



Rapport Vindbaarheid Rich Media Content

Versie 1.0

Datum 10 december 2009

SURFnet/Kennisnet Innovatieprogramma

INHOUD

Contents

1. Inleiding.....	3
1.1 Aanleiding	3
1.2 Doelstelling.....	3
1.3 Beoordelingscriteria	4
2. Methoden voor vindbaar maken	5
2.1 Inleiding	5
2.2 Webform.....	5
2.3 Externe component (m.b.v. FTP-bulkupload)	7
2.4 EGA (m.b.v. REST-calls)	8
2.5 OAI-harvester.....	8
2.6 LOREnet	10
3. Weging.....	11
4. Advies	12

1. Inleiding

Veel grote instellingen die aangesloten zijn bij SURFnet, werken met systemen om colleges op te nemen en via internet aan te bieden aan studenten. Deze internetcolleges worden Rich Media genoemd, omdat ze zijn verrijkt met bijvoorbeeld meelopende Powerpoint-presentaties of ondertiteling.

In het kader van het SURFnet / Kennisnet Innovatieprogramma 2009 wil SURFnet onderzoeken of het zinvol en haalbaar is om de Rich Media-content van de instellingen beter vindbaar te maken, bijvoorbeeld via SURFmedia.

Het onderzoek moet onder andere de volgende vragen beantwoorden:

- Willen instellingen hun materiaal ook buiten hun instelling beschikbaar maken?
- Willen instellingen hun Rich Media-content ook buiten hun instelling vindbaar maken?
- Is Rich Media-content bij instellingen zodanig opgeslagen dat gebruikers buiten de instelling het ook kunnen bekijken?
- Wat is de beste manier om Rich Media-content vindbaar te maken?

Dit adviesrapport maakt deel uit van dat onderzoek en zal de laatste vraag beantwoorden.

1.1 Aanleiding

In 2007 heeft SURFnet de beschikbare systemen in kaart gebracht die zoveel mogelijk automatisch colleges kunnen opnemen, tot Rich Media-content bewerken en online zetten. Daar kwamen als belangrijkste uit:

- MediaSite
- Podcast Producer
- Presentations2Go

Vervolgens heeft SURFnet in 2008 een haalbaarheidsstudie uitgevoerd naar de vraag: is het mogelijk om op basis van een van deze systemen een dienst op te zetten die instellingen ondersteunt bij het produceren en beschikbaar stellen van Rich Media-content?

Uiteindelijk bleek zo'n dienst niet te passen binnen de missie van SURFnet. De dienst zou niet innovatief genoeg zijn. Bovendien voorzagen de instellingen die al met Rich Media werkten, zelf al in deze dienstverlening.

SURFnet ziet wel een innovatie in het vindbaar maken van de Rich Media-content. Een betere vindbaarheid zou er namelijk voor zorgen dat gebruikers makkelijker toegang hebben tot de openbare Rich Media-content van verschillende instellingen.

Uit die gedachte is het huidige onderzoek gestart.

1.2 Doelstelling

Dit adviesrapport beantwoordt de volgende hoofdvraag:

Welke oplossing voor het vindbaar maken van Rich Media-content moet SURFnet verder ontwikkelen?

Hiervoor moeten de volgende subvragen worden beantwoord:

- Welke beoordelingscriteria zijn van belang om een keuze te maken voor een oplossing?
- Welke oplossingen zijn technisch mogelijk om Rich Media-content beter vindbaar te maken?
- Welke voor- en nadelen hebben deze oplossingen?

1.3 Beoordelingscriteria

In het advies worden de in hoofdstuk 0 besproken mogelijkheden gewogen aan de hand van de volgende criteria:

- Wordt met de oplossing het doel bereikt, namelijk het goed vindbaar maken van de Rich Media-content, waarbij geen of zo weinig mogelijk metadata verloren gaan?
- Wat is de time-to-market, m.a.w. hoeveel doorlooptijd kost het om de oplossing te ontwikkelen?
- Hoeveel geld kost het om de oplossing te ontwikkelen en te onderhouden?
- In hoeverre automatiseert de oplossing het proces waarmee Rich Media-content vindbaar gemaakt wordt?
- Hoeveel effort kost het de instelling om met de oplossing metadata vindbaar te maken via SURFmedia?
- Hoeveel geld kost het om de oplossing te beheren?
- Past de oplossing bij de architectuuruitgangspunten van de SURFnet diensten?



In de beoordeling weegt het criterium of het doel bereikt is, het zwaarst.

2. Methoden voor vindbaar maken

2.1 Inleiding

SURFnet ziet op dit moment vier methoden die het kan ontwikkelen voor het vindbaar maken van Rich Media-content. Deze worden in dit hoofdstuk uitgewerkt.

1. Metadata direct invoeren in SURFmedia door de Rich Media via een webform in SURFmedia op te nemen.
2. Metadata binnenhalen middels een XML-bestand dat via de FTP bulkupload-component van VP-Core wordt afgehandeld.
3. Metadata binnenhalen via een specifieke EGA die REST-calls doet aan VP-Core.
4. Metadata binnenhalen via een OAI-harvester.

Daarnaast wordt de mogelijkheid besproken om LOREnet te gaan gebruiken. Dit is een reeds bestaande harvester van Stichting SURF.

2.2 Webform

Hoe werkt het?

De gebruiker heeft het Rich Media bestand geüpload op de Rich Media-repository van de instelling en maakt via een webform in SURFmedia een link naar het bestand. Dit webform bestaat al.

1. UPLOAD VIDEO of AUDIO bestand

* verplicht invullen

Beschikbare ruimte
Je hebt **3 MB** verbruikt en **1021 MB** beschikbaar.

Het bestand
Waar staat jouw bestand? Het bestand staat op deze pc Het bestand is geplaatst op een andere website

* URL:

Omschrijf je bestand
Is dit een audio of video bestand? audio video

* Titel:

* Beschrijving:

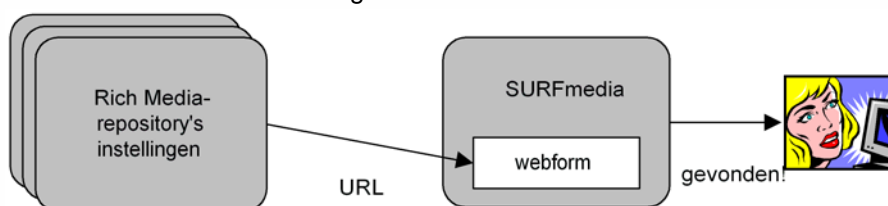
* Maker:

* Opname- of uitzenddatum (dd-mm-ijjj):

Tags:

Een tag is een trefwoord dat het mediabestand beschrijft.
Bijvoorbeeld: muziek, college, farmacie. Door het toevoegen van tags is het mediabestand gemakkelijk (terug) te vinden.
Meerdere tags kun je scheiden door een komma.

Hier vult hij de URL naar het bestand en alle benodigde of gewenste metadata in. De metadata en de URL naar het mediabestand worden in de VP-Core-database opgeslagen en het bestand is via SURFmedia vindbaar en beschikbaar voor gebruikers. Deze oplossing is met name geschikt als de instelling een beperkt aantal mediabestanden heeft die vindbaar moeten worden gemaakt.



Pro

- De oplossing is al beschikbaar, dus er is geen sprake van time-to-market en kosten voor ontwikkeling en beheer.
- Voor de gebruiker is dit een laagdrempelige oplossing; hij kan gemakkelijk bestanden vindbaar maken.

Contra

- De oplossing is arbeidsintensief want alle Rich Media-content moet handmatig vindbaar worden gemaakt (voor elk bestand moet een webform worden ingevoerd).

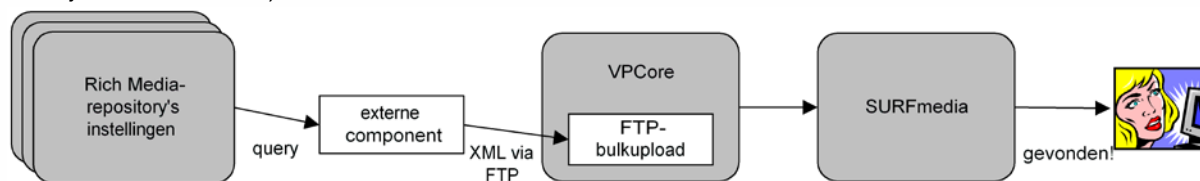
- De oplossing past slechts deels binnen de architectuuruitgangspunten van SURFnet. Voor kleine hoeveelheden bestanden is het een goede oplossing, maar voor het uploaden van grotere hoeveelheden bestanden is slimmere software nodig.

2.3 Externe component (m.b.v. FTP-bulkupload)

Hoe werkt het?

Bij deze methode wordt er een component ontwikkeld die fungeert als intermediair tussen het Rich Media-repository van de instelling en VP-Core.

De component doet periodiek een query op de Rich Media-repository van de instelling om te zien wat er gewijzigd is in metadata en URL's (content kan zijn toegevoegd, gewijzigd, verwijderd). Deze communicatie vindt plaats in het protocol van de Rich Media-repository. De wijzigingen worden in een XML-bestand gezet en dat XML-bestand wordt door de component verstuurd naar de FTP-bulkuploadcomponent van VP-Core. Deze component interpreteert de XML-code en genereert REST-calls waarmee de wijzigingen in de database van VP-Core worden uitgevoerd (assets creëren, assets van nieuwe metadata voorzien, assets verwijderen enzovoort).



Pro

- Er is weinig kennis van VP-Core nodig om deze oplossing te implementeren (het "echte" werk gebeurt in de standaardcomponenten van VP-Core).
- De instelling hoeft geen aanpassing te doen aan zijn systeem, aangezien communicatie met de Rich Media-repository plaatsvindt in het protocol van de repository.
- VP-Core hoeft niet aangepast te worden; er wordt immers een component buiten VP-Core ontwikkeld.

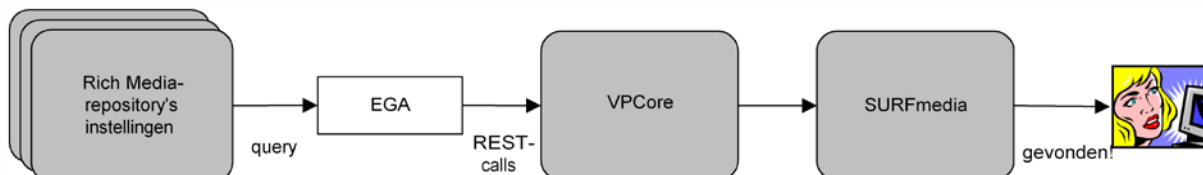
Contra

- Door het gebruik van applicatiespecifieke protocollen, bestaat het risico dat de koppeling tussen de Rich Media-repository van de instelling en de externe component niet meer werkt bij een software-update. SURFnet moet deze koppeling dan steeds herstellen door de externe component aan te passen.
- Er moet maatwerk geleverd worden om met de protocollen van de verschillende Rich Media repository's te kunnen communiceren. Denk aan Mediasite v4, Mediasite v5, Presentations2Go, Podcast Producer. Dit levert een aanzienlijke time-to-market op.
- De te ontwikkelen component moet actief beheerd worden op een separate server.

2.4 EGA (m.b.v. REST-calls)

Hoe werkt het?

Bij deze methode wordt er een nieuwe EGA voor VP-Core ontwikkeld. Deze EGA doet periodiek een query op de Rich Media-repository van de instelling om te zien wat er gewijzigd is in metadata en URL's. Vervolgens zet de EGA deze wijzigingen om in REST-calls waarmee VP-Core direct de wijzigingen kan doorvoeren in de database.



Pro

- Deze oplossing biedt extra mogelijkheden boven de oplossing behandeld in paragraaf 2.3. Nieuwe features in VP-Core worden namelijk eerst beschikbaar gemaakt via REST-calls (te gebruiken door EGA's) en alleen als het noodzakelijk is ook via de FTP-bulk upload.
- Deze oplossing is elegant omdat gebruikt wordt gemaakt van REST-calls en niet van de FTP-bulkuploadcomponent.
- De instelling hoeft geen aanpassing te doen aan zijn systeem, aangezien communicatie met de Rich Media-repository plaatsvindt in het protocol van de repository.

Contra

- De oplossing is complex qua onderhoud. Als er in VP-Core updates worden uitgevoerd, moet de EGA op forward compatibility worden gecontroleerd.
- Er moet maatwerk geleverd worden om met de protocollen van de verschillende Rich Media repository's te kunnen communiceren. Denk aan Mediasite v4, Mediasite v5, Presentations2Go, Podcast Producer.
- De te ontwikkelen EGA moet actief beheerd worden.

2.5 OAI-harvester

Hoe werkt het?

In deze oplossing wordt er een OAI-harvester gebruikt om de metadata bij de Rich Media-repository van de instelling op te halen. Er wordt dus gebruikgemaakt van een open, gestandaardiseerd protocol (OAI-PMH) dat alle instellingen moeten ondersteunen. De producenten van de gebruikte systemen moeten ervoor zorgen dat ze een OAI-repository hebben die gelezen kan worden door de OAI-harvester. SURFnet levert daarvoor geen maatwerk¹.

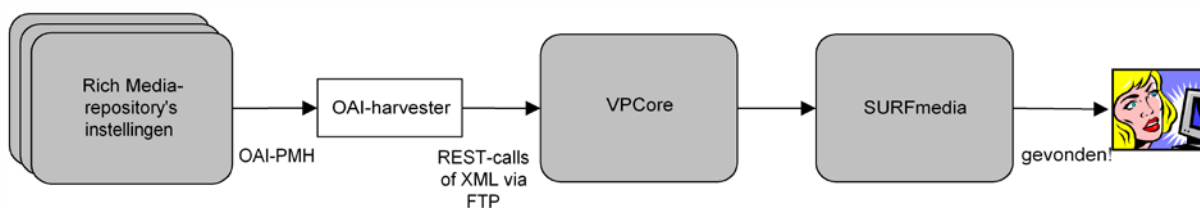
Er zijn verschillende standaarden voor het vastleggen van metadata in OAI, bijvoorbeeld Dublin Core, Qualified Dublin Core, LOM. Van groot belang is dat de OAI-repository van de instellingen

¹ Presentations2Go ondersteunt OAI in de standaardconfiguratie; Mediamission heeft voor Mediasite een add-on geschreven die OAI-functionaliteit toevoegt; Mass-IT heeft OAI geïmplementeerd op basis van Podcast Producer.

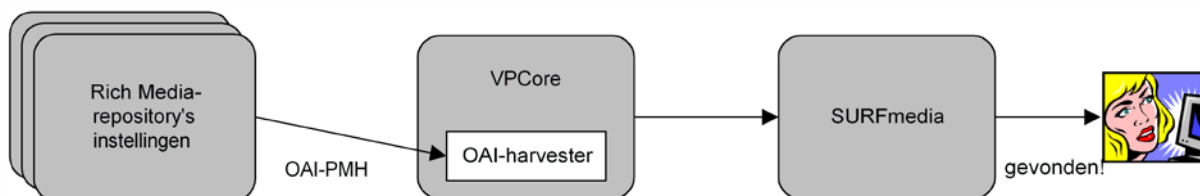
en de OAI-harvester van SURFnet met dezelfde standaard werken, anders bestaat er een grote kans op dataverarming. Bijvoorbeeld: LOM kan maar één datum vastleggen, Dublin Core kan er vier vastleggen. Als er in de OAI-repository vier datums beschikbaar zijn (opnamedatum, aanmaakdatum etc.), maar de OAI-harvester kan er maar één uitlezen, gaan er metadata verloren.

Er zijn twee mogelijkheden binnen deze oplossing:

1. Externe OAI-harvester. Deze communiceert met VP-Core via FTP of REST-calls, zoals in de oplossingen beschreven in paragraaf 2.3 en 2.4.



2. Interne OAI-harvester: een add-on op VP-Core die OAI kan lezen.



Pro

- De verantwoordelijkheid voor communicatie vanuit de systemen (ondersteuning van OAI) ligt bij de leveranciers van de systemen, niet bij SURFnet. Doorgaans zullen leveranciers er ook niet heel veel werk aan hebben, aangezien OAI door veel systemen al ondersteund wordt.
- Qua architectuur is dit de mooiste oplossing: er wordt een open standaard gebruikt, en deze standaard doet precies wat er gevraagd wordt: uitwisselen van gegevens tussen een repository en een 'zoekmachine'.
- De instelling heeft er niet veel werk aan: de het systeem moet één keer goed geconfigureerd worden voor het werken met OAI.
- Een deel van het ontwikkeltraject voor de add-on op VP-Core is al gedaan: MadCap heeft voor EDIT een OAI-harvester ontwikkeld. Bekeken moet worden in hoeverre deze OAI-harvester aangepast moet worden voor gebruik in VP-Core.

Contra

- Er moet veel afgestemd worden in de communicatie tussen de OAI-harvester en de Rich Media-repository. Wellicht is het moeilijker om instellingen hiertoe te bewegen omdat hun systemen mogelijk moeten worden aangepast.
- Het is onduidelijk hoeveel ontwikkeltijd en time-to-market deze oplossing vergt.

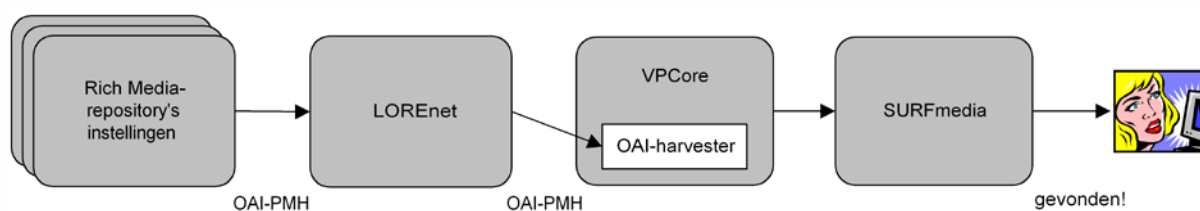
Optie 1 of optie 2

Als we kiezen voor het gebruik van een OAI-harvester, dan heeft optie 2 (interne OAI-harvester) de voorkeur, om de volgende redenen:

- De oplossing is eenvoudiger en eleganter aangezien alles binnen VP-Core gebeurt. Er is geen extra interface tussen VP-Core en een externe component.
- Een deel van het ontwikkeltraject is al gedaan voor deze oplossing (OAI-harvester voor EDIT).

2.6 LOREnet

LOREnet is een OAI-harvester van SURF. De vraag is of deze als kant-en-klaaroplossing kan worden toegepast voor SURFmedia. Helemaal kant-en-klaar is de oplossing in elk geval niet, want er VP-Core heeft een eigen OAI-harvester nodig om de OAI-repository van LOREnet te benaderen.



Pro

- In de eerder genoemde oplossingen is er een directe link tussen VP-Core en elk van de aangesloten instellingen. Bij deze oplossing hoeft VP-Core slechts rekening te houden met één partij, namelijk LOREnet. LOREnet staat vervolgens in verbinding met de Rich Media-repository's van de instellingen.
- Een deel van het ontwikkeltraject voor de OAI-harvester is al gedaan: MadCap heeft voor EDIT een OAI-harvester ontwikkeld. Bekeken moet worden in hoeverre deze OAI-harvester aangepast moet worden voor gebruik in VP-Core.

Contra

- LOREnet verwerkt alleen metadata volgens de LOM-standaard terwijl SURFmedia en VP-Core gericht zijn op (Qualified) Dublin Core. Er vinden dus veel metadataconversies plaats, waardoor het risico op verlies van metadata bestaat.
- LOREnet doorzoekt de repository's van een heleboel instellingen (met name document repository's) waardoor je een groot aantal zoekresultaten krijgt die wellicht minder relevant zijn.
- LOREnet vormt een extra schakel in de keten tussen media-aanbod van instellingen en de gebruiker. SURFnet heeft geen controle over deze schakel. Als er iets mis gaat in de communicatie tussen Rich Media-repository's en LOREnet, klopt de eindgebruiker die via SURFmedia zoekt, aan bij SURFnet. SURFnet moet dan weer contact opnemen met SURF om het probleem opgelost te krijgen.
- Het is onduidelijk hoeveel ontwikkeltijd en time-to-market deze oplossing vergt.

3. Weging

In onderstaand overzicht leest u hoe de besproken oplossingen scoren op de beoordelingscriteria. Dit overzicht vormt de basis voor het advies.

Oplossing	Beoordelingscriteria						
	Doel bereikt / geen metadata-verlies	Time to market	Ontwikkelkosten	In hoeverre automatisch	Gemak voor instelling	Beheerkosten	Architectuur
Webform	++	++	++	--	--	++	+/-
FTP bulk-upload	++	+	+/-	+	+	-	+/-
EGA	++	+	+/-	+	+	-	+
OAI-harvester	++	+/-	+/-	++	+/-	+	++
LOREnet	+/-	+/-	+/-	++	+/-	+	++



In de beoordeling weegt het criterium "Doel bereikt/geen metadata-verlies" het zwaarst.

4. Advies

In dit onderzoek hebben we verschillende mogelijke oplossingen beschreven waarmee de Rich Media-content van instellingen vindbaar gemaakt kan worden via SURFmedia.

Op basis van de scores op de verschillende beoordelingscriteria, adviseren we om de interne OAI-harvester voor VP-Core te ontwikkelen (zie paragraaf 2.5, optie 2). Deze optie sluit het beste aan bij de architectuurprincipes die SURFnet hanteert (o.a. werken met open source en open standaarden) en levert goede resultaten. De redenen om de interne OAI-harvester te verkiezen boven de externe zijn als volgt:

- De oplossing is eenvoudiger en eleganter aangezien alles binnen VP-Core gebeurt. Er is geen extra interface tussen VP-Core en een externe component.
- Een deel van het ontwikkeltraject is al gedaan voor deze oplossing (OAI-harvester voor EDIT).

Ontwikkeltraject

De basis voor de te ontwikkelen OAI-harvester is aanwezig in EDIT, en moet geschikt gemaakt worden voor gebruik in VP-Core. Het is zinvol om bij de ontwikkeling van de OAI-harvester een aantal instellingen te betrekken om de applicatie te testen.

In 2010 zijn 4 releases van VP-Core voorzien (versie 2.1 t/m 2.4). In welke release de OAI-harvester kan worden opgenomen, moet bepaald worden in overleg met betrokkenen van SURFnet en Kennisnet. Er kan voor gekozen worden om een eerste versie op te nemen in bijvoorbeeld VP-Core 2.1 en deze vervolgens uitgebreid te testen en door te ontwikkelen. Met een volgende versie van VP-Core (bijvoorbeeld 2.2) kan de tool dan grootschalig worden aangeboden.