



# **SURFmedia/MediaMosa**

## **Selectie Zoektechnologie**

**Auteurs: Martin Roest (Ibuildings) en Michiel Schok (SURFnet)**

**Versie: 1.1**

**Datum: 13 juli 2010**

**SURFnet/Kennisnet Innovatieprogramma**

## 1. Management Samenvatting

Video speelt een grote rol in de wijze waarop docenten kennis overbrengen en de manier waarop jongeren zelf kennis construeren en vastleggen. SURFnet en Kennisnet maken voor beheer en distributie van video gebruik van het VP-Core platform dat is gebaseerd op de gezamenlijk ontwikkelde MediaMosa software.

In 2009 en ook in 2010 wordt in verschillende projecten gewerkt aan het uitbouwen van het VP-Core platform, waarbij enerzijds op basis van technologieverkenningen extra functionaliteiten worden geïmplementeerd, en anderzijds in technologieverkenningen wordt verkend waar extra meerwaarde van de software kan worden gevonden. Dit project is een technologieverkenning naar de integratiemogelijkheden van bestaande zoektechnologie in MediaMosa en VP-Core.

In de huidige VP-Core omgeving zijn beperkte zoekmogelijkheden aanwezig. De belangrijkste beperkingen zijn:

- Er moet vooraf gespecificeerd worden in welk veld een zoekterm moet voorkomen (titel, beschrijving, onderwerpen);
- De resultaten bevatten geen 'ranking': niet de resultaten met meeste relevantie bovenaan, maar alfabetisch of chronologisch geordend.

SURFnet heeft Ibuildings gevraagd onderzoek te doen naar het inzetten van zoektechnologie voor het zoeken van mediabestanden binnen het VP-Core / MediaMosa platform. Na het samenstellen van een shortlist van 5 potentiële kandidaten is met de twee meest kansrijke zoekmachines een 'Proof of Concept' uitgevoerd. Zowel de POC met Apache solr als die met Sphinx zijn geslaagd. Apache solr lijkt de zoekmachine met de meeste potentie om succesvol met MediaMosa en VP-Core te integreren.

Het SURFnet/ Kennisnet Innovatieprogramma wordt financieel mogelijk gemaakt door het Ministerie van Onderwijs, Cultuur en Wetenschap.



Voor deze publicatie geldt de Creative Commons Licentie "Attribution 3.0 Unported".

Meer informatie over deze licentie is te vinden op <http://creativecommons.org/licenses/by/3.0/>

## Inhoudsopgave

1. Management Samenvatting .....	2
2. Rapportinformatie .....	5
2.1. Documenthistorie .....	5
2.2. Contactpersonen .....	5
3. Plan van aanpak .....	6
3.1. Activiteiten .....	6
3.1.1. Discovery .....	6
3.1.2. Technology Selection .....	6
3.1.3. Potential approaches .....	6
3.1.4. Initial Selection .....	6
3.1.5. Proof of Concept .....	6
3.1.6. Tests .....	6
3.1.7. Final recommendation .....	6
3.2. Input .....	6
4. Discovery .....	7
4.1. Achtergrondinformatie .....	7
4.2. Applicatiecontext .....	8
4.3. Relevante VP-Core functionaliteit .....	9
4.3.1. Delen van video's .....	9
4.3.2. EGA specifieke kenmerken .....	10
4.3.3. Collecties .....	10
4.3.4. Kanalen .....	11
4.4. Probleemstelling .....	11
4.5. Uitgangspunten .....	11
4.6. Wensen en randvoorwaarden .....	12
4.7. Doel van het rapport .....	12
5. Achtergrond zoektechnologie .....	13
5.1. Zoekindex .....	13
5.2. Indexeren .....	13
5.3. More-like-this .....	13
5.4. Spelling suggestions .....	13
5.5. Stemming / stopwords .....	14
5.6. Search operators .....	14
5.7. Facet search .....	14
5.8. Distributed search .....	14
5.9. Replication .....	15
6. Technology selection .....	16
6.1. Niet uitgewerkte zoekmachines .....	16
6.2. Sphinx ( <a href="http://www.sphinxsearch.com/">http://www.sphinxsearch.com/</a> ) .....	17
6.2.1. Pluspunten .....	17
6.2.2. Minpunten .....	18
6.2.3. Algemene indruk .....	18

6.3. MySQL Full-text search ( <a href="http://www.mysql.com/">http://www.mysql.com/</a> ) .....	18
6.3.1. Pluspunten.....	18
6.3.2. Minpunten .....	18
6.3.3. Algemene indruk.....	19
6.4. Apache Solr ( <a href="http://lucene.apache.org/solr/">http://lucene.apache.org/solr/</a> ).....	19
6.4.1. Pluspunten.....	20
6.4.2. Minpunten .....	20
6.4.3. Algemene indruk.....	20
6.5. ElasticSearch ( <a href="http://www.elasticsearch.com/">http://www.elasticsearch.com/</a> ) .....	20
6.5.1. Pluspunten.....	21
6.5.2. Minpunten .....	21
6.5.3 Algemene indruk.....	21
6.6. Xapian ( <a href="http://xapian.org/">http://xapian.org/</a> ).....	21
6.6.1. Pluspunten.....	22
6.6.2. Minpunten .....	22
6.6.3. Algemene indruk.....	22
7. Potential Approaches.....	23
7.1. Selection matrix.....	23
8. Initial Selection.....	24
9. Proof of Concept .....	25
9.1. Apache Solr .....	26
9.1.1. Solr installatie .....	26
9.1.2. Userinterface .....	26
9.1.3. Indexeerscript .....	28
9.2. Sphinx .....	28
9.2.1. Installatie/Configuratie Sphinx.....	28
9.2.2. Userinterface .....	28
9.2.3. Indexeerscript.....	29
10. Tests .....	30
10.1. Tijd volledige indexering.....	30
10.2. Bestandsgrootte van de index.....	30
10.3. Gemiddelde tijd per zoekopdracht .....	30
10.4. Query Syntax.....	31
10.5. Verschil in zoekresultaat .....	32
11. Final recommendation .....	33
11.1. Complexiteit.....	33
11.2. Indexeren .....	33
11.3. PHP ondersteuning .....	33
11.4. Software licenties .....	33
11.5. Functionele verschillen.....	33
11.6. Implementatie advies .....	34

## 2. Rapportinformatie

### 2.1. Documenthistorie

Versie	Datum	Opmerkingen
0.1	2 april 2010	Initiële versie
0.5	17 mei 2010	Proof read, spelfouten verbeterd, informatie over zoekmachines toegevoegd. Klaar voor initial selection
0.9	10 juni	Laatste informatie toegevoegd. Final recommendation geschreven
1.0	23 juni	QA Controle
1.1	13 juli	Final tweaks, SNKN template

### 2.2. Contactpersonen

Naam	Titel	Organisatie	email
Erno Tramper	Project Manager	Ibuildings	erno@ibuildings.nl
Patrick van der Velden	Software Engineer	Ibuildings	patrick@ibuildings.nl
Martin Roest	Consultant	Ibuildings	martin@ibuildings.nl
Michiel Schok	Technisch Product Manager	SURFnet	Michiel.Schok@surfnet.nl

## 3. Plan van aanpak

### 3.1. Activiteiten

#### 3.1.1. Discovery

De discoveryfase omvat het leren kennen van de relevante applicaties en functies. Daarnaast zal er een lijst van requirements worden opgesteld waaraan de zoektechnologie moet voldoen.

#### 3.1.2. Technology Selection

Deze stap bestaat uit het samenstellen van een lijst met beschikbare zoekmachines die zoveel mogelijk voldoen aan de in de vorige stap opgestelde requirements. Daarnaast worden mogelijke implementatie-oplossingen bekeken.

#### 3.1.3. Potential approaches

In deze stap worden een aantal valide oplossingen voorgesteld aan SURFnet. De oplossingen voldoen zoveel mogelijk aan de gestelde requirements en zijn te integreren in de bestaande architectuur.

#### 3.1.4. Initial Selection

Er wordt een selectie gemaakt van 2 oplossingen. Deze oplossingen worden nader uitgewerkt.

#### 3.1.5. Proof of Concept

In deze stap worden de 2 gekozen oplossingen verder uitgewerkt in een proof of concept. Het uitwerken geeft gedetailleerd inzicht in de haalbaarheid en complexiteit van de oplossing.

#### 3.1.6. Tests

De proof-of-concept-uitwerkingen worden in deze fase getest op stabiliteit en schaalbaarheid. Er zullen een aantal performance indicators worden opgesteld om inzicht te krijgen in systeemvereisten.

#### 3.1.7. Final recommendation

Op basis van de resultaten wordt samen met SURFnet gekeken welke oplossing de beste vooruitzichten biedt voor een uiteindelijke implementatie. Dit zal worden verwerkt in het eindrapport in de vorm van een advies voor implementatie.

### 3.2. Input

Als input voor het onderzoek worden gebruikt:

- Documentatie die van het systeem voorhanden is (programma van eisen, technisch ontwerp, achtergrondinformatie, etc.);
- Gesprekken gevoerd met personen die bij het project betrokken zijn;
- De broncode van het MediaMosa platform;
- Een testomgeving;
- Een databasedump met testgegevens.

## 4. Discovery

### 4.1. Achtergrondinformatie

SURFnet heeft Ibuildings gevraagd onderzoek te doen naar de mogelijkheden voor het inzetten van zoektechnologie voor het zoeken van mediabestanden binnen het VP-Core platform. Het VP-Core platform bestaat uit een MediaMosa (<http://www.mediamosa.org/>) installatie, een SAN voor opslag van mediafiles en diverse streaming- en transcoding servers. Het VP-Core platform wordt gebruikt door verschillende eindgebruikerapplicaties (EGA). Deze eindgebruikerapplicaties ontsluiten de mediabestanden naar gebruikers. SURFmedia (<http://www.surfmedia.nl/>) is één van de eindgebruikerapplicaties. Binnen SURFmedia kan er op dit moment gezocht worden naar mediabestanden. Dit is mogelijk op twee manieren. Er kan gezocht worden door één of meerdere sleutelwoorden in op te geven in een vrij zoekveld. Het versturen van deze zoekwoorden resulteert in een zoekaanvraag en er zal een lijst met zoekresultaten op het scherm gepresenteerd worden. De lijst met zoekresultaten kan gesorteerd worden op verschillende velden zoals titel, duur van het filmpje etc. Een voorbeeld is hieronder te zien:

The screenshot displays the SURFmedia website interface. At the top, there is a navigation bar with 'SURF MEDIA' logo, 'SURFmedia', and 'SURFgroepen' tabs. Below this is a secondary navigation bar with 'Mediatheek', 'Live', and 'Community & Support' buttons, along with a date 'Woensdag 7 april 2010'. A search bar is prominently featured with the text 'Doorzoek SURFmedia' and a search button. Below the search bar, there are tabs for 'Videos (854)', 'Collecties (7)', and 'Kanalen (0)'. The main content area shows search results for 'SURFnet' with 854 clips. A dropdown menu is open, showing sorting options: 'Gesorteerd op: Titel', 'Titel omgekeerd', 'Lengte', 'Waardering', 'Datum uitzending', and 'Datum plaatsing'. The search results list includes:

- Compilation film Enlighten Your Research**: SURFnet has filmed five socially relevant research trials that use lightpaths. These trials come from different scientific disciplines which make a social contribution to safety and research within the medical sector. The films show that scientists are aware of the added value of lightpaths for their research, for example by sharing scarce resources such as measurement instruments. The scientists expect that lightpaths will help speed up and... (0 stemmen, Onbekend uitzending, 00:04:13 lengte, 10/01/2000 plaatsing)
- 015-mpeg1-vhs.mpg**: this 015-mpeg1-vhs.mpg of surfnet was s/surfnet/transcoded/mpeg/015-mpeg1-vhs.mpg (0 stemmen, Onbekend uitzending, 00:11:04 lengte, 10/09/2004 plaatsing)
- 021-mpeg1-vhs.mpg**: this 021-mpeg1-vhs.mpg of surfnet was s/surfnet/transcoded/mpeg/021-mpeg1-vhs.mpg (0 stemmen, Onbekend uitzending, 00:11:04 lengte, 10/09/2004 plaatsing)
- 03[1].wmv**: this 03[1].wmv of surfnet was s/surfnet/netshow/03.wmv (0 stemmen, Onbekend uitzending, 00:01:22 lengte, 10/09/2004 plaatsing)

Illustratie 1 - eenvoudig zoeken

Er kan ook gekozen worden om geavanceerd te zoeken. Dit wordt gedaan door een complex formulier met verschillende velden in te vullen. De gebruiker kan onder andere kiezen om woorden uit te sluiten van de zoekresultaten en bijvoorbeeld op exacte datum zoeken. Vervolgens is het mogelijk om een zoekopdracht te verfijnen. Dit wordt gedaan door het zoekformulier bij de resultaten opnieuw te tonen en de gebruiker de mogelijkheid te geven het zoekformulier aan te passen en opnieuw te laten versturen. Ook is er de mogelijkheid om geavanceerde zoekopdrachten te bewaren. Zoekopdrachten kunnen worden opgeslagen onder een door de gebruiker te bepalen naam. De zoekopdrachten kunnen bij een volgend bezoek opnieuw worden geladen. Een voorbeeld van geavanceerd zoeken is hieronder te zien:

The screenshot displays the SURFmedia website's advanced search interface. At the top, there are navigation links for 'SURFmedia' and 'SURFgroepen'. The main navigation bar includes 'Mediatheek', 'Live', and 'Community & Support'. The search form is titled 'Geavanceerd zoeken' and includes a 'Standaard' button and a link to 'Meer criteria...'. The search criteria are organized into two sections. The first section has fields for 'Titel', 'Ondertitel', and 'Beschrijving', each with a dropdown menu set to 'met alle woorden' and a text input field containing 'surfnet'. The second section has fields for 'Titel', 'Ondertitel', and 'Beschrijving', each with a dropdown menu set to 'zonder de woorden' and an empty text input field. Below these are fields for 'Tags' (set to 'met alle woorden') and 'Eigenaar'. Further down, there are date range filters for 'Geplaatst tussen' and 'Uitzenddatum', both with empty input fields and 'en' connectors. There are also radio buttons for 'Video formaat' (All, Flash, Quicktime, Windows media) and 'Media type' (Audio, Video, Beide). An 'Externe URI' field is also present. At the bottom of the search form, there are navigation buttons '<<', '>>', 'Wis', and 'Zoek', along with a 'Mijn zoekopdrachten' dropdown. Below the search form, the search results are displayed, showing 'Zoekresultaten ( 161 clips )' and a 'Bewaar zoekopdracht' button. The first result is a video clip titled 'Reactie SURFnet Creative Lab - Fontys' by Eric Slaats and Peter Verbeek, with a rating of 0, a length of 00:04:15, and a date of 22/02/2010. The video is uploaded by Stephan Verveen. There is also a 'Voeg toe aan collectie' button.

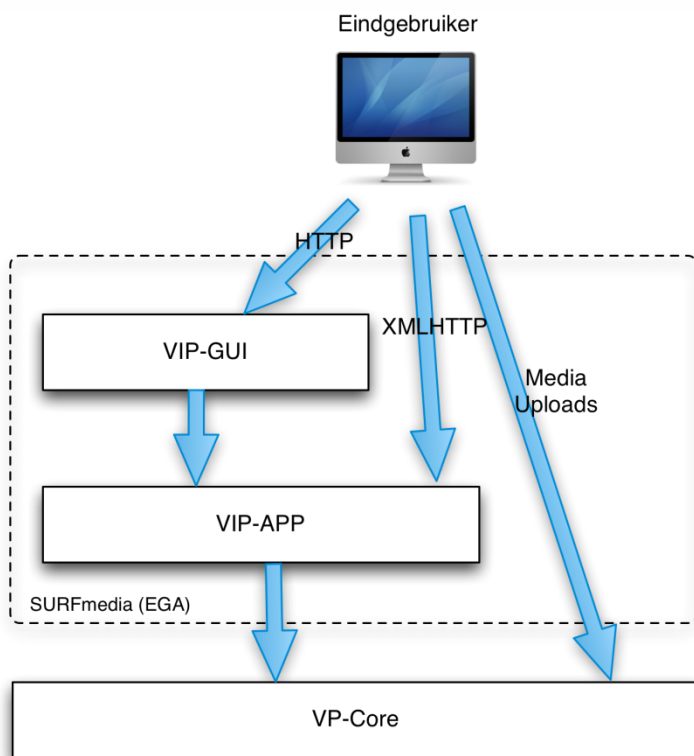
#### Illustratie 2 - geavanceerd zoeken

Beiden zoekmechanismen vertonen tekortkomingen, die hun achtergrond vinden in de opzet van het zoekmechanisme in MediaMosa. Zo kunnen zoekresultaten alleen gesorteerd worden op 'titel', 'datum', maar niet op relevantie of aantal keren dat de sleutelwoorden voorkomen in de metadata van de video's.

#### 4.2. Applicatiecontext

De meeste zoekmachines werken met aparte zoekindexen. Deze indexen moeten bijgewerkt worden op het moment dat de data verandert. Bij het inzetten van zoektechnologie is het van

belang te weten welke mogelijkheden er zijn om metadata en mediabestanden in het VP-Core platform te wijzigen, verwijderen of toe te voegen. Bij het wijzigen van data zal ook de zoekindex bijgewerkt moeten worden om zo snel mogelijk actuele resultaten te kunnen tonen. Het onderstaande diagram geeft inzicht in de communicatiemogelijkheden met VP-Core en schetst de context van de applicatie die relevant is voor dit onderzoek.



Illustratie 3 - applicatiecontext

De pijlen geven aan welke communicatierichtingen er zijn. De SURFmedia EGA kan globaal opgedeeld worden in twee componenten. De VIP-GUI en VIP-APP. De VIP-GUI verwerkt alle initiële pagina-aanvragen van de gebruiker. Bij AJAX (XMLHTTP) aanvragen vanuit SURFmedia zal de gebruiker direct contact maken met de VIP-APP. De VIP-APP geeft html-snipjets terug en vraagt VP-Core om de gegevens. In het geval van een media-upload zal de gebruiker voor de actuele file-upload direct contact maken met het VP-Core platform.

Naast de hierboven beschreven functies is er nog een extra mogelijkheid om via FTP batch uploads mediabestanden toe te voegen aan VP-Core.

### 4.3. Relevante VP-Core functionaliteit

#### 4.3.1. Delen van video's

Video's kunnen worden gedeeld met andere gebruikers, of afgeschermd zodat slechts één of meerdere personen de video kunnen bekijken. Tijdens het uploaden en bewerken van de metadata van een video kan aangegeven worden of de video publiekelijk toegankelijk is, of

afgeschermd wordt. Metadata zoals titel, omschrijving en uploaddatum is altijd voor elke gebruiker zichtbaar en verschijnt in de zoekresultaten, voor zowel openbaar als afgeschermd materiaal. Al bij het uitvoeren van zoek-opdrachten wordt door VP-Core bepaald of een gebruiker toegang heeft tot het videomateriaal. Is dit niet het geval, dan wordt dit aangegeven in de metadata. Het is aan de EGA om aan de gebruiker kenbaar te maken dat de video niet toegankelijk is. De SURFmedia applicatie doet dit door middel van het tonen van een 'slotje' in plaats van de thumbnail van de video in de zoekresultaten. Indien een gebruiker een video wil afspelen wordt de toegang tot de video gecontroleerd door het VP-Core platform. Als de gebruiker geen toegang heeft, volgt een foutmelding, anders krijgt de gebruiker de video te zien. Toegangsrechten kunnen op een aantal manieren worden aangegeven. Zo kunnen bepaalde groepen personen toegang krijgen tot een video. Dit gebeurt op basis van herkomst (IPadres/domein), loginnaam van een gebruiker of groepslidmaatschap.

#### 4.3.2. EGA specifieke kenmerken

Elke video, genaamd asset binnen VP-Core, heeft een aantal standaard kenmerken. Dit zijn kenmerken welke aangeven wie de eigenaar is, wat de titel van een video is, hoeveel mediabestanden aan de asset gekoppeld zijn, metagegevens zoals lengte en bestandsgrootte van de mediabestanden, etc. Naast deze kenmerken heeft elke EGA de mogelijkheid zelf extra kenmerken te definiëren en deze aan de video toe te voegen. In het geval van SURFmedia zijn dit kenmerken als de waardering van een video. Deze kenmerken kunnen voor de EGA van belang zijn bij het zoeken. Zo kan een video met een hogere waardering hoger in de zoekresultaten komen dan een video met een lagere waardering maar dezelfde titel en/of omschrijving. De voorkeur gaat dus uit naar de EGA te laten bepalen of het extra veld opgenomen moet worden in de zoekindex en welk gewicht het veld heeft in verband met het zoeken op relevantie.

#### 4.3.3. Collecties

Collecties worden onderverdeeld in twee groepen. Er zijn 'globale' collecties met het kenmerk 'categorie'. Deze collecties zijn voor alle EGA's beschikbaar. Tijdens het uploaden heeft de gebruiker de mogelijkheid om de video aan een of meerdere collecties te koppelen. Voorbeelden van deze collecties zijn: Documentaire, Informatief, Kunst en Cultuur, Medisch, etc.

Naast de 'globale' collecties zijn er collecties (zonder het kenmerk 'categorie') die door gebruikers worden aangemaakt. Een door een gebruiker aangemaakte collectie kan zowel privé als publiekelijk toegankelijk zijn. Een collectie kan één of meerdere video's bevatten. Dit kunnen zowel gerelateerde als ongerelateerde video's zijn, van de gebruiker zelf of van een andere gebruiker. Een voorbeeld van een door een gebruiker aangemaakte collectie kan zijn de collectie 'April 2010' waar de gebruiker al zijn in april toegevoegde video's aan koppelt.

SURFmedia stelt gebruikers in staat om in collecties te zoeken. Indien een gebruiker een zoekopdracht verstuurt zal het systeem niet alleen zoeken naar video's maar ook zoeken naar collecties. Het resultaat is te zien in Illustratie 1 waarbij het zoekresultaat verdeeld is in meerdere tabs. Onder de tab 'collecties' worden collecties getoond die voldoen aan de zoekopdracht. Door een gebruiker aangemaakte privécollecties zijn niet zichtbaar in de zoekresultaten.

#### 4.3.4. Kanalen

Illustratie 1 laat het resultaat zien van een zoekopdracht. In het scherm is er naast de tab 'collecties' ook een tab 'kanalen' te zien. Kanalen is op dit moment een functionaliteit buiten SURFmedia om en zal niet meegenomen worden in dit onderzoek.

#### 4.4. Probleemstelling

Het zoeken in VP-Core voldoet nu niet aan de wensen. Op dit moment is het niet mogelijk om zoekresultaten te sorteren op relevantie, maar alleen op velden als 'titel', 'eigenaar', 'uploaddatum' en 'speelduur'. De zoekfunctionaliteit in SURFmedia geeft hierdoor niet de resultaten die gebruikers verwachten. Het zoeken wordt al snel vergeleken met zoekmachines als Google waarbij de relevantie uiterst belangrijk is. Daarnaast zijn er verbeteringen nodig in de snelheid waarmee zoekresultaten getoond worden en kan de gebruikersinterface voor het verfijnen van zoekopdrachten verbeterd worden.

Het vervullen van deze wensen zal ertoe leiden dat docenten en studenten vaker succes hebben bij het zoeken naar multimediaal onderwijsmateriaal. Verwacht wordt dat het gebruik hierdoor toe zal nemen.

De programmeerinterface van VP-Core maakt op dit moment gebruik van CQL als querytaal (<http://www.loc.gov/standards/sru/specs/cql.html>). De CQL-query zal door VP-Core worden vertaald naar een databasequery. In de CQL-query wordt aangegeven in welke velden gezocht moet worden. Een van de wensen is om dit te vereenvoudigen zodat het niet meer nodig is om specifieke velden mee te geven. De door de gebruiker opgegeven zoekwoorden worden één op één doorgegeven aan de zoekmachine. De zoekmachine zal dan op alle beschikbare velden zoeken. Het verwerken van de door de gebruiker opgegeven zoekopdracht wordt dan door de zoekmachine zelf gedaan en zo is de directe relatie tot velden/indexen verdwenen.

#### 4.5. Uitgangspunten

In het onderzoek wordt de SURFmedia eindgebruikerapplicatie als uitgangspunt genomen. De SURFmedia applicatie is op dit moment verreweg de uitgebreidste applicatie die gebruik maakt van het VP-Core platform. Dit betekent dat bij de uitwerking van een proof of concept de SURFmedia applicatie als voorbeeld genomen wordt en dat we een vergelijkbare interface voor het zoeken in video's in de proof of concept uitwerken. Uiteraard kan het zijn dat bij het uitwerken van een proof of concept blijkt dat het niet mogelijk is om alle functionaliteiten te implementeren. Eventuele beperkingen in functionaliteiten zullen dan in de final recommendation worden meegenomen.

De nieuwe zoektechnologie zal worden ingezet naast de bestaande zoekmogelijkheid. Hierdoor hoeft er geen rekening gehouden worden met bestaande systemen die afhankelijk zijn van een bepaalde technologie.

Op het moment van dit onderzoek is de laatste stabiele versie van MediaMosa versie 1.7.3.2. Hoewel deze versie op dit moment gebruikt wordt door VP-Core zal het onderzoek zich richten op zowel de bestaande 1.7.3.2 als de nieuwe 2.x versie van MediaMosa. Binnen afzienbare tijd zal de 2.x versie als stabiele versie beschikbaar komen en zal het VP-Core platform bijgewerkt worden naar deze versie.

#### 4.6. Wensen en randvoorwaarden

Binnen SURFnet bestaat er de wens om de zoektechnologie op te nemen in het VP-Core platform. Het VP-Core platform functioneert als een middleware laag. Indien de nieuwe zoektechnologie in deze laag opgenomen kan worden kunnen alle andere eindgebruikerapplicaties hier profijt van hebben<sup>1</sup>.

Om een zoektechnologie in een later stadium succesvol toe te kunnen voegen aan MediaMosa en succesvol in productie te kunnen nemen zijn naast functionele criteria ook andere criteria van belang. Voor het opstellen van de lijst met beschikbare zoektechnologieën zijn onderstaande criteria gebruikt:

- Moderne technologie en actief in ontwikkeling;
- Bij voorkeur open-source;
- Portable voor Windows en Linux/Unix;
- Stabiele versie van de software beschikbaar;
- Goede referentieprojecten (proven technology);
- Geschikt voor het indexeren van 500.000+ documenten<sup>2</sup>;
- Goede integratie met PHP;
- Mogelijkheid tot integratie binnen VP-Core;
- Ondersteuning voor zoekoperators zoals +, -, AND, OR, etc.;
- Sorteren op relevantie;
- Commerciële ondersteuning.

#### 4.7. Doel van het rapport

Dit rapport zal inzicht geven in verschillende implementaties met verschillende zoektechnologieën. Het rapport zal op basis van de wensen en randvoorwaarden een advies uitbrengen voor een uiteindelijke implementatie van een bepaalde architectuur en zoektechnologie. Het advies zal in samenwerking met SURFnet worden opgesteld.

---

<sup>1</sup> SURFmedia is de EGA die op dit moment het meest geavanceerd gebruik maakt van de zoekinterface van VP-Core. Andere EGA's als Teleblik en Ed\*it maken geen gebruik van de zoekfunctionaliteit van VP-Core, maar kopiëren de metadata en zoeken in hun eigen database.

<sup>2</sup> Documenten moet in de context van dit rapport gelezen worden als 'metadata van een video'.

## 5. Achtergrond zoektechnologie

De binnen dit onderzoek beschreven zoekmachines hebben veel overeenkomstige kenmerken en/of functies. Om de juiste selectie te maken is het belangrijk om te weten wat deze kenmerken precies betekenen en hoe de algemene werking van een zoekmachine in elkaar steekt. In dit hoofdstuk worden een aantal belangrijke begrippen uitgelegd.

### 5.1. Zoekindex

Zoekmachines gebruiken een zoekindex om informatie snel te kunnen zoeken. De zoekindex kun je vergelijken met een inhoudsopgave van een boek. Op het moment dat een hoofdstuk aan het boek wordt toegevoegd zal de inhoudsopgave moeten worden bijgewerkt. Dit is hetzelfde bij een zoekindex. Na het toevoegen van een video in SURFmedia zal de zoekindex moeten worden bijgewerkt. De makkelijkste manier is om dit ene document aan de zoekindex toe te voegen. Dit wordt vaak beschreven als zogenaamde partial / incremental updates op de index. Soms kan dat echter niet en moet de hele zoekindex opnieuw opgebouwd worden. De snelheid waarmee de metadata van video's kunnen worden geïndexeerd is ook belangrijk, want in sommige gevallen zijn er miljoenen documenten. Als deze documenten allemaal opnieuw geïndexeerd moeten worden kan dit een behoorlijke belasting vormen. Partial updates op de zoekindex kunnen dan een oplossing zijn.

### 5.2. Indexeren

Het proces waarmee de zoekindex wordt gebouwd en/of bijgewerkt wordt indexeren genoemd. Informatie wordt verzameld door bijvoorbeeld het ophalen van een webpagina of het uitvoeren van een databasequery. Deze informatie wordt in veel gevallen eerst bewerkt. Zo kunnen alle hoofdletters omgezet worden naar kleine letters, er kunnen niet relevante woorden uit de tekst worden gefilterd, etc. Het bewerken en/of filteren van de tekst die wordt geïndexeerd heeft als doel later bij het zoeken alleen relevante informatie te tonen. Het indexeren heeft als doel nieuwe documenten aan de index toe te voegen, bestaande documenten in de index bij te werken en 'verlopen' documenten uit de index te verwijderen. Het is daarom een proces wat regelmatig en misschien wel continue uitgevoerd wordt.

### 5.3. More-like-this

De functie 'more like this' van een zoekmachine geeft de gebruiker de mogelijkheid om vergelijkbare documenten op te vragen op basis van een specifiek document uit de zoekresultaten. Dit kan gebruikt worden wanneer niet alle resultaten even relevant zijn aan de zoekopdracht. Het meest relevante resultaat kan dan gebruikt worden om meer van hetzelfde soort documenten te laten zien. De functie wordt echter niet door elke zoekmachine ondersteund en is een geavanceerde zoekfunctie. Vaak kan er wel handmatig ingesteld worden welke velden in het document gebruikt moeten worden om gelijkwaardige documenten te zoeken.

### 5.4. Spelling suggestions

Spellingsuggesties of beter gezegd zoekopdrachtsuggesties kunnen de gebruiker attenderen op een zoekopdracht met een beter resultaat. Vaak komt dit voort uit een verkeerde spelling van

het zoekwoord. De zoekmachine zal een suggestie doen van een andere 'verwante' zoekopdracht die tot meer of betere resultaten kan leiden.

### **5.5. Stemming / stopwords**

Wanneer er gezocht wordt op de zin 'De appel valt niet ver van de boom' dan zal dit misschien niet direct het gewenste resultaat opleveren. Uiteraard staat er een spelfout in het woord 'appel'. Met de functie 'stemming' heb je de mogelijkheid de zoekmachine te helpen om relevante resultaten te vinden. Je kan met de stemmingfunctie aangeven dat appel hetzelfde is als appel. De zoekmachine zal daarna ook resultaten met appel laten zien voor dezelfde query, wat de gebruiker waarschijnlijk bedoelde. In combinatie met de More-like-this functionaliteit zal het resultaat dichterbij de buurt komen van wat de gebruiker eigenlijk zou verwachten. In de zoekopdracht staan ook een aantal woorden die minder belangrijk zijn. Dit zijn zogenaamde stopwoorden. In de zoekopdracht kunnen dit bijvoorbeeld de woorden 'de', 'niet' en 'van' zijn. Deze woorden kunnen met de stopwoordenfunctie uit de zoekopdracht gefilterd worden om zo betere resultaten te kunnen tonen. De stopwoorden als 'de', 'van', etc. komen vaak voor en geven snel een minder relevant zoekresultaat.

### **5.6. Search operators**

Search operators zijn tekens die je voor of achter een zoekwoord kunt plaatsen en extra betekenis aan de zoekopdracht geven. Zo zijn er bijvoorbeeld de tekens +, - en ~. Het plusteken voor een zoekwoord, bijvoorbeeld +boom, betekent dat er alleen resultaten getoond worden waar boom in voorkomt. Dit is vooral handig als je zoekt op een zin als 'de appel valt niet ver van de boom'. De zoekopdracht bevat meerdere woorden en niet alle woorden hoeven voor te komen in het resultaat. Het zoekresultaat kan dus ook documenten bevatten waar het woord boom niet in voorkomt maar wel alle andere woorden uit het eerste deel van de zin. Door het plusteken voor het woord boom te zetten maak je dit een verplicht woord. Het tegenovergestelde kun je bereiken met het minteken. Dit kun je gebruiken om documenten met het woord boom te filteren uit het zoekresultaat.

### **5.7. Facet search**

Facet search is multi-dimensionaal zoeken, ook wel 'slicing' genoemd. Dit is een geavanceerde zoekfunctie die niet door iedere zoekmachine ondersteund wordt. Je ziet dit regelmatig bij e-commerce websites terug komen. Het stelt je in staat om je zoekopdracht te verfijnen op basis van kenmerken van een product. Bijvoorbeeld het filteren van producten binnen een bepaald prijsbereik. Of bijvoorbeeld het filteren op genre bij films. De zoekmachine zal alleen opties tonen waarin ook werkelijk resultaat te vinden is. Dit zie je vaak terug als een getal achter de verfijnoptie. Doordat alleen verfijnopties met resultaat getoond worden kan een gebruiker dus nooit op een 'geen resultaten' scherm uitkomen.

### **5.8. Distributed search**

Als de zoekindex zo groot wordt dat de server waarop de zoekindex staat niet meer groot en/of snel genoeg is kan distributed search een uitkomst bieden. Distributed search betekent dat de zoekindex over verschillende servers verdeeld staat. De zoekmachine zorgt ervoor dat bij een zoekopdracht alle delen van de zoekindex gebruikt worden en zal de resultaten samenvoegen en aan de gebruiker teruggeven. Voor gebruikers is distributed search transparant ten opzichte

van het normale zoeken en zij merken dus niets van deze functie. Hoe groot een totale zoekindex wordt ligt aan het aantal documenten dat geïndexeerd moet worden en de hoeveelheid daaruit opgenomen data in de zoekindex.

### **5.9. Replication**

Op het moment dat er zo veel gelijktijdige zoekopdrachten zijn dat één enkele server de aanvragen niet meer kan verwerken zal men extra servers moeten gaan inzetten. Het repliceren van de zoekindex is dan een oplossing om de zoekindex op meerdere servers te plaatsen en daarmee kopieën van de zoekmachine te hebben. De zoekopdrachten kunnen op dat moment verdeeld worden over de verschillende servers. Zoekmachines die replicatie ondersteunen kunnen delen van hun index synchroniseren met de aangesloten servers. Dit is vereist bij grote indexen waarbij het kopiëren van de index te veel tijd en/of belasting zou kosten. Replicatie kan ook een rol spelen bij het verbeteren van de beschikbaarheid. Stel dat een zoekmachine om welke reden ook niet meer beschikbaar is. Op dat moment kan een 'kopie'-omgeving de taken overnemen en zoekopdrachten verwerken.

## 6. Technology selection

In het onderzoek naar verschillende zoekmachines zijn veel zoekmachines bekeken en geëvalueerd. Uiteraard voldoen niet alle zoekmachines aan de gestelde randvoorwaarden. Er is een eerste selectie gemaakt op basis van een aantal belangrijke kenmerken die in paragraaf 4.6 staan. Zoekmachines die bij deze eerste selectie interessant bleken zijn verder uitgewerkt in paragrafen 6.3 en verder. Van deze zoekmachines is een korte lijst met plus- en minpunten samengesteld. Op basis van deze lijst is er een initiële selectie gemaakt van zoekmachines waarvan een 'proof of concept' is uitgewerkt.

### 6.1. Niet uitgewerkte zoekmachines

Tijdens het onderzoek en evaluatie van de verschillende zoekmachines is onderstaande lijst bijgehouden. Deze lijst met zoekmachines is onderdeel van het onderzoek maar zijn bij de eerste selectie afgevallen. De reden hiervoor is dat de zoekmachine om één of meerdere redenen niet voldeed aan de wensen en randvoorwaarden. Gezien de hoeveelheid zoekmachines is de lijst niet per zoekmachine uitgewerkt. Het merendeel van de zoekmachines zijn specifiek bedoeld om webpagina's te indexeren en zijn minder geschikt om specifieke documenten zoals video metadata te indexeren. Er zijn o.a. een aantal commerciële producten bekeken. Omdat vanuit SURFnet er een duidelijke wens is voor open-source-producten en de commerciële oplossingen vaak extra licentiekosten met zich meebrengen zijn deze zoekmachines niet verder uitgewerkt.

- mnoGoSearch (<http://www.mnogosearch.org/>)  
Bedoeld voor indexeren van webpagina's
- Fredhopper (<http://www.fredhopper.com/>)  
Commercieel product gericht op e-commerce
- Endeca Search (<http://www.endeca.com/>)  
Commerciële zoekmachine gericht op grote enterprises
- Flax (<http://www.flax.co.uk/index.shtml>)  
Gebaseerd op Xapian
- Google SA (<http://www.google.com/enterprise/search/gsa.html>)  
Google search appliance. Bedoeld voor het indexeren van webpagina's
- Microsoft Search (<http://www.microsoft.com/enterprisesearch/>)
- Webfeat (<http://www.webfeat.org/>)  
Commerciële federated zoekmachine gericht op bibliotheken
- Namazu (<http://www.namazu.org/>)

- Swish-E (<http://swish-e.org/>)  
Bedoeld voor het indexeren van webpagina's
- ht://dig (<http://www.htdig.org/>)  
Bedoeld voor het indexeren van webpagina's

## 6.2. Sphinx (<http://www.sphinxsearch.com/>)

*Licentie: GPLv2*

Sphinx is een open-source full-text zoekmachine. Sphinx is een stand-alone zoekmachine wat betekent dat sphinx als programma op een server in de achtergrond draait en de data indexeert. Sphinx is direct verbonden met een database om de data op te halen en te indexeren. Sphinx werkt met databases als MySQL en PostgreSQL. Ook is het mogelijk om data te indexeren via een 'XML pipe mechanism'. Het 'XML pipe mechanism' stelt je in staat om data in xml vorm aan te leveren en deze te laten indexeren. Sphinx is geschreven in C++.

Installatie en configuratie van Sphinx lijkt relatief eenvoudig. Er is een configuratiebestand waarin een SQL-query gedefinieerd kan worden. Deze query wordt door Sphinx als de data source gezien en Sphinx zal het resultaat indexeren. Incrementele updates op de index zijn mogelijk maar vergt wat extra configuratie. Sphinx kan bijhouden welke data voor het laatste geïndexeerd is en kan bij een volgende run alleen de delta (veranderingen) indexeren. Zo is het mogelijk om bijvoorbeeld 's nachts alle data te indexeren en overdag alleen de delta. Sphinx geeft aan dat het indexeren heel snel is en grote aantallen documenten geen probleem zijn. Hoewel Sphinx aangeeft dat databases met miljoenen records te indexeren zijn, lijkt het me toch een extra belasting op de database, zeker als de data in een complexe structuur is opgeslagen.

Sphinx heeft goede ondersteuning voor PHP. Sphinx levert een programmeerinterface die een developer kan gebruiken om vanuit PHP zoekopdrachten uit te voeren.

De programmeerinterface wordt geleverd als een PHP klasse in een apart PHP bestand. Het bestand kan gebruikt worden binnen een applicatie.

Het is mogelijk om op zoekwoorden binnen specifieke velden te zoeken. Ook is het mogelijk om over alle velden in de index te zoeken. Sphinx ondersteunt een groot aantal zoekwoord operators waarmee gebruikers krachtige zoekopdrachten kunnen uitvoeren.

### 6.2.1. Pluspunten

- Kan direct opereren op de database en in dat geval zijn er geen aparte indexerscripts nodig
- Goede integratie met PHP
- Commerciële ondersteuning is mogelijk
- Mogelijkheid tot distributed search (schalen in breedte)
- Goede referenties (Project gestart in 2001, nog steeds actief)
- Snelle indexering van documenten

### 6.2.2. Minpunten

- Geen ondersteuning voor het updaten van enkele documenten in de index
- Beperkte ondersteuning voor geavanceerde functies als facet search, 'More-like-this', etc.

### 6.2.3. Algemene indruk

Sphinx is een populaire open-source zoekmachine. De zoekmachine is ideaal in omgevingen waarbij MySQL gebruikt wordt. Sphinx kan direct op MySQL aangesloten worden waarbij Sphinx de informatie uit MySQL zal indexeren. Het is dus niet nodig om een apart indexeerprogramma te schrijven die alle informatie aan de zoekindex toevoegd. Hierdoor heb je al snel een vliegende start met Sphinx. Wel is hierdoor het indexeren van data minder flexibel als bij een zoekmachine als bijvoorbeeld Apache Solr. Sphinx levert een PHP programmeerinterface waarmee je als PHP developer snel aan de slag kan. Sphinx heeft een beperkte ondersteuning voor de geavanceerdere zoekfuncties.

## 6.3. MySQL Full-text search (<http://www.mysql.com/>)

*Licentie: GPLv2*

MySQL Full-text search is een feature van de MySQL database server. Het zoeken gebeurt door middel van een 'normale' SQL query die gebruik maakt van een full-text index en speciale full-text functies binnen de SQL query.

Het zoeken is vergelijkbaar met de manier zoals het nu in MediaMosa gebeurt. Een EGA zal nu de door een gebruiker ingegeven zoekopdracht vertalen naar een CQL-query. Deze CQL-query wordt vervolgens naar MediaMosa verstuurd. MediaMosa vertaalt de query op haar beurt naar een SQL-query en zal deze op de database uitvoeren. Het resultaat wordt terug naar de EGA gestuurd. Op dit moment wordt er geen gebruik gemaakt van full-text indexen binnen MediaMosa. Full-text indexen geven de mogelijkheid tot complexere zoekopdrachten en het (beperkt) sorteren op relevantie. Full-text indexen kunnen worden toegevoegd aan bestaande tabellen. Kolommen van het type char, varchar en text worden ondersteund door de full-text index. De index wordt op de gebruikelijke manier toegevoegd aan de tabel. Nieuwere versies van MySQL hebben native ondersteuning voor full-text indexen en is er dus geen extra installatie en/of configuratie nodig om van deze functie gebruik te maken.

Door de complexe database structuur van MediaMosa zal het gebruik van full-text indexen binnen MediaMosa aanzienlijk complex zijn.

### 6.3.1. Pluspunten

- Kan gebruikmaken van bestaande database tabellen
- Geen aparte zoekmachine software nodig

### 6.3.2. Minpunten

- Kostbaar om te schalen in de breedte
- Slechte ondersteuning voor het sorteren op relevantie
- Geen ondersteuning voor geavanceerde functies als facet search, 'More-like-this', etc.

### 6.3.3. Algemene indruk

MySQL full-text search leunt sterk op de MySQL databasetechnologie en is daardoor een stuk minder flexibel dan andere zoekmachines. Zoeken gebeurt met 'normale' SQL queries en er is geen parser die zoekopdrachten van gebruikers kan omzetten naar een SQL zoekopdracht. Hierdoor moet elk veld waarop gezocht moet worden expliciet genoemd worden. Inzetten van de full-text search is erg eenvoudig omdat MySQL al gebruikt wordt binnen MediaMosa. Er is geen extra software nodig. Helaas heeft full-text search een beperkte set van mogelijkheden en zal je al snel tegen deze beperkingen aanlopen<sup>3</sup>.

## 6.4. Apache Solr (<http://lucene.apache.org/solr/>)

*Licentie: Apache License, version 2.0*

Apache Solr is een open-source full-text zoekmachine met ondersteuning voor geavanceerde zoekfuncties. Solr is een stand-alone zoekmachine geschreven in Java en draait binnen een servlet container zoals Apache Tomcat of Jetty. Solr is gebaseerd op Apache Lucene. Solr heeft een REST-achtige API interface en is daardoor vrijwel in elke programmeertaal te gebruiken.

Installatie lijkt vrij eenvoudig door gebruik te maken van de op de Solr website aangeboden download. Deze download is een kant-en-klare oplossing en maakt gebruik van de Jetty servlet engine. De download kan draaien op een kleine memory footprint. Jetty wordt ook gebruikt op apparaten zoals mobiele telefoons.

Eerst zal er een schema moeten worden gedefinieerd waarin alle velden in de zoekindex worden beschreven. Solr heeft ook de mogelijkheid om dynamische velden te definiëren voor extra flexibiliteit. Solr beschikt over een admin interface waar informatie over geheugengebruik en indexen te zien is. Het toevoegen, bijwerken en verwijderen van een document in de zoekindex gaat via de REST-interface. Documenten moeten worden beschreven in XML. Deze XML zal via REST naar Solr gestuurd worden. Binnen Solr kan er gebruik gemaakt worden van een uniek document ID. Dit is een ID die je als beheerder zelf kun configureren en samenstellen. Het ID moet uniek zijn over alle documenten in de index. Met het unieke document ID kan een enkel document worden bijgewerkt in de Solr zoekindex. Mutaties op de Solr zoekindex kunnen vergeleken worden met databasetransacties. Solr ondersteunt commit en rollback acties en bijgewerkte documenten zijn pas zichtbaar als er een commit op de index heeft plaatsgevonden. Commits op de index zijn echter wel globaal: als er een commit plaatsvindt, is dit van invloed op alle wijzigingen op de index. Ook wijzigingen die door anderen in de wachtrij zijn gezet.

Het zoeken gaat ook via de REST-interface. Solr heeft verschillende zoekhandlers die ieder bepaalde kenmerken en/of functies bieden voor het zoeken. Zo heeft Solr een speciale zoekhandler die binnen alle velden van een document kan zoeken. De handler houdt rekening met het gewicht van de velden en het document. Zo is het niet nodig om in de zoekopdracht velden te specificeren.

---

<sup>3</sup> Bij Ed\*it (<http://www.ed-it.nu/>) dat gebaseerd is op MediaMosa v1 code, is voor MySQL Full Text Search gekozen. De genoemde beperkingen worden onderschreven door de ontwikkelaars van Ed\*it.

#### 6.4.1. Pluspunten

- Commerciële ondersteuning
- Goede integratie met PHP
- Gebaseerd op Apache Lucene (proven technology)
- Goede referentieprojecten
- Schaalbaar in de breedte d.m.v. replicatie
- Aanpasbaar door plugin architectuur
- Mogelijkheid tot geavanceerde zoekopties zoals facet search
- Mogelijkheid tot updaten van enkele documenten in de index

#### 6.4.2. Minpunten

- Aparte installatie Java servlet container nodig

#### 6.4.3. Algemene indruk

Binnen de PHP community is Apache Solr een van de meest besproken en populairste open-source zoekmachines. Dit komt vooral door het open karakter en de vele mogelijkheden. Solr gebruikt een REST-achtige interface waardoor integratie in vrijwel elke omgeving mogelijk is. Solr biedt op hun website een kant-en-klare oplossing aan. Je kunt hier eenvoudig mee aan de slag. Ook zonder Java voorkennis is Solr heel goed bruikbaar. Solr biedt veel flexibiliteit door de plugin architectuur. Je kunt met de standaard meegeleverde plugins de zoekmachine helemaal naar je hand zetten. Indien dat nog niet voldoende is kun je zelf ook plugins schrijven en deze binnen Solr gebruiken. Voor een goede implementatie vereist Solr uiteindelijk wel wat ervaring. Omdat er veel ingesteld kan worden is het waarschijnlijk dat de standaardinstellingen niet of onvoldoende aansluiten op de wensen en aanpassingen noodzakelijk zijn.

### 6.5. Elasticsearch (<http://www.elasticsearch.com/>)

*Licentie: Apache License, version 2.0*

ElasticSearch is een open-source distributed zoekmachine. ElasticSearch is een stand-alone zoekmachine geschreven in Java. ElasticSearch is ontworpen op het principe van 'cloud computing'. ElasticSearch heeft een REST-ful API en communiceert via JSON.

Installatie van ElasticSearch is eenvoudig. De op de website aangeboden download is een out-of-the-box oplossing en kan direct in gebruik genomen worden. Uiteraard zijn er Java environment settings waar rekening mee gehouden moet worden voor het starten van de zoekmachine. Direct na het starten kan de zoekmachine gebruikt worden. Er is weinig tot geen configuratie nodig. Het indexeren is vergelijkbaar met de Apache Solr machine. Het verschil is dat ElasticSearch een wat strictere REST-interface kent en dat documenten en/of zoekopdrachten in JSON-formaat verstuurd worden. ElasticSearch is gericht op cloud computing. Zo is het erg makkelijk om een cluster van zoekmachines te maken. Als er meerdere instanties worden gestart zal ElasticSearch dit detecteren. De index zal automatisch worden verdeeld en gerepliceerd over de instanties van ElasticSearch. Indexen kunnen volledig in het geheugen worden opgeslagen en zolang er altijd één van de ElasticSearch instanties blijft draaien zal de volledige zoekindex beschikbaar blijven. Door de verschillende ElasticSearch instanties met een loadbalancer te clusteren is het erg eenvoudig om in de breedte te schalen.

ElasticSearch ondersteunt een aantal belangrijke zoekmogelijkheden van Apache Lucene. Zo is het mogelijk facet search te doen en heeft ElasticSearch functies als MoreLikeThis.

#### 6.5.1. Pluspunten

- Eenvoudig in gebruik
- Goede integratie met PHP
- Gebaseerd op Apache Lucene (proven technology)
- Schaalbaar in de breedte d.m.v. distributed index
- Mogelijkheid tot het updaten van enkele documenten in de index

#### 6.5.2. Minpunten

- Relatief jong project en dus weinig referenties
- Nog geen full-featured / stabiele versie beschikbaar

#### 6.5.3 Algemene indruk

ElasticSearch is een kant-en-klaar product waarmee je snel aan de slag kunt. De eenvoud in het installeren en het gebruik maakt de zoekmachine erg aantrekkelijk. Omdat de zoekmachine gebaseerd is op de Apache Lucene technologie geeft dit meteen vertrouwen. Helaas is de zoekmachine nog vrij nieuw en is er op dit moment nog geen stabiele versie beschikbaar. Ook zal naar verwachting de ondersteuning en documentatie op dit moment minder zijn. De zoekmachine richt zich op 'cloud computing' en is daardoor uitermate geschikt in virtuele en/of geclusterde omgevingen. De zoekmachine maakt gebruik van een REST-interface voor alle communicatie. Dit zorgt ervoor dat de zoekmachine vrijwel overal kan worden geïntegreerd.

## 6.6. Xapian (<http://xapian.org/>)

*Licentie: GPL*

Xapian is een open-source programma-bibliotheek van zoekmachinefunctionaliteit. Xapian is geschreven in C++ en heeft ondersteuning voor PHP. Xapian kent geen out-of-the-box oplossing zoals ElasticSearch of Apache Solr. Xapian is een set van functies waarmee je als developer in staat bent je eigen zoekmachine te maken.

De ondersteuning in PHP betreft een PHP-extensie die geïnstalleerd moet worden op een bestaande PHP-installatie. Als de extensie is geïnstalleerd heb je de beschikking over een aantal classes en methodes om onder andere een Xapian database aan te maken en binnen de database te kunnen zoeken. Als eerste zal er een indexerprogramma geschreven moeten worden die een Xapian database kan aanmaken en documenten aan de database kan toevoegen. Er is op de Xapian site voldoende documentatie beschikbaar die je vertelt hoe je dit kunt doen. Xapian gebruikt een lockingmechanisme waardoor het niet mogelijk is meerdere keren tegelijk de database te openen voor het schrijven. Dit betekent dat er altijd maar één indexerprogramma kan zijn. In een gevulde Xapian database kun je gaan zoeken naar documenten. Hiervoor moet je een zoekprogramma maken die de database opent en een zoekopdracht uitvoert. Hierbij moet je rekening houden dat PHP-scripts meestal kort leven. Dit betekent dat je telkens de Xapian database opnieuw moet openen en je geen gebruik kan maken van de Xapian zoekopdrachten-cache. De zoekopdrachten-cache zal door Xapian worden

opgebouwd tijdens het zoeken. Xapian heeft een query parser die je in staat stelt om een zoekopdracht van een gebruiker om te zetten naar een Xapian query. De query parser ondersteunt verschillende zoekoperators waarmee gebruikers krachtige zoekopdrachten kunnen opstellen.

#### 6.6.1. *Pluspunten*

- Goede referentie projecten (proven technology)
- Integratie met PHP
- Commerciële ondersteuning
- Mogelijkheid tot het updaten van enkele documenten
- Mogelijkheid tot geavanceerde zoekopties als facet search

#### 6.6.2. *Minpunten*

- Hogere leercurve voor integrators
- Geen kant-en-klaar product

#### 6.6.3. *Algemene indruk*

Omdat Xapian een losse set van functies is waarmee een zoekmachine gemaakt kan worden is de implementatie complexer dan bij kant-en-klare zoekmachine oplossingen. Er moet meer software geschreven worden om functionaliteit te bouwen die de kant-en-klare zoekmachines standaard al hebben. Denk hierbij aan de geavanceerde zoekfuncties als facet search. Ook de leercurve is hoger dan bij de andere zoekmachines. Dit resulteert naar verwachting in een langere implementatietijd en hogere implementatiekosten. Ook moet rekening gehouden worden met de installatie van een PHP-extensie. De software is afhankelijk van deze extensie en kan dus niet zomaar op elke server functioneren.

## 7. Potential Approaches

### 7.1. Selection matrix

Om de besproken zoekmachines te kunnen vergelijken zijn de specifieke kenmerken in onderstaande matrix op een rij gezet. De belangrijkste punten uit de hoofdstukken 'Probleemstelling' en 'Wensen en randvoorwaarden' zijn opgenomen in de matrix.

	Sphinx	MySQL Full-text	Apache Solr	ElasticSearch	Xapian
<b>Stabiele versie beschikbaar</b>	✓	✓	✓	x	✓
<b>Commerciële ondersteuning</b>	✓	✓	✓	x	✓
<b>Simple search*</b>	✓	x	✓	✓	✓
<b>Proven technology</b>	✓	✓	✓	x	✓
<b>Sorteren op relevantie</b>	✓	x	✓	✓	✓
<b>Open-source</b>	✓	✓	✓	✓	✓
<b>Geschikt tot 500.000 documenten</b>	✓	✓	✓	✓	✓
<b>PHP ondersteuning</b>	✓	✓	✓	✓	✓

\* Zoeken in alle velden in de index zonder expliciet velden te specificeren in de zoekopdracht

De zoekmachines scoren over het algemeen goed op de punten in de matrix. De reden hiervoor is dat na de eerste selectie alleen de zoekmachines zijn uitgediept die goed aansluiten op de wensen en randvoorwaarden van de gevraagde zoektechnologie.

Alle in de matrix genoemde zoekmachines kunnen in het VP-Core platform worden geïntegreerd.

De matrix geeft geen inzicht in de complexiteit en kosten van de implementatie. Dit is uiteraard wel een belangrijk punt voor de uiteindelijke keuze. De complexiteit en kosten worden bepaald door een aantal factoren. Het kan zijn dat er voor een bepaalde zoekmachine specifieke en/of extra hardware nodig is. Of dat de PHP software die geschreven moet worden veel tijd kost. Deze punten worden duidelijk bij het uitwerken van de proof of concepts.

De zoekinterface zoals op dit moment beschikbaar in SURFmedia kan worden ondersteund door alle genoemde zoekmachines in de matrix.

## 8. Initial Selection

Op basis van de selectiematrix uit het vorige hoofdstuk hebben we de 2 volgende zoekmachines geselecteerd voor verdere uitwerking in een proof of concept:

- Sphinx (<http://www.sphinxsearch.com/>)
- Apache Solr (<http://lucene.apache.org/solr/>)

De zoekmachines sluiten het beste aan op punten besproken in de hoofdstukken 'Probleemstelling' en 'Wensen en randvoorwaarden'. Hoewel Xapian qua functionaliteit vergelijkbaar scoort wordt hier geen proof of concept mee gedaan. Omdat het geen kant en klaar product is zou veel tijd gestopt moeten worden in maatwerk.

In de selectiematrix is er geen rekening houden met de complexiteit van de integratie binnen SURFmedia, MediaMosa software of het VP-Core platform. Omdat de zoekmachines verschillend zijn is het waarschijnlijk dat elke uitwerking een aparte implementatiewijze vereist.

De zoekmachines zijn geschikt om te integreren als extern/apart component binnen bestaande software. Voor de integratie in het VP-Core platform is het waarschijnlijk dat VP-Core op een aantal punten zal moeten worden aangepast. De mate waarin aanpassingen nodig zijn zal blijken in de uitwerking van de proof of concept.

## 9. Proof of Concept

Voor de uitwerking van een proof of concept moet er eerst bepaald worden welke functies de proof of concept moet ondersteunen. Het doel van een proof of concept is om te bepalen of de zoektechnologie met succes geïmplementeerd kan worden.

Gewenste functionaliteit:

- Indexeren van minimaal 70.000 video's
  - Documenten ophalen via de VP-Core API
- Indexeren van collecties
  - Documenten ophalen via de VP-Core API
- Autorisatie (delen van video's) verwerken in de zoekresultaten
  - Door de rechten in de zoekindex op te slaan moet er gefilterd kunnen worden op toegankelijkheid in het zoekresultaat.
- EGA specifieke velden (aanmaken en zoeken)
  - Functionaliteit hoeft niet te worden uitgewerkt maar wel een plan/ontwerp
- Integratie binnen VP-Core
  - Triggers voor bijwerken, toevoegen en verwijderen van documenten. Niet alle triggers zullen geïmplementeerd worden.
  - Documenten ophalen via de VP-Core API
- UI voor eenvoudig en geavanceerd zoeken
  - Sorteren op relevantie, titel, lengte, datum etc.
  - 'Alleen voor mij toegankelijk' filter
  - Zoeken in specifieke velden (titel, beschrijving)
  - Zoeken binnen collecties

De verwachting is dat bij de uitwerking van de proof of concept niet alle functionaliteit gemaakt zal kunnen worden. Dit in verband met de beperkte tijd. De mate waarin functionaliteit gemaakt kan worden zal mede bepalen welke zoekmachines het meest geschikt is.

## 9.1. Apache Solr

De uitwerking van Apache Solr bestaat uit een aantal componenten te weten:

- Installatie/Configuratie Solr
- Zoek userinterface
- Indexeerscript

Functionaliteiten, beschreven in hoofdstuk 9, die zijn uitgewerkt in de proof of concept zijn hieronder beschreven.

### 9.1.1. Solr installatie

De Solr installatie bestaat uit de Solr java software en de volgende configuratiebestanden:

- schema.xml
- solr.xml

In schema.xml staat de definitie beschreven van alle velden in de Solr zoekindex. Er wordt beschreven welke velden er zijn en welk type ze hebben. Zo zijn er velden die alleen nummers kunnen bevatten, velden die tekst kunnen bevatten en velden die tekst bevatten en geoptimaliseerd kunnen worden voor het zoeken. Dit zijn de velden waarbij stopwoorden etc. eerst uit de tekst gefilterd worden.

### 9.1.2. Userinterface

De zoekinterface is vergelijkbaar met het huidige zoekformulier op SURFmedia. Het formulier maakt direct gebruik van de Solr REST interface. Dit betekent dat als er een zoekopdracht wordt uitgevoerd het zoekformulier de resultaten direct aan Apache Solr vraagt. De communicatie naar Solr loopt via de Solr REST interface. Voor de proof of concept maken we gebruik van een speciale PHP extensie voor Solr. Deze extensie is geen vereiste maar maakt de implementatie makkelijker. Meer over de extensie is te vinden op: <http://nl.php.net/solr> . Er zijn ook PHP bibliotheken beschikbaar die vergelijkbare functionaliteit bieden. Verder hebben we gebruik gemaakt van Zend Framework om het zoekformulier en andere elementen zoals paginanavigatie te maken. De zoekinterface ziet er als volgt uit:

Uw zoekopdracht:

zoek

 Alleen toegankelijk voor 

Aantal gevonden resultaten: 5158 in 0.03 sec

First | &lt; Previous | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Next &gt; Last &gt;&gt;

[still]	<p><a href="#">JA EN AMEN - DWARS DOOR DE BIJBEL (exp: id:24091)</a>            BG_37683-out.wmv            INTERVIEWS met: - Marion Kunstenaar, lid Joods-Liberale Gemeente in Amsterdam, over de persoon Rebecca; het verhaal in relatie tot de geschiedenis van de opstand van Israël; en de plaats van de vrouw in de bijbel; - Annet Schilder, publiciste Trouw en Opzij, over de verbondenheid van het Rebecca-verhaal met haar jeugd, onvruchtbaarheid en eerstgeborenen; haar boek "Van Paradijs naar koninkrijk" over de blijvende strijd tussen het materiële en immateriële; en de identificatie met Rebecca als moeder. SHOTS: 08:38 filmfragment "Genesis Project", Israëlische verfilming van het boek Genesis, 00:21:36 - 00:25:29 DE HELFT VAN HET SUCCES Portret van Marijke Selk in Deventer, moeder van Christian (19) en Alexander (16). Beiden turnen en bij Christian is er al duidelijk sprake van topsport. INTERVIEW met Marijke Selk. SHOTS: - div. shots, cu's en slowmotion turnen.            Teleblik</p>	Waardering: [waardering] Uitzending: 1991-03-10T23:00:00Z Lengte: 1715s Plaatsing: [plaatsingsdatum] Tags: [tags]	Geplaatst door: nibg-admin [logo]
[still]	<p><a href="#">BLIJF NATUUR (exp: id:24156)</a>            19:55:10 HET WADDENELAND TEXEL (het natuurgebied)            Aandacht voor de natuurgebieden op Texel en de vele vogelsoorten die er verblijven zoals meeuwen, rotganzen, lepelaars en visdieven. Shots: wadden; meeuwen; het Marsdiep; weilanden; duinen; natuurgebied Het oude Land van Texel; schaapskooien; bermen,tuinwallen; weilanden met schapen; natuurgebied de Zandkuil; wespen en hommels; de Texelse zandbij; div. van rotganzen; de Waddendijk; polder met molen; kwelder de Schorren; krekens; zee-alsen in lamsoor; div. van lepelaars; de bontbekplevier; div. van vogels; nietkragen; molen; dwergstern; Noorse stern; het duinmeer de Mui; de Sluftervallei; de Eilandse Duinen; velduil met jong; grote ratelaar en rode oegentroot; kemphanen; kolonie visdieven; Kluten. * 20:09:44 OUDE BUIZE HEIDE (cultuurhistorie) Wandeling over het landgoed Oude Buize Heide bij Zundert met Hans Hoffland (Vereniging Natuurmonumenten). Hij vertelt over de boerderijen en huizen op dit landgoed en met name over het huis waar de schrijfster Henriëtte Roland Holst heeft gewoond. Met div. (zww) foto's. Shots: div. van ext. en int. woningen op het landgoed Oude Buize Heide; atelier van Rick Roland Holst; uitzoorn ramenpartij; laan met holle beuken. Archiefmateriaal: (zww) Henriëtte Roland Holst declameert. * 20:15:51 GAUDEWILJN ODE (de specialist) Ode vertelt over zijn interesse voor bloemen en planten, zijn biologiestudie en zijn werk bij stichting Floron. Deze stichting deelt landschappen in advn kilometerhokken en beschrijft per vierkante kilometer nauwkeurig de aanwezige flora. Shots: div. van polderlandschappen; div. van bloemen; moerasplanten; distels; koekoeksbloem; woekerplanten; schietwilgen aan de Weal; konickpaarden(?).            BG_37715-out.wmv            Teleblik</p>	Waardering: [waardering] Uitzending: 2000-04-30T22:00:00Z Lengte: 1775s Plaatsing: [plaatsingsdatum] Tags: [tags]	Geplaatst door: nibg-admin [logo]

Initieel wordt er een 'lege' zoekopdracht uitgevoerd die de complete lijst met assets zal opleveren. Solr kan ook gebruikt worden om alleen de best beoordeelde assets te laten zien zoals dat gebeurt op SURFmedia. Het weergeven van thumbnails voor de video's is op dit moment nog niet geïmplementeerd. De interface ondersteunt het zoeken binnen alle velden of op specifieke velden. Door een zoekterm in te geven als 'werken' zal binnen alle velden gezocht worden. Het is mogelijk om binnen een specifiek veld te zoeken door een zoekopdracht als: 'title:werken' te gebruiken.

De zoekinterface houdt rekening met de rechten van video's. Zo kan het zoekresultaat verfijnd worden met video's die de gebruiker ook werkelijk kan bekijken. Dit wordt nu gedaan door een gebruikersnaam mee te geven bij de zoekopdracht. In de werkelijke situatie zal dit uiteraard de ingelogde gebruiker zijn en kun je dit niet zelf invullen. De rechten van een asset zijn opgenomen in de zoekindex. Door zoekfilters op gebruiker of domein te zetten kan er verfijnd worden op video's waar de gebruiker ook werkelijk rechten voor heeft.

Standaard wordt er gesorteerd op relevantie. Hierbij is het titelveld het belangrijkste. Een hit in het titelveld zal daarom de hoogste relevantiescore opleveren.

### 9.1.3. Indexeerscript

Het indexeerscript heeft als taak periodiek alle assets uit de database op te halen en de zoekindex volledig bij te werken. Dit betekent dat assets die verwijderd zijn uit de Solr index gehaald worden. Assets die zijn toegevoegd of bewerkt worden bijgewerkt in de zoekindex. Het script kan bijvoorbeeld dagelijks draaien. Het script wordt ook gebruikt om de zoekindex initieel te vullen. Op dit moment maakt het script gebruik van de Mediamosa REST interface. Het initieel vullen van de zoekindex kan enkele uren in beslag nemen. Dit zal idealiter versneld kunnen worden door op een andere / snellere manier de assets uit MediaMosa op te halen. Tijdens het indexeren is het mogelijk een bepaald gewicht toe te kennen aan assets. Zo is het mogelijk om assets met een hoge beoordeling een hoger gewicht te geven. Dit zal resulteren in een hogere relevantie score tijdens het zoeken. Gebruikers zien dus assets met een hoge beoordeling eerder in de zoekresultaten verschijnen.

## 9.2. Sphinx

Sphinx heeft de mogelijkheid om, vergelijkbaar met Apache Solr, assets te indexeren via XML. Om gebruik te maken van de unieke en specifieke kenmerken van Sphinx is er gekozen om met de database koppeling vanuit sphinx assets te indexeren. De uitwerking bestaat uit de volgende componenten/werkzaamheden:

- Installatie/Configuratie Sphinx
- Indexeerscript
- Uitwerking zoekinterface

### 9.2.1. Installatie/Configuratie Sphinx

Een installatie van Sphinx bestaat uit software en een configuratiebestand. Het configuratiebestand beschrijft de zogenoemde datasource. De datasource is de query die door sphinx tijdens het indexeren wordt uitgevoerd. Het resultaat van de query zal door Sphinx geïndexeerd worden.

### 9.2.2. Userinterface

Ook voor deze zoekinterface hebben we Zend Framework gebruikt om het formulier en elementen zoals pagina navigatie te maken. Het formulier is dan ook gebaseerd op die van de andere POC. De communicatie naar Sphinx gebeurt via de door Sphinx meegeleverde PHP API bibliotheek. Dit is een PHP bestand met daarin functies/klassen om eenvoudig met Sphinx te communiceren. De zoekinterface ziet er als volgt uit:

Uw zoekopdracht:

test

Alleen toegankelijk voor

---

Aantal gevonden resultaten: 191 in 0.002 sec

First | < Previous | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Next > Last >>

[still]	<a href="#">test mpa</a> test mpa	Waardering: [waardering] Utzending: 2004-09-10 12:59:44 Lengte: 3s Plaatsing: [plaatsingdatum] Tags: [tags]	Geplaatst door: nibg-admin [logo]
[still]	<a href="#">test 223</a> test ma course december	Waardering: [waardering] Utzending: 2004-09-10 12:59:29 Lengte: 3s Plaatsing: [plaatsingdatum] Tags: [tags]	Geplaatst door: nibg-admin [logo]
[still]	<a href="#">Davideon test</a> test	Waardering: [waardering] Utzending: 2004-09-10 12:59:29 Lengte: 3s Plaatsing: [plaatsingdatum] Tags: [tags]	Geplaatst door: nibg-admin [logo]
[still]	<a href="#">test mark</a> test mark	Waardering: [waardering] Utzending: 2004-09-10 12:59:29 Lengte: 3s Plaatsing: [plaatsingdatum] Tags: [tags]	Geplaatst door: nibg-admin [logo]
[still]	<a href="#">Test upload bulk nr.1</a> Dit is een test met de bulkupload	Waardering: [waardering] Utzending: 2009-10-20 07:54:10 Lengte: 3s Plaatsing: [plaatsingdatum] Tags: [tags]	Geplaatst door: SURFgroepen.frans [logo]

Ook bij deze uitwerking is het weergeven van thumbs niet geïmplementeerd. Er is de mogelijkheid om op specifieke velden te zoeken. Dit kan door de veldnaam te specificeren in de zoekopdracht. De syntax is als volgt: '@title jeugdjournaal'. Standaard zal Sphinx binnen alle tekstvelden zoeken.

### 9.2.3. Indexeerscript

Omdat Sphinx direct op de database is aangesloten vereist het indexeren een speciale aanpak. Het denormalisatiescript is een SQL script dat een aparte MySQL-tabel aanmaakt. In de tabel wordt alle informatie van een asset opgeslagen. Per asset is er één rij in de tabel. De assetinformatie uit de MediaMosa database wordt in meerdere tabellen opgeslagen. Het script zal de relevante informatie uit meerdere tabellen ophalen en in een enkele tabel opslaan. Alle informatie wordt op deze manier gedenormaliseerd. Voordat de Sphinx zoekindex geupdate kan worden zal uiteraard het denormalisatiescript moeten draaien. Op dit moment wordt er nog geen rekening gehouden met assets die zijn verwijderd.

Sphinx heeft voor elke asset in de zoekindex een uniek ID nodig. Dit ID moet van het type integer zijn. Omdat in MediaMosa unieke ID's van assets van het type varchar zijn is er een extra tabel die voor elke asset een uniek ID van het type integer opslaat. Deze tabel wordt bijgewerkt met een SQL opdracht. Deze SQL opdracht zal uiteraard voor elke run van het denormalisatiescript moeten draaien.

Sphinx ondersteunt alleen multivalued velden van het type integer. Multivalued velden zijn velden waar meerdere waarden voor bestaan. Vergelijkbaar met 1 op n relaties in een database. Multivalued velden worden idealiter gebruikt bij bijvoorbeeld de autorisatie van assets op applicatie niveau. Bijvoorbeeld in het geval van realm autorisatie waarbij meerdere email adressen zijn opgegeven. Omdat Sphinx voor multivalued velden alleen integers ondersteund hebben we ervoor gekozen de waarden kommagescheiden als string in de index op te slaan. Het filteren op een emailadres of andere waarden doen we door middel van een reguliere expressie op de waarde van het veld.

## 10. Tests

Elke zoektechnologie heeft zijn specifieke kenmerken. Voor de tests is een aantal performance indicatoren gedefinieerd die inzicht geven of er afwijkingen zijn in de systeemeisen en schaalbaarheid. De resultaten van elke proof of concept worden met elkaar vergeleken.

### 10.1. Tijd volledige indexering

Met dit punt wordt de totale tijd bedoeld die nodig is voor het opbouwen van de volledige index. De test is uitgevoerd met zo'n 34.000 assets in de database.

Proof of concept	Totale tijd
Apache Solr	+/- 5 uur
Sphinx	+/- 3 min

Het verschil tussen de twee is aanzienlijk. De reden hiervoor is dat Apache Solr gebruik maakt van de MediaMosa REST interface en voor elke asset meerdere REST calls moet doen, waar Sphinx de gegevens rechtstreeks uit de database verzamelt maar ook in de zelfde database opslaat.

De belasting van VP-Core tijdens het indexeren bij Apache Solr is aanzienlijk. Ook dit komt voort uit het gebruik van de REST interface.

Om vergelijkbare resultaten te hebben is er voor Apache Solr een script gemaakt die alle assets uit de voor Sphinx gemaakte aparte MySQL tabel haalt. Het indexeren duurt dan ook enkele minuten.

### 10.2. Bestands grootte van de index

De bestands grootte van de index is belangrijk omdat dit mogelijk afwijkende eisen stelt aan de server.

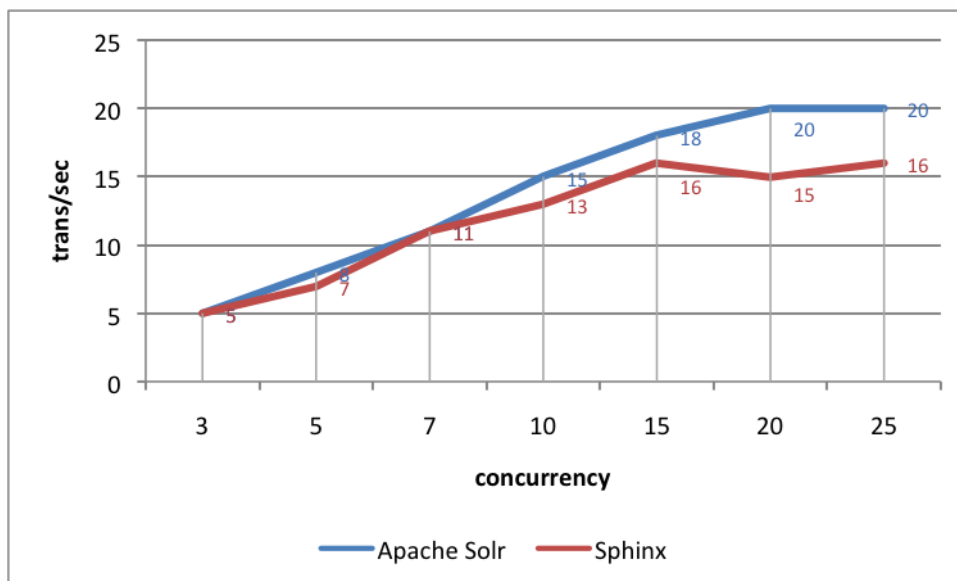
Proof of concept	Totale grootte
Apache Solr	35 MB
Sphinx	15 + 20 MB*

\* schijfruimte plus de ruimte die voor de aparte MySQL tabel nodig is.

### 10.3. Gemiddelde tijd per zoekopdracht

Onder de gemiddelde tijd per zoekopdracht wordt verstaan de totale tijd die nodig is om een zoekopdracht uit te voeren en de resultaatpagina op het scherm te tonen. De tijd die nodig is voor het uitvoeren van de zoekopdracht en het opbouwen van het resultaat komt overeen met de beleving van de gebruiker. De gegevens worden gemeten met de tool siege (<http://www.joedog.org/index/siege-home>). Hiervoor is een lijst met URL's (zoekopdrachten) samengesteld. De zoekopdrachten worden gedurende 1 minuut uitgevoerd. Om de belasting

zoveel mogelijk naar de werkelijkheid te simuleren worden de zoekopdrachten gelijktijdig uitgevoerd. Dit wordt de gelijktijdigheidsfactor genoemd die in relatie staat tot het aantal gebruikers. De gelijktijdigheidsfactor wordt bij elke test opgevoerd. Bij een gelijktijdigheidsfactor van 10 is in werkelijkheid het aantal gebruikers vele malen hoger. Om het gebruik zoveel mogelijk naar de werkelijkheid te simuleren is er 1 seconde vertraging na het opvragen van elke zoekopdracht. Na een seconde vertraging zal de volgende zoekopdracht worden uitgevoerd. De vertraging simuleert het lezen van de resultaten door de gebruiker. De test gaat er vanuit dat eventuele zoekmachinecaches zijn opgewarmd. De test is meerdere keren gedraaid om het gemiddelde te bepalen.



De grafiek laat zien dat Apache Solr iets sneller is. Het verschil wordt groter bij een hogere gelijktijdigheidsfactor.

#### 10.4. Query Syntax

Apache Solr heeft een zgn. "Google-achtige" manier van zoeken. Men kan 1 of meer zoektermen opgeven waarbij men kan aangeven of een term verplicht moet voorkomen in het resultaat (+) of juist niet mag voorkomen in het resultaat (-). Boolean operatoren kunnen daarbij ook gebruikt worden (AND/OR). Als men in specifieke velden wil zoeken, dan kan dat door het veld, gevolgd door een dubbele punt, voor de zoekterm te plakken. Daarnaast kan men nog zoeken op zinnen of andere combinaties van woorden door er dubbele quotes omheen te plaatsen.

Een voorbeeld: "title:nieuws content:nieuws vandaag gisteren -eergisteren".

Sphinx kijkt op een aantal punten af van deze syntax. Bij Solr wordt tussen de zoektermen de operator 'OR' aangenomen. Het voorbeeld leest als: "Ik zoek op nieuws in de titel of in de content waar vandaag of gisteren in voorkomt maar niet eergisteren". In Sphinx wordt echter 'AND' aangenomen en leest dit dan als "Ik zoek op nieuws in titel en in content waarbij vandaag en gisteren voor moeten komen maar niet eergisteren.". Dit leidt dus tot hele andere resultaten.

Een ander verschil is dat Sphinx in plaats van AND en OR gebruik maakt van & en |. Dit vormt alleen voor de gebruiker een verschil.

Tenslotte is de syntax voor het zoeken in een specifiek veld in Sphinx anders dan in Solr. Het eerdergenoemde voorbeeld zou er in Sphinx als volgt uitzien:

“@title nieuws | @content nieuws | vandaag | gisteren -eergisteren”.

Naast deze zoekopties zijn er zowel in Solr als in Sphinx nog meer geavanceerde zoekopties, maar omdat deze zelden door een gemiddelde gebruiker toegepast zullen worden hebben deze nauwelijks invloed op de uiteindelijke aanbeveling.

### **10.5. Verschil in zoekresultaat**

In onze Proof of Concept fase zoeken we vrijwel uitsluitend op de velden dublin core titel en dublin core omschrijving. Dit geeft ons voldoende variatie om de relevantie van de zoekresultaten te beoordelen.

Zowel Solr en Sphinx geven 'relevantiescores' aan de resultaten en standaard doen ze dat aan de hand van hoe vaak termen voorkomen en hoe dicht deze bij elkaar in de buurt staan. Bij gebruik van de standaard opties zijn de verschillen in relevantie score niet heel groot.

Apache Solr heeft echter veel meer mogelijkheden dan Sphinx om de scores te beïnvloeden, zoals het zwaarder laten wegen van bepaalde zoektermen of zoekvelden, het uitbreiden van de zoekquery (voor de berekening), de 'maximale afstand' opgeven m.b.t. hoe dicht termen bij elkaar te vinden zijn en zelfs het gebruik van eigen (Solr) scorefuncties (tijdens de berekening).

## 11. Final recommendation

Beide zoekmachines presteren voor het merendeel van de wensen en randvoorwaarden vrijwel gelijk. Beide zoekmachines geven een stabiele en volwassen indruk. Ook hebben beide zoekmachines goede referentieprojecten en is er commerciële ondersteuning beschikbaar.

### 11.1. Complexiteit

Voor wat betreft de complexiteit van de implementatie was er voor Sphinx extra inzet nodig om asset informatie te indexeren. Sphinx heeft een database tabel nodig waarin alle asset informatie gedenormaliseerd opgeslagen is. Door de complexe database structuur van MediaMosa was hiervoor extra tijd nodig. Ook was in het geval van Sphinx wat meer tijd nodig voor de juiste implementatie van multivalued velden. Multivalued velden zijn in database termen 1 op n relaties. Velden die meerdere waardes kunnen bevatten. Sphinx is hierin enigszins beperkt en ondersteunt alleen multivalued velden van het type integer.

### 11.2. Indexeren

Beide zoekmachines zijn geschikt om de gevraagde hoeveelheid documenten te indexeren. Een minpuntje van Sphinx is het feit dat deze zoekmachine geen gedeeltelijke updates ondersteunt. Dit betekent in de praktijk dat het langer duurt voordat nieuwe of gewijzigde assets zoekbaar zijn. Omdat het indexeren bij Sphinx erg snel is hoeft dit niet direct een probleem op te leveren.

### 11.3. PHP ondersteuning

Zowel Sphinx als Apache Solr hebben goede PHP ondersteuning. Er zijn meerdere bibliotheken voor beide zoekmachines beschikbaar. De bibliotheken hebben een rijke API. Hoewel we in dit rapport gebruik hebben gemaakt van een PHP extensie is het voor beide zoekmachines niet nodig om specifieke PHP extensies aan de PHP software toe te voegen. Dit betekent dat voor wat betreft de integratie met beide zoekmachines er geen extra eisen aan de hosting omgeving zijn.

### 11.4. Software licenties

MediaMosa en Sphinx gebruiken dezelfde software licentie GPLv2. Apache Solr maakt gebruik van de Apache license v2. Deze licentie is niet compatibel met GPLv2. Dit wil zeggen dat Apache Solr niet als 1 software bundel gedistribueerd mag worden. In de praktijk zal dit waarschijnlijk niet heel relevant zijn. Het is waarschijnlijk dat een dergelijke zoekmachine als extra module of extensie apart aangeboden wordt. Daarnaast is het zo dat de PHP code die als onderdeel van MediaMosa geschreven wordt wel onder GPLv2 uitgebracht kan worden. Het verschil van licentie is alleen van toepassing op de Solr java server installatie.

### 11.5. Functionele verschillen

Voor gebruikers zijn er in eerste instantie weinig functionele verschillen. Beide zoekmachines sorteren op relevantie. Er zijn uiteraard kleine verschillen in de relevantie score. Een pluspunt van Apache Solr is dat deze een google-achtige zoekopdrachtsyntax heeft. Veel gebruikers kennen de functies van google en zullen dit gebruik snel overnemen bij andere zoekmachines. Apache Solr biedt echter meer mogelijkheden om het sorteren op relevantie te beïnvloeden en dus eventueel te verbeteren. Dit kan voor zowel een gebruik van de zoekmachine door extra

gewicht aan een zoekterm te geven of door de applicatie door middel van specifieke regels. Er kan op assetniveau en zelfs op veldniveau aangegeven worden hoe de relevantie score van een asset moet worden berekend.

#### **11.6. Implementatie advies**

Het advies voor de keuze voor implementatie van een zoektechnologie gaat naar Apache Solr. De reden hiervoor is dat Apache Solr voor wat betreft functionaliteit op dit moment meer te bieden heeft. Apache Solr is door zijn pluginarchitectuur flexibel en relatief eenvoudig aan te passen voor een perfecte integratie met een extern systeem. Een belangrijk aspect is dat de huidige SURFmedia zoekinterface aanzienlijk verbeterd kan worden. Apache Solr biedt hiervoor meer mogelijkheden en lagere investeringskosten. Geavanceerde zoekfuncties als facetzoeken zijn relatief eenvoudig te implementeren. Deze functies zijn beperkt beschikbaar in Sphinx en kosten meer implementatietijd. Apache Solr zal naar de toekomst toe de betere keuze zijn.

Bij de implementatie zal beoordeeld moeten worden op welke manier de index gevuld zal worden: een losse koppeling via REST of via tussentabellen rechtstreeks op de database.