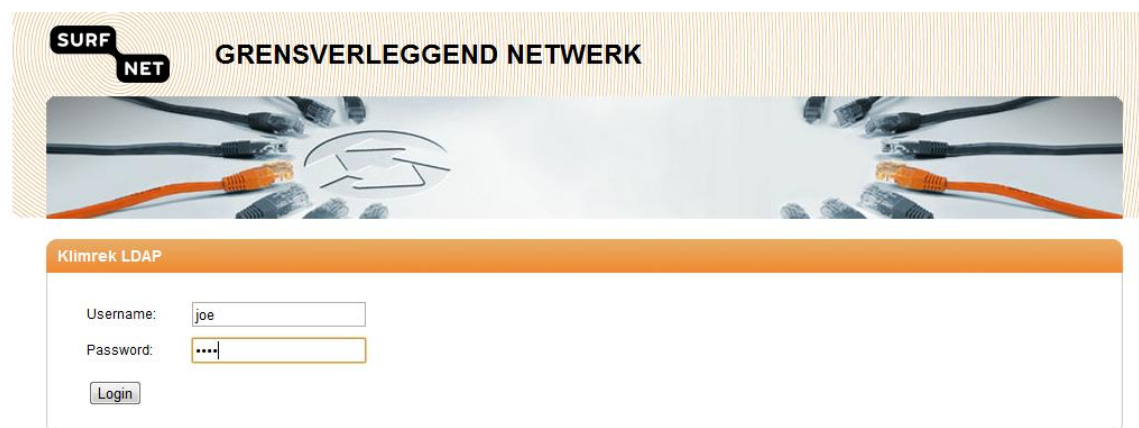
[Federated logout](#)

Figuur 11 Weergave van succesvol aangemeld in front end

3.5.1 Extra: scenario met gebruiker van andere Identity Provider



Figuur 12 Identity Provider SURFnet (test) wordt gekozen



Figuur 13 Inloggen met testaccount bij SURFnet

simpleSAMLphp
login

Logged in as:
joe@showcase.surfnet.nl
[Federated logout](#)

Figuur 14 Weergave van succesvol aangemeld in front end met testaccount SURFnet

4 Object Code tbv Streaming

- Eerste poging is Flash, indien niet aanwezig overschakelen naar HTML5 player modus.
- Flash streaming via het RTMP protocol, Wowza Media Server.
- HTTP Pseudo Streaming, applicatieserver.

```
<div id="{MEDIAFILE_ID}">Loading the player ...</div>
<script type="text/javascript"
src="http://job1.mediamosa.uvt.nl/mmplayer/jwplayer/jwplayer.js"></script>
<script type="text/javascript">
jwplayer("{MEDIAFILE_ID}").setup({
height: {HEIGHT},
width: {WIDTH},
image: "{STILL_URI}",
modes: [
{ type: "flash",
src: "http://job1.mediamosa.uvt.nl/mmplayer/jwplayer/player.swf",
config: {
file: "{TICKET_URI}",
streamer: "rtmp://137.56.72.176:1935/mediamosa",
provider: "rtmp"
}
},
{ type: "html5",
"controlbar.position": "bottom",
config: {
file: "http://app.mediamosa.uvt.nl/ticket/{TICKET_URI}",
'provider': 'http',
'x-webkit-airplay': 'allow'
}
}
]
});
</script>
```
