

Contract:
Classificatie:
Code:

Status informatie

Vertrouwelijkheidsgraad:	TRASYYS Confidentieel	Stand:	Draft
Huidig versienummer:	V03.00	Datum eerste uitgave:	03/04/2010
Opgesteld door:	Reginald Amade	Datum:	13/10/2010
Geverifieerd door:	--	Datum:	--/--/----
Goedgekeurd door:	--	Datum:	--/--/----

Verdelingslijst

Naam	Adres	I/A	Naam	Adres	I/A
Johan Le Maire	VMM	A	Pieter Borremans	VMM	I
			Diederik Stubbe	Trasys	I

(I = ter informatie, A = voor actie)

Opgvolgingsformulier wijzigingen

Gelieve de gewijzigde tekstfragmenten per versie van het document te markeren. (Klik op 'Wijzigingen bijhouden' in het menu 'Extra')

Versie	Datum	Omschrijving	Aangepaste paragrafen
0.1	04/03/2010	Ontwerp (intern)	
1.0	10/05/2010	Ter goedkeuring (tussenresultaat)	
1.50	13/08/2010	Ter goedkeuring	
2.00	11/10/2010	Goedgekeurd tussenresultaat	
3.00	13/10/2010	Goedgekeurd eindresultaat	10

INHOUDSTABEL

1. INLEIDING	5
1.1 Doel van dit document	5
1.1.1 Het portaal	5
1.1.2 Metadata Harvesting	5
1.1.3 FELNET beheer	6
1.2 Begrippen	6
1.3 Afkortingen	6
2. DATABANKEN	7
2.1 Operationele databank	7
2.1.1 Overzicht	7
2.1.2 Package Raadplegen en Lenen	7
2.1.3 Package Metadata Harvesting	7
2.1.4 Package FELNET beheer	8
2.2 Staging databank	8
2.2.1 Overzicht	8
2.2.2 Package <staging package1>	8
2.2.3 Package < staging package2>	8
2.3 Data warehouse	8
2.3.1 Overzicht	8
2.3.2 Package <data warehouse package1>	8
3. SCHERMEN	9
3.1 FELNET beheermodule	9
3.1.1 Documentbeheer	9
4. RAPPORTEN	13
4.1 Operationele rapportering	13
4.1.1 Overzicht van de operationele rapportering	13
4.1.2 <Operationeel rapport 1>	13
4.2 Data warehouse rapportering	13
4.2.1 Overzicht van de data warehouse rapportering	13
4.2.2 <Data warehouse rapport 1>	13
5. BATCH PROCEDURES	14
5.1 Operationele databank	14
5.1.1 Overzicht van de batch procedures	14
5.1.2 <Batch procedure 1>	14
5.2 Staging databank	14
5.2.1 Overzicht van de batch procedures	14
5.2.2 <Batch procedure 1>	14
5.3 Data warehouse databank	14
5.3.1 Overzicht van de batch procedures	14
5.3.2 <Batch procedure 1>	14

6. FUNCTIES	15
6.1 Business services	16
6.1.1 Scheduler BS	16
6.1.2 Service Provider BS	17
6.1.3 Data Provider BS	19
6.2 Business Connector services	21
6.2.1 De Service Provider BCS	21
6.2.2 De Data Provider BCS	23
6.3 Entity services	24
6.3.1 Oogststrategie ES	24
6.3.2 Oogstproces ES	24
6.3.3 Oogstaanvraag ES	25
6.3.4 Partner ES	25
6.3.5 Document ES	25
6.4 Connection services	26
6.4.1 Data Provider CS	26
6.4.2 Document CS	26
6.5 Infrastructuur services	26
6.5.1 EmailIS	26
6.5.2 Logging	26
6.6 Operationele databank	26
6.6.1 Triggers	26
6.6.2 Stored procedures	26
6.6.3 Andere functies	27
6.7 Staging databank	27
6.8 Data warehouse databank	27
7. CONVERSIE	28
7.1 Brongegevens	28
7.2 Conversiebeslissingen	28
7.3 Automatische conversieprocedures	28
7.3.1 Operationele databank	28
7.3.2 Staging databank	28
7.3.3 Data warehouse databank	28
7.4 Manuele conversieprocedures	28
8. TRANSITIE	29
8.1.1 Operationele databank	29
8.1.2 Staging databank	29
8.1.3 Data warehouse databank	29
9. ARCHITECTUUR	30
9.1 Huidige architectuur (AS-IS)	30
9.1.1 Deelsystemen	30
9.1.2 Lagen	30
9.2 Gewenste architectuur (TO-BE)	30
9.2.1 Lagen	30
9.2.2 Deelsystemen	31
9.2.3 Use Case Realizations	33

10. INSCHATTING VAN DE ONTWIKKELINGEN	39
10.1 Aandachtspunten	39
10.2 Gefaseerde implementatie	40
10.2.1 Release 1	42
10.2.2 Release 2	43
11. INFORMATIEBRONNEN	44
11.1 Referentiedocumenten	44
11.2 Interviews	44
12. BIJLAGEN	45
12.1 Technische beschrijving van de OAI-PMH implementatie	45
12.1.1 Gebruik	45
12.1.2 OAI-PMH protocol	46
12.1.3 Aandachtspunten	56
12.2 Bedrijfsregels	59
12.3 Domains	59
12.4 Foutcodes en foutboodschappen	59
12.5 Tabellen en velden vanuit PowerDesigner	59
12.5.1 Conversie	60
12.5.2 Transitie	60
12.6 Data voor <Referentietabel 1>	60
12.7 Data voor <Referentietabel 2>	60

1. INLEIDING

FELNET is een vrijwillig verband van bibliotheken met een milieuachtergrond, met als doel om milieu-informatie zo breed mogelijk te verspreiden, en intern kennis uit te wisselen. Het belangrijkste instrument is het portaal www.FELNET.eu.

Dit document bevat de technische analyse van de oplossing die in de functionele analyse "FELNET2" werd voorgesteld.

1.1 Doel van dit document

In dit document worden de volgende punten in detail beschreven

1. de voorgestelde oplossing, die er van uit gaat dat de synchronisatie, volgens het OAI concept, tussen de partner- en de FELNET bibliotheek, automatisch verloopt.
2. de functionele domeinen zoals de VuFind zoekmotor in samenhang met het te ontwikkelen portaal en het beheer ervan, het metadata harvesting proces en het FELNET beheer

1.1.1 Het portaal

Het portaal geeft de (publieke) gebruiker toegang tot de bibliotheekcatalogi van de FELNET partners. Dit wordt bereikt door incorporatie van de open source zoekmotor VuFind en een nieuwe website met vernieuwde content en een geïntegreerd content management systeem.

1.1.2 Metadata Harvesting

Om de toegang tot de partner bibliotheekcatalogi mogelijk te maken synchroniseren zij geregeld hun catalogus met de centrale FELNET databank. Dit proces, Metadata Harvesting, wordt in detail beschreven en streeft transparant te zijn voor OAI compliant partners. Voor niet OAI compliant partners wordt een manuele procedure voorgesteld.

Metadata Harvesting maakt gebruik van het *aanbieder - consument* principe, waarbij de partners de (metadata)aanbieders zijn en FELNET de consument. De laatste is verantwoordelijk voor de synchronisatie van de FELNET databank met de deelnemende partnerbibliotheken.

FELNET synchroniseert, op in te plannen tijdstippen, de FELNET databank met de partnerbibliotheken. De aanbieders zullen de aanvraag ontvangen en de metadata, die aan de eisen van de aanvraag voldoen, uit hun systeem opvragen en opsturen naar FELNET, die de FELNET databank up-to-date brengt.

Op overeen te komen tijdstippen wordt de gesynchroniseerde FELNET databank gepubliceerd, zodat de data beschikbaar is voor de VuFind zoekmotor en het publiek.

De aanbieders, FELNET partners, zullen de diensten, zoals aangegeven, moeten aanbieden, zodat FELNET deze kan consumeren, , met de door haar geïmplementeerde diensten.

1.1.3 FELNET beheer

Met behulp van deze toepassing kan de gebruiker met beheerrechten de data die nodig is om het Metadata Harvesting proces goed te laten verlopen. De data wordt opgeslagen in een nieuwe databank waarin referentietabellen en gebruikertabellen onderhouden worden.

1.2 Begrippen

Zie FELNET2_FA_Woordenlijst.doc.

1.3 Afkortingen

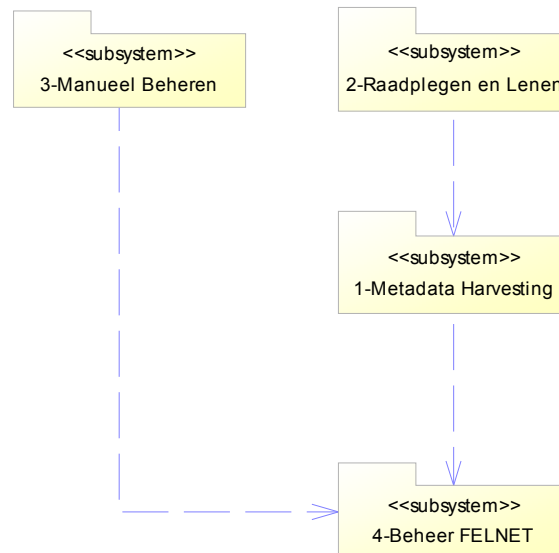
Zie FELNET2_FA_Woordenlijst.doc.

2. DATABANKEN

Zie [Conceptual Report.html](#)

2.1 Operationele databank

2.1.1 Overzicht



Figuur 2-1: package overzicht

2.1.2 Package Raadplegen en Lenen

2.1.2.1 ERD diagram voor Raadplegen en Lenen

Zie [Conceptual Report.html](#)

2.1.2.2 Tabellen en velden voor Raadplegen en Lenen

Zie [Conceptual Report.html](#)

2.1.2.3 Bedrijfsregels voor Raadplegen en Lenen

Zie [Conceptual Report.html](#)

2.1.3 Package Metadata Harvesting

2.1.3.1 ERD diagram voor Metadata Harvesting

Zie [Conceptual Report.html](#)

2.1.3.2 Tabellen en velden voor Metadata Harvesting

Zie [Conceptual Report.html](#)

2.1.3.3 Bedrijfsregels voor Metadata Harvesting

Zie [Conceptual Report.html](#)

2.1.4 Package FELNET beheer

2.1.4.1 ERD diagram voor FELNET beheer

Zie [Conceptual Report.html](#)

2.1.4.2 Tabellen en velden voor FELNET beheer

Zie [Conceptual Report.html](#)

2.1.4.3 Bedrijfsregels voor Metadata Harvesting

Zie [Conceptual Report.html](#)

2.2 Staging databank

N.V.T.

2.2.1 Overzicht

2.2.2 Package <staging package1>

2.2.2.1 ERD diagram voor <staging package 1>

2.2.2.2 Tabellen en velden voor <staging package 1>

2.2.3 Package < staging package2>

2.3 Data warehouse

N.V.T.

2.3.1 Overzicht

2.3.2 Package <data warehouse package1>

2.3.2.1 ERD diagram voor <data warehouse package 1>

2.3.2.2 Tabellen en velden voor <data warehouse package 1>

3. SCHERMEN

De schermen, waar van toepassing, worden middels een navigatiekaart overzichtelijk weergegeven. Daarnaast wordt de mapping van de data en de functionaliteit aangegeven.

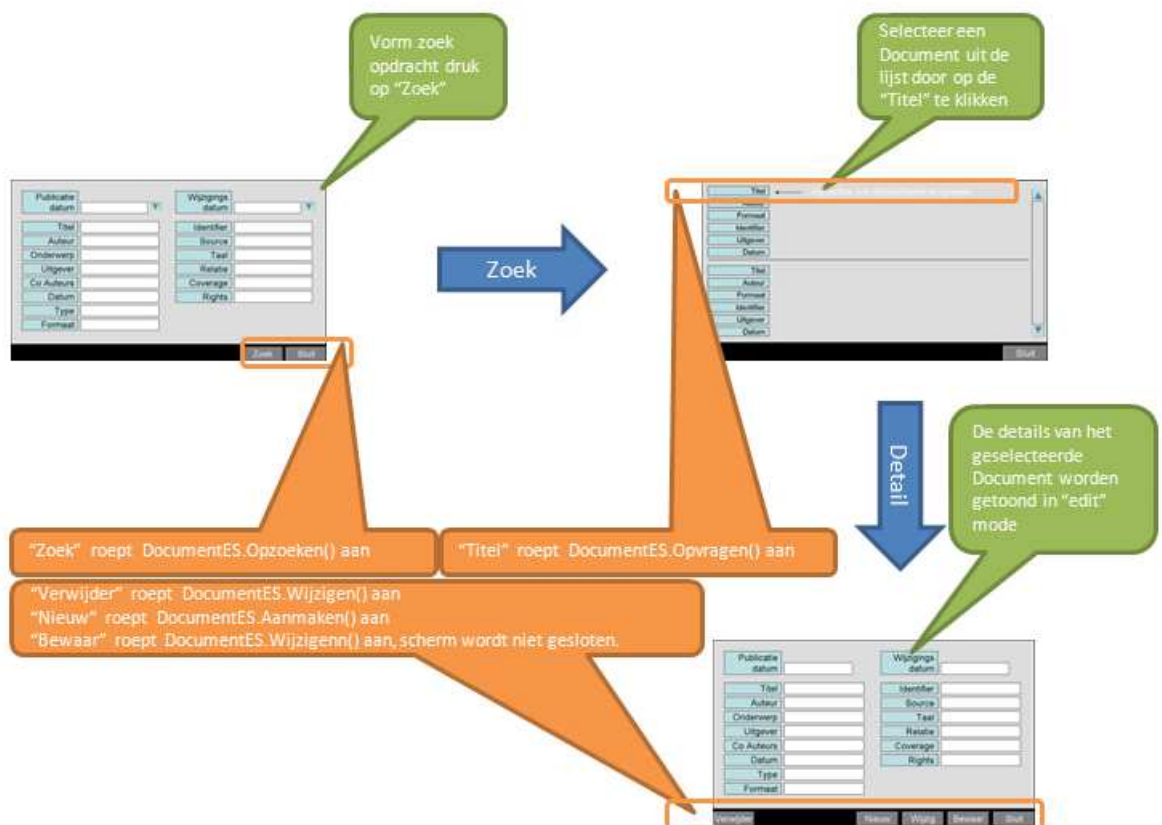
De schermen van de VuFind portal worden niet meer detail beschreven: voor de functionele specificatie verwijs ik naar de FA.

In dit hoofdstuk worden hoofdzakelijk de schermen van de FELNET beheermodule verder gedetailleerd.

3.1 FELNET beheermodule

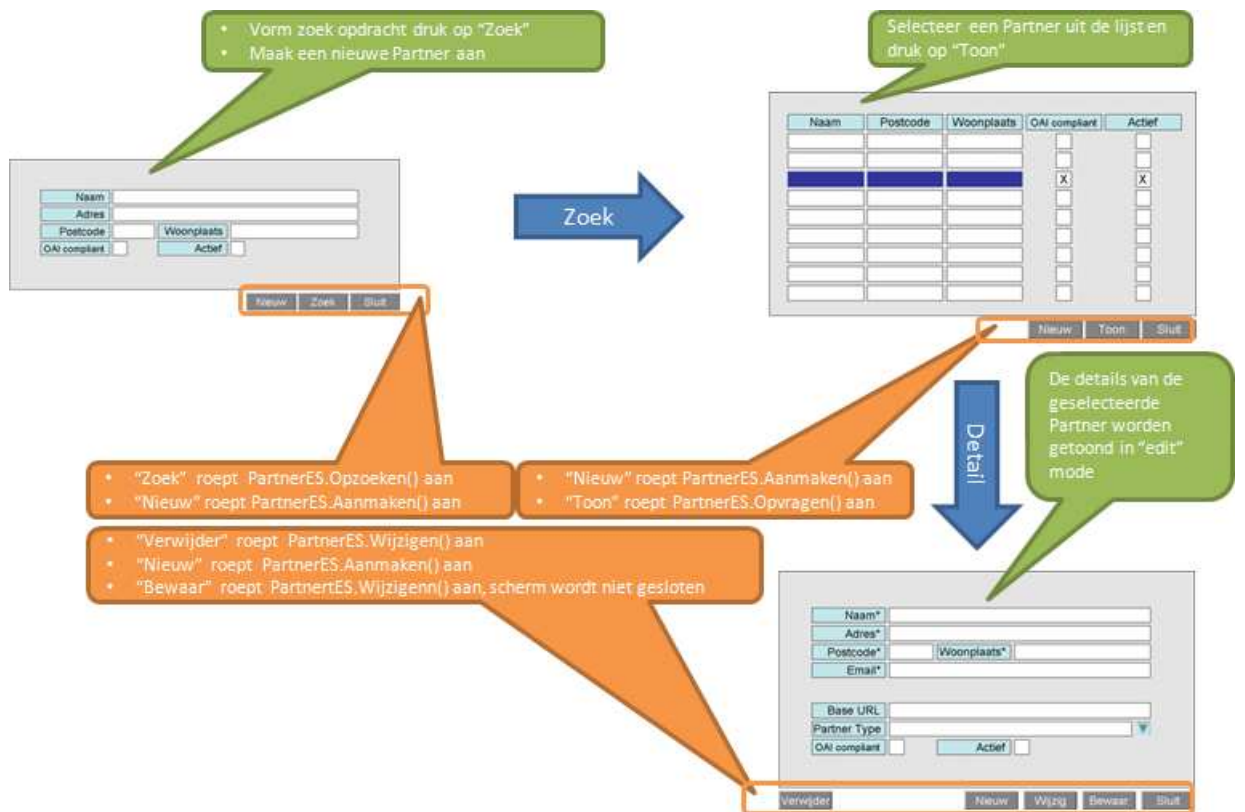
3.1.1 Documentbeheer

Onder (manueel) documentbeheer in de FELNET bibliotheek worden eenvoudige taken bedoeld die deel zijn van de data levenscyclus: creëren, lezen, wijzigen en verwijderen (CRUD) en uitgevoerd worden door de bibliotheekbeheerder.



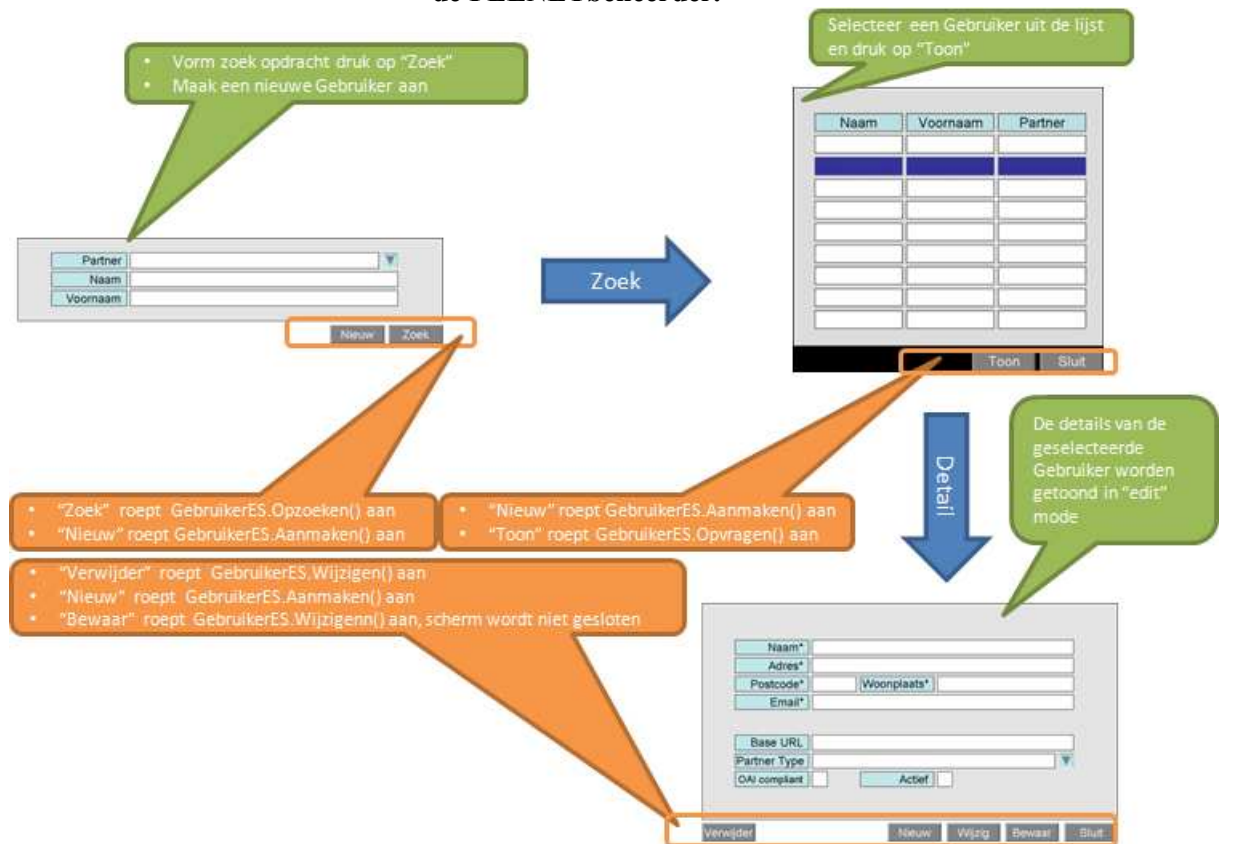
Figuur 3-1
Partnerbeheer

Onder Partnerbeheer worden eenvoudige taken bedoeld die deel zijn van de data levenscyclus: creëren, lezen, wijzigen en verwijderen (CRUD) en uitgevoerd worden door de FELNETbeheerder.



Figuur 3-2
Gebruikerbeheer

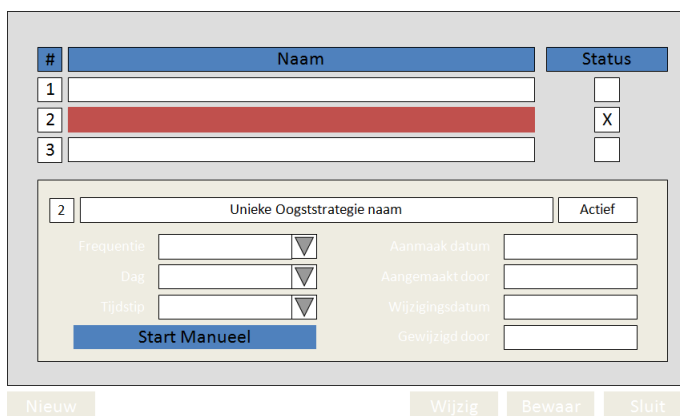
Onder Gebruikerbeheer worden eenvoudige taken bedoeld die deel zijn van de data levenscyclus: creëren, lezen, wijzigen en verwijderen (CRUD) en uitgevoerd worden door de FELNETbeheerder.



Figuur 3-3: Oogstprocesbeheer

Eens een Oogstproces is gestart dan heeft de FELNET beheerder een aantal beheerschermen ter beschikking. Deze beheerfunctionaliteit stelt de FELNET beheerder in staat om

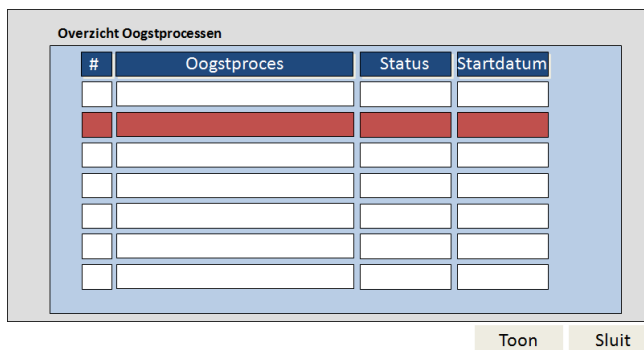
Een oogstproces manueel te starten



Figuur 3-4

Het Oogststrategie beheerscherf stelt de gebruiker in staat om een Oogststrategie naar keuze te activeren (aanroepen `OogststrategieES.Wijzigen()`) en een Oogstproces manueel te starten (aanroepen `OogstprocesES.Aanmaken()`).

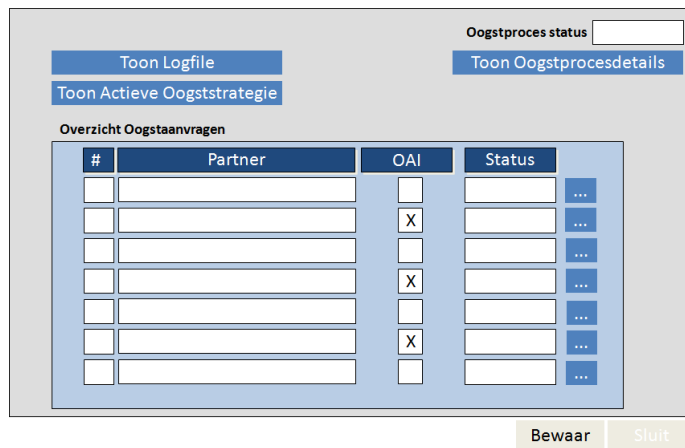
Een Oogstproces te monitoren



Figuur 3-5

Het Oogstproces zoekscherm toont de lijst van geïnitieerde Oogstprocessen in aflopend gesorteerd op startdatum. De gebruiker kan in de lijst een Oogstproces selecteren en dan op de knop “Toon” drukken (aanroepen `OogstprocesES.Opvragen()`) om het detail te zien.

Een Oogstproces te wijzigen.



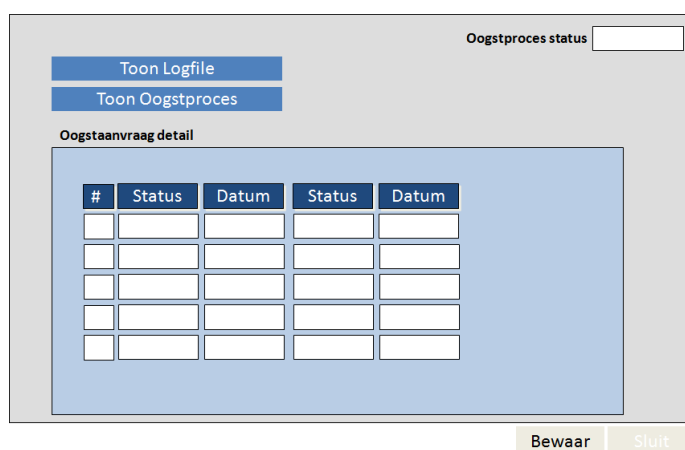
Figuur 3-6

Het Oogstproces beheerscherms toont de details van het geselecteerde Oogstproces. Het toont een lijst van de door het systeem gegenereerde Oogstaanvragen. Elk van de Oogstaanvragen werd naar een specifieke Partner gestuurd waarvan de OAI compliancy aangeduid is. Tevens wordt van de Oogstaanvraag de status getoond.
 De gebruiker kan van een Oogstaanvraag naar keuze de details opvragen (aanroepen `OogstaanvraagES.Opvragen()`)

Daarnaast kan de gebruiker

- de logfile opvragen: aanroepen `LoggingIS.Opvragen()`
- de details van de actieve Oogststrategie opvragen: `OogststrategieES.Opvragen()`
- de details van het geselecteerde Oogstproces opvragen: `OogstprocesES.Opvragen()`

Een Oogstaanvraag te wijzigen



Figuur 3-7

Het Oogstaanvraag beheerscherms toont de historiek: de statusovergangen worden in detail getoond : aanroepen `OogstaanvraagES.Opvragen()`

4. RAPPORTEN

N.V.T.

4.1 Operationele rapportering

4.1.1 Overzicht van de operationele rapportering

4.1.2 <Operationeel rapport 1>

4.2 Data warehouse rapportering

4.2.1 Overzicht van de data warehouse rapportering

4.2.2 <Data warehouse rapport 1>

5. BATCH PROCEDURES

N.V.T.

5.1 Operationele databank

5.1.1 Overzicht van de batch procedures

5.1.2 <Batch procedure 1>

5.2 Staging databank

5.2.1 Overzicht van de batch procedures

5.2.2 <Batch procedure 1>

5.3 Data warehouse databank

5.3.1 Overzicht van de batch procedures

5.3.2 <Batch procedure 1>

6. FUNCTIES

De functionaliteit die in dit hoofdstuk wordt beschreven gaat uit van de volgende assumpties

1. de Oogstfrequentie wordt vastgelegd middels de definitie van een Oogststrategie
 - a. Een timer leest de actieve Oogststrategie automatisch in. Deze timer vergelijkt de Oogstdatum met de systeemdatum
 - i. Indien Oogstdatum = systeemdatum dan wordt het Oogstproces gestart
 - ii. Indien Oogstdatum \neq systeemdatum dan gebeurt er niets
 - b. De FELNET beheerder start het Oogstproces manueel
2. Niet alle FELNET Partners zullen OAI Compliant zijn. De functionaliteit voorziet in de volgende scenario's
 - a. Alle Partners zijn OAI Compliant
 - b. Niet alle Partners zijn OAI Compliant
 - c. Geen van de Partners is OAI Compliant

De volgende paragrafen beschrijven het technisch ontwerp aan de hand van scenario's:

1. Scenario 1: de assumpties 1(a en/of) en 2a zijn waar
2. Scenario 2: de assumpties 1(a en/of) en 2 (b en/of c) zijn waar

Het technisch ontwerp is, in feite de FELNET implementatie van de bedrijfsdiensten, die het FELNET systeem realiseert en wordt beschreven aan de hand van:

1. de contracten die zij aanbieden,
2. de berichten die tussen de bedrijfsdiensten worden uitgewisseld en
3. de business logica die zij realiseren.

De bedrijfsdiensten zijn in hoofdstuk 9 "Architectuur" gevisualiseerd, opgelijst en beknopt beschreven. In de volgende paragrafen worden de bedrijfsdiensten telkens aan de hand van de 3 hierboven genoemde specificaties beschreven.

De bedrijfsdiensten zijn opgedeeld in

- Business Services,
- Business Connector Services,
- Entity Services
- Connectivity Services

en worden in de volgende 5 paragrafen beschreven

6.1 Business services

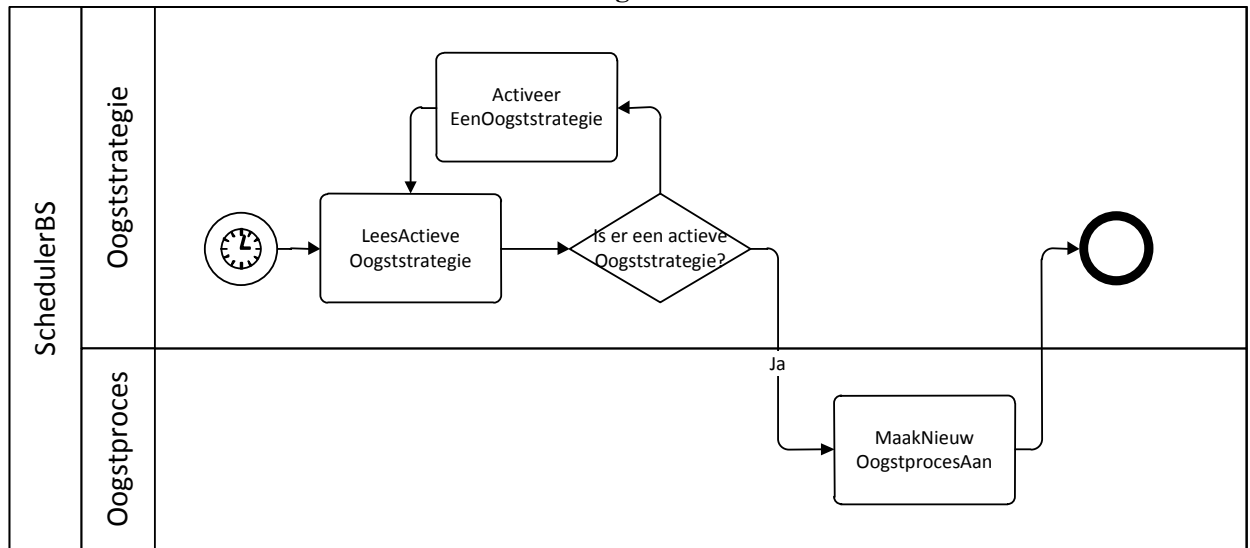
6.1.1 Scheduler BS

De Scheduler Business service heeft de verantwoordelijkheid om aan de hand van de actieve Oogststrategie een nieuw Oogstproces aan te maken met de status “*geïnitieerd*”. Na het succesvol aanmaken van een nieuw Oogstproces stuurt de Scheduler Business service een bericht met het OogstprocesID naar de Business Connector service.

6.1.1.1 Contract

Aangeboden diensten	Berichttype		Capability
	Input	Output	
LeesActieveOogststrategie	-	het resultaat van de raadpleging, een Oogststrategie record	Deze dienst implementeert het raadplegen van de Oogststrategie tabel, het selecteren van de actieve Oogststrategie en het aanroepen van <code>NieuwOogstprocesAanmaken()</code> .
NieuwOogstprocesAanmaken	Oogststrategie	Boolean	Deze dienst bepaalt of er een Oogstproces aangemaakt moet worden en vergelijkt dag, tijdstip en frequentie met de systeemdatum: Ja – Aanroepen <code>OogstprocesES.Aanmaken()</code> Nee – Aanroepen <code>ActiveerEenOogststrategie()</code>
ActiveerEenOogststrategie	-	Oogststrategie	Deze dienst bepaalt eerst in welke modus het systeem zich bevindt: Modus = “ <i>manueel</i> ”: roept <code>OogststrategieES.Wijzigen()</code> aan, die het Oogststrategie beheerschermpent. Modus = “ <i>automatisch</i> ”: schrijft een foutbericht in de logfile en roept de <code>emailIS.FoutberichtVersturen()</code> aan die de mail, met het foutbericht, naar de FELNET beheerder stuurt

Tabel 6-1: het Scheduler Business service contract Business logica



Figuur 6-1: Scheduler Business service Business Logica

6.1.2 Service Provider BS

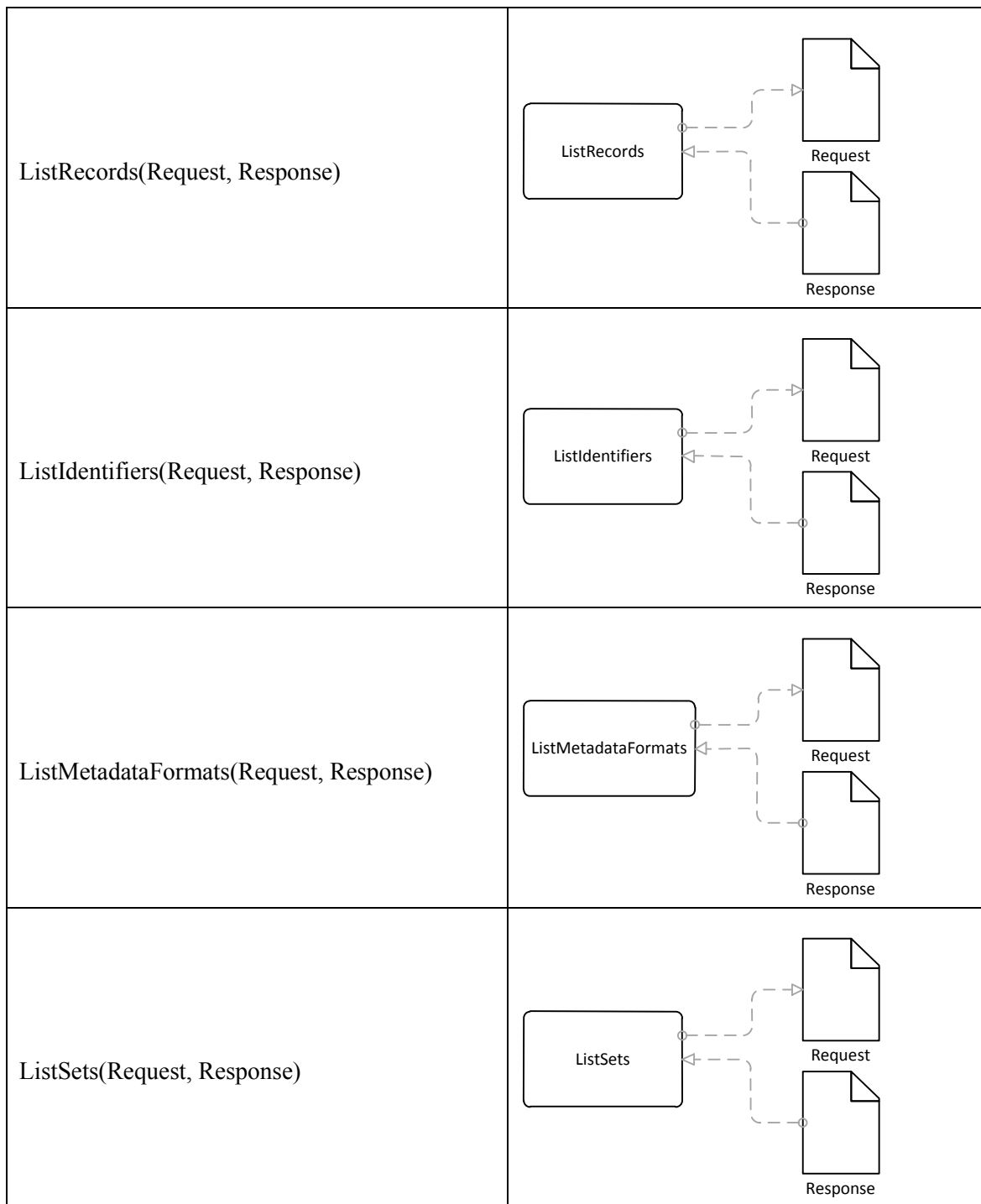
De Service Provider Business service heeft de verantwoordelijkheid om de ontvangen OAI compliant berichten door te sturen naar de Partners.

Response

De Service Provider Business service ontvangt het OAI compliant resultaat en stuurt het door naar de Business Connector service (Service Provider).

6.1.2.1 Contract

Aangeboden diensten	Visuele representatie
Identify(Request, Response)	
GetRecord(Request, Response)	



Tabel 6-2: OAI-PMH compliant contract

6.1.2.2 Berichten

Bericht	Inhoud
ListRecordsRequest	Het bericht bevat minstens het argument Verb met de waarde "ListRecords" en metadatataprefix met de waarde "felnet"
ListRecordsResponse	De response bevat een bericht dat voldoet aan oai-pmh.xsd met metadata volgens

	marexml
--	---------

Tabel 6-3: Belangrijkste berichten

6.1.2.3 Business Logica

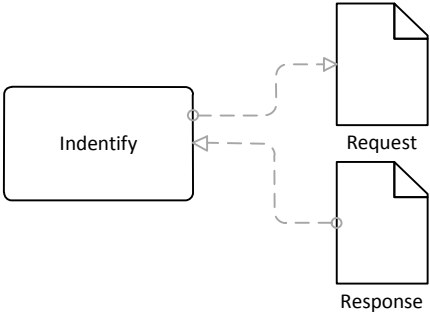
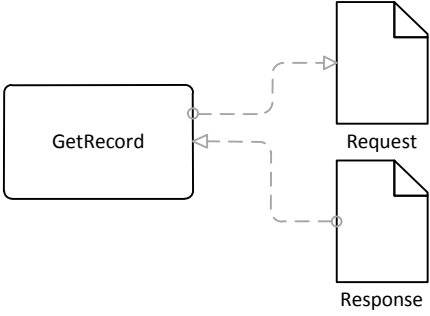
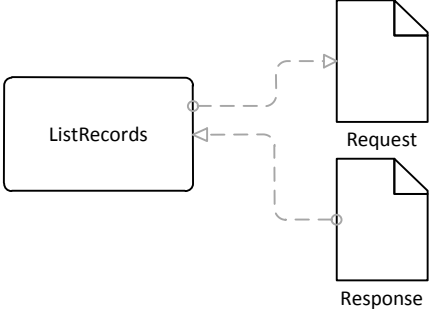
Zie OAI-PMH, OSS project OAICat en OAIHarvester2 specificaties

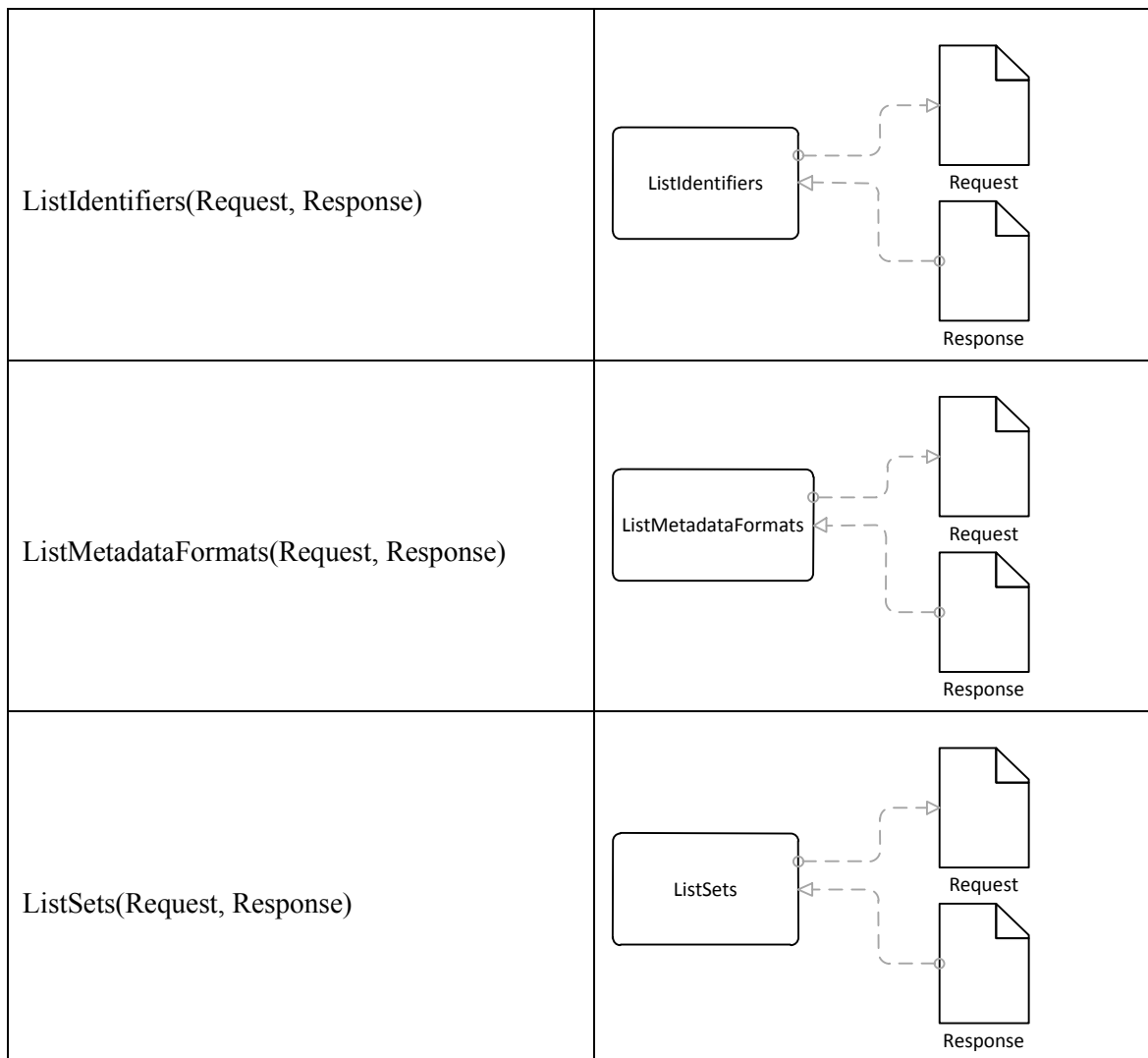
6.1.3 Data Provider BS

De Data Provider Business service heeft de verantwoordelijkheid om berichten, ontvangen van de Service Provider Business service, te controleren. OAI compliant berichten worden doorgestuurd naar de Data Provider Business Connector service. Foute berichten worden volgens OAI – PMH afgehandeld.

De Data Provider Business service ontvangt het OAI compliant resultaat van de Partner databankbevraging en stuurt het door naar de Service Provider.

6.1.3.1 Contract

Aangeboden diensten	Visuele representatie
Identify(Request, Response)	 <p>The diagram shows a service boundary labeled 'Identify'. A dashed arrow points from the service to a document icon labeled 'Request'. A return dashed arrow points from the 'Request' back to the service. Another dashed arrow points from the service to a second document icon labeled 'Response'. A return dashed arrow points from the 'Response' back to the service.</p>
GetRecord(Request, Response)	 <p>The diagram shows a service boundary labeled 'GetRecord'. A dashed arrow points from the service to a document icon labeled 'Request'. A return dashed arrow points from the 'Request' back to the service. Another dashed arrow points from the service to a second document icon labeled 'Response'. A return dashed arrow points from the 'Response' back to the service.</p>
ListRecords(Request, Response)	 <p>The diagram shows a service boundary labeled 'ListRecords'. A dashed arrow points from the service to a document icon labeled 'Request'. A return dashed arrow points from the 'Request' back to the service. Another dashed arrow points from the service to a second document icon labeled 'Response'. A return dashed arrow points from the 'Response' back to the service.</p>



Tabel 6-4: OAI-PMH compliant contract

6.1.3.2 Berichten

Bericht	Inhoud
ListRecordsRequest	Het bericht bevat minstens het argument Verb met de waarde “ListRecords” en metadataprefix met de waarde “felnet”
ListRecordsResponse	De response bevat een bericht dat voldoet aan oai-pmh.xsd met metadata volgens marxml

Tabel 6-5:

6.1.3.3 Business Logica

Zie OAI-PMH, OSS project OAICat en OAIHarvester2 specificaties

6.2 Business Connector services

6.2.1 De Service Provider BCS

De ServiceProviderBCS heeft de verantwoordelijkheid om Oogstaanvragen in het kader van een Oogstproces aan te maken en OAI berichten door te sturen.

Voor niet OAI compliant Partners wordt alternatieven aangeboden. Deze Partners ontvangen en email met het verzoek om nieuwe/gewijzigde documenten naar de FELNET beheerder te sturen.

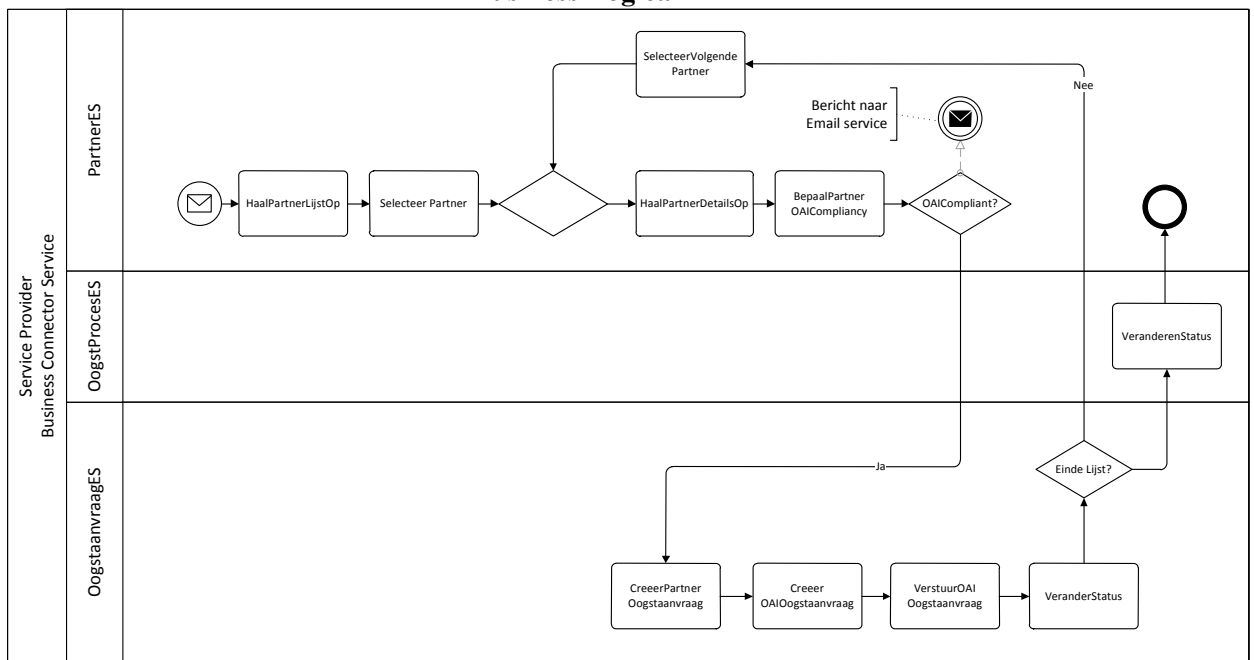
Anderzijds zal de ServiceProviderBCS berichten ontvangen van de DataProviderBS OAI RequestResponses, die ze doorstuurt naar de ServiceProviderBS.

6.2.1.1 Contract

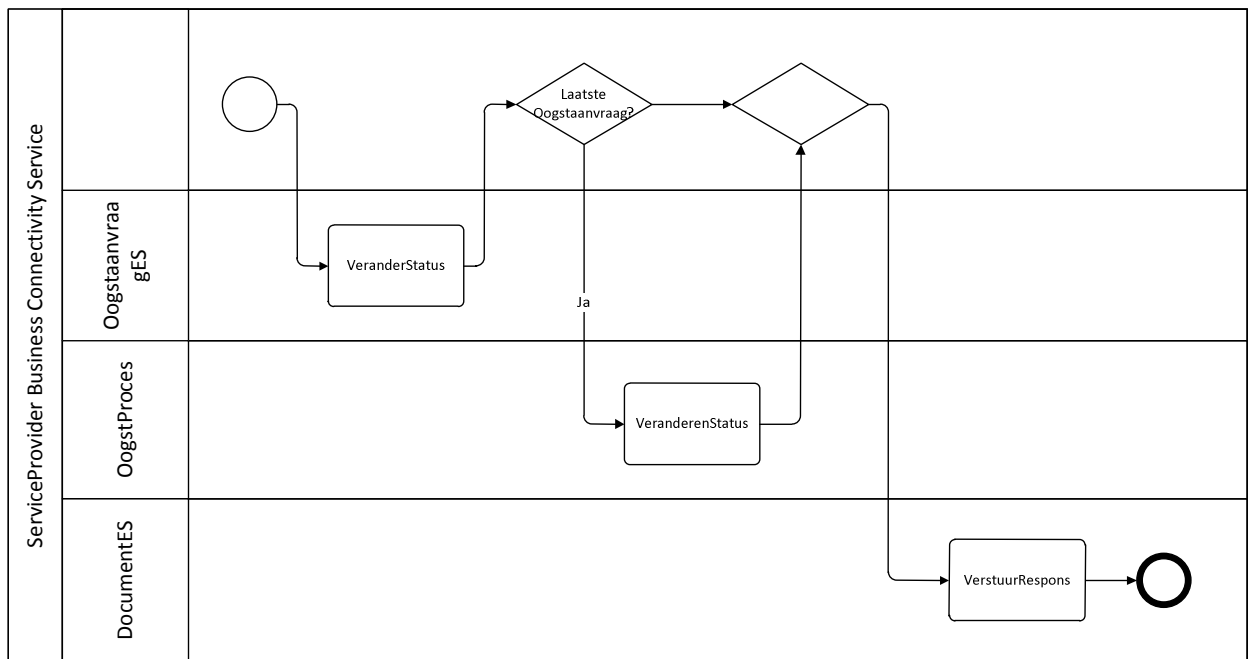
Aangeboden diensten	Berichttype		Capability
	Input	Output	
HaalPartnerLijstOp	-	PartnerLijst	bepaalt actieve Partners en geeft Partnerlijst terug: Roep PartnerES.GeefActievePartners () aan
SelecteerPartner	-	PartnerID	Selecteer de eerste Partner in de lijst, geef aan de hand van de geselecteerde Partner, roep HaalPartnerDetailsOp () aan. Herhaal voor alle Partners
HaalPartnerDetailsOp	PartnerID	Partner	Roep PartnerES.GeefPartner () aan
BepaalPartnerOAICompliance	Partner	boolean	Roep PartnerES.IsOAICompliant () aan "Ja" – Roep CreeerPartnerOogstaanvraag () aan "Nee" – Roep EmailIS.GeefGewijzigdeDocumenten () aan
CreeerPartnerOogstaanvraag	OogstProces	Oogstaanvraag	Roep eerst OogstaanvraagES.Aanmaken () aan en dan CreeerOAI Oogstaanvraag ()
CreeerOAI Oogstaanvraag	Oogstaanvraag	OAI bericht	Roep OogstaanvraagES.Transformeren () aan en dan VerstuurOAI Oogstaanvraag ()
AlleOogstaanvragenAangemaakt	Partnerlijst	boolean	Vergelijk het aantal Oogstaanvragen en emails met het aantal actieve Partners

			<p>“Ja” – ze zijn gelijk: roep OogstprocesES.VeranderenStatus () aan</p> <p>“Nee” – ze zijn ongelijk: doe niets</p>
VerstuurOAI Oogstaanvraag	OAI bericht	-	Roep OogstaanvraagES.Versturen () aan
VeranderOogstaanvraagStatus	Oogstaanvraag	-	Roep Oogstaanvraag.VeranderenStatus () aan
VeranderOogstprocesStatus	Proces	-	Roep Oogstproces.VeranderenStatus () aan
VerstuurOAI Respons	XML resultaat	XML resultaat	Stuur resultaat door: roep DocumentES.Verwerken () aan

**Tabel 6-6:
Business Logica**



Figuur 6-2: BPMN diagram verwerken Oogstaanvraag



Figuur 6-3: BPMN diagram verwerken OogstaanvraagRespons

6.2.2 De Data Provider BCS

De Business Connector service heeft als verantwoordelijkheid het ontvangen OAI bericht te transformeren in een doelsysteem query, zodat het Partner doelsysteem haar databank kan bevragen.

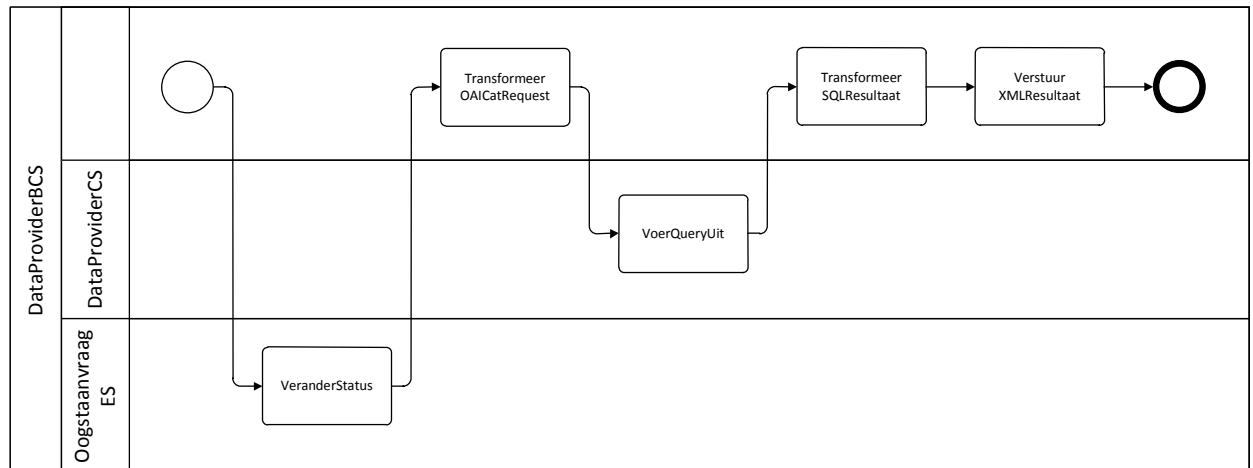
De Business Connector service ontvangt/leest het resultaat, transformeert het, indien nodig, naar MARCXML formaat, verrijkt het zodat het OAI compliant is en stuurt het bericht naar de Data Provider Business service.

6.2.2.1 Contract

Aangeboden diensten	Berichttype		Capability
	Input	Output	
VeranderStatus	Oogstaanvraag	-	Verander de status van de Oogstaanvraag naar <i>"In verwerking"</i> : roep <code>OogstaanvraagES.VeranderStatus ()</code> aan
TransformeerOAICatRequest	String	SQLString	Transformeer de String (OIA Request met deze opbouw: <code>url?verb=requesttype[&argument]</code>) naar een Partnerspecifieke SQLString
VoerQueryUit	SQLString	SQLResultaat	Roep <code>DataProviderCS.Execute ()</code> aan

TransformeerSQLResultaat	SQLResultaat	XMLResultaat	Transformeer het SQLResultaat naar XMLResultaat (MARCXML)
VerstuurXMLResultaat	XMLResultaat		Verstuur het XMLResultaat naar de DataProviderBS

6.2.2.2 Business Logica



Figuur 6-4: BPMN diagram verwerken Oogstaanvraag (Request&Respons)

6.3 Entity services

6.3.1 Oogststrategie ES

```

OogststrategieES.Aanmaken()
OogststrategieES.Wijzigen()
OogststrategieES.Opzoeken()
OogststrategieES.Opvragen()
  
```

6.3.2 Oogstproces ES

```

OogstprocesES.Aanmaken()
OogstprocesES.Wijzigen()
OogstprocesES.Opzoeken()
OogstprocesES.Opvragen()
  
```

OogstprocesES.VeranderenStatus(): Als alle Oogstaanvragen succesvol zijn aangemaakt, wordt de status van het Oogstproces veranderd naar “gestart”.

6.3.3 Oogstaanvraag ES

OogstaanvraagES.Aanmaken()

OogstaanvraagES.Wijzigen()

OogstaanvraagES.Opzoeken()

OogstaanvraagES.Opvragen()

OogstaanvraagES.VeranderenStatus(): Als alle Oogstaanvragen succesvol zijn aangemaakt, wordt de status van het Oogstproces veranderd naar “gestart”.

OogstaanvraagES.Transformeren() elk van de Oogstaanvragen wordt getransformeerd naar een OAI compliant bericht, Het bericht bevat Partnerinformatie en de periode gedefinieerd door een begindatum en een einddatum.

OogstaanvraagES.Versturen() stuur OAI bericht door naar de ServiceProviderBS.

6.3.4 Partner ES

PartnerES.Aanmaken()

PartnerES.Wijzigen()

PartnerES.Opzoeken()

PartnerES.GeefActievePartnerLijst()

PartnerES.GeefPartner()

PartnerES.IsOAICompliant()

6.3.5 Document ES

DocumentES.verwerkenResultaat(): De Business Connector service ontvangt het OAI compliant resultaat en stuurt het door naar de DocumentES die het resultaat per record beoordeelt en de juiste operatie (CRUD) uitvoert.

DocumentES.Aanmaken()

DocumentES.Wijzigen()

DocumentES.Opzoeken()

DocumentES.Opvragen()

6.4 Connection services

6.4.1 Data Provider CS

`DataProviderCS.ExecuteSQL()`: bevaag de Partner databank met de ontvangen `SQLString` en geef het resultaat terug

6.4.2 Document CS

`DocumentCS.ExecuteSQL`: voeg de nieuwe documenten toe en vervang de gewijzigde documenten

6.5 Infrastructuur services

6.5.1 EmailIS

`EmailBS.FoutberichtVersturen()`

`EmailIS.GeefGewijzigdeDocumenten()`

Voor niet OAI compliant Partners worden de volgende alternatieven aangeboden. De Business Connector Service stuurt een bericht naar de emailIS service die, een email, op basis van een sjabloon, aanmaakt en naar de Partner verstuurt met het verzoek om

1. nieuwe/gewijzigde documenten manueel te beheren met behulp van de FELNET beheermodule
2. met behulp van RVOT gecreëerde export (XML), naar de FELNET beheerder te sturen.
3. een txt-bestand te genereren met nieuwe/gewijzigde documenten en naar de FELNET beheerder te sturen

6.5.2 Logging

Nog verder uit te werken.

6.6 Operationele databank

N.V.T.

6.6.1 Triggers

6.6.1.1 Overzicht

6.6.1.2 <Trigger 1>

6.6.2 Stored procedures

6.6.2.1 Overzicht

6.6.2.2 <Stored procedure/functie 1>

6.6.3 Andere functies

6.6.3.1 Overzicht

6.6.3.2 <Andere functie 1>

6.6.3.3 <Andere functie 2>

6.7 Staging databank

N.V.T.

6.8 Data warehouse databank

N.V.T.

7. CONVERSIE

N.V.T.

7.1 Brongegevens

7.2 Conversiebeslissingen

7.3 Automatische conversieprocedures

7.3.1 Operationele databank

7.3.2 Staging databank

7.3.3 Data warehouse databank

7.4 Manuele conversieprocedures

8. TRANSITIE

N.V.T.

8.1.1 Operationele databank

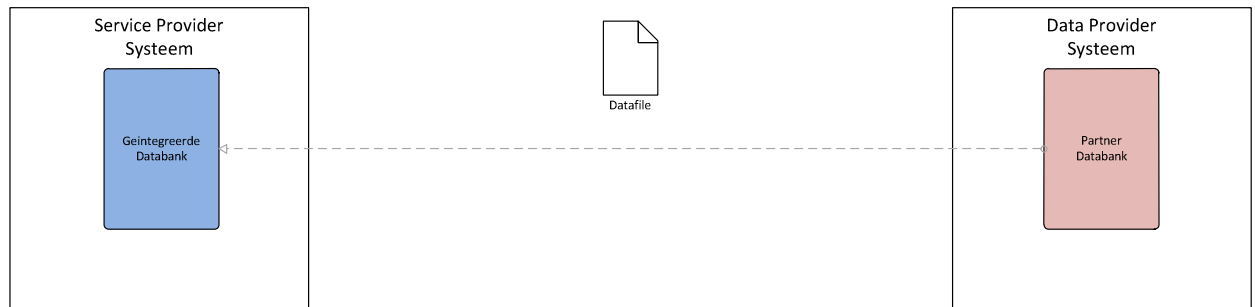
8.1.2 Staging databank

8.1.3 Data warehouse databank

9. ARCHITECTUUR

9.1 Huidige architectuur (AS-IS)

9.1.1 Deelsystemen



Figuur 9-1: huidige architectuur

De huidige architectuur bestaat uit een Service Provider Systeem, een Access toepassing, dat onderhouden wordt door de FELNET beheerder, VITO.

In het onderhoud is inbegrepen het hosten en onderhouden van de webapplicatie, het verwerken:toevoegen van gewijzigde/nieuwe documenten aan de felnet databank.

De Partners zullen op verzoek, of periodiek een bestand genereren. Dit bestand bevat alle felnet records, dan wel enkel de veranderde records. Het gegenereerde bestand wordt via email of op CD-ROM verstuurd naar de FELNET beheerder, die de wijzigingen doorvoert in de felnet databank met behulp van de Access toepassing.

9.1.2 Lagen

De Access toepassing is een klassieke client/server implementatie bestaande uit een presentatielaag en een data laag.

De presentatielaag is op een client PC geïnstalleerd terwijl de databank deel is van de webapplicatie die de data beschikbaar stelt voor de felnet website.

9.2 Gewenste architectuur (TO-BE)

9.2.1 Lagen

9.2.1.1 Presentatie

De presentatie wordt gerealiseerd door de Drupal CMS en VuFind zoekmotor componenten.

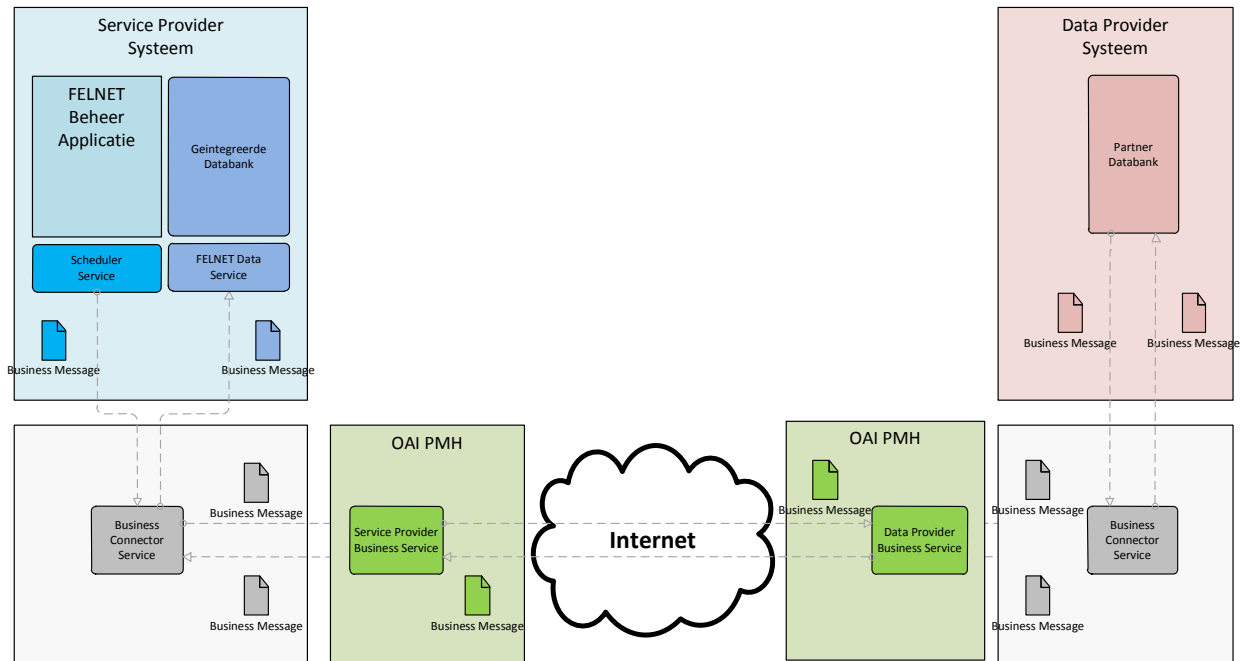
9.2.1.2 Bedrijfslogica

De bedrijfslogica wordt geïmplementeerd door de vrij beschikbare Open source componenten OAI Harvester en OAI Cat en te ontwikkelen webservice's.

9.2.1.3 Connectiviteit

Gespecialiseerde Connectivity Services verzorgen de systeemspecifieke database communicatie met de verschillende systemen.

9.2.2 Deelsystemen



Figuur 9-2: gewenste architectuur

De gewenste architectuur volgt het OAI-PMH protocol die spreekt over een Service Provider en Data Provider, de FELNET Partners.

9.2.2.1 Service Provider

In de gewenste architectuur zijn de volgende componenten voor de Service Provider te onderscheiden

1. de FELNET beheer applicatie
met deze applicatie kunnen FELNET beheerders referentie data en documenten manueel beheren (=FELNET Manueel beheer). Deze applicatie is een multi layered webapplicatie, volgens het MVC pattern, die de beheerders voorziet van een grafische userinterface, de view layer, een control layer voor de implementatie van bedrijfslogica en –regels en ten slotte de model layer die web services aanbiedt op basis van Entity Services.
Voor een gedetailleerde functionele specificatie van de FELNET beheer applicatie verwijs ik naar de werkproducten van de Functionele Analyse.
2. de geïntegreerde databank
eXist is een native xml databank. Het voordeel ervan is dat er geen transformaties (marshalling/unmarshalling) nodig zijn. Hierdoor blijven de complexiteit en de omvang van de ontwikkeling beperkt.
Tevens kan de data (xml) zondermeer geïmporteerd worden door de VuFind component solrmarc.

3. de Service Provider Business Connector service functioneert als voorbereider van het oogstproces. De Service Provider Business Connector service ontvangt een procesinitiatiebericht en zal oogstaanvragen aanmaken op basis van de beschikbare Partnerdata. Voor Partners die niet OAI compliant zijn wordt geen oogstaanvraag aangemaakt. Zij ontvangen een email met het verzoek wijzigingen op te sturen. Het formaat waarin zij dit doen zal met de individuele Partners overeengekomen moeten worden. Tot de mogelijkheden behoren marcxml, dc en textfiles. De Service Provider Business Connector service wordt geïmplementeerd als web service. De bedrijfslogica wordt gerealiseerd in Java.
4. de Service Provider Business service (OAI Harvester) zorgt in feite voor de metadata harvesting. Voor OAI compliant Partners wordt een bericht ontvangen met Partnerdata, waarmee een OAI compliant request wordt gecreeerd en opgestuurd naar de juiste Partner. De OAI Harvester component implementeert het OAI-PMH framework en is gerealiseerd in Java en het kan eenvoudig worden geïmplementeerd als web service.

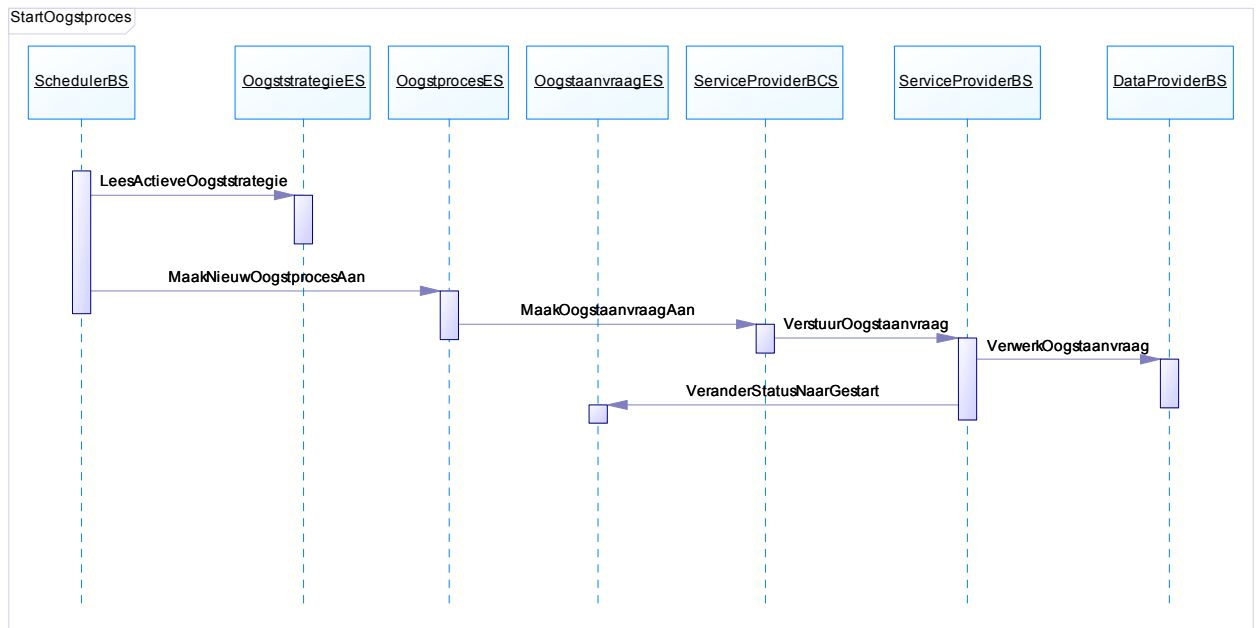
9.2.2.2 Data Provider

In de gewenste architectuur zijn de volgende componenten voor de Data Provider te onderscheiden

1. de Partner databank
De FELNETgemeenschap telt meer dan 24 leden met elk een specifieke context. Zo is er tijdens de definitiestudie geïnventariseerd dat er zo'n 8 verschillende systemen zijn mogelijk met een eigen implementatie.
2. de Data Provider Business Connector service
ontvangt OAI compliant requests. Requests worden volgens het OAI-PMH protocol verwerkt: goede request worden in behandeling genomen, terwijl requests die niet voldoen afgehandeld worden als een fout. De requests die de verificatie doorstaan worden getransformeerd naar een database specifieke query die uitgevoerd wordt. De ontvangen database responses worden volgens afspraak ontvangen. We onderscheiden xml files, zoals dc en marcxml, en text files, zoals comma separated en marc21. Ontvangen files worden, indien nodig, getransformeerd naar het doelformaat: marcxml.
3. de Data Provider Business service (OAI Cat)
ontvangt OAI compliant responses die doorgestuurd worden naar de Service Provider Business service. De OAI Cat is een Java component die het OAI-PMH framework implementeert.

9.2.3 Use Case Realizations

9.2.3.1 Sequence diagram: StartOogstproces



Figuur 9-3: StartOogstproces Sequence diagram

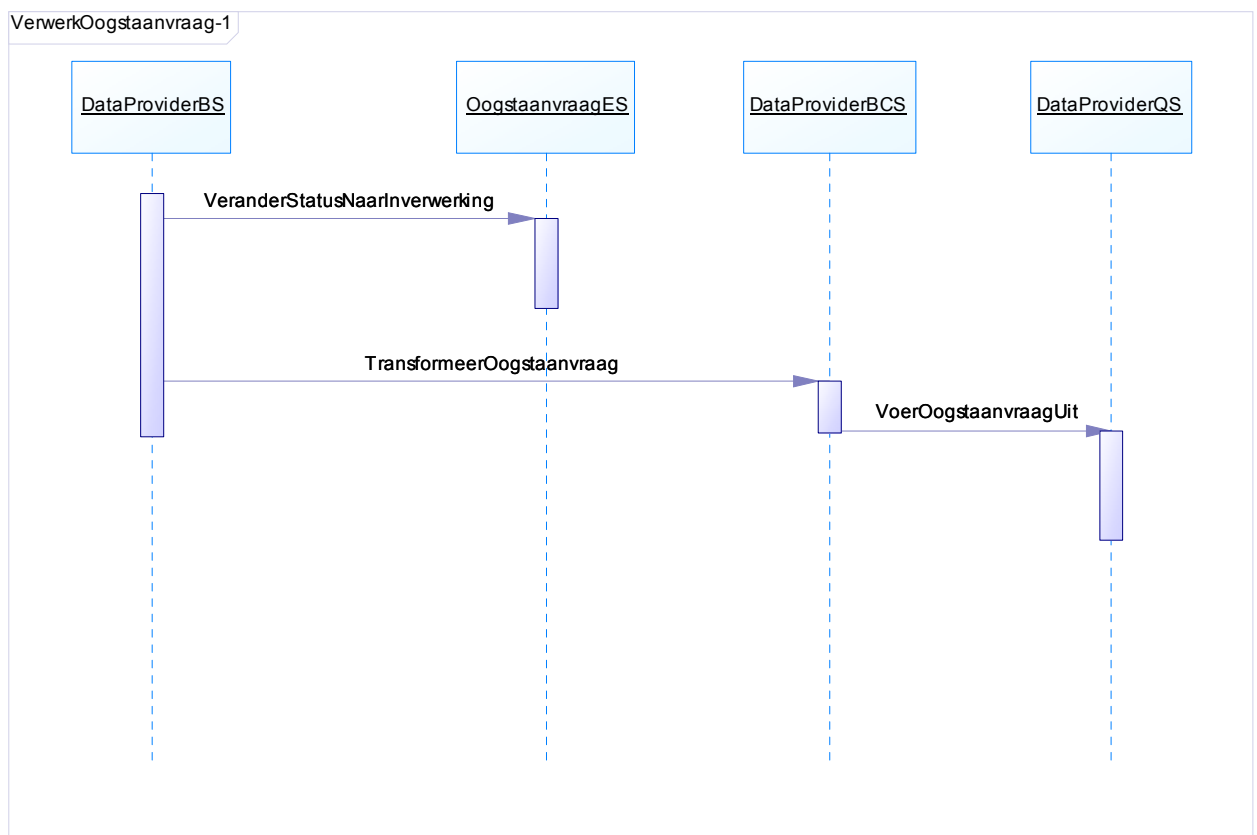
9.2.3.1.1 Geïdentificeerde objecten

Class	Objecttype	Objectnaam	Korte Beschrijving
Scheduler	Business Service	SchedulerBS	leest de Oogststrategie met status "Actief" en stuurt een xml bericht met Oogststrategiedata naar OogstProcesES
Oogststrategie	Entity Service	OogststrategieES	Stelt de actieve Oogststrategie ter beschikking
Oogstproces	Entity Service	OogstprocesES	Maakt een nieuw Oogstproces aan, op basis van de Oogststrategiedata en kent automatisch de volgende Oogstproces attributen toe: <ul style="list-style-type: none"> status "Geïnitieerd" startdatum "<systemdatum>" Na aanmaken van het Oogstproces wordt een xml bericht, met Oogstprocesdata, naar ServiceProviderBCS gestuurd
ServiceProvider	Business Connector Service	ServiceProviderBCS	Maakt een Oogstaanvraag aan voor elke actieve OAI compliant Partner en kent automatisch de volgende attributen, status "Geïnitieerd", baseUrl en OAI Partner info toe

Oogstaanvraag	Entity Service	OogststrategieES	<p>Creëert voor elke Oogstaanvraag een OAI Compliant bericht en stuurt dit naar ServiceProviderBS</p> <p>Als voor alle OAI Compliant Partners een Oogstaanvraag is aangemaakt met status “Geïnitieerd”, verandert het systeem automatisch de Oogstproces status van “Geïnitieerd” naar “Gestart”</p>
ServiceProvider	Business Service	ServiceProviderBS	<p>Maakt Oogstaanvragen voor een specifiek Oogstproces aan</p> <p>Ontvangt een OAI Compliant bericht, controleert het en stuurt het door naar de juiste Partner op basis van de baseURL en verandert de Oogstaanvraag status naar “Gestart”</p>
DataProvider	Business Service	DataProviderBS	<p>Ontvangt een OAI Compliant Partner bericht</p>

Tabel 9-1: Korte beschrijving van de geïdentificeerde objecten van het StartOogstproces

9.2.3.2 Sequence diagram: VerwerkOogstaanvraag



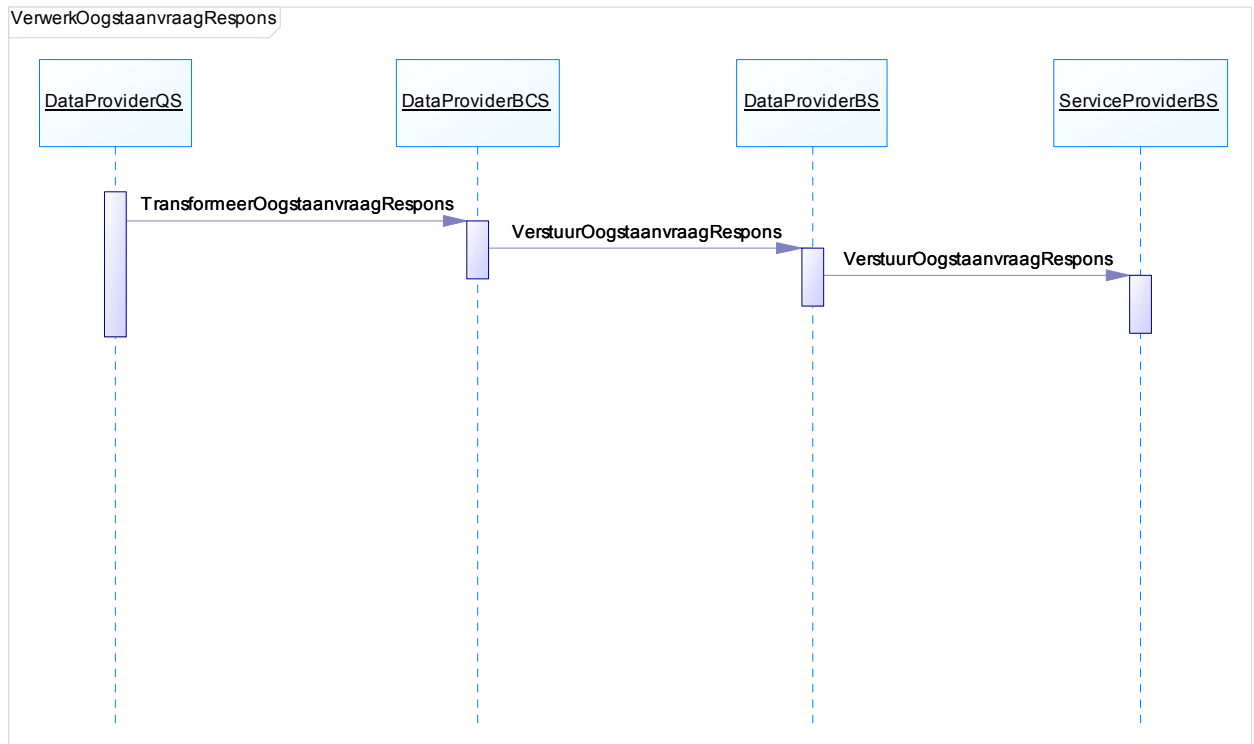
Figuur 9-4: VerwerkOogstaanvraag Sequence diagram

9.2.3.2.1 Geïdentificeerde objecten

<i>Class</i>	<i>Objecttype</i>	<i>Objectnaam</i>	<i>Korte Beschrijving</i>
DataProvider	Business Service	DataProviderBS	<p>Ontvangt een OAI compliant Partner bericht en controleert de correctheid van het bericht</p> <p>En stuurt het door naar de Business Connector Service, DataProviderBCS en verandert de Oogstaanvraag status naar "In verwerking"</p> <p>Indien het bericht niet OAI Compliant is wordt het als fout afgehandeld en verandert de Oogstaanvraag status naar "Geweigerd"</p>
DataProvider	Business Connector Service	DataProviderBCS	<p>Ontvangt het OAI Compliant Partner bericht</p> <p>Transformeert het naar een Partner specifieke query</p> <p>Stuurt de query door naar DataProviderQS</p>
DataProvider	Connectivity Service	DataProviderCS	<p>Ontvangt de query en voert de query vervolgens uit</p>

Tabel 9-2: Korte beschrijving van de geïdentificeerde objecten van VerwerkOogstAanvraag

9.2.3.3 Sequence diagram: VerwerkOogstaanvraagRespons-1



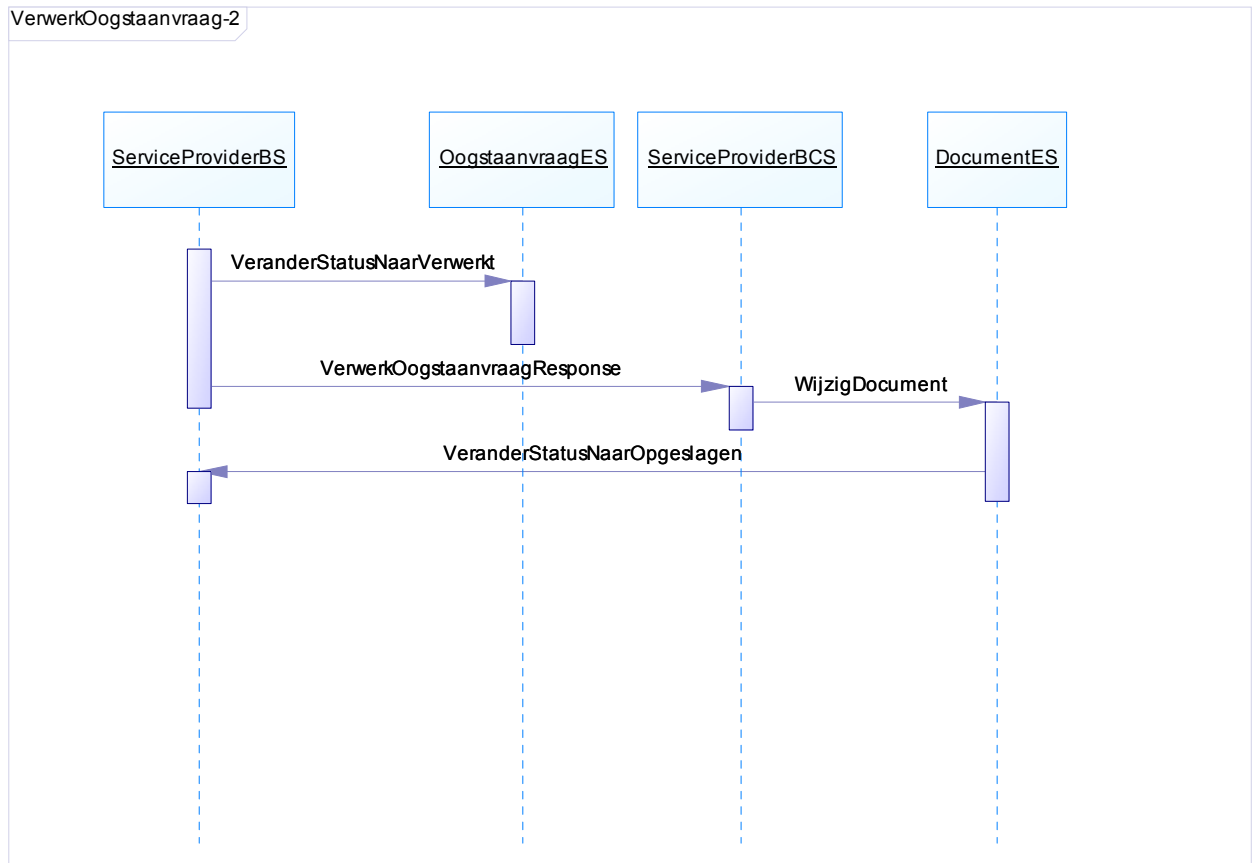
Figuur 9-5: VerwerkOogstaanvraagResponse-1 Sequence diagram

9.2.3.3.1 Geïdentificeerde objecten

<i>Class</i>	<i>Objecttype</i>	<i>Objectnaam</i>	<i>Korte Beschrijving</i>
DataProvider	Query Service	DataProviderQS	Ontvangt het resultaat en stuurt het door naar de DataProviderBCS
DataProvider	Business Connector Service	DataProviderBCS	Ontvangt het resultaat Transformeert het naar het gewenste MARC XML formaat En stuurt het door naar de DataProviderBS
DataProvider	Business Service	DataProviderBS	Ontvangt het resultaat En stuurt het door naar de ServiceProviderBS

Tabel 9-3: Korte beschrijving van de geïdentificeerde objecten van VerwerkOogstAanvraagResponse-1

9.2.3.4 Sequence diagram: VerwerkOogstaanvraagRespons-2



Figuur 9-6: VerwerkOogstaanvraagResponse-2 Sequence diagram

9.2.3.4.1 Geïdentificeerde objecten

Class	Objecttype	Objectnaam	Korte Beschrijving
ServiceProvider	Business Service	ServiceProviderBS	Ontvangt de Oogstaanvraagrespons in MARC XML formaat Verandert de Oogstaanvraag status naar "Verwerkt" En stuurt het door naar de ServiceProviderBCS
ServiceProvider	Business Connector Service	ServiceProviderBCS	Ontvangt het bericht en stuurt het door naar de Entity Service, DocumentES
Document	Entity Service	DocumentES	Ontvangt de lijst met gewijzigde documenten Slaat de gewijzigde documenten één voor één op in de Aggregator.(Dit is de native xml database eXist) En verandert de Oogstaanvraag status naar "Opgeslagen"

			Als alle Oogstaanvragen behorend bij een Oogstproces opgeslagen zijn dan wordt de Oogstproces status gewijzigd naar " <i>Gesloten</i> "
--	--	--	---

Tabel 9-4: Korte beschrijving van de geïdentificeerde objecten van VerwerkOogstAanvraagResponse-2

10. INSCHATTING VAN DE ONTWIKKELINGEN

Voor het ontwikkelen van de nieuwe FELNET toepassing worden zoals al eerder genoemd en beschreven, een aantal subsystemen onderscheiden. In de volgende paragrafen geef ik een aantal aandachtspunten en voorstellen om de nieuwe FELNET toepassing te realiseren.

10.1 Aandachtspunten

Een deel van deze punten kan direct worden gerealiseerd, maar voor een aantal zal nog bijkomend onderzoek en ontwerp gedaan moeten worden.

1. FELNET Portaal: voor de realisatie van het portaal moet de huidige FELNET website gemigreerd worden naar DRUPAL. Daarvoor moet er consensus zijn over de look&feel, de content en de structuur.
2. De VuFind zoekmotor moet nog in het portaal geïntegreerd worden
3. De schermen voor Oogststrategie- en Oogstprocesbeheer, deel van de in het portaal geïntegreerde Oogst beheermodule, worden met behulp van Drupal als webpagina's ontworpen en ontwikkeld.
4. De schermen voor FELNETbeheer, deel van de in het portaal geïntegreerde FELNET beheermodule, worden met behulp van Drupal als webpagina's ontworpen en ontwikkeld.
5. De xml schema's voor Gebruiker, Partner, Rol, Roltype en Partnertype moeten nog ontwikkeld worden.
6. De xml schema's voor Oogststrategie, Oogstproces, Oogstaanvraag en Document moeten nog ontwikkeld worden
7. De services, Business, Business Connector, Entity, Connector and Infrastructure worden als webservices ontwikkeld. Eens webservices zijn vervolledigd met operaties en berichten kunnen de interfaces (wsdl) afgeleid/ontwikkeld worden.
8. Het ontwikkeld xml schema wordt gebruikt om een native xml database te ontwikkelen en te implementeren met de Open Source product genaam eXist.
9. De transformatie en conversie van de verschillende data van Partner systemen moet nog geïnventariseerd, gedocumenteerd en ontwikkeld worden
10. Er moet nog een functioneel en technisch ontwerp van de Infrastructuur Services EmailIS en LoggingIS gedaan worden.
11. De synchronisatie tussen eXist, de Aggregator, en de VuFind, solr, databanken moet nog worden uitgeklaard.
12. De DataProviderBCS is de webservice die alle functionaliteit bevat om berichten van en naar Partnersystemen te transformeren en te verrijken.
13. Het verwerken van updates van niet OAI compliant Partners moet nog meer in detail beschreven worden. Er wordt gebruik gemaakt van de ontwikkelde webservices.

10.2 Gefaseerde implementatie

De voorgestelde oplossing voor het FELNET 2 ecosysteem op basis van open source applicaties en standaarden kent 3 subsystemen

1. Het portaal, bestaande uit
 - a. het Drupal Content Management Systeem en
 - b. de VuFind component om de FELNET data te visualiseren
 - c. de solrmarc tool voor synchronisatie van SOLR en Aggregator
 - d. SOLR onderhoud: optimalisering en indexering
2. FELNET beheer: functionaliteit voor de aggregator
 - a. Partnerbeheer
 - b. Gebruikerbeheer
 - c. Documentbeheer
 - i. Manueel
 - ii. Harvesting
3. Harvesting: implementatie van OAI-PMH
 - a. Service Provider bedrijfscomponent
 - i. Oogststrategiebeheer
 - ii. Oogstprocesbeheer
 - iii. Oogstaanvraagbeheer
 - b. Data Provider bedrijfscomponent
 - i. Ontvangen Oogstaanvragen
 - ii. Verwerken Oogstaanvragen
 - iii. Ontvangen Response
 - iv. Transformeren Response
 - v. Versturen Response
 1. Via email
 2. Via OAI-PMH

Het beschreven ecosysteem kent door het aantal subsystemen, open source applicaties en standaarden een hogere complexiteit tov de huidige FELNET toepassing. Tevens zijn gebruikte subsystemen en technologieën nieuw voor het FELNET onderhoudsteam, waardoor rekening gehouden moet worden met een steile leercurve.

Rekening houdend met het bovenstaande is ons voorstel om het FELNET 2 ecosysteem gefaseerd te implementeren.

1. Release 1 levert een werkend ecosysteem op: een nieuw portaal met een geïntegreerde VuFind visualisatiecomponent waarvan de SOLR databank van de manueel beheerd (CRUD) wordt me behulp van de FELNET beheer bedrijfscomponent. Met deze bedrijfscomponent kan de FELNET beheerder de ontvangen MARC-XML bestanden inlezen en in de eXist aggregator opladen. Indien alle ontvangen parterbestanden zijn opgeladen kan de verantwoordelijke de VuFind SOLR databank en de eXist databank synchroniseren.
2. Release 2 breidt het ecosysteem opgeleverd door Fase 1 uit door implementatie van het OAI-PMH. Hierdoor kan de verantwoordelijke oogstrategien definiëren die als doel hebben om de synchronisatie tussen Partners en FELNET volledig automatisch te laten verlopen.

10.3 Technische profielen voor implementatie.

Trasys stelt de volgende profielen voor voor de realisatie van de voorgestelde oplossing

1. Architect: staat borg voor de correcte vertaling en implementatie van de vereisten die in de functionele en technische analyse zijn beschreven. Hij/zij heeft ruime ervaring met de toepassing van design patterns in applicatie architectuur gebruikmakend van Java technologieën, Webservices en integratie.
2. Analist/programmeur: is een senior persoon die ruime ervaring heeft in de realisatie van gedistribueerde applicaties gebruik makend van Java technologieën en Webservices.

10.3.1 Release 1

FELNET 2 Release 1						
Activiteit	Taak	Architect	Analist/ Programmeur	Project beheer	Kwaliteit beheer	IMPLEMENTATIE TOTAAL
Implementatie		35,5	74,5	3,0	6,0	119,0
	Kennisneming Definitiestudie, Functionele analyse en Technische analyse	1,0	1,0			
	Installeren en testen Basiscomponenten					
	Kennisneming Basiscomponenten installatie en gebruik	2,0	2,0			
	Installeren VuFind					
	Testen/optimaliseren/indexeren	2,0	5,0			
	Gebruik SOLRMARC tool	1,0	5,0			
	Installeren eXist	0,5	0,5			
	Data Provider bedrijfscomponent				0,5	
	Ontwerpen/creëren van een Zoekschermbasis op basis van de documentspecificatie	0,5	2,0			
	Transformatie van de ingegeven zoekopdracht naar Partnerspecifiek SQL statement (25)	5,0	10,0			
	Transformatie van de Partner databankresponse naar MARC-XML (25)	5,0	10,0			
	Versturen van email met MARC-XML bestand	0,5	2,0			
	Service Provider bedrijfscomponent				0,5	
	Ontwerpen/creëren van een Document beheerschermbasis	0,5	2,0			
	Het opladen (aggregator) van ontvangen MARC-XML bestanden	2,0	5,0			
	Ontwerpen/creëren van een Document zoekschermbasis op basis van de document specificatie, voor het raadplegen van de aggregator	0,5	2,0			
	Gebruikerbeheer				0,5	
	Ontwerpen/creëren van een Zoekschermbasis op basis van de gebruikerspecificatie	0,5	2,0			
	Ontwerpen/creëren van een Beheerschermbasis op basis van de gebruikerspecificatie	0,5	2,0			
	Partnerbeheer				0,5	
	Ontwerpen/creëren van een Zoekschermbasis op basis van de partnerspecificatie	0,5	2,0			
	Ontwerpen/creëren van een Beheerschermbasis op basis van de partnerspecificatie	0,5	2,0			
	Webservices					
	Selecteer Messaging Model (SOAP, REST)	1,0			1,0	
	Gebruikerservice					
	Definieer Service operaties	1,0				
	Definieer voor elke Service operatie de Request en Response	1,0				
	Definieer de service interface (wsdl)	1,0				
	Definieer voor elke Service operatie Request en Response, het XML Schema Definition(xsd)	1,0				
	Partnerservice					
	Definieer Service operaties	1,0				
	Definieer voor elke Service operatie de Request en Response	1,0				
	Definieer de service interface (wsdl)	1,0				
	Definieer voor elke Service operatie Request en Response, het XML Schema Definition(xsd)	1,0				
	Documentservice					
	Definieer Service operaties	1,0				
	Definieer voor elke Service operatie de Request en Response	1,0				
	Definieer de service interface (wsdl)	1,0				
	Definieer voor elke Service operatie Request en Response, het XML Schema Definition(xsd)	1,0				
	Java ontwikkeling					
	Realiseer voor elke Service Operatie een Java implementatie		20,0			
	Validate			3,0	3,0	
	Unit testen					
	Component testen					
	Integratie testen					
	Gebruiker Acceptatie testen					

10.3.2 Release 2

FELNET 2 Release 2						
Activiteit	Taak	Architect	Analist/ Program	Projectbe heer	Kwaliteit beheer	IMPLEMENTATIE TOTAAL
Implementatie		25,5	68,0	4,0	7,5	105,0
	Kennisneming Definitiestudie, Functionele analyse en Technische analyse	1,0	1,0			
	Data Provider bedrijfscomponent				0,5	
	Ontwerpen/creëren van functionaliteit voor het ontvangen van een Oogstaanvraag	0,5	2,0			
	Aanpassen Zoekscherms op basis van de documentspecificatie	0,5	2,0			
	Aanpassen Transformatiefunctie	5,0	10,0			
	Aanpassen functionaliteit voor het de Response (MARC-XML bestand)	0,5	2,0			
	Service Provider bedrijfscomponent				0,5	
	Aanpassen Document beheerscherms	0,5	2,0			
	Aanpassen van de functionaliteit voor het opladen (aggregator) van ontvangen MARC-XML bestanden	2,0	5,0			
	Aanpassen Document zoekscherms, op basis van de document specificatie, voor het raadplegen van de aggregator	0,5	2,0			
	Oogststrategiebeheer				0,5	
	Ontwerpen/creëren van een Zoekscherms op basis van de Oogststrategiespecificatie	0,5	2,0			
	Ontwerpen/creëren van een Beheerscherms op basis van de Oogststrategiespecificatie	0,5	2,0			
	Oogstprocesbeheer				0,5	
	Ontwerpen/creëren van een Zoekscherms op basis van de Oogstprocesspecificatie	0,5	2,0			
	Ontwerpen/creëren van een Beheerscherms op basis van de Oogstprocesspecificatie	0,5	2,0			
	Oogstaanvraagbeheer				0,5	
	Ontwerpen/creëren van een Zoekscherms op basis van de Oogstaanvraagspecificatie	0,5	2,0			
	Ontwerpen/creëren van een Beheerscherms op basis van de Oogstaanvraagspecificatie	0,5	2,0			
	Webservices				1,0	
	Oogststrategieservice					
	Definieer Service operaties	1,0				
	Definieer voor elke Service operatie de Request en Response	1,0				
	Definieer de service interface (wsdl)	1,0				
	Definieer voor elke Service operatie Request en Response, het XML Schema Definition(xsd)	1,0				
	Oogstprocesservice					
	Definieer Service operaties	1,0				
	Definieer voor elke Service operatie de Request en Response	1,0				
	Definieer de service interface (wsdl)	1,0				
	Definieer voor elke Service operatie Request en Response, het XML Schema Definition(xsd)	1,0				
	Oogstaanvraagservice					
	Definieer Service operaties	1,0				
	Definieer voor elke Service operatie de Request en Response	1,0				
	Definieer de service interface (wsdl)	1,0				
	Definieer voor elke Service operatie Request en Response, het XML Schema Definition(xsd)	1,0				
	Java ontwikkeling			1,0	1,0	
	Realiseer voor elke Service Operatie een Java implementatie		20,0			
	Realiseer de OAI-PMH functionaliteit		10,0			
	Validatie			3,0	3,0	
	Unit testen					
	Component testen					
	Integratie testen					
	Gebruiker Acceptatie testen					

11. INFORMATIEBRONNEN

11.1 Referentiedocumenten

<u>Bestandsnaam</u>	<u>Onderwerp</u>	<u>Code</u>
P_DVP_7_naamgeving_dbs tructuur.doc	Procedure voor de naamgeving van de databankstructuur (18/04/2005).	[Ref. 1]
Procedure TA met PowerDesigner_v1.0.doc	Procedure technische analyse met PowerDesigner	[Ref. 2]
Felnet2_definitiestudie.doc	Definitie Studie	[Ref. 3]
FELNET2_FA_XXX_V3.0 0.doc	Functionele analyse	[Ref. 4]

11.2 Interviews

<u>Geïnterviewde(n)</u>	<u>Onderwerp</u>	<u>Datum</u>

12. BIJLAGEN

12.1 Technische beschrijving van de OAI-PMH implementatie

De aard en inhoud van de afspraak “Beschikbaar stellen en verzamelen van metadata” is besproken in de Functionele Analyse. In deze bijlage wordt de uitwisseling van metadata tussen de metadata verzamelende applicatie (harvester) en metadata aanbiedende repository (repository) op technisch niveau besproken.

De afspraak “Beschikbaar stellen en verzamelen van metadata” behelst het volgende:

- Er is communicatieverbinding tussen de harvester en de repository;
- De communicatie met de repository gebeurt volgens het OAI-PMH protocol;
- De repository levert de metadata volgens het formaat van content-zoekprofiel.

Eerst zullen de OAI-PMH requests en responses worden beschreven inclusief de algemene structuren, de argumenten die meegegeven kunnen worden met een verzoek en het antwoord wat de repository zou moeten geven. Het antwoord omvat tevens de antwoorden op fout geformuleerde requests of requests zonder een passend record (een record dat voldoet aan de specificaties van het verzoek). Vervolgens wordt ingegaan op enkele aanvullende zaken als het gebruik van het content-zoekprofiel, unieke identifiers, verwijdering van items in repositories en selectieve harvesting.

12.1.1 Gebruik

Deze bijlage is in eerste instantie een verduidelijking op het document “The Open Archives Initiative Protocol for Metadata Harvesting”, “Protocol Version 2.0 of 2002-06-14”, “Document Version 2004/10/12T15:31:00Z”. Dit document is verkrijgbaar via: <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>.

Het enige verschil hiermee is dat het FELNET-zoekprofiel als marcxml metadataformaat moet worden toegepast in aanvulling op het metadataformaat Dublin core.

Het protocol eist dat de repository minimaal Dublin Core als metadataformaat ondersteunt, d.w.z. metadata volgens Dublin Core kan aanleveren. Optioneel kunnen meerdere andere metadataformaten worden ondersteund. Binnen de afspraak is het content-zoekprofiel als verplicht metadataformaat toegevoegd aan de ondersteunde metadataformaten.

Deze technische beschrijving is bedoeld voor ontwikkelaars van zowel harvesters als repositories. Nadere informatie over het toepassen van dit protocol kan worden gevonden in het document “Implementation Guidelines”, verkrijgbaar via <http://www.openarchives.org/OAI/2.0/guidelines.htm>.

Specifiek voor de eerste doelgroep (ontwikkelaars van harvesters) is het document “Guidelines for Harvester implementers”, verkrijgbaar via <http://www.openarchives.org/OAI/2.0/guidelines-harvester.htm>. Voor de tweede doelgroep (ontwikkelaars van repositories) is er het document “Guidelines for Repository implementers”, verkrijgbaar via <http://www.openarchives.org/OAI/2.0/guidelines-repository.htm>.

12.1.2 OAI-PMH protocol

Het communicatie protocol OAI-PMH voorziet in een raamwerk voor de uitwisseling van gegevens op basis waarvan metadata kan worden verzameld. OAI-PMH is gebaseerd op de open standaarden HTTP (Hypertext Transport Protocol) en XML (Extensible Markup Language), en kan vergeleken worden met een nadere invulling van het SOAP protocol. HTTP heeft als nadeel dat het stateless is, en er daarom geen garantie is dat een verzoek (Request) wordt beantwoord of in welke toestand de ontvanger verkeert (verzoek ontvangen, verzoek begrepen, bezig te beantwoorden, etc.).

De harvester stuurt een verzoek (een zogenaamd “request”) aan een repository. Een repository antwoordt in een “response”. Het “request” van de harvester is één van de 6 verschillende type requests die kunnen worden ingediend. De response van de repository is in XML formaat. Hierna volgt een gedetailleerdere beschrijving van het request en de response.

12.1.2.1 Request

Het request wordt gedaan in de vorm van een HTTP request. Een repository moet zowel de HTTP GET als HTTP POST methode ondersteunen. POST heeft als voordeel dat er geen beperking aan de lengte van de argumenten stelt en dat de details van een request voor de gebruiker onzichtbaar gemaakt kunnen worden.

De opbouw van de request is als volgt: **url?verb=requesttype[&argument]**

De onderdelen van de request zijn:

- **url** is het URL van de repository,
- **verb** is het vaste sleutelwoord van OAI-PMH,
- **requesttype** geeft het type request aan (mogelijke waarden volgen hieronder),
- **argument** is het argument die behoort bij het request (kan meerdere keren voorkomen); ieder argument bestaat uit een argumentsleutel en een waarde, gescheiden door ‘=’.

Voorbeelden van een correct request zijn:

<http://pubs.es.uct.ac.za/perl/oai2?verb=Identify>

http://ir.lib.cbs.dk/oai.php?verb=GetRecord&Identifier=42&metadatataprefix=oai_lom

<http://srb2.bib.iteso.mx/oai/ite/default.aspx?verb=ListSets>

De mogelijke waarden die “verb” kan hebben als aanduiding van het requesttype worden hieronder genoemd. Let hierbij op de hoofdletters die aan het begin en in het requesttype overeenkomstig moeten worden gehanteerd. Nadere uitleg over de argumenten die kunnen worden meegegeven, de resultaten van de requests en voorbeelden worden in de volgende paragrafen uitgelegd.

Verb	Beschrijving
Identify	ophalen van informatie over de repository (Geef mij informatie over jou)
GetRecord	ophalen van metadata van één specifiek item in de repository (Geef mij de metadata van een bepaald item)
ListRecords	ophalen van een reeks records uit een repository (Geef mij de

	metadata van de gevraagde item). Optioneel kan ‘selectieve harvesting’ plaatsvinden op basis van metadataformaat, rubricering (sets) en wijzigingsdatum
ListIdentifiers	een variant van ListRecords, waarbij alleen de headers van de records opgehaald worden (Geef mij de identifiers van de gevraagde items). Optioneel kan ‘selectieve harvesting’ plaatsvinden op basis van metadataformaat, rubricering (sets) en wijzigingsdatum
ListMetadataFormats	het opvragen van de metadataformaten waarin een repository metadata kan aanleveren (Vertel mij welke metadataformaten je kent)
ListSets	het ophalen van set-structuren (aanduidingen van rubrieken) die gebruikt kunnen worden voor ‘selective harvesting’ (Vertel mij welke rubrieken je kent)

Tabel 12-1: OAI-PMH verbs

12.1.2.2 Response

De response van de repository is een XML-bericht zoals getoond in onderstaande codelisting:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
  http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-05-17T07:39:28Z</responseDate>
  <request verb="Identify">
    http://pubs.cs.uct.ac.za/perl/oai2
  </request>
  <Identify>
    <repositoryName>
      UCT CS Research Document Archive
    </repositoryName>
    <baseURL>http://pubs.cs.uct.ac.za/perl/oai2</baseURL>
    <protocolVersion >2.0</protocolVersion>
    <adminEmail>mailto:techrep@pubs.cs.uct.ac.za</adminEmail>
    <earliestDatestamp >0001-01-01</earliestDatestamp>
    <deletedRecord >persistent</deletedRecord>
    <granularity>YYYY-MM-DD</granularity>
  </Identify>
</OAI-PMH>
```

Dit XML-bericht is als volgt opgebouwd:

- Het eerste element is een XML declaratie, waarbij het attribuut ‘version’ altijd de waarde “1.0” heeft en encoding altijd “UTF-8”,
- De overige inhoud van het bericht wordt omvat door het container root element <OAI-PMH>,
- Het eerste deelelement <responseDate> van het root element <OAI-PMH> is de indicatie van de tijd en datum waarop het antwoord was verzonden,

- Het tweede deelelement <request> van het root element <OAI-PMH> is de indicatie van het request dat tot dit antwoord heeft geleid,
- Het derde deelelement van het root element <OAI-PMH> is het element <error> (in geval van fout of uitzondering) of het element met dezelfde naam als de ‘verb’ van het request.

De definitie van het gegevenselement <OAI-PMH>, met name de specificaties van de attributen en deelelementen, volgt verder in de volgende tabel (zie Tabel).

Alle responses van de repository moeten positief gevalideerd kunnen worden tegen het XSD bestand dat geleverd is door OAI, het bestand ‘OAI-PMH.xsd’. Dit XSD bestand is te vinden in <http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd>.

In de komende tabellen met betrekking tot de definitie van het gegevenselement <OAI-PMH> wordt per tabel een bepaald gegevenselement of deelelement gedefinieerd. Al deze tabellen hebben een vaste volgorde van kolommen:

- 1^e kolom ‘**Nr**’: het hiërarchische volgnummer. Wanneer het element op het laagste niveau een veld betreft dan eindigt dit nummer op een punt gevolgd door een cijfer (‘.1’ of ‘.2’) en wanneer dit een attribuut betreft dan eindigt dit nummer op een punt gevolgd door een letter (‘.a’ of ‘.b’).
- 2^e kolom ‘**Elementnaam**’: de naam van het betrokken gegevenselement uit de specificatie.
- 3^e kolom ‘**O/V**’: beschrijft of het gegevenselement binnen deze afspraak mag of moet voorkomen
 - ‘**O**’, voor een optioneel gegevenselement,
 - ‘**V**’, voor een verplicht gegevenselement.
- 4^e kolom ‘**Max**’: beschrijft of het gegevenselement binnen deze afspraak één of meerdere keren mag voorkomen:
 - ‘**1**’, voor een gegevenselement dat maximaal één keer mag voorkomen,
 - ‘**N**’, voor een gegevenselement dat meerdere keren mag voorkomen.
- 5^e kolom ‘**Elementtype**’: beschrijft het type van het gegevenselement; in deze kolom worden de volgende schrijfwijzen gebruikt:
 - ‘**zie <elementtype>**’, voor verwijzing naar een algemeen elementtype van het gegevenselement; bijvoorbeeld ‘zie <dateString>’, dit element is dan van aangegeven type. Alle mogelijke algemene elementtypen worden in een aparte tabel gedefinieerd (zie Tabel).
 - ‘**ATTRIBUUT**’ voor attribuut met bijbehorende gedefinieerde waardetype; voor het overzicht wordt deze rij tevens in italics weergegeven.
‘**CONTAINER**’, voor verzameling van velden; voor het overzicht wordt deze rij tevens grijs gekleurd.
 - ‘**String**’, voor waardeveld met als waarde een string datatype dat reeks van karakters in XML weergeeft (zie <http://www.w3.org/TR/xmlschema-2/#string>).
‘**Vaste waarde**’, voor waardeveld waarvan de waarde in de huidige afspraak

vast is. Is de waarde kort dan volgt deze direct, is de waarde lang dan volgt deze in de kolom ‘Opmerking / toelichting’.

- ‘**Enumeratie**’, voor het waardeveld waarvan de waarde afkomstig is uit een vooraf vastgestelde lijst met namen of codes. Is de lijst kort dan volgt deze direct, is de lijst lang dan wordt deze in de kolom ‘Opmerking / toelichting’ opgesomd. Bij complexe enumeraties wordt de lijst met bijbehorende betekenissen in een afzonderlijke tabel vermeld; de verwijzing naar de tabel volgt eveneens in de kolom ‘Opmerking / toelichting’.

- 6^e kolom ‘**Opmerking / toelichting**’ wordt een opmerking of toelichting op het element gegeven.

#	Elementnaam	O/V	Max	Elementtype	Opmerking/toelichting
1	OAI-PMH	V	1	CONTAINER	Bevat exact 1 van de deelelementen 1.3 t/m 1.9
1.a	xmlns	V	1	ATTRIBUUT Vaste waarde	De namespace URI van het OAI-PMH. De waarde is altijd <i>"http://www.openarchives.org/OAI/2.0/"</i>
1.b	xmlns: xsi	V	1	ATTRIBUUT Vaste waarde	De namespace URI van het XML schema De waarde is altijd <i>"http://www.w3.org/2001/XMLSchema-instance"</i>
1.c	xsi: schema location	V	1	ATTRIBUUT Vaste waarde	Een paar van namespace URI en URL van het XML schema, gescheiden door een spatie. De waarde is altijd <i>"http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd"</i>
1.1	responseDate	V	1	Zie <dateTimeString>	Het datum tijdstip van de response
1.2	request	V	1	Zie <uriString>	De indicatie van het request dat tot dit antwoord heeft geleid
1.2.a	verb	O	1	ATTRIBUUT enumeratie	Aanduiding van het request. Mogelijke waarden <i>"Identify"</i> , <i>"ListMetadataFormats"</i> , <i>"ListSets"</i> , <i>"GetRecord"</i> , <i>"ListIdentifiers"</i> en <i>"ListRecords"</i>
1.2.b	identifier	O	1	ATTRIBUUT zie <idString>	De unieke identifier die een record in de repository identificeert
1.2.c	metadataPrefix	O	1	ATTRIBUUT zie <prefixString>	Indicatie welk type metadata bij het record moet worden gevoegd
1.2.d	from	O	1	ATTRIBUUT zie <dateString>	Deze datum of dit datumtijdstip geeft de ondergrens aan van de laatste wijziging van de te ontvangen records (t.b.v.

					selectieve harvesting).
1.2.e	until	O	1	ATTRIBUUT zie <dateString>	Deze datum of dit datumtijdstip geeft de bovengrens aan van de laatste wijziging van de te ontvangen records (t.b.v. selectieve harvesting).
1.2.f	set	O	1	ATTRIBUUT String	<p>Een setSpec waarde. Deze waarde maakt het mogelijk data (records) op te vragen binnen een bepaalde set (rubriek) t.b.v. selectieve harvesting.</p> <p>Een setSpec waarde is een door dubbele punt ‘:’ gescheiden lijst van URI reserved3 karakters waarin geen dubbele punt voorkomt. Het setSpec element moet uniek zijn binnen een gegevensverzameling. Het is handig hiervan een korte benaming te maken gezien het feit dat dit een onderdeel is van de OAI-identificer.</p>
1.2.g	resumptionToken	O	1	ATTRIBUUT String	Hiermee kunnen lange lijsten ondersteund worden, het zogenaamde “flow control” ⁴ . De repository kan een lijst van gegevens gedeeltelijk teruggeven. De aanvrager kan dan een verzoek (request) doen naar de rest van de lijst. Dit kan nuttig zijn als de opgevraagde lijsten bijvoorbeeld erg groot zijn. De stringwaarde van dit element moet worden gebruikt bij het request naar het vervolg van de lijst. De inhoud en betekenis van deze stringwaarde is geheel voor rekening van de repository en bevat dus een reeks karakters die onderscheidend zijn voor de repository.
1.3	response			CONTAINER	Wordt verder gedefinieerd de volgende paragrafen. De naam van dit element is één van de volgende tags: Identify, GetRecord, ListRecords, ListIdentifiers, ListMetadataFormats, ListSets of error.

Tabel 12-2: definitie van gegevenselement <OAI-PMH>

In onderstaande tabel (Tabel 12-3) staat de definitie van alle elementtypen die bij de typering van diverse elementen in de gegevensstructuur op diverse plaatsen worden gebruikt.

Elementnaam	Type (en voorbeeld)	Opmerking/toelichting
dateString	String volgens formaat "YYYY-MM-DD" of "YYYY-MM-DDThh:mm:ssZ" (b.v. "2006-03-08" of "2006-03-08T08:25:00Z")	Bevat een UTC tijdsnotatie volgens ISO8601; het is een datum of datumtijdstip in uren, minuten en seconden voor zulu tijd ('Z'). Dus let op, wanneer uren, minuten en seconden worden gebruikt dan moet de indicator 'Z' worden gebruikt. De keuze datum of datumtijdstip in een request moet overeenkomen met de 'granularity' van de repository.
datetimeString	String volgens formaat "YYYY-MM-DDThh:mm:ssZ" (b.v. "2006-03-08T08:25:00Z")	Bevat een UTC tijdsnotatie. Het is een datumtijdstip in datum en tijdstip in uren, minuten en seconden voor zulu tijd ('Z'). Komt alleen voor bij het tijdstip van response (responseDate) en
emailString	String volgens formaat "(string)@(string).(string)" (b.v. "somebody@organisation.nl")	Bevat een string dat een geldig e-mailadres is. Komt alleen voor bij specificatie van het e-mailadres van de administrator van de repository bij de Identify request.
idString	String volgens formaat "oai:(namespace-id):(local-id)" zie <uriString> (b.v. "oai:foo.org:some-local-id-53" of "oai:kennisnet.org:afd-oi/0601001")	Is een unieke identifier welke foutloos een item in de repository identificeert. Het formaat van deze string voldoet aan dat van een URI en de oai-identifier syntax (zie paragraaf Unieke identifier).
prefixString	String van karakters uit de volgende reeks 'A-Za-z0-9-._!~*()' (b.v. "oai_dc")	Is een unieke identifier ter identificatie van het metadataformaat.
uriString	String volgens formaat "(scheme):(path)" (b.v. "http://xxx.yyy.zzz/oai")	Een URI (Uniform Resource Identifier) is een string volgens een bepaald formaat voor de identificatie van een abstracte of fysieke resource.

Tabel 12-3: definitie van algemene gegevenselementen

Enkele belangrijke begrippen hierbij zijn:

- **Resource:** het document zelf, waarover de metadata beschrijving in de repository is opgenomen.
- **Item:** de metadata beschrijving van een document in een repository. Deze beschrijving mag binnen een repository in elk formaat gedaan worden zolang de aangeleverde metadataformaten voldoen.

- **Record:** presentatie van de metadata beschrijving in de response aan de harvester, volgens een afgesproken XML-schema (b.v. Dublin core of marcxml).
- **Set** Is de indeling van items in hiërarchische klassen (rubrieken) door de repository. Dit kan worden gebruikt om te zoeken binnen een bepaalde rubriek.
- **setSpec** Is de aanduiding van een bepaalde klasse (rubriek) in de hiërarchische structuur van klassen middels het pad. Opsomming van alle bovenliggende klassen gescheiden door dubbele punt ‘:’.
- **Selectieve harvesting** Er wordt gesproken over selectieve harvesting indien een harvester metadata van slechts een deel van de totale metadagegevens ophaalt. De selectie vindt plaats op basis van metadata schema, wijzigingsdatum en/of rubricering (set) van de metadata in de repository.

In de volgende paragrafen wordt voor de eerder genoemde request ListRecords het bijbehorende response behandeld. Er wordt een korte uitleg gegeven van de benodigde response door de repository.

Verder worden de argumenten die meegegeven kunnen worden aan het request opgesomd en uitgelegd. Daarna wordt uitgewerkt welke velden een response (dus een XML-record) kan en moet bevatten en wordt een voorbeeld gegeven van een mogelijke response.

De overige responses worden niet in detail beschreven. Voor een gedetailleerde beschrijving verwijs ik naar <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>.

12.1.2.3 ListRecords

In de response op het ListRecords request worden records opgestuurd. Om de beschrijving van een record goed te volgen zijn de begrippen resource, item en record van belang.

12.1.2.3.1 Argumenten van ListRecords request

Het ListRecords request heeft de volgende argumenten

Argument sleutel	O/V	Max	Waarde	Opmerking/toelichting
verb	V	1	“ListRecords”	
from	O	1	Zie <dateString>	Deze datum of dit datums tijdstip geeft de ondergrens aan van de laatste wijziging van de te ontvangen records (t.b.v. selectieve harvesting).
until	O	1	Zie <dateString>	Deze datum of dit datums tijdstip geeft de bovengrens aan van de laatste wijziging van de te ontvangen records (t.b.v. selectieve harvesting).
set	O	1	String	een setSpec (rubriek) waarde. Dit argument maakt het mogelijk

				items te zoeken binnen een bepaalde rubriek t.b.v. selectieve harvesting
metadataPrefix	V	1	Zie <prefixString>	geeft aan welk type metadata bij het record moet worden gevoegd.

Tabel 12-4: definitie van argumenten van ListRecords request

Hier een voorbeeld van de invulling van het veld <dateStamp>.

```
(...)
<header>
<identificier>oai:xxx.yyy.zzz:abc_res001</identificier>
<datestamp>2010-02-28</datestamp>
</header>
(...)
```

12.1.2.3.2 Response op ListRecords request

De response van het ListRecords request is een XML-bericht met het gegevens-element <OAI-PMH> volgens de algemene structuur in Tabel 12-2. Hierbij is het container-element 1.3 <ListRecords>, een aantal container-elementen van het type <record> gevolgd door een aantal velden van het type <resumptionToken> zoals in de volgende tabel Tabel 12-5 gedefinieerd.

#	Elementnaam	O/V	Max	Elementtype	Opmerking/toelichting
1.3	ListRecords	O	1	CONTAINER	
1.3.1	Record	O	N	CONTAINER	metadata in één specifiek formaat van een resource. Deze metadata staat in deelelement 1.3.1.2.
1.3.1.1	Header	V	1	CONTAINER	bevat de unieke identificatie en de eigenschappen voor selectieve harvesting.
1.3.1.1.a	status	O	1	ATTRIBUUT vaste waarde	<i>Aanduiding van verwijdering van het record</i>
1.3.1.1.1	identificier	V	1	Zie <idString>	De unieke identificier – de unieke identiteit van een item in een repository.
1.3.1.1.2	datestamp	V	1	Zie <dateString>	Datum (of datumtijdstip) van laatste wijziging van het item.
1.3.1.1.3	setSpec	O	N	String	0, 1 of meer setSpec elementen van alle sets

					(rubrieken) waartoe het item behoort worden weergegeven.
1.3.1.2	metadata	O	N	CONTAINER met vrije invullingen van elementen	Deze container wordt gebruikt om de metadata volgens het marcxml te leveren. In deze container kunnen volgens het protocol vrij XML tags worden opgenomen; welke XML tags (containers en velden) dit mogen/moeten zijn, is volgens marcxml beschreven
1.3.1.3	about	O	N	CONTAINER	Container met aanvullende gegevens over de metadata van het record
1.3.2	resumptionToken	O	1	String	Hiermee kan een lange gegevenslijst worden geregeld over verschillende requests
1.3.2.a	expirationDate	O	1	ATTRIBUUT zie <dateString>	geeft het datumtijdstip aan wanneer de resumtiontoken verlopen is.
1.3.2.b	completeListSize	O	1	ATTRIBUUT integer	Indicatie van het aantal van de totale lijst; dit aantal kan variëren gedurende de requests.

□ Tabel 12-5: definitie van <ListRecords> binnen <OAI-PMH> bij ListRecords request

Een request voorbeeld:

http://xxx.yyy.zzz/oai?verb=ListRecords&from=2003-01-01&set=physics.hep&metadata=oai_felnet

Een response voorbeeld

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:oai_felnet
"http://www.imsi-global.org/xsd/imsmd_rootv1p2p1"
xsi:schemaLocation="##other
http://download.felnet.eu/xsd/fndocument.xsd
```

```

http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2010-02-08T08:55:46Z</responseDate>
<request verb="ListRecords" from="2003-01-01"
  metadataPrefix="oai_felnet ">
  http://xxx.yyy.zzz/oai/
</request>
<ListRecords>
  <record>
    <header>
      <identifiser> oai:xxx.yyy.zzz:0001</identifiser>
      <datestamp>2010-02-14</datestamp>
    </header>
    <metadata>
      <oai_felnet:felnet>
        Hier komt de invulling van het marcxml record.
      </oai_felnet:felnet>
    </metadata>
  </record>
  Hier komt de rest van de records.
</ListRecords>
</OAI-PMH>

```

12.1.2.4 Antwoord op foute requests

Het OAI-PMH protocol voorziet ook in foutafhandeling wanneer er fouten optreden in de communicatie of wanneer geen gepaste records zijn gevonden. In dit geval is element 1.3 ‘(response)’ uit Tabel 12-2 het element <error> dat wordt gedefinieerd in de volgende tabel.

Waarde van code	Verbs van toepassing	Opmerking/toelichting
“badArgument”	Allen	Het request bevat illegale argumenten, verplichte argumenten ontbreken, bevat dubbele argumenten of de waarde van argumenten hebben een illegale syntax.
“badResumptionToken”	ListIdentifiers, ListRecords, ListSets	De waarde van het argument ‘resumptionToken’ is niet valide of verlopen.
“badVerb”	n.v.t.	De waarde van het argument ‘verb’ is niet een valide OAI-PMH verb, het argument ontbreekt of het argument wordt herhaald.
“cannotDisseminateFormat”	GetRecord, ListIdentifiers, ListRecords	Het metadataformaat zoals aangegeven in het argument ‘metadataPrefix’ wordt niet ondersteund door het item of de repository.
“idDoesNotExist”	GetRecord, ListMetadataFormats	De waarde van het argument ‘identifiser wordt niet herkend

		of illegaal in de repository.
“noMetadataFormats”	ListMetadataFormats	Er zijn geen metadata formaten beschikbaar voor het gespecificeerde item
“noRecordsMatch”	ListIdentifiers, ListRecords	De combinatie van waarden van de argumenten ‘from’, ‘until’, ‘set’ en ‘metadataPrefix’ resulteren in een lege lijst.
“noSetHierarchy”	ListSets, ListIdentifiers, ListRecords	De repository ondersteunt geen sets.

Tabel 12-6: definitie van vocabulaire voor “code” binnen <error>

Een voorbeeld van een foute request is: *http://xxx.yyy.zzz/oai?verb=fouteVerb*

Het antwoord van de repository kan er als volgt uitzien:

```
<?xml version="1.0" encoding="UTF-8"?>
  <OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
      http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
    <responseDate>2012-05-01T09:18:29Z</responseDate>
    <request>http://arXiv.org/oai2</request>
    <error code="badVerb">Illegal OAI verb</error>
  </OAI-PMH>
```

12.1.3 Aandachtspunten

In deze paragraaf wordt ingegaan op verdere afspraken en aandachtspunten relevant voor OAI-PMH implementatie.

12.1.3.1 Gebruik van marcxml binnen OAI-PMH

De metadata moet volgens de felnet afspraak op zijn minst conform de worden uitgeleverd maar marcxml is geprefereerd.

Metadata wordt alleen uitgeleverd bij de aanroep met de verbs “GetRecord” en Listrecords. Als prefix bij het ophalen van data volgens de felnet afspraak dient “oai_felnet” te worden gebruikt. Een voorbeeld van de juiste aanroep is als volgt (let hierbij op het laatste argument ‘metadataprefix’):

```
http://xxx.yyy.zzz/oai?verb=GetRecord&Identifier=oai:xxx.yyy.zzz:cs/0112017&metadataprefix=oai_felnet
```

12.1.3.2 Unieke identifier

Voor harvesting met OAI-PMH moet elk item (de beschrijving van het document tabase) een unieke identifier hebben. Deze unieke identifier betreft de identificatie van het record binnen de OAI-PMH omgeving Deze identifier moet gedurende de hele levenscyclus van het item hetzelfde blijven; het moet altijd naar hetzelfde item blijven verwijzen. Deze identifier wordt gebruikt als argument bij het GetRecord en ListMetadataFormats request en in de headers

binnen een response. Het betreft hier de response op de GetRecord, ListRecords of ListIdentifiers request. Omdat items verschillende typen metadata kunnen bevatten, kan dezelfde identifier in verschillende records voorkomen.

Als formaat wordt de oai-identifier gebruikt volgens de richtlijnen⁹. Hierbij voldoet een identifier aan de specificaties van een URI als de oai-identifier syntax. Deze syntax is als volgt:

“oai:(namespace-id):(local-id)”

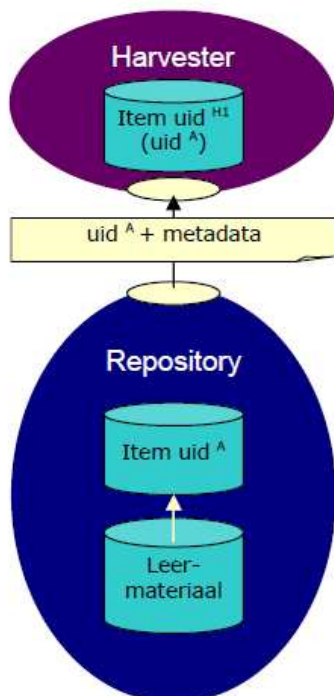
waarbij voor (namespace-id) het betrokken domein (b.v. baseURL in Identify reponse) en voor (local-id) een binnen dit domein unieke code kan worden gebruikt. Deze unieke code zal gedurende de hele levenscyclus van at item gehandhaafd blijven. Voor deze code kan bijvoorbeeld het database-id van het item in de repository worden gebruikt.

Voorbeelden zijn “oai:foo.org:some-local-id-53” of “oai:davindi.nl:0601901”

Een repository moet dus elk item een oai-identifier kunnen geven. Op basis van de unieke identifier kan de harvester bepalen of aangeleverde gegevens nieuwe leermiddelen of een update van leermiddelen betreft. Voor meer informatie over de oai-identifiers zie het betrokken document “Implementation guidelines. Specification and XML schema for the OAI identifier format”

Let op

De unique identifier betreft dus niet de identiteit van het leermiddel zelf (veld 1.1 Identifier van IEEE-LOM): dat veld is onderdeel van de metadata en valt verder buiten de beschouwing van OAI-PMH. Voor de unique identifier wordt de ‘oai-identifier’-syntax gebruikt zoals die in OAI-PMH is gedefinieerd. Voordeel hiervan is dat elke repository zelfstandig unique identifiers kan bepalen.



De aanbieder beschrijft een document deze beschrijving vast als item in de repository. Het item krijgt als unique identifier (uid) de code A. Bij harvesting wordt een record aangemaakt waarin de metadata zijn opgenomen en de unique identifier van item A (uid_A). De harvester slaat het record op als eigen item met een eigen interne unique identifier (H1), en een verwijzing naar de identifier van de aanbieder (A).

Figuur 12-1: Een item in repository en harvester.

Een eenvoudig voorbeeld:

Stel, er is een centrale verwijzingsbank met daarbij een harvester die data ophaalt bij een tweetal aanbieders, aanbieder 1 en aanbieder 2. De URL naar de verwijzingsbank is <http://centraleverwijzingsbank.data.be/oai>.

Bij de aanbieders is de volgende, lokaal uniek genummerde metadata aanwezig:

Aanbieder	metadatarrecord
Aanbieder1	metadatarrecord1
	metadatarrecord2
Aanbieder2	metadatarrecord1
	metadatarrecord2

In de centrale verwijzingsbank is de volgende metadata aanwezig na harvesten:

1. <http://aanbieder1.aanbiedingen.be/oai> metadatarrecord1
2. <http://aanbieder1.aanbiedingen.be/oai> metadatarrecord2
3. <http://aanbieder2.aanbiedingen.be/oai> metadatarrecord1
4. <http://aanbieder2.aanbiedingen.be/oai> metadatarrecord2

Nu kunnen unieke records opgevraagd worden doordat de centrale verwijzingsbank steeds een unieke identifier krijgt die verwijst naar het unieke item van de aanbieder. De harvester kan nu de data op de volgende wijze opvragen bij de aanbieders:

- http://aanbieder1.aanbiedingen.be/oai?verb=getRecord&Identifier=oai:aanbieder1.be:1&metadataprefix=oai_felnet

Het resultaat van deze vraag is metadatarrecord 1 van aanbieder 1.

- http://aanbieder2.aanbiedingen.be/oai?verb=getRecord&Identifier=oai:aanbieder2.be:1&metadataprefix=oai_felnet

Het resultaat van deze vraag is metadatarrecord 1 van aanbieder 2.

Bij de harvester kan metadata opgevraagd worden op de volgende manier:

- http://centraleverwijzingsbank.data.be/oai?verb=getRecord&Identifier=oai:centraleverwijzingsbank.be:3&metadataprefix=oai_felnet

Het resultaat van deze vraag is metadatarrecord 1 van aanbieder 2.

De uiteindelijke vorm van de unieke identifier is zeer bepalend voor communicatie met de centrale repository. Daarom dient deze vorm in overleg met de centrale partij tot stand te komen.

12.1.3.3 Verwijdering van items in repositories

Binnen OAI-PMH is gedefinieerd dat een repository kan aangeven dat een item is verwijderd. De harvester weet dan dat het item uit de eigen database kan worden verwijderd. Indien

mogelijk worden repositories geadviseerd om verwijderingen aan te geven, dit vergemakkelijkt namelijk het onderhoud en dus de betrouwbaarheid van de metadata verzameling bij de harvester.

Indien een repository verwijdering niet op itemniveau kan aangeven moet de harvester periodiek de headers van alle items ophalen om te kunnen bepalen welke items niet meer in de repository aanwezig zijn.

12.1.3.4 Flow Control

Is een principe voor het gedeeltelijke aanleveren van lange lijsten als zoekresultaat bij de zoekopdrachten ListRecords, ListIdentifiers en ListSets. Bij FELNET zal de FELNET beheerder hiermee geconfronteerd worden bij de eerste harvesting activiteiten. De FELNET databank is immers (nog) leeg en alle items uit de partner repository zullen moeten worden geoogst.

De repository kan een lange lijst van gegevens gedeeltelijk teruggeven tezamen met een zogenaamd ‘resumptionToken’. De aanvrager kan dan een verzoek (request) doen naar de rest van de lijst met als parameter het resumptionToken. Nadere informatie is verkrijgbaar op <http://www.openarchives.org/OAI/openarchivesprotocol.html#FlowControl>.

De specificatie van het OAI-PMH protocol specificeert niet de syntax of implementatie strategie van het resumptionToken. Dit is geheel de verantwoordelijkheid van de repository en hierdoor heeft de repository maximale flexibiliteit voor het eigen mechanisme.

Bijvoorbeeld, een repository kan ervoor kiezen om “voor het resumptionToken een goed leesbare syntax te gebruiken (“resultSet=157&nextRange=1001-2000”) of een onleesbare syntax (“9023A210CD007”).

12.2 Bedrijfsregels

Zie [Conceptual Report.html](#)

12.3 Domains

Zie [Conceptual Report.html](#)

12.4 Foutcodes en foutboodschappen

Zie [Conceptual Report.html](#)

12.5 Tabellen en velden vanuit PowerDesigner

Zie [Conceptual Report.html](#)

12.5.1 Conversie

12.5.2 Transitie

12.6 Data voor <Referentietabel 1>

12.7 Data voor <Referentietabel 2>