



Ministerie van Binnenlandse Zaken en  
Koninkrijksrelaties

## Software Architectuur Document Migratiesoftware BRP

Versie 1.0

Datum	29 maart 2017
Status	Definitief



## Inhoud

<b>1</b>	<b>INLEIDING .....</b>	<b>10</b>
1.1	SCOPE .....	10
1.1.1	<i>Wat is softwarearchitectuur?</i> .....	10
1.1.2	<i>Elementen en relaties</i> .....	10
1.1.3	<i>Verschillende views</i> .....	10
1.1.4	<i>Overige technische documentatie</i> .....	10
1.2	LEESWIJZER .....	11
1.3	OVERZICHT VAN DE VIEWS .....	12
1.4	BELANGHEBBENDEN EN VIEWS .....	12
<b>2</b>	<b>WIJZE WAAROP EEN VIEW IS GEDOCUMENTEERD.....</b>	<b>13</b>
2.1	TEMPLATE VOOR EEN VIEW .....	13
2.1.1	<i>Deel 1. Primaire presentatie</i> .....	13
2.1.2	<i>Deel 2. Element catalogus</i> .....	13
2.1.3	<i>Deel 3. Context diagram</i> .....	13
2.1.4	<i>Deel 4. Variabiliteit</i> .....	13
2.1.5	<i>Deel 5. Motivatie</i> .....	14
2.1.6	<i>Deel 6. Interfaces</i> .....	14
2.2	TEMPLATE VOOR INTERFACE DOCUMENTATIE .....	14
2.2.1	<i>Interface identiteit</i> .....	14
2.2.2	<i>Resources</i> .....	14
2.2.3	<i>Data types en constanten</i> .....	15
2.2.4	<i>Foutafhandeling</i> .....	15
2.2.5	<i>Variabiliteit</i> .....	15
2.2.6	<i>Kwaliteitsattributen</i> .....	15
2.2.7	<i>Motivatie</i> .....	15
2.2.8	<i>Gebruikshandleiding</i> .....	15
<b>3</b>	<b>SYSTEEM OVERZICHT.....</b>	<b>16</b>
3.1	INITIËLE VULLING .....	16
3.2	INTERSTELSELCOMMUNICATIE(ISC) & SYNCHRONISATIE .....	16
3.3	GGO VIEWER .....	17
<b>4</b>	<b>VIEWS.....</b>	<b>18</b>
4.1	LOGICAL VIEW .....	18
4.1.1	<i>Omschrijving</i> .....	18
4.1.2	<i>View-delen overzicht</i> .....	18
4.1.3	<i>View-deel 1: Migratiesoftware</i> .....	19
4.1.3.1	<i>Primaire presentatie</i> .....	19
4.1.3.2	<i>Element catalogus</i> .....	19
4.1.3.3	<i>Context diagram</i> .....	22
4.1.3.4	<i>Variabiliteit</i> .....	22
4.1.3.5	<i>Motivatie</i> .....	23
4.1.3.6	<i>Interfaces</i> .....	23
4.1.4	<i>View-deel 2: Initiële vulling</i> .....	24
4.1.4.1	<i>Primaire presentatie</i> .....	24
4.1.4.2	<i>Element catalogus</i> .....	24
4.1.4.3	<i>Context diagram</i> .....	25
4.1.4.4	<i>Variabiliteit</i> .....	25
4.1.4.5	<i>Motivatie</i> .....	25
4.1.4.6	<i>Interfaces</i> .....	26
4.1.5	<i>View-deel 3: Conversie</i> .....	27

4.1.5.1	Primaire presentatie .....	27
4.1.5.2	Element catalogus .....	27
4.1.5.3	Context diagram.....	28
4.1.5.4	Variabiliteit.....	28
4.1.5.5	Motivatie.....	28
4.1.5.6	Interfaces .....	28
4.1.6	<i>View-deel 4: BRP Synchronisatie Service</i> .....	35
4.1.6.1	Primaire presentatie .....	35
4.1.6.2	Element catalogus .....	35
4.1.6.3	Context diagram.....	36
4.1.6.4	Variabiliteit.....	36
4.1.6.5	Motivatie.....	36
4.1.6.6	Interfaces .....	36
4.1.7	<i>View-deel 5: Verzend &amp; Ontvangst Interstelsel Communicatie (VOISC)</i> .....	41
4.1.7.1	Primaire presentatie .....	41
4.1.7.2	Element catalogus .....	41
4.1.7.3	Context diagram.....	42
4.1.7.4	Variabiliteit.....	42
4.1.7.5	Motivatie.....	42
4.1.7.6	Interfaces .....	42
4.1.8	<i>View-deel 6: Interstelsel Communicatie (ISC)</i> .....	43
4.1.8.1	Primaire presentatie .....	43
4.1.8.2	Element catalogus .....	43
4.1.8.3	Context diagram.....	44
4.1.8.4	Variabiliteit.....	44
4.1.8.5	Motivatie.....	44
4.1.8.6	Interfaces .....	45
4.2	IMPLEMENTATION VIEW .....	46
4.2.1	<i>Omschrijving</i> .....	46
4.2.2	<i>Primaire presentatie</i> .....	46
4.2.3	<i>Element catalogus</i> .....	46
4.2.4	<i>Context diagram</i> .....	47
4.2.5	<i>Variabiliteit</i> .....	47
4.2.6	<i>Motivatie</i> .....	47
4.2.6.1	Werkwijze.....	47
4.2.6.2	Ontwikkelomgeving .....	48
4.2.6.3	Maven structuur .....	48
4.2.6.4	Werkverdeling .....	49
4.2.6.5	Lagen in de architectuur .....	49
4.2.6.6	Code richtlijnen .....	51
4.2.6.7	Ontwikkelaarshandleiding .....	51
4.2.7	<i>Interfaces</i> .....	51
4.3	PROCES VIEW .....	52
4.3.1	<i>Omschrijving</i> .....	52
4.3.2	<i>View-delen overzicht</i> .....	52
4.3.3	<i>View-deel 1: Initiële vulling</i> .....	53
4.3.3.1	Primaire presentatie .....	53
4.3.3.2	Element catalogus .....	53
4.3.3.3	Context diagram.....	58
4.3.3.4	Variabiliteit.....	59
4.3.3.5	Motivatie.....	59
4.3.3.6	Interfaces .....	59
4.3.4	<i>View-deel 2: Interstelsel Communicatie (ISC) &amp; Synchronisatie</i> .....	66
4.3.4.1	Primaire presentatie .....	66
4.3.4.2	Element catalogus .....	66

4.3.4.3	Context diagram.....	70
4.3.4.4	Variabiliteit.....	70
4.3.4.5	Motivatie.....	71
4.3.4.6	Interfaces .....	72
4.4	DEPLOYMENT VIEW.....	84
4.4.1	<i>Omschrijving</i> .....	84
4.4.2	<i>View-delen overzicht</i> .....	84
4.4.3	<i>View-deel 1: Initiële vulling</i> .....	85
4.4.3.1	Primaire presentatie .....	85
4.4.3.2	Element catalogus .....	85
4.4.3.3	Context diagram.....	86
4.4.3.4	Variabiliteit.....	86
4.4.3.5	Motivatie.....	87
4.4.3.6	Interfaces .....	87
4.4.4	<i>View-deel 2: Interstelsel communicatie (ISC) &amp; Synchronisatie</i> .....	88
4.4.4.1	Primaire presentatie .....	88
4.4.4.2	Element catalogus .....	88
4.4.4.3	Context diagram.....	89
4.4.4.4	Variabiliteit.....	89
4.4.4.5	Motivatie.....	89
4.4.4.6	Interfaces .....	90
<b>5</b>	<b>RELATIE TUSSEN VIEWS .....</b>	<b>91</b>
5.1	PROCES VIEW $\leftrightarrow$ LOGICAL VIEW .....	91
5.2	PROCES VIEW $\leftrightarrow$ DEPLOYMENT VIEW .....	92
<b>6</b>	<b>MOTIVERING .....</b>	<b>93</b>
6.1	ONTWERPBESLISSINGEN .....	93
6.2	RELEVANTE USE CASES.....	94
6.2.1	<i>Use Case 101: Vul persoonsgegevens in BRP initieel</i> .....	94
6.2.2	<i>Use Case 202: Verwerken synchronisatie-cyclus</i> .....	96
<b>7</b>	<b>WOORDENLIJST .....</b>	<b>98</b>
<b>8</b>	<b>AFKORTINGEN.....</b>	<b>99</b>
<b>9</b>	<b>REFERENTIES.....</b>	<b>100</b>
<b>10</b>	<b>APPENDIX A: LOCKING AANPAK IN VERSIE 2.2 VAN DE BRP .....</b>	<b>101</b>
10.1	INLEIDING.....	101
10.2	DOELSTELLING .....	101
10.3	VRAGEN EN AFBAKENING .....	101
10.4	LOCKING PROBLEMATIEK .....	101
10.5	BOP STAP 2.2 .....	103
10.6	CONCLUSIE .....	103
10.7	AANBEVELINGEN .....	103
<b>11</b>	<b>APPENDIX B: NFR KRUISTABEL .....</b>	<b>105</b>
11.1	EISEN TEN AANZIEN VAN PRESTATIE-EFFICIËNTIE.....	105
11.2	EISEN TEN AANZIEN VAN OVERDRAAGBAARHEID .....	105
11.3	EISEN TEN AANZIEN VAN ONDERHOUDBAARHEID .....	105
11.4	EISEN TEN AANZIEN VAN DOCUMENTATIE .....	106
11.5	EISEN TEN AANZIEN VAN BEVEILIGBAARHEID .....	108
11.6	EISEN TEN AANZIEN VAN BETROUWBAARHEID .....	110
<b>12</b>	<b>APPENDIX C: LOGGEN IN DE MIGRATIEVOORZIENING.....</b>	<b>111</b>

12.1	GESTRUCTUREERDE LOGGING .....	111
12.2	EVENT LOGGING .....	111
12.2.1	<i>Functionele event logging</i> .....	111
12.2.2	<i>Technische event logging</i> .....	111
12.2.3	<i>Logging niveaus</i> .....	111
12.3	LOGGEN OP EXTERN SYSTEEM .....	112
12.4	ACTIEVE MELDINGEN.....	112
<b>13</b>	<b>APPENDIX D: MONITORING.....</b>	<b>113</b>
13.1	JAVA MANAGEMENT EXTENSIONS.....	113
13.2	COMMUNICATIE PROTOCOL .....	113
<b>14</b>	<b>APPENDIX E: TOELICHTING BIJ OWASP TOP TIEN .....</b>	<b>114</b>

## Lijst met afbeeldingen

Afbeelding 1 Context Diagram Migratiesoftware BRP .....	16
Afbeelding 2 Module View Migratiesoftware .....	19
Afbeelding 3 Module View Initiële vulling .....	24
Afbeelding 4 Module View Conversie .....	27
Afbeelding 5 Module View Synchronisatie .....	35
Afbeelding 6 Module View VOISC .....	41
Afbeelding 7 Module View ISC .....	43
Afbeelding 8 Werkverdeling migratiesoftware .....	49
Afbeelding 9 Lagen view van componenten migratiesoftware .....	50
Afbeelding 10 Component & Connector View Initiële Vulling .....	53
Afbeelding 11 Entiteit relatie diagram van GBA-V database (gebruikt deel) .....	64
Afbeelding 12 Entiteit relatie diagram van de IV database .....	65
Afbeelding 13 Component & Connector View ISC & Synchronisatie .....	66
Afbeelding 14 Entiteit relatie diagram van de VOISC database .....	73
Afbeelding 15 Entiteit relatie diagram van de ISC database .....	78
Afbeelding 16 Deployment view van de initiële vulling .....	85
Afbeelding 17 Deployment view van ISC & Synchronisatie .....	88
Afbeelding 18 Use Case Model Initieel vullen BRP .....	94
Afbeelding 19 Activiteiten diagram van heen-conversie in initiële vulling .....	95
Afbeelding 20 Activiteiten diagram van terug-conversie in initiële vulling .....	96
Afbeelding 21 Use Case diagram van verwerken synchronisatie-cyclus .....	96
Afbeelding 22 Activiteiten diagram van ISC en Synchronisatie .....	97
Afbeelding 23 Beslisboom voor locking methode .....	102

## Documentbeheer

### Documentinformatie

Documenttitel	Software Architectuur Document Migratiesoftware BRP
Documentdatum	16 maart 2016
Versienummer	0.9
Versiestatus	Definitief
Documentlocatie	<a href="#">migratievoorziening/03 Analysis &amp; Design/01 Software Architecture Document/</a>

### Documenthistorie

#### Versiehistorie

Versienummer	Datum	Auteur	Aanpassingen
0.1	17 april 2014	Operatie BRP	Initiële versie.
0.2	3 juni 2014	Operatie BRP	Review commentaar op versie 0.1 verwerkt.
0.3	17 september 2014	Operatie BRP	Review commentaar op versie 0.2 verwerkt en aanpassingen in code verwerkt. Document definitief gemaakt.
0.4	augustus 2015	Operatie BRP	Wijzigingen t.bv. release 3.1.
0.5	28 september 2015	Operatie BRP	INTEST-818: In hoofdstuk 3 kleine tekstuele aanpassingen. De motivatie in hoofdstuk 4.1.3.5 aangepast. De lijst met afkortingen uitgebreid in hoofdstuk 8.
0.6	29 februari 2016	Operatie BRP	Wijzigingen in ISC verwerkt n.a.v. uitfasering JBoss.
0.7	5 oktober 2016	Operatie BRP	INTEST-2324 Tijdzone conversie naar Nederlandse tijd verwijderd.
0.8	18 januari 2017	Operatie BRP	Wijzigingen t/m release Fred verwerkt.
0.9	16 maart 2017	Operatie BRP	Review commentaar op NFR kruistabel verwerkt.
1.0	29 maart 2017	Operatie BRP	Review commentaar I&T verwerkt en opmerkingen op NFR kruistabel m.b.t. RD-DOC 11/13/14/25/26.



## Reviewhistorie

Versienummer	Datum	Namen
0.1	17 april 2014	Operatie BRP
0.2	22 juli 2014	Agentschap BPR
0.8	18 januari 2017	Interne review door operatie BRP
0.9	23 maart 2017	Review door I&T

## Goedkeuring

Versienummer	Datum	Rol	Akkoord

## Distributie

Versienummer	Naam	Rol

# 1 Inleiding

Dit document is gebaseerd op de methode van het Software Engineering Instituut genaamd "Views and Beyond" zoals deze beschreven is door Clements, et al. [1]. Bij de keuze van de te documenteren views is uitgegaan van het "4+1" view model van Philippe Kruchten [2]. Voor dit model is gekozen omdat dit model gehanteerd wordt bij het Agentschap BPR en ook binnen RUP waarop veel documenten van Operatie BRP zijn gebaseerd.

## 1.1 Scope

In dit document wordt de software architectuur beschreven van de migratiesoftware van de centrale BRP voorziening. Dit document richt zich met name op de migratiesoftware die als onderdeel van stap 3.1 uit het BRP Opleveringsplan (BOP) [14] wordt opgeleverd.

### 1.1.1 *Wat is softwarearchitectuur?*

Als definitie van software architectuur wordt hier uitgegaan van de definitie van Len Bass [3]:

*De softwarearchitectuur van een systeem is de structuur van dat systeem, bestaande uit software elementen, de extern zichtbare eigenschappen van deze elementen, en de relaties daartussen.*

Met extern zichtbare eigenschappen wordt hier bedoeld op de aannames die andere elementen van een bepaald element mogen maken zoals geleverde services, prestaties, foutafhandeling, resource gebruik, enzovoort. Deze definitie is de leidraad waarmee wordt bepaald wat wel en niet gedocumenteerd wordt in dit software architectuur document (SAD).

### 1.1.2 *Elementen en relaties*

Het SAD beschrijft met name de elementen in relatie tot elkaar. Dit betekent dat de beschreven architectuur bewust bepaalde informatie van elementen weglaat omdat deze niet relevant zijn voor haar omgeving. Kortom, een softwarearchitectuur is een abstractie van een systeem waarbij bepaalde details van elementen worden weggelaten omdat deze niets zeggen over de wijze waarop een element gebruik maakt van of gebruikt wordt door andere elementen of informatie uitwisselt met andere elementen. Informatie die alleen betrekking heeft op de interne werking van elementen wordt niet beschreven in het SAD. Dit soort informatie is beschreven in het technisch ontwerp (TO) van het betreffende element.

### 1.1.3 *Verschillende views*

Software is te complex om deze vanuit een enkele structuur te beschrijven en uit te leggen. In plaats van alles in één enkele plaat uit te leggen beschrijft dit SAD de architectuur m.b.v. verschillende structuren of views.

Dit document volgt de methode van het Software Engineering Instituut wat betekent dat dit document de relevante views beschrijft en daarna datgene wat van toepassing is op meerdere views. Vandaar de naam: "Views and Beyond".

### 1.1.4 *Overige technische documentatie*

Zoals hierboven beschreven wordt sommige informatie niet in het SAD opgenomen. Informatie met betrekking tot de interne werking van elementen dient in de vorm

van Javadoc bij de broncode te zijn opgenomen of in één van de volgende documenten:

- Technisch Ontwerp Initiële vulling (IV) [4]
- Technisch Ontwerp Conversie [5]
- Technisch Ontwerp BRP Synchronisatie Service [6]
- Technisch Ontwerp Interstelselcommunicatie (ISC) [7]
- Technisch Ontwerp Verzend & Ontvangst Interstelselcommunicatie (VOISC) [8]
- Technisch Ontwerp Routing [26]
- Technisch Ontwerp Deltabepaling [27]

## 1.2

### Leeswijzer

Dit SAD is bedoeld voor diegenen die kennis nodig hebben over de softwarearchitectuur van de migratiesoftware van de centrale BRP voorziening. In paragraaf 1.4 staat een overzicht van de verschillende onderdelen van dit SAD en voor wie het relevant is.

Benodigde voorkennis is een goede bekendheid met de gegevensmodellen van zowel GBA[15] als BRP [16]. Tevens is enige kennis van de specificatie van de conversie software nodig[13] en kennis van de use case modellen voor ISC[17] en GGO[18].

Hieronder staan de hoofdstukken uit het SAD opgesomd om de lezer te ondersteunen bij het vinden van de juiste informatie.

- Hoofdstuk 1 ("Inleiding") beschrijft het doel en scope van het SAD. Welke informatie kan je wel en welke informatie kan je niet vinden en voor wie is deze informatie relevant is.
- Hoofdstuk 2 ("Wijze waarop een view is gedocumenteerd") geeft de lezer inzicht in het raamwerk dat gehanteerd is om dit SAD te schrijven. Lezers die voor het eerst het SAD lezen doen er verstandig aan dit even door te lezen. Voor een schrijver van dit document is dit hoofdstuk erg belangrijk en zou daarom deze informatie zich goed eigen moeten maken voordat men het SAD gaat wijzigen.
- Hoofdstuk 3 ("Systeem overzicht") is een korte omschrijving en introductie van de migratiesoftware BRP. Dit hoofdstuk beschrijft de migratiesoftware vanuit het hoogste niveau en maakt de relatie tot haar omgeving inzichtelijk.
- Hoofdstuk 4 ("Views") beschrijven die verschillende views op de softwarearchitectuur. Niet elke view is even relevant voor de lezer. Aan de hand van hoofdstuk 1 kan de lezer nagaan welke view en meerdere of mindere mate interessant voor hem of haar is.
- Hoofdstuk 5 ("Relatie tussen views") helpt de lezer te begrijpen wat de relatie van een element in de ene view is met een element uit een andere view. Tenslotte beschrijven alle views hetzelfde systeem.
- Hoofdstuk 6 ("Motivering") gaat in op de ontwerpbeslissingen die betrekking hebben op alle views. Om dus een totaal overzicht van de motivering te krijgen dient de lezer ook te kijken naar de motivering van de verschillende views. Daarnaast beschrijft dit hoofdstuk het gedrag van het systeem om zo beter te kunnen analyseren of de gekozen architectuur passend is. Dit komt overeen met de +1 view van het 4+1 model.
- Hoofdstuk 7 ("Woordenlijst") toont een overzicht begrippen en hun betekenis.
- Hoofdstuk 8 ("Afkortingen") beschrijft wat de verschillende afkortingen in het SAD betekenen.

- Hoofdstuk 9 ("Referenties") toont een overzicht van documenten waarnaar gerefereerd wordt in het SAD.

### 1.3

#### Overzicht van de views

Hieronder volgt een beschrijving van de views die in het SAD zijn beschreven. De gekozen views komen overeen met de view uit het 4+1 model [2].

- Logical View: een decompositie van de migratiesoftware in modules en de afhankelijkheden daartussen.
- Implementation View: een overzicht van de wijze waarop ontwikkelaars (programmeurs en testers) hun werk organiseren, welke artefacten horen bij welke modules en de principes die worden gehanteerd bij de ontwikkeling daarvan.
- Proces View: deze view geeft inzicht in het runtime gedrag van de componenten binnen de migratiesoftware. Hierbij wordt gebruik gemaakt van de component en connector (C&C) stijl om dit inzichtelijk te maken.
- Deployment View: welke componenten worden gedeployed op welke hardware.
- Relevante Use Cases: alhoewel dit in het 4+1 model als view wordt genoemd is de beschrijving van de relevante use cases bij de motivatie opgenomen en niet bij de views.

### 1.4

#### Belanghebbenden en views

Hier staat beschreven welke views interessant zijn voor de verschillende belanghebbenden en welke mate van detail relevant is.

#### Overzicht van views en relevantie voor belanghebbenden

	Logical View	Implementation View	Process View	Deployment View
<b>Belanghebbenden</b>				
Projectmanager	e	o	-	d
Analist	d	e	d	d
Softwareontwikkelaar	d	d	d	e
Tester	d	d	e	e
Applicatie beheerder	d	d	d	e
Technisch beheerder	o	o	e	d
Softwarearchitect	d	d	d	d
<b>Legenda: d = detail informatie, e = enig detail, o = overzicht info</b>				

## 2 Wijze waarop een view is gedocumenteerd

In dit hoofdstuk staat beschreven hoe een view is gedocumenteerd. Dit dient enerzijds als toelichting voor de lezer zodat deze weet hoe bepaalde informatie uit een view te halen is. Anderzijds dient dit als raamwerk voor het documenteren van views en ondersteunt bij het uniform documenteren van verschillende views.

### 2.1 **Template voor een view**

Een beschrijving van een view bestaat uit zes delen. Voor elke view die in dit document beschreven is wordt onderstaand template gevolgd. De delen twee tot en met zes zijn ondersteunend aan het eerste deel – de primaire presentatie – van de view.

#### 2.1.1 *Deel 1. Primaire presentatie*

Dit is meestal een grafische weergave van de elementen en relaties van de view, zoals een UML diagram. Maar dit kan ook een tabel of beschrijving zijn. Deze weergave dient – samen met de legenda – datgene te beschrijven wat deze view over het systeem wil communiceren aan de lezer. Deze weergave hoeft niet alle elementen te beschrijven. Wanneer niet alle elementen zijn beschreven dan is verwezen naar aanvullende documentatie. De primaire presentatie is vaak een schets van de uiteindelijke architectuur en hoeft niet volledig te zijn.

#### 2.1.2 *Deel 2. Element catalogus*

Hierin staan tenminste de elementen opgesomd uit de primaire presentatie. Van elk element is het doel of de rol dat dit element in deze view speelt beschreven. Als het element niet beschreven is in de primaire presentatie dan wordt het element hier geïntroduceerd. De volgende zaken worden beschreven:

##### a) *Elementen en hun eigenschappen*

In dit deel is elk element uit de view genoemd met daarbij de eigenschappen van het element.

##### b) *Relaties en hun eigenschappen*

Elke view heeft specifieke soorten relaties, deze zijn meestal opgenomen in de primaire presentatie. Zo niet dan kunnen ze hier beschreven zijn anders blijft dit deel leeg.

##### c) *Element interfaces*

Hierin zijn de interfaces van de elementen beschreven.

##### d) *Element gedrag*

Sommige elementen vertonen complex gedrag en interacties met hun omgeving. Dit gedrag kan hier nader uitgelegd zijn.

#### 2.1.3 *Deel 3. Context diagram*

Het context diagram laat het (deel)systeem zien in relatie tot haar omgeving.

#### 2.1.4 *Deel 4. Variabiliteit*

Hier staat beschreven welke variatiemogelijkheden zijn geïntroduceerd en hoe deze dienen te worden toegepast. Een variatie mogelijkheid is een bewust aangebrachte flexibiliteit in de architectuur die tijdens de implementatie verder concreet gemaakt wordt.

### 2.1.5 *Deel 5. Motivatie*

Hier staan de relevante ontwerpbeslissingen voor deze view gedocumenteerd. Hoe is dit ontwerp tot stand gekomen en waarom. Daarnaast wordt uitgelegd waarom een bepaald architectuurpatroon of -stijl is gehanteerd.

### 2.1.6 *Deel 6. Interfaces*

Beschrijvingen van interfaces van elementen uit deze view. Vaak zijn interfaces in meerdere views zichtbaar. In sommige gevallen is verwezen naar de beschrijving van de interface in een andere view.

## 2.2 **Template voor interface documentatie**

Om te komen tot een uniforme wijze van interfacebeschrijvingen is hier een standaard indeling beschreven voor interfacedocumentatie. Niet alle onderdelen zijn verplicht, maar wanneer een deel niet is ingevuld, is verantwoordt waarom dit niet relevant is.

### 2.2.1 *Interface identiteit*

Om een unieke verwijzing naar een specifieke interface mogelijk te maken heeft een interface een naam. Wanneer een interface één-op-één overeenkomt met het element waartoe de interface behoort dan is de naam van het element voldoende.

### 2.2.2 *Resources*

Dit is de kern van de interface beschrijving. Dit deel beschrijft hoe de interface gebruikt kan worden, meestal in termen van operaties die op de interface kunnen worden aangeroepen (zoals methoden, procedures en functies).

Hier zijn alle mogelijke resources opgesomd en per resource zijn de onderstaande delen beschreven:

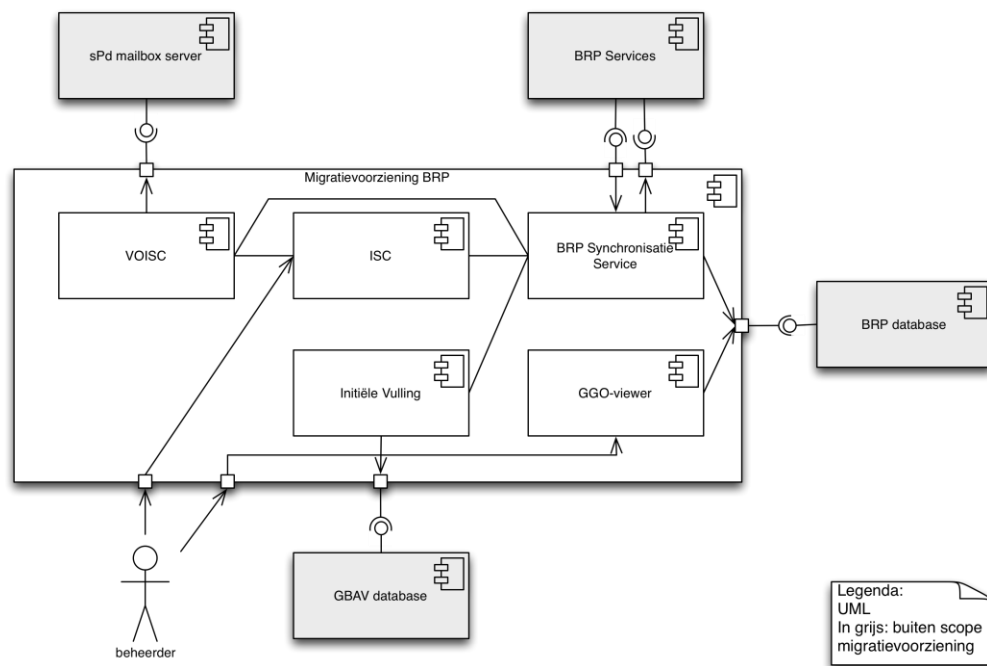
- a) *resource syntax*: dit betreft de naam, de data typen, argumenten, resultaat, enzovoort;
- b) *resource semantiek*: wat levert het gebruik van een bepaalde resource op vanuit het perspectief van de gebruiker? Deze semantiek kan zich op de volgende wijze uiten:
  - I. het toekennen van waarden aan parameters en resultaat, en wat deze betekenen (bijvoorbeeld een gegeven lezen uit de database of schrijven naar de database);
  - II. veranderingen in de – extern merkbare – status van een element;
  - III. gebeurtenissen of berichten die worden verzonden als gevolg van het gebruik van deze resource;
  - IV. de randverschijnselen of effecten op andere elementen als gevolg van het gebruik van deze resource;
  - V. voor de gebruiker merkbare resultaten (bijvoorbeeld het tonen van een melding aan een gebruikers als resultaat van het gebruik van deze resource);
  - VI. of het gedrag atomair is of niet, synchroon of asynchroon, te onderbreken of niet, enzovoort;
  - VII. gebruiksbependingen. Onder welke randvoorwaarden mag deze resource gebruikt worden (bijvoorbeeld dat een operatie B pas mag worden uitgevoerd na operatie A);
- c) *foutafhandeling*: beschrijf de fouten die kunnen optreden als gevolg van het gebruik van deze resource.

- 2.2.3 *Data types en constanten*  
Voor het gebruik van bepaalde resources kunnen data types of constanten gedefinieerd zijn. Deze zijn hier beschreven. Bijvoorbeeld de definitie van een *PersoonEntiteit* die moet worden meegegeven aan de *persoonOpslaan* service aanroep.
- 2.2.4 *Foutafhandeling*  
Hier staat de foutafhandeling beschreven die gelijk is voor alle resources. Per resource staat dus alleen foutafhandeling beschreven die specifiek is voor deze resource.
- 2.2.5 *Variabiliteit*  
Hier staan de configuratiemogelijkheden beschreven van de interface in termen van parameters en hoe de verschillende waarden van deze parameters het gedrag van de interface bepalen.
- 2.2.6 *Kwaliteitsattributen*  
Wat zijn de karakteristieken van de kwaliteitsattributen van deze interface? Welke prestaties mag je verwachten bij een bepaald gebruik? Dus zowel eisen aan het gebruik maar ook beloften van het gebruik.
- 2.2.7 *Motivatie*  
Hier dient uitgelegd te worden wat de redenatie is achter het ontwerp van de interface. Wat waren beperkingen? Zijn er alternatieven overwogen? Welke compromissen zijn gemaakt? Kortom, welke ontwerpbeslissingen zijn genomen en waarom.
- 2.2.8 *Gebruikshandleiding*  
Voor elke resource is eerder beschreven hoe deze gebruikt dient te worden. Toch is het in sommige gevallen handig om voor het interface in het algemeen toe te lichten hoe deze gebruikt dient te worden en hoe bepaalde resources van elkaar afhankelijk zijn. Hier is het gebruik van voorbeelden bijzonder nuttig als aanvulling op de syntax/semantiek beschrijving per resource.

### 3 Systeem overzicht

Het doel van de migratiesoftware is het fungeren als intermediair zodat gemeenten en afnemers geleidelijk, gecontroleerd en m.b.v. een centrale voorziening kunnen overschakelen van het LO3-stelsel naar het BRP-stelsel en binnen beide stelsels zo veel mogelijk hun eigen 'taal' kunnen blijven spreken. De centrale BRP voorziening (inclusief migratiesoftware) wordt gefaseerd opgeleverd.

Deze migratiesoftware biedt functionaliteit voor het initieel vullen van de centrale BRP voorziening, het synchroon houden van de BRP database door het verwerken van LO3 synchronisatie berichten en tenslotte het monitoren van de kwaliteit van de conversie (GGO viewer, zie paragraaf 3.3). Deze functionaliteit wordt geleverd door vijf componenten die in Afbeelding 1 zijn weergegeven.



**Afbeelding 1 Context Diagram Migratiesoftware BRP**

De functionaliteit van de migratiesoftware is verdeeld over drie groepen die hierna elk kort worden toegelicht.

#### 3.1 Initiële vulling

Tijdens de initiële vulling wordt de BRP-database gevuld met de gegevens uit de GBA-V database, zodat de inhoud van beide databases gelijk is. Deze gegevens bestaan uit persoonsgegevens, autorisatieregels en afnemerindicaties. Tevens wordt het gehanteerde gegevensmodel binnen het GBA stelsel geconverteerd naar het BRP gegevensmodel.

#### 3.2 Interstelselcommunicatie(ISC) & Synchronisatie

Vanuit het GBA netwerk worden synchronisatieberichten (bijvoorbeeld Lg01 en La01) [15] via het ISC en Synchronisatie component in de BRP verwerkt. Dit proces staat verder beschreven in de documentatie van use case 202 [9]. Het ontvangen en verzenden van LO3 berichten via het GBA netwerk wordt uitgevoerd door het



VOISC (Verzend & Ontvangst ISC) component. Alle interactie met de BRP verloopt via de BRP Synchronisatie Service. Tenslotte levert het ISC component functionaliteit voor het uitwisselen van berichten tussen migratiecomponenten en het uitvoeren van (langlopende) bedrijfsprocessen. Voor het daadwerkelijke transport van berichten tussen migratie componenten wordt gebruik gemaakt van de migratie routing centrale [26].

### 3.3

#### **GGO viewer**

De GGO viewer maakt inzichtelijk hoe LO3 persoonslijsten als BRP personen zijn opgeslagen in de BRP database. Naast het inzichtelijk maken welke gegevens onderdeel zijn van de LO3 persoonslijst en de BRP persoon laat dit hulpmiddel ook zien hoe gegevens in de BRP afgeleid zijn van LO3 gegevens. Daarbij worden logmeldingen m.b.t. de gegevenskwaliteit getoond. Hiermee kan men de kwaliteit van de conversie monitoren.

## 4 Views

In dit hoofdstuk staan achtereenvolgend de views gedocumenteerd conform het template uit hoofdstuk 2.

### 4.1 Logical View

#### 4.1.1 *Omschrijving*

Deze view toont een decompositie van de migratiesoftware in modules en heeft als doel om inzichtelijk te maken hoe de functionaliteit van de migratiesoftware is verdeeld over de verschillende modules en hun onderlinge afhankelijkheden.

Deze view kan worden gebruikt als:

- uitleg hoe de migratiesoftware is gestructureerd door deze te presenteren in hapklare delen;
- input voor het verdelen van werk;
- ondersteuning bij het bepalen van de impact van wijzigingen;
- ondersteuning bij testen en debuggen.

Deze view toont dus alleen decompositie en afhankelijkheid. Het runtime gedrag is beschreven in de proces view.

#### 4.1.2 *View-delen overzicht*

Voor de overzichtelijkheid is deze view opgedeeld in meerdere delen, verdeeld over twee abstractie niveaus. Op het eerste en tevens hoogste niveau is de migratiesoftware weergegeven in modules en op het tweede niveau is elke migratie module opgedeeld in meerdere sub-modules.

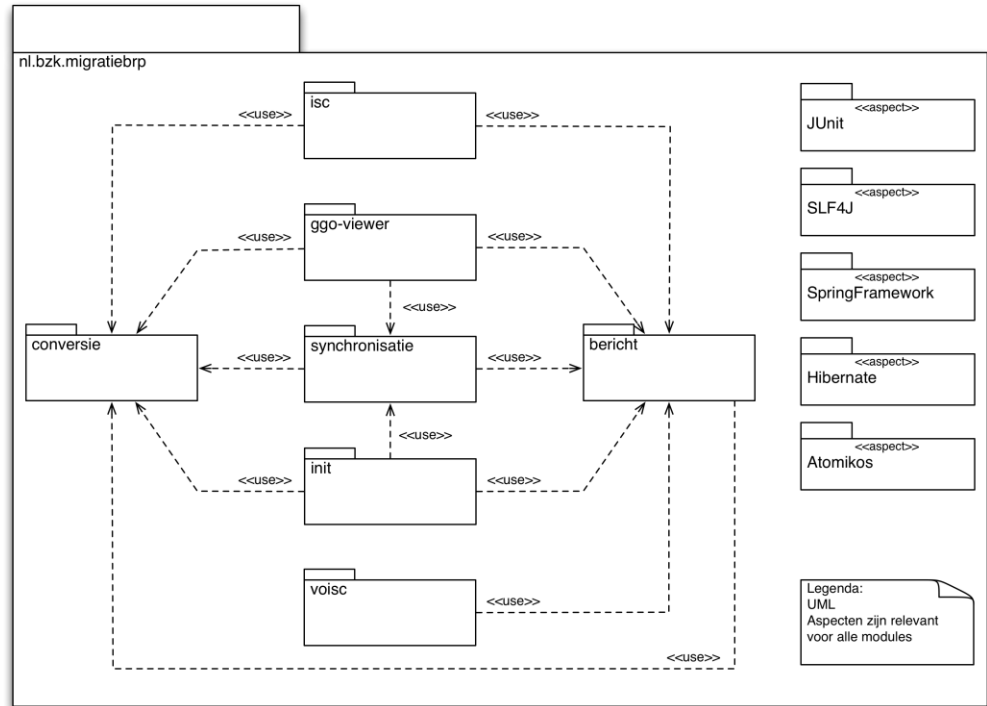
In totaal bestaat deze view uit de volgende delen:

1. Migratiesoftware (migratie);
2. initiële vulling (init);
3. conversie;
4. synchronisatie;
5. verzend & ontvangst interstelsel communicatie (voisc);
6. interstelsel communicatie (isc).

#### 4.1.3 View-deel 1: Migratiesoftware

Dit is het overzichts niveau van de migratiesoftware.

##### 4.1.3.1 Primaire presentatie



**Afbeelding 2 Module View Migratiesoftware**

In bovenstaande afbeelding zijn de verschillende modules van de migratiesoftware weergegeven met hun onderlinge relatie. Daarnaast zijn een aantal relevante afhankelijkheden met externe softwarebibliotheken als aspecten weergegeven. Deze aspecten hebben betrekking op de meeste modules en hebben daarom een significante impact op de softwarearchitectuur.

##### 4.1.3.2 Element catalogus

###### 4.1.3.2.1 Migratiesoftware (nl.bzk.migratiebrp)

Eigenschappen	<b>Verantwoordelijkheid</b> Het migratiesysteem fungeert als intermediair zodat partijen geleidelijk, gecontroleerd en m.b.v. centrale voorzieningen kunnen overschakelen van LO3 naar BRP en binnen beide stelsels zo veel mogelijk hun eigen 'taal' kunnen blijven spreken.
Relaties	Zie Afbeelding 2
Interfaces	Deze zijn beschreven in de process view.
Gedrag	n.v.t.

###### 4.1.3.2.2 Initiële vulling (init)

Eigenschappen	<b>Verantwoordelijkheid</b> Het initieel vullen van de centrale BRP-voorziening om levering aan en bevraging door gemeenten en landelijke afnemers vanuit de centrale BRP-voorziening mogelijk te
---------------	--

	maken.
Relaties	Zie Afbeelding 2
Interfaces	Deze zijn beschreven in de process view.
Gedrag	n.v.t.

## 4.1.3.2.3 Conversie

Eigenschappen	<i>Verantwoordelijkheid</i> Het converteren van persoonsgegevens tussen LO GBA en LO BRP formaat.
Relaties	Zie Afbeelding 2
Interfaces	Zie de interface van conversie regels beschreven in paragraaf 4.1.5.2.2.
Gedrag	n.v.t.

## 4.1.3.2.4 BRP Synchronisatie Service (synchronisatie)

Eigenschappen	<i>Verantwoordelijkheid</i> Voorziet de migratiesoftware van een eenvoudige manier om de BRP database te lezen of schrijven. Dit zijn o.a. eenvoudige 'doorgeef' operaties voor het opslaan of lezen van een persoonslijst maar deze service levert ook complexere operaties die vanuit het proces gedelegeerd zijn naar deze service omdat zij database intensief zijn zoals het verwerken van delta's op de BRP database. <i>Implementatie informatie</i> Alle interactie met de BRP database en de migratiesoftware vindt via de BRP Synchronisatie Service plaats.
Relaties	Zie Afbeelding 2
Interfaces	Deze zijn beschreven in de process view.
Gedrag	n.v.t.

## 4.1.3.2.5 Verzend &amp; ontvangst interstelsel communicatie (voisc)

Eigenschappen	<i>Verantwoordelijkheid</i> Deze service biedt operaties voor het versturen en ontvangen van berichten naar gemeentes en afnemers in het GBA stelsel via de mailbox server (MBS).
Relaties	Zie Afbeelding 2
Interfaces	Deze zijn beschreven in de process view.
Gedrag	n.v.t.

## 4.1.3.2.6 Interstelsel communicatie (isc)

Eigenschappen	<i>Verantwoordelijkheid</i> Het optreden als intermediair bij de berichtuitwisseling tussen partijen (gemeenten en afnemers) die op enig moment niet in hetzelfde stelsel (LO3 of BRP) actief zijn en het verwerken van synchronisatie berichten van GBA gemeentes in de centrale BRP voorziening. <i>Implementatie informatie</i> De ISC module bestaat uit de volgende functionaliteit: <ul style="list-style-type: none"> <li>ISC-runtime: ondersteunt de uitwisseling en transformatie van berichten tussen de migratie componenten;</li> </ul>
---------------	--

	<ul style="list-style-type: none"> <li>• jBPM-engine: executeert processen (in model vorm bijvoorbeeld JPDL);</li> <li>• maatwerk software ter ondersteuning van het uitvoeren van processen door de BPM-engine;</li> <li>• beheerfunctionaliteit in de vorm van webpagina's.</li> </ul>
Relaties	Zie Afbeelding 2
Interfaces	Deze zijn beschreven in de process view.
Gedrag	n.v.t.

## 4.1.3.2.7 GGO-viewer

Eigenschappen	<i>Verantwoordelijkheid</i> Het bieden van een user interface die inzicht geeft in (proef)conversieresultaten. Dit hulpmiddel kan in twee modi opereren: <ul style="list-style-type: none"> <li>• stand-alone: de gebruiker gebruikt een bestand met daarin een persoonslijst ter analyse;</li> <li>• centraal: de gebruiker gebruikt een persoonslijst uit de centrale BRP-voorziening ter analyse.</li> </ul>
Relaties	Zie Afbeelding 2
Interfaces	n.v.t.
Gedrag	n.v.t.

## 4.1.3.2.8 Bericht

Eigenschappen	<i>Verantwoordelijkheid</i> Bevat het berichtenmodel dat gebruikt wordt door alle componenten binnen de migratiesoftware om onderling te kunnen communiceren.
Relaties	Zie Afbeelding 2
Interfaces	n.v.t.
Gedrag	n.v.t.

## 4.1.3.2.9 JUnit

Eigenschappen	<i>Verantwoordelijkheid</i> JUnit is een softwarebibliotheek voor het uitvoeren van geautomatiseerde unit testen. Door Java classes te annoteren met JUnit annotaties kunnen de testen in deze Java classes als onderdeel van een test suite worden uitgevoerd op de ontwikkel en testomgeving en worden hiervan rapportages gemaakt.
Relaties	Zie Afbeelding 2
Interfaces	n.v.t.
Gedrag	n.v.t.

## 4.1.3.2.10 SLF4J

Eigenschappen	<i>Verantwoordelijkheid</i> Deze softwarebibliotheek wordt gebruikt om logmeldingen weg te schrijven naar een bepaald log medium (bijvoorbeeld een logbestand). Het behulp van een configuratiebestand (log4j.properties) kan de wijze van loggen worden geconfigureerd.
Relaties	Zie Afbeelding 2

Interfaces	n.v.t.
Gedrag	n.v.t.

#### 4.1.3.2.11 SpringFramework

Eigenschappen	<i>Verantwoordelijkheid</i> Het springframework bestaat uit een aantal softwarebibliotheken die onder andere functionaliteit bieden voor dependency injection, transaction management en templates voor het versturen van JMS berichten.
Relaties	Zie Afbeelding 2
Interfaces	n.v.t.
Gedrag	n.v.t.

#### 4.1.3.2.12 Hibernate

Eigenschappen	<i>Verantwoordelijkheid</i> Hibernate is een Object Relational Mapping (ORM) tool die het opslaan van objecten in een relationele database en lezen van objecten mogelijk maakt door classes te mappen op tabellen.
Relaties	Zie Afbeelding 2
Interfaces	n.v.t.
Gedrag	n.v.t.

#### 4.1.3.2.13 Atomikos

Eigenschappen	<i>Verantwoordelijkheid</i> De migratiecomponenten maken gebruik van de Java Transaction API (JTA) om het versturen van berichten via JMS en het bijwerken van de database (JDBC) in één transactie te kunnen uitvoeren. Als JTA provider is gekozen voor Atomikos.
Relaties	Zie Afbeelding 2
Interfaces	n.v.t.
Gedrag	n.v.t.

#### 4.1.3.3 Context diagram

Het context diagram van de migratiesoftware BRP toont de context. Zie Afbeelding 1 voor een weergave van de context.

#### 4.1.3.4 Variabiliteit

De functionaliteit van de migratiesoftware kan in drie groepen worden verdeeld zoals beschreven in hoofdstuk 3. Elk van deze groepen heeft zijn eigen installatie van de migratiesoftware:

1. initiële vulling (IV);
2. interstelselcommunicatie (ISC);
3. GGO viewer (GGO).

Dit heeft tot gevolg dat er drie configuraties bestaan voor de migratiesoftware. Elke configuratie heeft een installatie- en beheerhandleiding. Onderstaande tabel geeft weer welke modules gebruikt worden binnen een type installatie.

Configuratie	Modules
IV	conversie sync init
ISC	conversie sync voisc isc routing
GGO	conversie ggo-viewer

#### 4.1.3.5 Motivatie

De verdeling van de functionaliteit over de verschillende modules van de migratiesoftware is voornamelijk beïnvloed door de volgende factoren:

1. groepering o.b.v. verantwoordelijkheid;
2. groepering o.b.v. executeerbare eenheid (zie process view).

#### 4.1.3.6 Interfaces

De interfaces zijn beschreven in de proces view. De volgende (externe) interfaces worden geleverd door de migratiesoftware:

- IscBeheer, zie 4.3.4.6.13
- IscLeveringInbound, zie 4.3.4.6.7
- GGO viewer

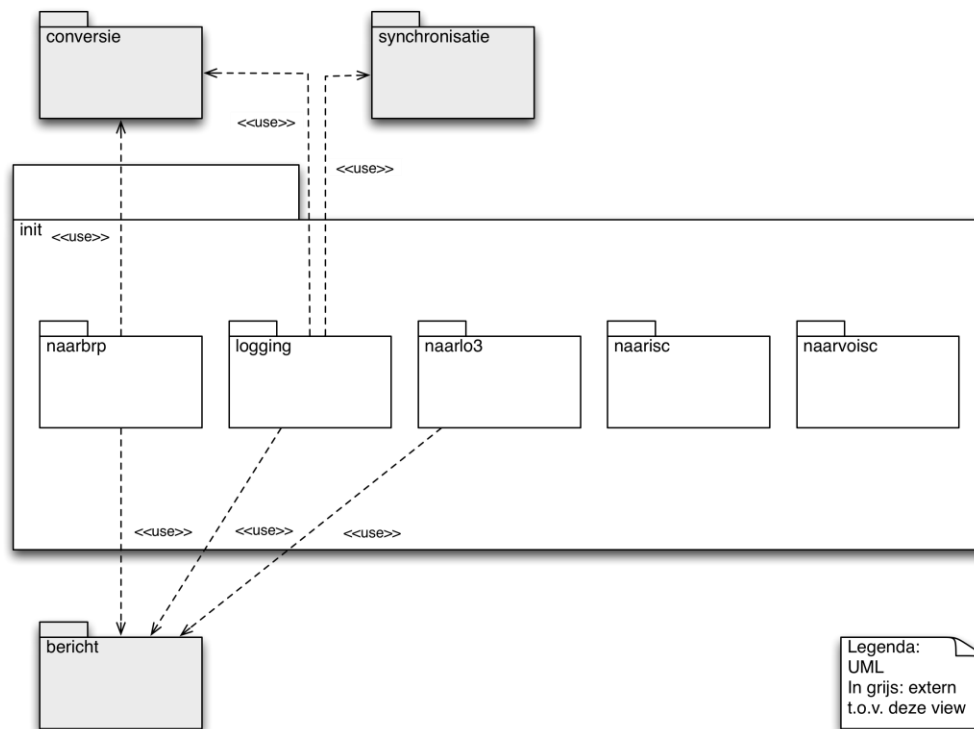
Daarnaast worden de volgende (externe) interfaces gebruikt door de migratiesoftware:

- Mailbox Server, zie 4.3.4.6.3
- BRP database, zie 4.3.3.6.7
- GBAV database, zie 4.3.3.6.9

#### 4.1.4 View-deel 2: Initiële vulling

Dit deel van de logical view beschrijft de module die verantwoordelijk is voor het initieel vullen van de centrale BRP voorziening.

##### 4.1.4.1 Primaire presentatie



**Afbeelding 3 Module View Initiële vulling**

##### 4.1.4.2 Element catalogus

###### 4.1.4.2.1 Conversie naar BRP (naarbrp)

Eigenschappen	<p><i>Verantwoordelijkheid</i></p> <p>Leest informatie in LO3-formaat uit een GBA-V database en stuurt deze naar de BRP Synchronisatie Service om opgeslagen te worden in de BRP database. Deze informatie bestaat uit:</p> <ul style="list-style-type: none"> <li>• persoonslijsten;</li> <li>• autorisaties;</li> <li>• afnemer indicaties;</li> <li>• protocollering.</li> </ul>
Relaties	Zie Afbeelding 3
Interfaces	Deze zijn beschreven in de process view.
Gedrag	n.v.t.

###### 4.1.4.2.2 Logging en verschillenanalyse (logging)

Eigenschappen	<p><i>Verantwoordelijkheid</i></p> <p>Het loggen van de resultaten van de conversie terug naar LO3-formaat en de eventuele verschillen met de oorspronkelijke Lo3 persoonslijst.</p>
Relaties	Zie Afbeelding 3. De afhankelijkheid met de conversie



	module is ontstaan doordat de verschillenanalyse de syntaxcontrole en precondities op de oorspronkelijke Lo3 persoonslijst moet uitvoeren alvorens een zuivere vergelijking met het resultaat van de terugconversie te kunnen uitvoeren. [4]
Interfaces	Deze zijn beschreven in de process view.
Gedrag	n.v.t.

#### 4.1.4.2.3 Conversie naar LO3 (naarlo3)

Eigenschappen	<i>Verantwoordelijkheid</i> Zoekt in de BRP database naar alle ingeschreven personen en stuurt voor elke ingeschreven persoon een bericht naar de BRP Synchronisatie Service met als doel de persoon uit de BRP database te lezen en te converteren naar LO3 formaat.
Relaties	Zie Afbeelding 3
Interfaces	Deze zijn beschreven in de proces view.
Gedrag	n.v.t.

#### 4.1.4.2.4 Conversie naar ISC (naarisc)

Eigenschappen	<i>Verantwoordelijkheid</i> Vult de mig_verzender tabel die door ISC gebruikt wordt om de verzender te bepalen o.b.v. de ontvanger van een bericht.
Relaties	Zie Afbeelding 3
Interfaces	n.v.t.
Gedrag	n.v.t.

#### 4.1.4.2.5 Conversie naar VOISC (naarvoisc)

Eigenschappen	<i>Verantwoordelijkheid</i> Vult de lo3_mailbox tabel die door VOISC gebruikt wordt om voor een verzendende of ontvangende instantie het mailbox nummer te bepalen.
Relaties	Zie Afbeelding 3
Interfaces	n.v.t.
Gedrag	n.v.t.

#### 4.1.4.3 Context diagram

De module view van de migratiesoftware toont de initiële vulling module in zijn context. Zie Afbeelding 2 voor een weergave van de context.

#### 4.1.4.4 Variabiliteit N.v.t.

#### 4.1.4.5 Motivatie

De verschillende submodules binnen initiële vulling zijn verdeeld o.b.v. de stappen die uitgevoerd worden als onderdeel van de initiële vulling:

- stap 1: het lezen van persoonslijsten, autorisatieregels, afnemerindicaties en protocolleringsgegevens in LO3-formaat en aanbieden aan de BRP Synchronisatie Service zodat deze gegevens kunnen worden geconverteerd naar BRP-formaat en worden opgeslagen in de BRP database (naarbrp);

- stap 2: log het resultaat van de synchronisatie naar BRP (logging);
- stap 3: het lezen van administratienummers uit de BRP database van alle ingeschreven personen en deze in de vorm van een LeesUitBrpVerzoek bericht aanbieden aan de BRP Synchronisatie Service zodat voor elk administratienummer de persoon uit de BRP wordt gelezen en geconverteerd wordt naar LO3-formaat (naarlo3);
- stap 4: log het resultaat van de conversie van BRP naar LO3 en de eventuele verschillen met de oorspronkelijke LO3 persoonslijst (logging).

Elke submodule is separaat uitvoerbaar en komt zodanig één-op- één terug in de process view.

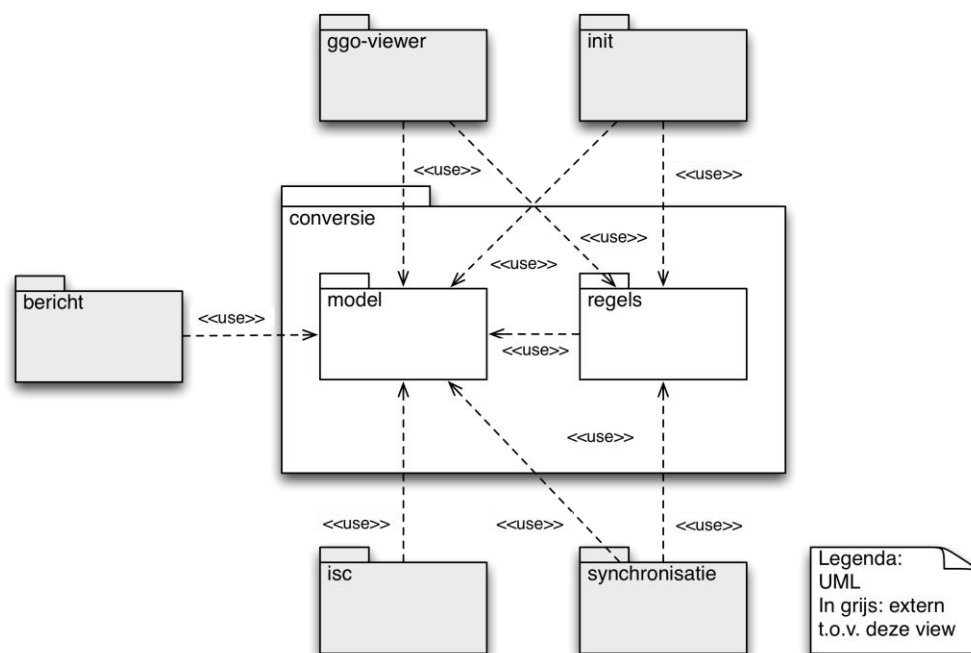
#### 4.1.4.6 Interfaces

N.v.t.

#### 4.1.5 View-deel 3: Conversie

De conversie module bevat enerzijds het *model* waarin LO3 en BRP informatie (persoonslijsten, autorisaties en afnemerindicaties) wordt uitgedrukt. Anderzijds bevat deze module de *regels* die worden uitgevoerd als onderdeel van de conversie van LO3 naar BRP. Voor wat betreft de persoonslijsten is er ook conversie van BRP naar LO3 mogelijk.

##### 4.1.5.1 Primaire presentatie



**Afbeelding 4 Module View Conversie**

##### 4.1.5.2 Element catalogus

###### 4.1.5.2.1 Model

Eigenschappen	<i>Verantwoordelijkheid</i> Het tot uitdrukking brengen van de LO3 en BRP informatie (persoonslijsten, autorisaties en afnemerindicaties) op een zodanige wijze dat deze geconverteerd kunnen worden van LO3 naar BRP.
Relaties	Zie Afbeelding 4
Interfaces	n.v.t.
Gedrag	n.v.t.

###### 4.1.5.2.2 Regels

Eigenschappen	<i>Verantwoordelijkheid</i> Het converteren van GBA informatie in LO3-formaat naar BRP-formaat en vice versa. Daarnaast levert deze submodule functionaliteit voor het controleren van de GBA informatie o.b.v. een verzameling precondities en het loggen van fouten en herleidbaarheid informatie. Het laatst genoemde is informatie die gebruikt kan worden om te herleiden waar persoonsgegevens in het BRP-
---------------	---

	formaat hun herkomst hebben in het LO3-formaat. Voor wat betreft autorisaties en afnemerindicaties is er alleen conversie van LO3 naar BRP mogelijk.
Relaties	Zie Afbeelding 4
Interfaces	Lo3SyntaxControle, zie paragraaf 4.1.5.6.1 PreconditieService, zie paragraaf 4.1.5.6.2 ConverteerLo3NaarBrpService, zie paragraaf 4.1.5.6.3 ConverteerBrpNaarLo3Service, zie paragraaf 4.1.5.6.5 ConverteerNaarExpressieService, zie paragraaf 4.1.5.6.6
Gedrag	n.v.t.

4.1.5.3 Context diagram  
De module view van de migratiesoftware toont de conversie module in zijn context. Zie Afbeelding 2 voor een weergave van de context.

4.1.5.4 Variabiliteit  
N.v.t.

4.1.5.5 Motivatie  
Er is gekozen voor een migratie specifiek model (conversie model) van LO3 en BRP informatie om de conversie logica niet te belasten met informatie of software constructies die niet direct noodzakelijk zijn om een conversie uit te voeren. Dit model dient ter ondersteuning van de conversie zodat de conversie regels eenvoudiger kunnen worden uitgedrukt. Daarnaast heeft dit conversie model als voordeel dat wijzigingen in het fysieke model (zie paragraaf 4.1.6.2.1) niet altijd gevolgen heeft voor het conversie model. De regels zijn in een andere submodule geplaatst om de afhankelijkheden met andere modules te verkleinen. Oorspronkelijk is deze module ontwikkeld voor de conversie van persoonslijsten die bi-directioneel kan worden uitgevoerd. Later is hier de conversie van autorisaties en afnemerindicaties aan toegevoegd. Voor deze laatste twee geldt echter dat er geen conversie van BRP naar LO3 nodig is.

4.1.5.6 Interfaces

4.1.5.6.1 Lo3SyntaxControle

Interface identiteit	Lo3SyntaxControle
Resources	
Resource 1	
Syntax	List< <b>Lo3CategorieWaarde</b> > controleer(List< <b>Lo3CategorieWaarde</b> >categorieWaarden)
Semantiek	Controleert de syntax van een lijst van LO3 voorkomens. Als er fouten worden gevonden van het niveau ERROR op een onjuist voorkomen[13], dan wordt dit voorkomen verwijderd en wordt de logmelding aangepast naar het niveau SUPPRESSED. Het resultaat van deze aanroep is dus een verzameling logmeldingen (benaderbaar via de Logging context) en een geschoonde lijst met LO3 voorkomens.
Foutafhandeling	N.v.t.
Data types/constanten	Class Lo3CategorieWaarde Class BerichtSyntaxException Class OngeldigePersoonslijstException
Foutafhandeling	Tijdens de syntax controle worden meldingen gelogd van het soort: <ul style="list-style-type: none"> <li>SYNTAX</li> </ul>

	<p>Hierbij wordt ook de ernst van de melding gelogd (van klein naar groot):</p> <ul style="list-style-type: none"> <li>• INFO</li> <li>• WARNING</li> <li>• SUPPRESSED</li> <li>• ERROR</li> <li>• CRITICAL</li> </ul> <p>Wanneer er fouten worden geconstateerd tijdens de syntax controle dan worden deze gelogd en gaat de controle verder. Pas als de controle is afgerond wordt bepaald of er meldingen zijn met een ernst van ERROR of hoger. Als dit zo is dan zal er een foutsituatie optreden in de vorm van een OngeldigePersoonslijstException. De gebruiker van de Lo3SyntaxControle heeft zowel voor een normale als een foutieve situatie de beschikking over de volledige logging.</p>
Variabiliteit	N.v.t.
Kwaliteitsattributen	Zie 4.1.5.6.3
Motivatie	Syntaxfouten kunnen zo ernstig zijn dat deze het onmogelijk maken om een Lo3Persoonlijst object te maken. Omdat alle fouten gelogd moeten worden is er een structuur nodig om de wat ruwere vorm van een LO3 persoonslijst uit te drukken.
Gebruikshandleiding	<p><i>Randvoorwaarde</i></p> <p>Voordat deze methode kan worden aangeroepen dient men eerst de logging context te initialiseren.</p> <pre>Logging.initContext();</pre> <p>Vervolgens dient eerst het ruwe Lg01 of La01 bericht in de vorm van een teletex string te worden verwerkt.</p> <pre>final List&lt;Lo3CategorieWaarde&gt; lo3CategorieWaarden =     Lo3Inhoud.parseInhoud(lo3BerichtAsTeletexString);</pre> <p>Als dit niet mogelijk is dan treedt er een fout op in de vorm van een BerichtSyntaxException. Kan het ruwe bericht goed worden gelezen dan kan met het resultaat van deze aanroep de controleer methode worden aangeroepen.</p>

## 4.1.5.6.2 PreconditieService

Interface identiteit	PreconditieService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>Lo3Persoonslijst</b> verwerk( <b>Lo3Persoonslijst</b> )
Semantiek	De Lo3Persoonslijst die aan alle syntax controles voldoet wordt nu gecontroleerd aan de hand van een verzameling structuurregels en precondities. Als er overtredingen worden gevonden dan worden deze gelogd in de Logging context. Overtredingen van het niveau ERROR die betrekking hebben op onjuist LO3 voorkomens hebben tot gevolg dat het betreffende LO3 voorkomen wordt verwijderd en dat het niveau van de overtreding wordt aangepast naar SUPPRESSED.

	Foutafhandeling	N.v.t.
	<i>Resource 2</i>	
	Syntax	<b>Lo3Afnemersindicatie</b> verwerk( <b>Lo3Afnemersindicatie</b> )
	Semantiek	De Lo3Afnemersindicatie wordt gecontroleerd aan de hand van een verzameling structuurregels en precondities. Als er overtredingen worden gevonden dan worden deze gelogd in de Logging context. Overtredingen van het niveau ERROR die betrekking hebben op onjuiste LO3 voorkomens hebben tot gevolg dat het betreffende LO3 voorkomen wordt verwijderd en dat het niveau van de overtreding wordt aangepast naar SUPPRESSED.
	Foutafhandeling	N.v.t.
	<i>Resource 3</i>	
	Syntax	verwerk( <b>Lo3Autorisatie</b> )
	Semantiek	De Lo3Autorisatie wordt gecontroleerd aan de hand van een verzameling structuurregels en precondities. Als er overtredingen worden gevonden dan worden deze gelogd in de Logging context.
	Foutafhandeling	N.v.t.
	Data types/constanten	Class Lo3Persoonslijst Class Lo3Afnemersindicatie Class Lo3Autorisatie
	Foutafhandeling	Tijdens de preconditie controle worden meldingen gelogd van het soort: <ul style="list-style-type: none"> <li>• VERWERKING</li> <li>• STRUCTUUR</li> <li>• BIJZONDERE_SITUATIE</li> <li>• PRECONDITIE</li> </ul> Hierbij wordt ook de ernst van de melding gelogd (van klein naar groot): <ul style="list-style-type: none"> <li>• INFO</li> <li>• WARNING</li> <li>• SUPPRESSED</li> <li>• ERROR</li> <li>• CRITICAL</li> </ul> Wanneer er fouten worden geconstateerd tijdens de preconditie controle dan worden deze gelogd en gaat de controle verder. Pas als de controle is afgerond wordt bepaald of er meldingen zijn met een ernst van ERROR of hoger. Als dit zo is dan zal er een foutsituatie optreden in de vorm van een OngeldigePersoonslijstException. De gebruiker van de PreconditieService heeft zowel voor een normale als een foutieve situatie de beschikking over de volledige logging.
	Variabiliteit	N.v.t.
	Kwaliteitsattributen	Zie 4.1.5.6.3
	Motivatie	Omdat alle preconditie meldingen voor een persoonslijst moeten worden gelogd, ook als daar meldingen tussen zitten die zo ernstig zijn dat conversie onmogelijk is, dient de controle toch verder te worden uitgevoerd. Daarom wordt eerst de gehele Lo3Persoonlijst gecontroleerd voordat de conversie wordt gestart.
	Gebruikshandleiding	<i>Randvoorwaarde (alleen Lo3Persoonslijst)</i>

	<p>Voordat de methode voor de verwerking van Lo3Persoonslijsten kan worden aangeroepen dient eerst de Lo3SyntaxControle te zijn uitgevoerd. Het resultaat van deze controle dient te worden omgezet in de Lo3Persoonslijst alvorens de verwerk methode van de PreconditieService kan worden aangeroepen. Dit kan met de volgende aanroep:</p> <pre>new Lo3PersoonslijstParser().parse(schoneLo3CategorieWaarden);</pre>
--	---

## 4.1.5.6.3 ConverteerLo3NaarBrpService

Interface identiteit		ConverteerLo3NaarBrpService
<i>Resources</i>		
	<i>Resource 1</i>	
	Syntax	<b>BrpPersoonslijst</b> <i>converteerLo3Persoonslijst(Lo3Persoonslijst)</i>
	Semantiek	Converteert de Lo3Persoonslijst parameter naar een BrpPersoonslijst object door het toepassen van de conversieregels.
	Foutafhandeling	n.v.t.
	<i>Resource 2</i>	
	Syntax	<b>BrpAutorisatie</b> <i>converteerLo3Autorisatie(Lo3Autorisatie)</i>
	Semantiek	Converteert de Lo3Autorisatie parameter naar een BrpAutorisatie object door het toepassen van de conversieregels.
	Foutafhandeling	n.v.t.
	<i>Resource 3</i>	
	Syntax	<b>BrpAfnemerindicaties</b> <i>converteerLo3Afnemerindicaties(Lo3Afnemerindicaties)</i>
	Semantiek	Converteert de Lo3Afnemerindicaties parameter naar een BrpAfnemerindicaties object door het toepassen van de conversieregels.
	Foutafhandeling	n.v.t.
	<i>Resource 4</i>	
	Syntax	<b>BrpToevalligeGebeurtenis</b> <i>converteerLo3ToevalligeGebeurtenis(Lo3ToevalligeGebeurtenis)</i>
	Semantiek	Converteert de Lo3ToevalligeGebeurtenis parameter naar een BrpToevalligeGebeurtenis object door het toepassen van de conversieregels.
	Foutafhandeling	n.v.t.
	Data types/constanten	Class BrpAfnemerindicaties Class BrpAutorisatie Class BrpPersoonslijst Class BrpToevalligeGebeurtenis Class Lo3Afnemerindicaties Class Lo3Autorisatie Class Lo3Persoonslijst Class Lo3ToevalligeGebeurtenis
	Foutafhandeling	Tijdens de conversie worden meldingen gelogd van een bepaalde soort: <ul style="list-style-type: none"> <li>VERWERKING</li> </ul>

	<ul style="list-style-type: none"> <li>• STRUCTUUR</li> <li>• SYNTAX</li> <li>• BIJZONDERE_SITUATIE</li> <li>• PRECONDITIE</li> </ul> <p>Hierbij wordt ook de ernst van de melding gelogd (van klein naar groot):</p> <ul style="list-style-type: none"> <li>• INFO</li> <li>• WARNING</li> <li>• SUPPRESSED</li> <li>• ERROR</li> <li>• CRITICAL</li> </ul> <p>Wanneer er fouten worden geconstateerd tijdens de conversie dan worden deze gelogd en gaat de conversie verder. Pas als de conversie is afgerond wordt gecontroleerd of er meldingen zijn met een ernst van ERROR of hoger. Als dit zo is dan zal er een foutsituatie optreden in de vorm van een Java RuntimeException. De gebruiker van de ConversieService heeft zowel voor een normale als een foutieve situatie de beschikking over de volledige logging.</p>
Variabiliteit	n.v.t.
Kwaliteitsattributen	<p><i>Testbaarheid</i></p> <p>Om de testbaarheid van de conversiesoftware te verbeteren kan het model naar XML worden geëxporteerd. Ook kan XML worden geïmporteerd. Speciaal ontwikkelde test tooling voor conversie maakt hiervan gebruik.</p> <p><i>Prestaties</i></p> <p>De snelheid van conversie is van groot belang om de kwaliteitseisen m.b.t. initiële vulling en synchronisatie te kunnen behalen. Omdat conversie geheel in geheugen plaatsvindt wordt de snelheid van initiële vulling of synchronisatie vooral beperkt door de interactie met de database. Dit wordt verder uitgewerkt in de proces view.</p>
Motivatie	<p>Deze interface is in de loop der tijd geëvolueerd. In eerste instantie is de conversie software ontwikkeld. Hierbij kwam later de wens om meldingen te loggen. De conversie van persoonslijsten is bi-directioneel. Later is hier de conversie van autorisaties en afnemerindicaties aan toegevoegd. Omdat hiervoor geen terugconversie nodig is wijkt dit af van het standaard patroon.</p>
Gebruikshandleiding	<p>Voordat de conversie kan worden uitgevoerd dient eerst de PreconditieService te zijn aangeroepen voor de te converteren LO3 informatie. Hieronder staat een stukje voorbeeldcode wat het gebruik van de conversie service illustreert.</p> <pre> @Inject private Lo3SyntaxControle syntaxControle; @Inject private PreconditiesService preconditieService; @Inject private ConverterLo3NaarBrpService conversieService;  public BrpPersoonslijst gebruikConversieService(final String lo3BerichtAsTeletexString) {     try {         Logging.initContext();         final List&lt;Lo3CategorieWaarde&gt; lo3CategorieWaarden = </pre>



	<pre> Lo3Inhoud.parseInhoud(lo3BerichtAsTeletexString);      final List&lt;Lo3CategorieWaarde&gt;         schoneLo3CategorieWaarden =             syntaxControle.controleer(lo3CategorieWaarden);      final Lo3Persoonslijst lo3Persoonslijst =         new Lo3PersoonslijstParser().parse(             schoneLo3CategorieWaarden);      final Lo3Persoonslijst schoneLo3Persoonslijst =         preconditionieService.verwerk(lo3Persoonslijst);     return conversieService.converteerLo3Persoonslijst(         schoneLo3Persoonslijst); } catch (BerichtSyntaxException           InputValidationException           OngeldigePersoonslijstException e) {     // TODO: mogelijkheid om logging af te handelen     Logging logging = Logging.getLogging();     // TODO: handel fout af     return null; } finally {     Logging.destroyContext(); } </pre>
--	---

## 4.1.5.6.5 ConverteerBrpNaarLo3Service

Interface identiteit	ConverteerBrpNaarLo3Service
Resources	
Resource 1	
Syntax	<b>Lo3Persoonslijst</b> <i>converteerBrpPersoonslijst(BrpPersoonslijst)</i>
Semantiek	Converteert de BrpPersoonlijst parameter naar een Lo3Persoonslijst object door het toepassen van de conversieregels.
Foutafhandeling	n.v.t.
Data types/constanten	Class BrpPersoonslijst Class Lo3Persoonslijst
Foutafhandeling	In tegenstelling tot de conversie van LO3-formaat naar BRP-formaat, wordt niet de verder gegaan met het valideren van de persoonsgegevens als er een fout is gevonden. De migratiesoftware gaat ervan uit dat de persoonsgegevens uit de BRP aan alle precondities voldoet. Wanneer dit niet het geval is dan treedt er een runtime exception op van het type 'PreconditieException'.
Variabiliteit	n.v.t.
Kwaliteitsattributen	Deze zijn gelijk aan die van de ConverteerLo3NaarBrpService interface. Zie hiervoor paragraaf 4.1.5.6.3.
Motivatie	Deze is gelijk aan die van de ConverteerLo3NaarBrpService interface. Zie hiervoor paragraaf 4.1.5.6.3.
Gebruikshandleiding	N.v.t.

## 4.1.5.6.6 ConverteerNaarExpressieService

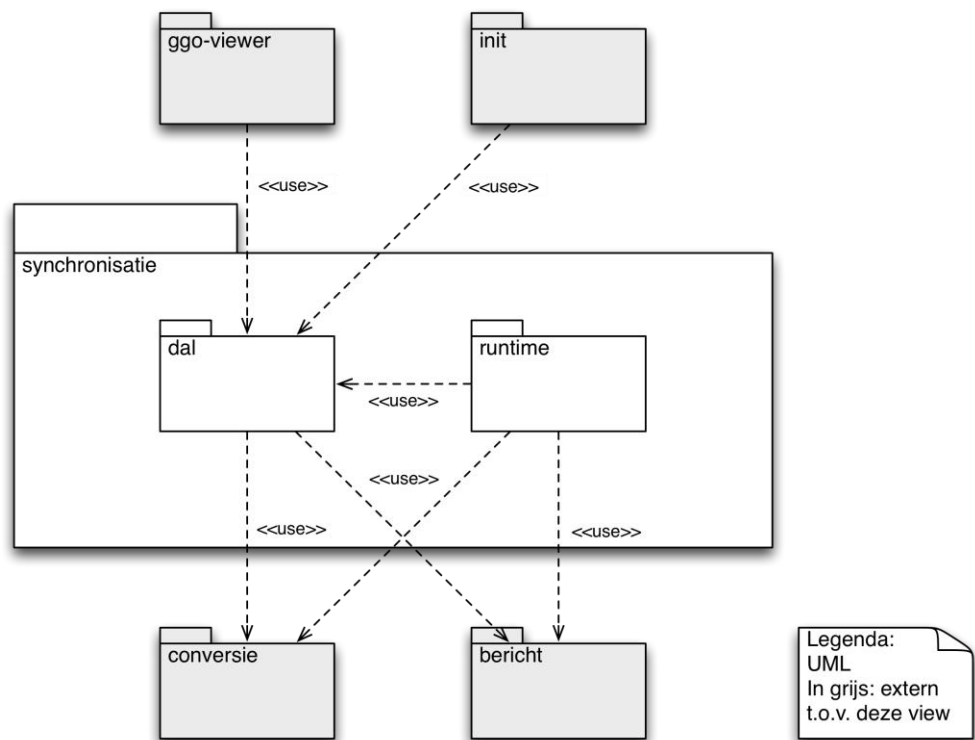
Interface identiteit	ConverteerNaarExpressieService
Resources	
Resource 1	
Syntax	<b>String</b> converteerSleutelRubrieken( <b>autorisatieRubrieken,</b>

		<b>herkomst)</b>
	Semantiek	Converteert een lijst met autorisatierubrieken naar een attenderingscriterium door voor elke rubriek de bijbehorende expressie te bepalen.
	Foutafhandeling	Als voor een LO3 rubriek geen BRP expressie gevonden kan worden dan wordt een fout gelogd. Hiervoor wordt het herkomst argument gebruikt. De fout wordt gelogd als AUT007. Vervolgens wordt er verder gegaan de overige rubrieken.
	<i>Resource 2</i>	
	Syntax	<b>String</b> converteerVoorwaardeRegel( <b>lo3VoorwaardeRegel</b> )
	Semantiek	Converteert een LO3 voorwaarderegel naar een BRP expressie.
	Foutafhandeling	Wanneer de LO3 voorwaarde regel niet geconverteerd kan worden wordt een exceptie gegooid en stop de verwerking.
	Data types/constanten	String autorisatieRubrieken Lo3Herkomst herkomst String lo3Voorwaarderegel Lo3VoorwaardeRegelOnvertaalbaarExceptie
	Foutafhandeling	Deze is specifiek per resource.
	Variabiliteit	n.v.t.
	Kwaliteitsattributen	Deze zijn gelijk aan die van de ConverteerLo3NaarBrpService interface. Zie hiervoor paragraaf 4.1.5.6.3.
	Motivatie	Deze is gelijk aan die van de ConverteerLo3NaarBrpService interface. Zie hiervoor paragraaf 4.1.5.6.3.
	Gebruikshandleiding	N.v.t.

#### 4.1.6 View-deel 4: BRP Synchronisatie Service

De BRP Synchronisatie Service is het enige element van de migratiesoftware dat toegang heeft tot de BRP database en queues. Deze service biedt een interface voor het lezen en schrijven van personen naar de BRP database maar bevat ook complexere operaties die naar deze service zijn gedelegeerd omdat ze database intensief zijn.

##### 4.1.6.1 Primaire presentatie



**Afbeelding 5 Module View Synchronisatie**

##### 4.1.6.2 Element catalogus

###### 4.1.6.2.1 Data Access Layer (DAL)

Eigenschappen	<p><b>Verantwoordelijkheid</b></p> <p>Deze submodule bevat functionaliteit die BRP database intensief is. Dit is de enige module die kennis heeft van de BRP database. De functionaliteit van deze submodule bestaat o.a. uit:</p> <ul style="list-style-type: none"> <li>• Create, Read, Update en Delete (CRUD) operaties op de BRP tabellen;</li> <li>• Mapping van logisch model op fysiek model en vice versa;</li> <li>• het bepalen van delta's voor personen in de BRP.</li> </ul>
Relaties	Zie Afbeelding 5
Interfaces	BrpDalService, zie paragraaf 4.1.6.6.1
Gedrag	n.v.t.

## 4.1.6.2.2 Runtime

Eigenschappen	<i>Verantwoordelijkheid</i> Deze submodule bevat de functionaliteit om de BRP Synchronisatie Service JMS berichten te laten verwerken en versturen. De interface van de DAL wordt via deze service ontsloten voor andere componenten. Zie hiervoor de proces view.
Relaties	Zie Afbeelding 5
Interfaces	Zie hiervoor de proces view.
Gedrag	n.v.t.

## 4.1.6.3 Context diagram

De module view van de migratiesoftware toont de synchronisatie module in zijn context. Zie Afbeelding 2 voor een weergave van de context.

## 4.1.6.4 Variabiliteit

Zie hiervoor de variabiliteit van de migratiesoftware in paragraaf 4.1.3.4.

## 4.1.6.5 Motivatie

Door de functionaliteit m.b.t. de BRP database te concentreren in een aparte module worden de volgende voordelen bereikt:

- de impact van wijzigingen op het fysieke BRP model blijven beperkt;
- de wijzigingen op de BRP database vanuit de migratiesoftware zijn beter beheersbaar;
- door de DAL interface te ontsluiten via JMS kan de BRP Synchronisatie Service als los component draaien en dus mogelijkwerwijs 'dichterbij' de BRP database worden gedeployed dan andere migratie componenten;
- de kennis m.b.t. de BRP database is geïsoleerd binnen één module. Dit bevordert de onderhoudbaarheid van de software.

Hierbij zijn de volgende nadelen in overwogen genomen:

- verslechtering van de performance van de migratiesoftware omdat andere componenten via JMS met de BRP Synchronisatie Service moeten communiceren om gegevens uit de BRP database te kunnen benaderen;
- het uitvoeren van bedrijfslogica dient binnen ISC te geschieden, wanneer deze echter database intensief is dan moet deze vanwege performance redenen worden gedelegeerd naar de BRP Synchronisatie Service.

## 4.1.6.6 Interfaces

## 4.1.6.6.1 BrpDalService

Interface identiteit	BrpDalService (uitgewerkt in: BrpAfnemerIndicatiesService, BrpAutorisatieService, BrpPersoonslijstService, LeveringsaantekeningService, PersoonService)
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>Persoon</b> persisteerPersoonslijst( <b>BrpPersoonslijst, Lo3Bericht</b> )
Semantiek	Deze functie mapped de BrpPersoonslijst op een Persoon entiteit en relateert de betrokkenen van deze persoon met eventuele bestaande personen in de database alvorens de persoon op te slaan in de BRP database. Het kan zijn dat een nieuw persoon wordt aangemaakt of dat een bestaand

		persoon wordt bijgewerkt.
	Foutafhandeling	N.v.t.
	<i>Resource 2</i>	
	Syntax	<b>Persoon</b> persisterPersoonslijst( <b>BrpPersoonslijst, ANummer, Lo3Bericht</b> )
	Semantiek	Deze methode doet hetzelfde als de persisterPersoonslijst aanroep zonder a-nummer. Hierbij is echter sprake van een a-nummer wijziging en zal dus het meegegeven a-nummer gebruikt worden om te bepalen welke persoon in de BRP moet worden bijgewerkt. Deze persoon <u>moet</u> bestaan in de database. Deze methode gaat ervan uit dat de gebruiker dit voldoende heeft geverifieerd.
	Foutafhandeling	Als met het gegeven ANummer geen persoon gevonden wordt zal er een technische fout optreden.
	<i>Resource 3</i>	
	Syntax	<b>BrpPersoonslijst</b> bevaagPersoonslijst( <b>ANummer</b> )
	Semantiek	Leest een persoon uit de BRP database voor het meegegeven ANummer en mapped de persoon entiteit op het conversie model. Deze persoon <u>moet</u> bestaan in de database. Deze methode gaat ervan uit dat de gebruiker dit voldoende heeft geverifieerd.
	Foutafhandeling	Als met het gegeven ANummer geen persoon gevonden wordt zal er een technische fout optreden.
	<i>Resource 4</i>	
	Syntax	<b>BrpPersoonslijst</b> bevaagPersoonslijstOpTechnischeSleutel( <b>techischesleutel</b> )
	Semantiek	Leest een persoon uit de BRP database voor de meegegeven technische sleutel en mapped de persoon entiteit op het conversie model. Deze persoon <u>moet</u> bestaan in de database. Deze methode gaat ervan uit dat de gebruiker dit voldoende heeft geverifieerd.
	Foutafhandeling	Als met de gegeven technische sleutel geen persoon gevonden wordt zal er een technische fout optreden.
	<i>Resource 5</i>	
	Syntax	<b>BrpPersoonslijst</b> zoekPersoonOpAnummer( <b>ANummer</b> )
	Semantiek	Zoekt een persoon in de BRP database voor het meegegeven ANummer en mapped de persoon entiteit op het conversie model. Wanneer de persoon niet wordt gevonden wordt er een NULL waarde geretourneerd.
	Foutafhandeling	N.v.t.
	<i>Resource 6</i>	
	Syntax	<b>BrpPersoonslijst</b> zoekPersoonOpHistorischAnummer( <b>ANummer</b> )
	Semantiek	Zoekt een persoon met een historisch ANummer dat gelijk is aan het meegegeven ANummer. Wanneer de persoon niet wordt gevonden wordt er een NULL waarde geretourneerd.
	Foutafhandeling	N.v.t.
	<i>Resource 7</i>	
	Syntax	<b>Lo3Bericht</b> zoekLo3BerichtOpAnummer( <b>ANummer</b> )
	Semantiek	Zoekt het laatst toegevoegde Lo3Bericht voor het meegegeven ANummer op. Als er niet gevonden wordt dan wordt een NULL waarde geretourneerd.
	Foutafhandeling	N.v.t.

	<i>Resource 8</i>	
	Syntax	<b>Set&lt;ANummer&gt;</b> zoekBerichtLogAnrs( <b>vanaf, tot</b> )
	Semantiek	Zoekt naar de laatst toegevoegde Lo3Berichten die zijn toegevoegd tussen: <i>vanaf</i> (inclusief) en <i>tot</i> (exclusief) en retourneert hiervan het ANummer.
	Foutafhandeling	N.v.t.
	<i>Resource 9</i>	
	Syntax	<b>Lo3Bericht</b> persisteerLo3Bericht( <b>Lo3Bericht</b> )
	Semantiek	Deze methode slaat het meegegeven Lo3Bericht op in de database.
	Foutafhandeling	N.v.t.
	<i>Resource 10</i>	
	Syntax	<b>Blokkering</b> persisteerBlokkering( <b>Blokkering</b> )
	Semantiek	Deze methode slaat het meegegeven Blokkering object op in de database.
	Foutafhandeling	N.v.t.
	<i>Resource 11</i>	
	Syntax	<b>Blokkering</b> vraagOpBlokkering( <b>ANummer</b> )
	Semantiek	Zoekt naar een Blokkering object voor het meegegeven ANummer. De methode retourneert een NULL waarde als er geen Blokkering wordt gevonden.
	Foutafhandeling	N.v.t.
	<i>Resource 12</i>	
	Syntax	verwijderBlokkering( <b>Blokkering</b> )
	Semantiek	Verwijdert het meegegeven Blokkering object uit de database.
	Foutafhandeling	N.v.t.
	<i>Resource 13</i>	
	Syntax	<b>List&lt;Persoon&gt;</b> zoekPersoon( <b>Persoon</b> )
	Semantiek	Zoekt naar Personen die voldoen aan de criteria van de meegegeven persoon.
	Foutafhandeling	N.v.t.
	<i>Resource 14</i>	
	Syntax	<b>Collection&lt;Gemeente&gt;</b> geefAlleGemeenten()
	Semantiek	Retourneert een verzameling van alle gemeentes uit de BRP database.
	Foutafhandeling	N.v.t.
	<i>Resource 15</i>	
	Syntax	persisteerAutorisatie( <b>BrpAutorisatie</b> )
	Semantiek	Deze methode mapped het meegegeven BrpAutorisatie object op de BRP entiteiten en slaat deze op in de database. Dit heeft effect op de volgende BRP entiteiten: <ul style="list-style-type: none"> <li>• partij;</li> <li>• autorisatietabel;</li> <li>• autorisatiebesluit;</li> <li>• abonnement.</li> </ul>
	Foutafhandeling	N.v.t.
	<i>Resource 16</i>	
	Syntax	<b>BrpAutorisatie</b> bevrageAutorisatie( <b>partijCode, naam, ingangsDatumRegel</b> )
	Semantiek	Zoekt naar autorisatie besluiten waarvan de partijCode en naam overeenkomen met de meegegeven partijCode en naam. Daarnaast wordt gezocht naar autorisatie regels waarvan de

		partijCode en ingangsdatum overeenkomt met de meegegeven partijCode en ingangsDatumRegel. Deze informatie wordt vervolgens gemapped op het conversie model (BrpAutorisatie) en geretourneerd.
Foutafhandeling		N.v.t.
<i>Resource 17</i>		
Syntax		<b>persisteerAfnemersindicaties(BrpAfnemersindicaties)</b>
Semantiek		Deze methode mapped een BrpAfnemerindicaties object op de BRP entiteiten en slaat deze op in de database.
Foutafhandeling		N.v.t.
<i>Resource 18</i>		
Syntax		<b>BrpAfnemersindicaties</b> bevraagAfnemersindicaties( <b>ANummer</b> )
Semantiek		Zoekt o.b.v. het meegeven ANummer naar afnemer indicaties en mapped deze op het conversie model (BrpAfnemerIndicaties).
Foutafhandeling		N.v.t.
<i>Resource 19</i>		
Syntax		<b>List&lt;Rubriek&gt;</b> geefAlleRubriekenVoorBrpExpressie( <b>BrpExpressie</b> )
Semantiek		Zoekt naar de rubrieken voor een bepaalde BRP expressie en retourneert een lijst van gevonden rubrieken.
Foutafhandeling		N.v.t.
<i>Resource 20</i>		
Syntax		<b>List&lt;Rubriek&gt;</b> geefAlleRubriekenVoorBrpExpressie( <b>Expressie</b> )
Semantiek		Zoekt naar de rubrieken voor een bepaalde expressie en retourneert een lijst van gevonden rubrieken.
Foutafhandeling		N.v.t.
<i>Resource 21</i>		
Syntax		<b>Expressie</b> geefExpressieOpBasisVanExpressieTekst( <b>expressieTekst</b> )
Semantiek		Zoekt naar een expressie op basis van de meegegeven expressie tekst.
Foutafhandeling		Als er niet exact één Expressie wordt gevonden voor de meegegeven expressie tekst treedt er een technische fout op.
Data types/constanten		<i>Conversie model</i> Class BrpAutorisatie Class BrpAfnemerindicaties Class BrpExpressie Class BrpPersoonslijst <i>BRP entiteiten</i> Class Blokkering Class Expressie Class Gemeente Class Lo3Bericht Class Rubriek <i>Java typen</i> Long ANummer String technischesleutel, naam, expressieTekst Date vanaf, tot,ingangsDatumRegel Integer partijCode
Foutafhandeling		Bij het gebruik van deze methodes kunnen fouten optreden

	<p>bij de interactie met de BRP database. Dit gaat om onverwachte fouten die zullen resulteren in zogenaamde 'unchecked exceptions'. Daarnaast worden er ook controles uitgevoerd op de inhoud van de database die niet door de database zelf worden afgedwongen. Bijvoorbeeld op administratienummer of burgerservicenummer, hiervan dwingt de database niet af dat deze uniek zijn. Toch gaat de migratiesoftware ervan uit dat ze uniek zijn als de betrokken persoonslijst niet opgeschort is met reden 'F'. Blijkt dit niet het geval dan treedt er een technische fout op. We gaan er namelijk vanuit dat de bedrijfsregels van de BRP en migratie software zullen afdwingen dat aan deze voorwaarde wordt voldaan.</p> <p>Alle methodes van deze service worden in een transactie uitgevoerd. Dat betekent dat een wijziging in de BRP database alleen wordt doorgevoerd indien de methode volledig succesvol is afgerond. Deze interface is niet beschikbaar buiten de Synchronisatie module en de verdere foutafhandeling vindt dus binnen de module plaats.</p>
Variabiliteit	N.v.t.
Kwaliteitsattributen	<p><i>Prestaties</i></p> <p>Omdat de methodes van deze interface database intensief zijn, zijn ze relatief langzaam. Het is daarom van groot belang om het aantal database operaties zo klein mogelijk te houden. De gegevens die een methode raakt zijn wel geïsoleerd. Een wijziging heeft vaak betrekking op een specifiek persoon en relaties met andere personen zijn relatief beperkt. Dit maakt de kans op conflicten klein en daarom kunnen er dus veel van dit soort methodes parallel worden uitgevoerd. Dit wordt verder beschreven in de proces view.</p>
Motivatie	Om de wijzigingen in de BRP database vanuit de migratiesoftware beheersbaar te houden is besloten dat alleen de BRP Synchronisatie Service toegang heeft tot de BRP database. Binnen deze module heeft alleen de DAL submodule kennis van de BRP database. En daarom mogen andere modules geen gebruik maken van deze interface.
Gebruikshandleiding	Deze interface is alleen beschikbaar binnen de Synchronisatie module.



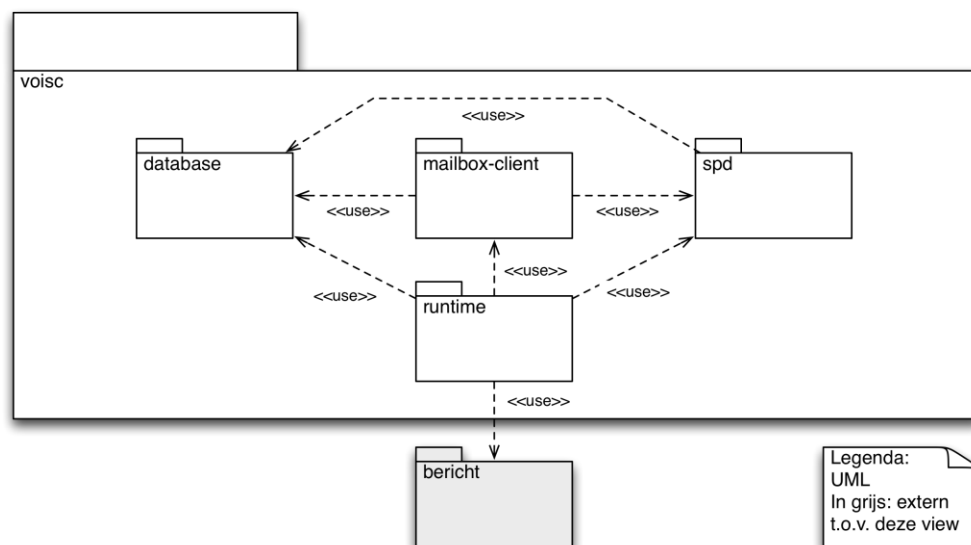
## 4.1.7

*View-deel 5: Verzend & Ontvangst Interstelsel Communicatie (VOISC)*

De VOISC module bevat protocol "bridging" functionaliteit om communicatie tussen een sPd mailbox en JMS queue mogelijk te maken. Deze module bevat daarnaast functionaliteit om te kunnen fungeren als een "middle-tier" berichtenmakelaar die berichten uit het ene kanaal ontvangt, deze vertaalt m.b.v. een protocol adapter en vervolgens doorstuurt naar het andere kanaal.

## 4.1.7.1

## Primaire presentatie

**Afbeelding 6 Module View VOISC**

## 4.1.7.2

## Element catalogus

## 4.1.7.2.1

## Database

Eigenschappen	<i>Verantwoordelijkheid</i> Berichten worden tijdelijk opgeslagen in een database alvorens te worden doorgestuurd naar het andere kanaal. Deze submodule bevat de functionaliteit om CRUD operaties op de database te kunnen uitvoeren.
Relaties	Zie Afbeelding 6
Interfaces	n.v.t.
Gedrag	n.v.t.

## 4.1.7.2.2

## Runtime

Eigenschappen	<i>Verantwoordelijkheid</i> Deze submodule bevat de functionaliteit om de VOISC als stand-alone component te kunnen draaien. Dit bestaat uit de volgende functionaliteit: <ul style="list-style-type: none"> <li>planning functionaliteit om periodiek de sPd mailbox server te bevragen voor nieuwe berichten;</li> <li>JMS functionaliteit om te kunnen communiceren met ISC via queues;</li> <li>adapter functionaliteit om ISC berichten te vertalen naar sPd berichten en vice versa.</li> </ul>
---------------	--

Relaties	Zie Afbeelding 6
Interfaces	Deze zijn beschreven in de proces view.
Gedrag	n.v.t.

## 4.1.7.2.3 mailbox-client

Eigenschappen	<i>Verantwoordelijkheid</i> Bevat functionaliteit om de mailbox server te kunnen gebruiken.
Relaties	Zie Afbeelding 6
Interfaces	n.v.t.
Gedrag	n.v.t.

## 4.1.7.2.4 sPd

Eigenschappen	<i>Verantwoordelijkheid</i> Bevat functionaliteit om sPd operaties te kunnen uitvoeren.
Relaties	Zie Afbeelding 6
Interfaces	n.v.t.
Gedrag	n.v.t.

## 4.1.7.3

## Context diagram

De module view van de migratiesoftware toont de VOISC module in zijn context. Zie Afbeelding 2 voor een weergave van deze context.

## 4.1.7.4

## Variabiliteit

N.v.t.

## 4.1.7.5

## Motivatie

Deze module bevat specifieke kennis om berichten tussen ISC en de GBA-mailbox mogelijk te maken door gebruik te maken van het sPd protocol. Deze functionaliteit loskoppelen van ISC functionaliteit heeft als voordeel dat wijzigingen in ISC weinig impact hebben op de VOISC en vice versa. Daarnaast wordt het makkelijker om software ontwikkel inspanning tussen de VOISC module en ISC module te verdelen omdat iedere module zijn eigen specifieke probleemgebieden kent. Kortom, door het toepassen van het "separation of concerns" principe is tot deze opdeling gekomen. Hierbij is in overweging genomen dat een extra component nadelige gevolgen heeft voor de prestaties van de migratiesoftware, maar deze effecten zullen minimaal zijn gezien het gedrag van de GBA-mailbox server.

## 4.1.7.6

## Interfaces

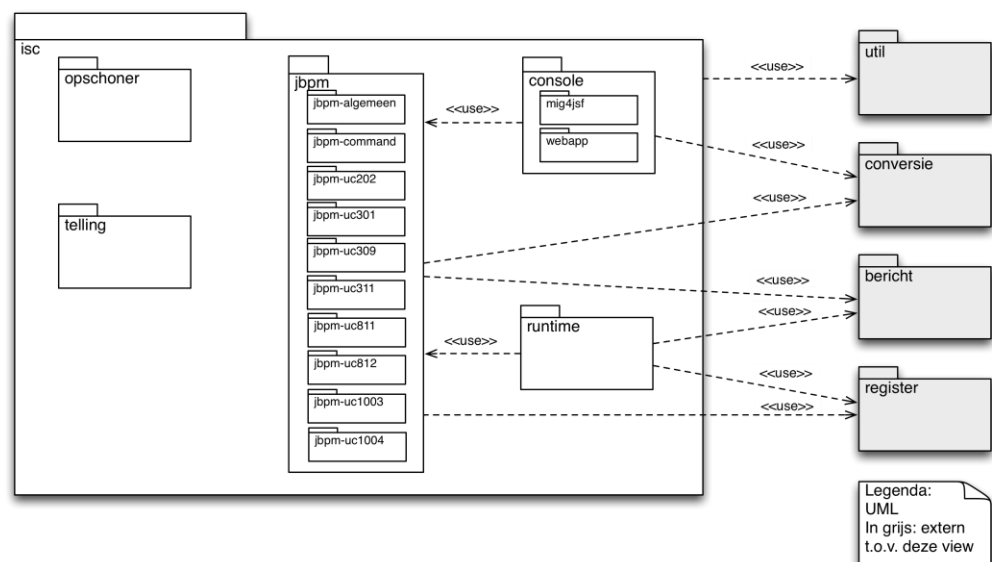
N.v.t.

#### 4.1.8 View-deel 6: Interstelsel Communicatie (ISC)

De ISC module bevat de bedrijfslogica van de migratiesoftware. Hier vindt men de functionaliteit m.b.t. verschillende ISC use cases die betrekking hebben op de interstelsel communicatie en synchronisatie van de migratiesoftware. Deze module orkestreert berichtenstromen tussen GBA gemeentes en afnemers enerzijds en de centrale BRP voorziening anderzijds. Daarnaast levert deze module rapportage functionaliteit en beheerfunctionaliteit voor de toekomstige beheerder van de migratiesoftware.

Deze orkestratie logica is uitgewerkt in JPDL diagrammen die worden uitgevoerd door een jBPM service die als onderdeel van een Spring applicatie is gedeployed op een applicatieserver. Daarnaast bestaat ISC uit maatwerk software die meestal vanuit jBPM processen wordt aangeroepen.

##### 4.1.8.1 Primaire presentatie



**Afbeelding 7 Module View ISC**

##### 4.1.8.2 Element catalogus

###### 4.1.8.2.1 Opschoner

Eigenschappen	<i>Verantwoordelijkheid</i> Het periodiek opschonen van management informatie m.b.t. jBPM processen en berichten in de ISC database.
Relaties	Zie Afbeelding 7
Interfaces	n.v.t.
Gedrag	n.v.t.

###### 4.1.8.2.2 Console

Eigenschappen	<i>Verantwoordelijkheid</i> Levert rapportage- en beheerfunctionaliteit voor de migratiesoftware. Deze webapplicatie is gebaseerd op de standaard jBPM console met daaraan maatwerk software toegevoegd.
Relaties	Zie Afbeelding 7
Interfaces	n.v.t.

Gedrag	n.v.t.
--------	--------

## 4.1.8.2.3 JBPM

Eigenschappen	<p><i>Verantwoordelijkheid</i></p> <p>Deze submodule bevat algemene functionaliteit ter ondersteuning van de JBPM processen. Deze functionaliteit bestaat o.a. uit:</p> <ul style="list-style-type: none"> <li>• maatwerk acties;</li> <li>• integratie met ISC specifieke tabellen;</li> <li>• spring integratie;</li> <li>• foutafhandeling;</li> <li>• locking;</li> <li>• logging.</li> </ul> <p>Daarnaast bevat deze module de functionaliteit om de verschillende ISC processen te executeren.</p>
Relaties	Zie Afbeelding 7
Interfaces	n.v.t.
Gedrag	n.v.t.

## 4.1.8.2.4 Runtime

Eigenschappen	<p><i>Verantwoordelijkheid</i></p> <p>Bevat functionaliteit voor het afhandelen van berichten die van en naar ISC worden gestuurd en de integratie van JMS met jBPM.</p>
Relaties	Zie Afbeelding 7
Interfaces	Deze zijn beschreven in de proces view.
Gedrag	n.v.t.

## 4.1.8.2.5 Telling

Eigenschappen	<p><i>Verantwoordelijkheid</i></p> <p>Het periodiek bijwerken van management informatie m.b.t. verwerkte jBPM processen en berichten in ISC. Deze informatie wordt opgeslagen in de ISC database en kan d.m.v. de console worden ingezien.</p>
Relaties	Zie Afbeelding 7
Interfaces	n.v.t.
Gedrag	n.v.t.

## 4.1.8.3 Context diagram

De module view van de migratiesoftware toont de ISC module in zijn context. Zie Afbeelding 2 voor een weergave van deze context.

## 4.1.8.4 Variabiliteit

N.v.t.

## 4.1.8.5 Motivatie

Voor elke use case binnen ISC is een submodule die de specifieke proceslogica voor die use case bevat. Generieke functionaliteit die door meerdere processen worden gebruikt bevinden zich in ondersteunende submodules. Daarnaast zijn er nog submodules die beheerfunctionaliteit bevatten voor de processen.

#### 4.1.8.6 Interfaces

N.v.t.

## 4.2 Implementation View

### 4.2.1 Omschrijving

Deze view maakt inzichtelijk welke maven projecten in de ontwikkelomgeving horen bij de verschillende modules uit de logical view. Naast deze mapping van software artefacten op modules beschrijft deze view hoe wordt ontwikkeld en hoe de werkzaamheden aan modules over de verschillende teams zijn verdeeld. De naamgeving van de maven projecten voldoet aan een conventie (zie 4.2.6.3) zodat meteen duidelijk is op welke (sub)module het project betrekking heeft.

### 4.2.2 Primaire presentatie N.v.t.

### 4.2.3 Element catalogus

Maven project	Module	Sub projecten
migr-bericht	bericht	migr-bericht-model
migr-conversie	conversie	migr-conversie-model migr-conversie-regels
migr-ggo	ggo-viewer	migr-ggo-viewer
migr-init	init	migr-init-logging migr-init-logging-model migr-init-logging-runtime migr-init-logging-verschilanalyse migr-init-naarbrp migr-init-naarisc migr-init-naarlo3 migr-init-naarvoisc
migr-isc	isc	migr-isc-console migr-isc-console-mig4jsf migr-isc-console-webapp migr-isc-database migr-isc-jbpm migr-isc-jbpm-algemeen migr-isc-jbpm-command migr-isc-jbpm-uc202 migr-isc-jbpm-uc301 migr-isc-jbpm-uc309 migr-isc-jbpm-uc311 migr-isc-jbpm-uc811 migr-isc-jbpm-uc812 migr-isc-jbpm-uc1003 migr-isc-jbpm-uc1004 migr-isc-opschoner migr-isc-runtime migr-isc-telling
migr-register	isc	migr-register-client
migr-routering	isc	migr-routering-runtime
migr-synchronisatie	synchronisatie	migr-synchronisatie-dal migr-synchronisatie-runtime
migr-util	util	migr-util-common migr-util-excel
migr-voisc	voisc	migr-voisc-database migr-voisc-runtime migr-voisc-spdl

Maven project	Module	Sub projecten
		migr-voisc-mailbox-client

4.2.4 *Context diagram*  
N.v.t.

4.2.5 *Variabiliteit*  
N.v.t.

4.2.6 *Motivatie*  
De projecten zijn op een zodanige wijze opgedeeld dat zij een weergave zijn van de verdeling in modules en sub-modules en hun onderlinge hiërarchie. Een uitzondering hierop zijn de test projecten. Deze zijn alleen onderdeel van de ontwikkel- en testomgeving en worden niet ge-deployed op productie.

4.2.6.1 *Werkwijze*  
Bij het ontwikkelen van de migratiesoftware dienen de onderstaande principes in acht genomen te worden.

Principe 1: Minimaliseer afhankelijkheden

Zorg ervoor dat de impact van een wijziging zo lokaal mogelijk is. Beperk daarom de afhankelijkheden tussen modules, packages en classes.

Principe 2: Maximaliseer cohesie

Een class, package of module dient een eenduidige verantwoordelijkheid te hebben. De code binnen een class, package of module dient daarom alleen hieraan bij te dragen. Splits daarom software elementen op indien deze meerdere verantwoordelijkheden hebben. Hou hierbij rekening met principe 1.

Principe 3: Vermijd herhaling

Als dezelfde code op meerdere plaatsen voorkomt wordt het moeilijker om code aan te passen omdat deze code duplicaten impliciet afhankelijk van elkaar zijn. Pas je de code op één plaats aan dan moet je vervolgens nagaan of de andere plaatsen ook moeten worden aangepast. Ditzelfde principe geldt ook voor documentatie.

Principe 4: Logica en data horen bij elkaar

Bij wijzigingen van de data moet waarschijnlijk ook de logica worden aangepast probeer daarom deze zo dicht mogelijk bij elkaar te houden.

Principe 5: Schrijf "zelf-documenterende" code

Code wordt veel vaker gelezen dan geschreven. Het is daarom een goede investering om leesbare code te produceren. Bovendien maakt men makkelijker fouten in slecht leesbare code. Het is makkelijk om ingewikkelde code te maken, maar het kost een stuk meer denkwerk om code te vereenvoudigen tot zijn essentie.

Naast deze algemene principes worden ook de volgende tactieken toegepast:

- Continuous integration;
- Test driven development;
- Code is pas af als alle testen draaien, coverage voldoende is en gereviewed is door een collega;
- Gebruik altijd een versiebeheersysteem voor code en documentatie;

- Gebruik prototypes om onzekerheden op te lossen en te leren, niet om code te produceren;
- Maak code geschikt voor concurrency;
- Gebruik code analyse tools zoals checkstyle, PMD en Sonar, en de configuratie zoals die voor Operatie BRP is vastgesteld;
- Pas de "boy scout"<sup>1</sup> regel toe;
- Zorg dat een bug maar één keer wordt gevonden. Start altijd met het schrijven van een geautomatiseerde test die de bug reproduceert voordat je de bug oplost;
- Refactor snel en vaak, maak hier een gewoonte van. Uitstellen van refactoren creëert technische schuld die op een later moment niet meer eenvoudig in te lossen is;
- Zorg dat je code voldoet aan de code richtlijnen van Operatie BRP (bijvoorbeeld de richtlijnen voor defensief programmeren);
- Vermijd handmatige procedures, maak gebruik van shell scripts waar mogelijk;
- Zorg dat eisen traceerbaar zijn in de code. Hierbij kan gebruik gemaakt worden van speciale annotaties (@Requirement) of door duidelijke naamgeving;
- Gebruik alleen de Nexus repository op modernodam voor het binnenhalen van third-party libraries.

#### 4.2.6.2

##### Ontwikkelomgeving

De volgende hulpmiddelen worden gebruikt bij het ontwikkelen van de migratiesoftware:

- Maven, voor het bouwen en managen van Java code;
- Java SDK 1.8.x;
- Eclipse IDE for Java EE Developers (of een andere IDE);
- jPDL designer voor jBPM;
- Git;
- Jira;
- Confluence;
- Nexus;
- pgAdmin (of andere database tooling);
- PostgreSQL 9.x.

#### 4.2.6.3

##### Maven structuur

De maven projecten zoals die hierboven zijn beschreven, zijn volgens de onderstaande regels gestructureerd:

- de maven root pom definieert dependencies en versies voor de gehele migratiesoftware;
- de module pom definieert dependencies die alleen gelden binnen een module;
- de submodule pom definieert hoe een project gebouwd en gemanaged wordt;
- de modules pom definieert profielen voor het bouwen van verzamelingen projecten.

De namen van de projecten voldoen aan de volgende conventie:

<sup>1</sup> Met het toepassen van de "boy scout" regel binnen softwareontwikkeling wordt bedoeld dat men bij het wijzigen van broncode deze netter achterlaat dan de staat waarin deze is aangetroffen.

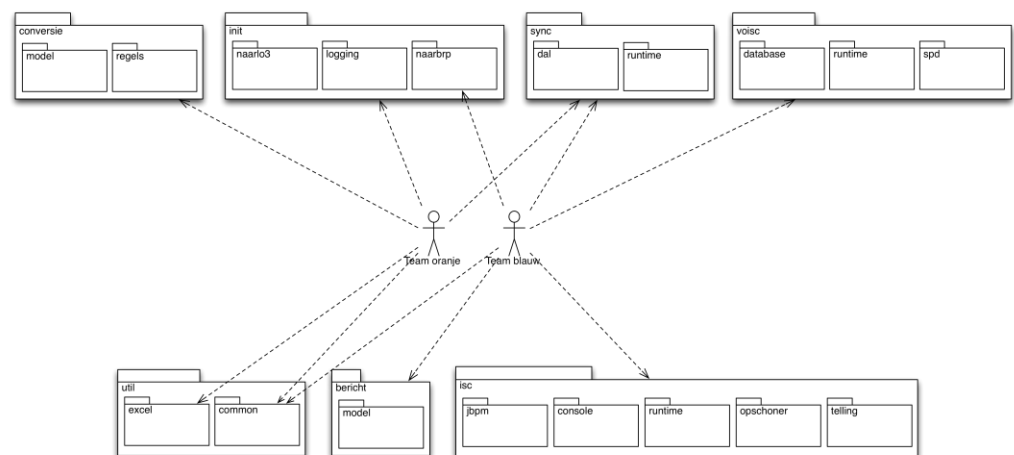


- migr-[module], bijvoorbeeld: migr-synchronisatie  
of
- migr-[module]-[submodule], bijvoorbeeld: migr-synchronisatie-dal

## 4.2.6.4

## Werkverdeling

Er zijn meerdere ontwikkelteams binnen het programma die werken aan de migratiesoftware. Omdat de materie van de migratiesoftware complex is en de inwerktijd daardoor lang is, is afgesproken het werk te verdelen zodat de kennis die een team opdoet efficiënt ingezet kan worden bij de werkzaamheden.



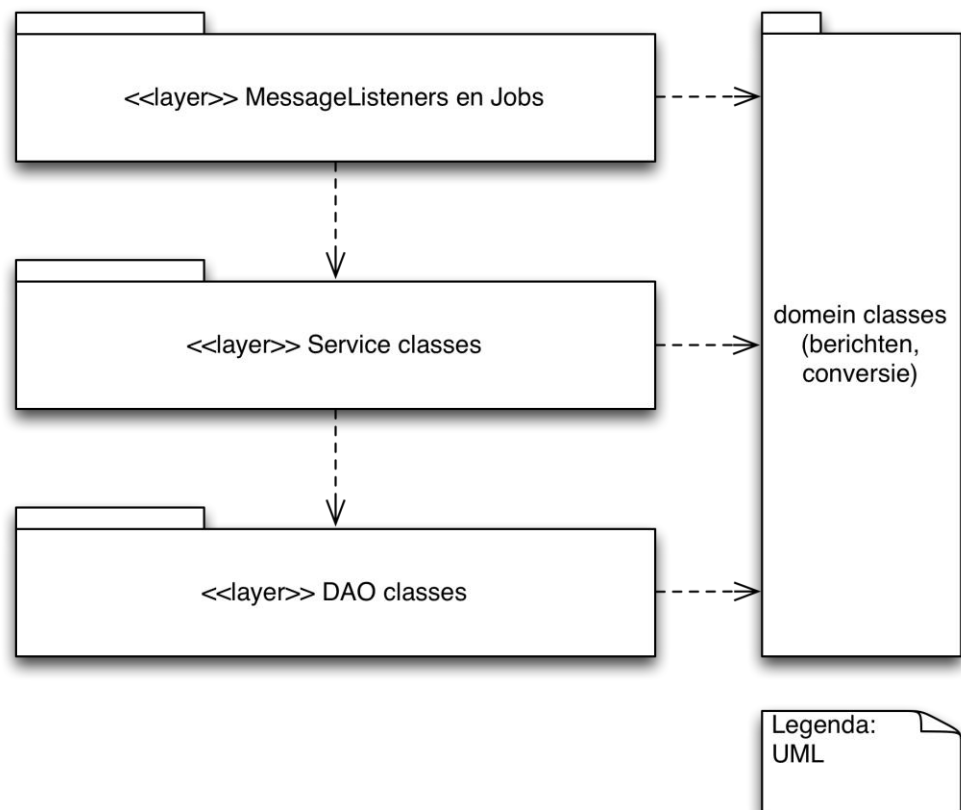
**Afbeelding 8 Werkverdeling migratiesoftware**

In Afbeelding 8 is te zien dat aan sommige (sub)modules twee teams werken. Dit diagram kan daarom gebruikt worden voor teams om in te schatten voor welke werkzaamheden extra afstemming tussen teams nodig is.

## 4.2.6.5

## Lagen in de architectuur

De migratiesoftware bestaat uit geautomatiseerde bedrijfsprocessen die worden uitgevoerd binnen het jBPM component. Deze processen maken gebruik van services die verdeeld zijn over een aantal componenten en maatwerk software (actions) die direct vanuit processen kunnen worden aangeroepen. Voor het maatwerk geldt dat de classes verdeeld zijn over lagen, waarbij er restricties gelden m.b.t. de afhankelijkheden tussen classes o.b.v. de laag waarin de class hoort. Over het algemeen geldt dat een class gebruik mag maken van alle classes uit dezelfde laag en de laag eronder. Uitzonderingen zijn domein classes, hulp classes en expliciete 'callback' mechanismen om vanuit lagere lagen te communiceren met hogere lagen.



**Afbeelding 9 Lagen view van componenten migratiesoftware**

#### 4.2.6.5.1 Messagelisteners en Jobs

De bovenste laag bestaat uit de integratie-laag met de runtime omgeving, zoals MessageListeners voor het ontvangen van berichten uit een JMS queue, Quartz jobs voor functionaliteit die periodiek moet worden uitgevoerd. Voor deze classes geldt dat er een nette scheiding moet zijn tussen de interface en implementerende class.

#### 4.2.6.5.2 Service classes

De service classes voor het uitvoeren van de business logica. Hierbij geldt dat methoden uit service classes de kleinste atomaire operaties zijn die in hun geheel worden uitgevoerd of in zijn geheel niet worden uitgevoerd wanneer er fouten optreden. Ook hier geldt dat services bestaan uit een interface en een implementerende class.

#### 4.2.6.5.3 DAO classes

Deze classes leveren functionaliteit om entiteiten in de database op te slaan, te wijzigen of te laden. Ook hier geldt dat deze classes opgesplitst worden in een interface en implementerende class.

#### 4.2.6.5.4 Domein classes

De domein classes bestaan uit bericht classes en classes van het migratiemodel die als basis dienen voor de conversie. Deze classes mogen geen gebruik maken van classes uit de verschillende lagen maar de lagen mogen wel gebruik maken van de domein classes.

- 4.2.6.6      Code richtlijnen  
De code richtlijnen die gehanteerd moeten worden zijn bevat in de checkstyle en PMD configuratie. Voor de richtlijnen die niet kunnen worden uitgedrukt in checkstyle of PMD regels zijn aanvullende beschrijvingen op de modernodam Wiki[10] geplaatst.
- 4.2.6.7      Ontwikkelaarshandleiding  
Voor het opzetten van een nieuwe ontwikkelomgeving is een ontwikkelaarshandleiding gemaakt. Deze is te vinden op de modernodam Wiki[11].
- 4.2.7        *Interfaces*  
N.v.t.

## **4.3        Proces View**

### *4.3.1        Omschrijving*

De proces view toont een decompositie van de migratiesoftware in executeerbare eenheden. Deze componenten en hun onderlinge communicatiestromen geven inzicht in het runtime gedrag, schaalbaarheid en beschikbaarheid van de migratiesoftware.

Deze view kan worden gebruikt om:

- uit te leggen hoe de migratiesoftware werkt;
- als ondersteuning voor softwareontwikkelaars door inzicht te verschaffen in het runtime gedrag van de verschillende componenten;
- als ondersteuning bij het redeneren over de runtime kwaliteitsaspecten zoals prestaties, betrouwbaarheid en beschikbaarheid.

### *4.3.2        View-delen overzicht*

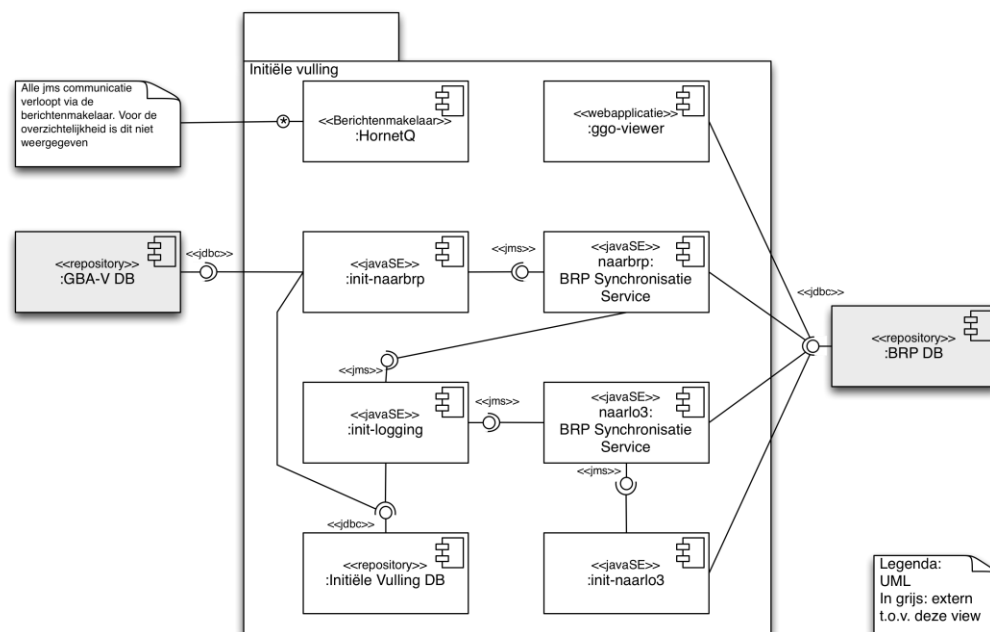
De migratiesoftware bestaat uit twee delen die elk hun eigen toepassing en karakteristieken hebben:

1. initiële vulling;
2. interstelselcommunicatie (ISC) & synchronisatie.

#### 4.3.3 View-deel 1: Initiële vulling

De initiële vulling componenten dienen er samen voor te zorgen dat de circa tweeëntwintig miljoen persoonslijsten, tweeduizend autorisatietabelregels en vierhonderd miljoen afnemerindicaties uit de GBA-V database worden geëxtraheerd zodat ze kunnen worden getransformeerd naar het BRP formaat om vervolgens te worden opgeslagen in de BRP database.

##### 4.3.3.1 Primaire presentatie



**Afbeelding 10 Component & Connector View Initiële Vulling**

##### 4.3.3.2 Element catalogus

###### 4.3.3.2.1 Initiële Vulling Naar BRP (init-naarbrp)

Eigenschappen	<p><b>Functionaliteit</b></p> <p>De functionaliteit van deze component valt in twee stappen uiteen die na elkaar worden uitgevoerd.</p> <ol style="list-style-type: none"> <li>1. Het lezen van het laatst verstuurd Lg01/La01 bericht uit een GBA-V database voor alle personen. Deze berichten worden opgeslagen in de initiële vulling database met daarbij status informatie.</li> <li>2. Elk bericht wordt vervolgens doorgestuurd naar de BRP Synchronisatie Service om te worden verwerkt in de BRP database. Het resultaat van deze actie wordt bijgewerkt in de IV database.</li> </ol> <p><b>Betrouwbaarheid</b></p> <ol style="list-style-type: none"> <li>1. Het lezen van Lg01/La01 berichten uit GBA-V en het opslaan hiervan in de IV database is één query die binnen één database transactie wordt uitgevoerd. De kans op falen is klein met name als de twee logische databases op één fysieke database draaien.</li> <li>2. Het lezen van te versturen berichten uit de IV</li> </ol>
---------------	--

	<p>database en deze vervolgens via JMS versturen naar de BRP Synchronisatie Server is gevoeliger voor fouten door mogelijke problemen met het netwerk en belasting van de berichtenmakelaar. Het bijwerken van de status in de IV database en het versturen van berichten gebeurt in één atomaire actie m.b.v. JTA.</p> <p><i>Prestaties</i> Dit proces is niet horizontaal schaalbaar omdat de daadwerkelijke conversie en opslag van gegevens significant meer verwerkingstijd vergt en in andere componenten plaatsvindt. Dit proces vooral fungeert als 'voeder' van andere verwerkende processen.</p> <p><i>Resource benodigdheden</i> <i>Beveiliging</i> De initiële vulling wordt in een veilige omgeving uitgevoerd. Hiervoor zijn geen extra maatregelen in de software getroffen.</p> <p><i>Concurrency</i> Deze component draait als één proces met één verwerkende thread.</p>
Relaties	Zie Afbeelding 10
Interfaces	<p><i>Gebruikt de volgende interfaces</i></p> <ul style="list-style-type: none"> <li>• GBA-V database, zie 4.3.3.6.9</li> <li>• Initiële vulling database, zie 4.3.3.6.10</li> <li>• SynchroniseerNaarBrpService, zie 4.3.3.6.1</li> <li>• AfnemersindicatieService, zie 4.3.3.6.3</li> <li>• AutorisatieService, zie 4.3.3.6.4</li> <li>• ProtocolleringService, zie 4.3.3.6.7</li> </ul>
Gedrag	N.v.t.

#### 4.3.3.2.2 BRP Synchronisatie Service

Eigenschappen	<p><i>Functionaliteit</i> Deze component biedt functionaliteit voor het converteren en opslaan van personen, afnemerindicaties en autorisaties in de BRP database. Daarnaast biedt hij functionaliteit voor het lezen van personen uit de BRP</p> <p><i>Betrouwbaarheid</i> Doordat deze component communiceert met een database en berichtenmakelaar is deze component gevoelig voor OS en netwerk fouten. Omdat deze component ook binnen de context van ISC wordt gebruikt en daar aan hogere betrouwbaarheidseisen dient te voldoen vindt het verwerken van JMS berichten en bewerkingen op de BRP database binnen één atomaire actie plaats. Dit is opgelost met JTA.</p> <p><i>Prestaties</i> <i>Resource benodigdheden</i> <i>Beveiliging</i> Hiervoor geldt dezelfde argumentatie als voor de Initiële vulling runtime, zie 4.3.3.2.1.</p> <p><i>Concurrency</i> Deze component is zo ontworpen dat hij horizontaal schaalbaar is. Deze component kan parallel en verdeeld</p>
---------------	--

	over meerdere nodes berichten verwerken. Daarnaast is het mogelijk om het aantal verwerkende threads binnen een proces aan te passen. Het voordeel van parallelle threads is dat deze geheugen kunnen delen zoals bijvoorbeeld de hibernate cache. Dit kan schelen in het geheugengebruik. Het voordeel van parallelle processen is de mogelijkheid om de processen over meerdere nodes te verdelen.
Relaties	Zie Afbeelding 10
Interfaces	<p><i>Gebruikt de volgende interfaces</i></p> <ul style="list-style-type: none"> <li>• BRP DB, zie 4.3.3.6.7</li> </ul> <p><i>Levert de volgende interfaces</i></p> <ul style="list-style-type: none"> <li>• SynchroniseerNaarBrpService, zie 4.3.3.6.1</li> <li>• AfnemersindicatieService, zie 4.3.3.6.3</li> <li>• AutorisatieService, zie 4.3.3.6.4</li> <li>• LeesUitBrpService, zie 4.3.3.6.2</li> <li>• ProtocolleringService, zie 4.3.3.6.7</li> </ul> <p>Dit component biedt meerdere interfaces maar alleen die relevant zijn voor initiële vulling zijn hier opgesomd.</p>
Gedrag	N.v.t.

#### 4.3.3.2.3 Initiële Vulling Logging (init-logging)

Eigenschappen	<p><i>Functionaliteit</i></p> <p>Deze component leest berichten van twee queues. Namelijk de queue met daarop antwoordberichten op synchronisatieberichten en de queue met daarop antwoorden van LeesUitBrp berichten. Afhankelijk van het antwoordbericht doet deze component het volgende:</p> <ul style="list-style-type: none"> <li>• het resultaat (OK/FOUT) van een synchronisatie naar de BRP database, inclusief eventuele foutmelding wordt opgeslagen in de initiële vulling database;</li> <li>• het verwerken van een antwoord op het LeesUitBrp bericht met daarin het resultaat van een conversie van een persoon uit de BRP naar een persoonslijst in LO3-formaat. Dit verwerken bestaat uit het uitvoeren van een verschillenanalyse tussen deze persoonslijst en de oorspronkelijke persoonslijst uit het GBA die geleid heeft tot deze persoon in de BRP. Daarnaast worden de eventuele verschillen opgeslagen in de initiële vulling database.</li> </ul> <p><i>Betrouwbaarheid</i></p> <p>Hiervoor geldt dezelfde argumentatie als voor de Initiële vulling runtime, zie 4.3.3.2.1.</p> <p><i>Prestaties</i></p> <p><i>Resource benodigdheden</i></p> <p><i>Beveiliging</i></p> <p>Hiervoor geldt dezelfde argumentatie als voor de Initiële vulling runtime, zie 4.3.3.2.1.</p> <p><i>Concurrency</i></p>
---------------	--

	Hiervoor geldt dezelfde argumentatie als voor de BRP Synchronisatie Service, zie 4.3.3.2.2.
Relaties	Zie Afbeelding 10
Interfaces	<i>Gebruikt de volgende interfaces</i> <ul style="list-style-type: none"> <li>• Initiële vulling database, zie 4.3.3.6.10</li> </ul> <i>Levert de volgende interfaces</i> <ul style="list-style-type: none"> <li>• LogSynchronisatieResultaatService, zie 4.3.3.6.5</li> <li>• LogTerugConversieResultaatService, zie 4.3.3.6.6</li> </ul>
Gedrag	N.v.t.

#### 4.3.3.2.4 Initiële Vulling Database

Eigenschappen	<p><i>Functionaliteit</i> De opslag van logging informatie m.b.t. de initiële vulling in een PostgreSQL DBMS zodat hierop analyses kunnen worden uitgevoerd m.b.t. de kwaliteit van de conversie en initiële vulling. Informatie de wordt opgeslagen bestaat uit:</p> <ul style="list-style-type: none"> <li>• de LO3 berichten die uit de GBA-V database zijn geëxtraheerd;</li> <li>• de status van de verwerking van de LO3 berichten;</li> <li>• het resultaat van de terugconversie van persoon uit de BRP database;</li> <li>• het verschil tussen de LO3 persoonslijst die initieel gevuld is in de BRP en de LO3 persoonslijst die het resultaat is van de terugconversie;</li> <li>• de te autorisatie en afnemerindicaties uit de GBA-V database.</li> </ul> <p><i>Betrouwbaarheid</i> Zie hiervoor de PostgreSQL documentatie.</p> <p><i>Prestaties</i> Zie hiervoor de PostgreSQL documentatie.</p> <p><i>Resource benodigdheden</i> Zie hiervoor de PostgreSQL documentatie.</p> <p><i>Beveiliging</i> Hiervoor geldt dezelfde argumentatie als voor de Initiële vulling runtime, zie 4.3.3.2.1.</p> <p><i>Concurrency</i> Zie hiervoor de PostgreSQL documentatie.</p>
Relaties	Zie Afbeelding 10
Interfaces	<i>Levert de volgende interfaces</i> <ul style="list-style-type: none"> <li>• Initiële vulling database, zie 4.3.3.6.10</li> </ul>
Gedrag	N.v.t.

#### 4.3.3.2.5 Initiële Vulling Naar LO3 (naarlo3)

Eigenschappen	<p><i>Functionaliteit</i> Selecteert uit de BRP database alle administratienummers van de ingeschreven personen en stuurt per administratienummer een bericht naar de BRP Synchronisatie Service met het verzoek deze persoon uit de BRP te lezen en terug te converteren naar het LO3 formaat.</p>
---------------	---



	<p><i>Betrouwbaarheid</i></p> <p>1. Het lezen van te verwerken BRP personen en het vervolgens sturen van JMS berichten t.b.v. de terug-conversie is gevoeliger voor fouten door mogelijke problemen met het netwerk en belasting van de berichtenmakelaar. Het versturen van alle berichten gebeurt niet in één atomaire actie vanwege het grote aantal berichten dat wordt gegenereerd.</p> <p>Het betreft hier de initiële vulling die wordt uitgevoerd in een 'laboratorium' opstelling waarbij de condities waaronder dit wordt uitgevoerd nauwkeurig te beïnvloeden zijn en technische fouten eenvoudig gedetecteerd kunnen worden in de logbestanden. Er zijn geen extra maatregelen genomen om het effect van technische fouten te beperken. Vanwege het karakter van de initiële vulling is het eenvoudiger om na een initiële vulling de log te controleren op technische fouten en in dit geval het resultaat als niet betrouwbaar te beschouwen waarna eventuele problemen kunnen worden opgelost en de initiële vulling eventueel opnieuw kan worden uitgevoerd.</p> <p><i>Prestaties</i></p> <p><i>Resource benodigdheden</i></p> <p><i>Beveiliging</i></p> <p>Hiervoor geldt dezelfde argumentatie als voor de Initiële vulling runtime, zie 4.3.3.2.1.</p> <p><i>Concurrency</i></p> <p>Hiervoor geldt dezelfde argumentatie als voor de Initiële vulling runtime, zie 4.3.3.2.1.</p>
Relaties	Zie Afbeelding 10
Interfaces	<p><i>Gebruikt de volgende interfaces</i></p> <ul style="list-style-type: none"> <li>• BRP DB, zie 4.3.3.6.7</li> <li>• SynchroniseerNaarBrpService, zie 4.3.3.6.1</li> </ul>
Gedrag	N.v.t.

#### 4.3.3.2.6 GGO Viewer (ggo-viewer)

Eigenschappen	<p><i>Functionaliteit</i></p> <p>Deze webapplicatie maakt de conversie van LO3 naar BRP inzichtelijk door persoonslijsten in beide formaten te tonen en daarbij de informatie hoe gegevens uit LO3 zijn terechtgekomen in het BRP formaat (tracing). Daarnaast worden de logmeldingen weergegeven die optreden tijdens het conversieproces. Dit hulpmiddel helpt bij het analyseren van de conversie van een persoonslijst. Deze webapplicatie kan gegevens uit een BRP database lezen maar kan ook in een standalone modus worden gebruikt waarbij geen database met daarin persoonsgegevens nodig is maar de gebruiker een LO3 persoonslijst uploadt om vervolgens de conversie van deze persoonslijst te analyseren.</p> <p><i>Betrouwbaarheid</i></p> <p>De kans op falen is afhankelijk van het gebruik van een database. Over het algemeen is de kans op falen klein</p>
---------------	--

	<p>gezien het gebruik van standaard technologie en het kleine aantal concurrent gebruikers en informatie die wordt ingelezen. Daarnaast heeft het ontstaan van een fout niet direct verstrekende consequenties.</p> <p><i>Prestaties</i>  <i>Resource benodigdheden</i>  <i>Beveiliging</i>  <i>Concurrency</i></p> <p>Berichten van gebruikers (http requests) worden gelijktijdig afgehandeld door een pool van worker threads die door Tomcat wordt beheerd. Zie hiervoor de Tomcat documentatie. Vanwege het kleine aantal gebruikers is er geen rekening gehouden met het clusteren van meerdere Tomcat instanties omdat niet een dusdanige belasting van dit systeem is voorzien dat dat nodig zal zijn.</p>
Relaties	Zie Afbeelding 10
Interfaces	<p><i>Gebruikt de volgende interfaces</i></p> <ul style="list-style-type: none"> <li>BRP DB, zie 4.3.3.6.7</li> </ul>
Gedrag	N.v.t.

#### 4.3.3.2.7 Berichtenmakelaar (HornetQ)

Eigenschappen	<p><i>Functionaliteit</i></p> <p>Het beheren van de JMS queues die gebruikt worden voor de informatie-uitwisseling tussen de componenten en de afhandeling van de JMS communicatie tussen de componenten.</p> <p><i>Betrouwbaarheid</i></p> <p>Voor het beheren van de JMS queues dienen berichten te worden opgeslagen. HornetQ slaat berichten op het filesysteem op. Het kan dat berichten sneller op een queue worden geplaatst dan dat ze ervan gelezen worden en daarom kan het aantal berichten dat wordt opgeslagen op het filesysteem groot worden. De kans op falen wordt met name bepaald door de beschikbaarheid van voldoende ruimte op het filesysteem.</p> <p><i>Prestaties</i>  <i>Resource benodigdheden</i>  <i>Beveiliging</i></p> <p>Hiervoor geldt dezelfde argumentatie als voor de Initiële vulling runtime, zie 4.3.3.2.1.</p> <p><i>Concurrency</i></p> <p>De HornetQ berichtenmakelaar is geconfigureerd als stand-alone en non-clustered. Zie voor meer informatie de HornetQ documentatie.</p>
Relaties	Zie Afbeelding 10
Interfaces	N.v.t.
Gedrag	N.v.t.

#### 4.3.3.3

##### Context diagram

Het systeemoverzicht beschrijft de migratiesoftware in relatie tot haar omgeving. Dit is schematisch weergegeven in Afbeelding 1.

## 4.3.3.4 Variabiliteit

Van de componenten die hierboven zijn beschreven is alleen de BRP Synchronisatie Server op verschillende manieren inzetbaar. Deze component wordt ook gebruikt als onderdeel van de ISC configuratie, zie paragraaf 4.3.4. De BRP Synchronisatie Service bestaat uit meerdere services waarvan bij de initiële vulling alleen gebruik gemaakt wordt van de volgende services:

- SynchroniseerNaarBrpService
- LeesUitBrpService
- AfnemersindicatieService
- AutorisatieService
- ProtocolleringService

Welke services beschikbaar zijn kan worden geconfigureerd in een Spring configuratie bestand. Daarnaast dient bij het uitvoeren van deze component te worden opgegeven of deze in de initiële vulling of ISC modus moet draaien.

## 4.3.3.5 Motivatie

Het initiële vulling proces is ontworpen conform de ETL stijl. Daarnaast is er bij het ontwerp rekening gehouden met veranderende eisen qua prestaties en biedt de gekozen oplossing flexibiliteit om de prestaties te verbeteren zonder dat er hiervoor wijzigingen in de software nodig zijn. Hierbij zijn de volgende tactieken toegepast:

1. vergroten van resources: de gekozen oplossing is zowel verticaal als horizontaal schaalbaar. Horizontale schaalbaarheid wordt bereikt doordat parallelle processen over meerdere nodes kunnen worden verdeeld en verticale schaalbaarheid doordat het aantal verwerkende threads binnen één proces kan worden aangepast om zo efficiënt mogelijk gebruik te maken van de beschikbare resources binnen één node;
2. parallelle verwerking: taken binnen het proces die veel verwerkingstijd kosten kunnen parallel worden uitgevoerd.

## 4.3.3.6 Interfaces

## 4.3.3.6.1 SynchroniseerNaarBrpService

Interface identiteit	SynchroniseerNaarBrpService
Resources	
Resource 1	
Syntax	<b>SynchroniseerNaarBrpAntwoordBericht</b> verwerkBericht( <b>SynchroniseerNaarBrpVerzoekBericht</b> )
Semantiek	Verwerkt het binnenkomende lo3 bericht dat eerst wordt geparsed en op syntax- en preconditiefouten wordt gecontroleerd alvorens geconverteerd te worden naar een BRP persoon. Als de conversie naar BRP succesvol is verlopen wordt de persoon in de BRP database opgeslagen.
Foutafhandeling	N.v.t.
Data types/constanten	Class SynchroniseerNaarBrpVerzoekBericht Class SynchroniseerNaarBrpAntwoordBericht
Foutafhandeling	Wanneer er een technische fout optreedt dan wordt een antwoord bericht gestuurd van het type: SynchronisatieFoutBericht. Wanneer er fouten zitten in het verzoek bericht en dit daarom niet verwerkt kan worden

	dan wordt een antwoord bericht gestuurd van het standaard type: SynchroniseerNaarBrpAntwoordBericht met daarin als status: Fout.
Variabiliteit	Deze service wordt gebruikt als onderdeel van initiële vulling maar ook als onderdeel van ISC & Synchronisatie. Deze service kan dus in twee modi opereren.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

## 4.3.3.6.2 LeesUitBrpService

Interface identiteit	LeesUitBrpService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>LeesUitBrpAntwoordBericht</b> verwerkBericht( <b>LeesUitBrpVerzoekBericht</b> )
Semantiek	Zoekt op basis van het binnenkomende bericht naar een BRP persoon in de BRP database en retourneert deze persoon – afhankelijk van het gewenste formaat – als BRP of LO3 persoonslijst.
Foutafhandeling	N.v.t.
Data types/constanten	Class LeesUitBrpVerzoekBericht Class LeesUitBrpAntwoordBericht
Foutafhandeling	Wanneer er een technische fout optreedt dan wordt een antwoord bericht gestuurd van het type: SynchronisatieFoutBericht.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

## 4.3.3.6.3 AfnemersindicatieService

Interface identiteit	AfnemersindicatieService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>AfnemersindicatiesAntwoordBericht</b> verwerkBericht( <b>AfnemersindicatiesBericht</b> )
Semantiek	Ontvangt LO3 afnemerindicaties die vervolgens worden gecontroleerd op preconditioniefouten, om vervolgens geconverteerd te worden in BRP afnemerindicaties. Na een succesvolle conversie worden de BRP afnemerindicaties opgeslagen in de BRP database.
Foutafhandeling	N.v.t.
Data types/constanten	Class AfnemersindicatiesBericht Class AfnemersindicatiesAntwoordBericht
Foutafhandeling	Wanneer er een technische fout optreedt dan wordt een antwoord bericht gestuurd van het type: SynchronisatieFoutBericht. Wanneer er fouten zitten in het verzoek bericht en dit daarom niet verwerkt kan worden dan wordt een antwoord bericht gestuurd van het standaard type: SynchroniseerNaarBrpAntwoordBericht met daarin als status: Fout.

Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

## 4.3.3.6.4 AutorisatieService

Interface identiteit	AutorisatieService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>AutorisatieAntwoordBericht</b> verwerkBericht( <b>AutorisatieBericht</b> )
Semantiek	Ontvangt LO3 autorisaties die vervolgens worden gecontroleerd op preconditioniefouten, om vervolgens geconverteerd te worden in BRP autorisaties. Na een succesvolle conversie worden de BRP autorisaties gemapped op Autorisatietabel en Autorisatiebesluit entiteiten en opgeslagen in de database.
Foutafhandeling	N.v.t.
Data types/constanten	Class AutorisatieAntwoordBericht Class AutorisatieBericht
Foutafhandeling	Wanneer er een technische fout optreedt dan wordt een antwoord bericht gestuurd van het type: SynchronisatieFoutBericht. Wanneer er fouten zitten in het verzoek bericht en dit daarom niet verwerkt kan worden dan wordt een antwoord bericht gestuurd van het standaard type: SynchroniseerNaarBrpAntwoordBericht met daarin als status: Fout.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

## 4.3.3.6.5 LogSynchronisatieResultaatService

Interface identiteit	LogSynchronisatieResultaatService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>void</b> verwerk( <b>SynchroniseerNaarBrpAntwoordBericht</b> )
Semantiek	Zoekt o.b.v. het administratienummer in het bericht naar de initiële vulling logging van personen in de IV database. Vervolgens wordt deze logging aangevuld met de status melding uit dit bericht en opgeslagen in de IV database.
Foutafhandeling	N.v.t.
<i>Resource 2</i>	
Syntax	<b>void</b> verwerk( <b>AfnemersindicatiesAntwoordBericht</b> )
Semantiek	Zoekt o.b.v. het administratienummer in het bericht naar de initiële vulling logging van afnemerindicaties in de IV database. Vervolgens wordt deze logging aangevuld met de status melding uit dit bericht en opgeslagen in de IV database.
Foutafhandeling	N.v.t.
<i>Resource 3</i>	

	Syntax	<b>void</b> verwerk( <b>AutorisatieAntwoordBericht</b> )
	Semantiek	Zoekt o.b.v. het autorisatie-id in het bericht naar de initiële vulling logging van autorisaties in de IV database. Vervolgens wordt deze logging aangevuld met de status melding uit dit bericht en opgeslagen in de IV database.
	Foutafhandeling	N.v.t.
	Data types/constanten	Class SynchroniseerNaarBrpAntwoordBericht Class AfnemersindicatiesAntwoordBericht Class AutorisatieAntwoordBericht Class SynchronisatieFoutBericht
	Foutafhandeling	In het geval van technische fouten worden deze gelogd in een logbestand.
	Variabiliteit	N.v.t.
	Kwaliteitsattributen	N.v.t.
	Motivatie	De motivatie voor de minimale foutafhandeling is gelijk aan de motivatie van de Initiële vulling runtime, zie 4.3.3.2.1.
	Gebruikshandleiding	Zie Technisch Ontwerp Initiële Vulling[4].

## 4.3.3.6.6 LogTerugConversieResultaatService

	Interface identiteit	LogTerugConversieResultaatService
	<i>Resources</i>	
	<i>Resource 1</i>	
	Syntax	<b>void</b> verwerk( <b>LeesUitBrpAntwoordBericht</b> )
	Semantiek	Zoekt o.b.v. het administratienummer in het bericht naar de initiële vulling logging van personen in de IV database. Vervolgens wordt deze logging aangevuld met het resultaat van de terugconversie en wordt er een verschillenanalyse uitgevoerd t.o.v. het oorspronkelijke LO3 bericht. Het resultaat van deze analyse wordt vervolgens ook bij deze logging opgeslagen.
	Foutafhandeling	N.v.t.
	Data types/constanten	Class LeesUitBrpAntwoordBericht Class SynchronisatieFoutBericht
	Foutafhandeling	In het geval van technische fouten worden deze gelogd in een logbestand.
	Variabiliteit	N.v.t.
	Kwaliteitsattributen	N.v.t.
	Motivatie	De motivatie voor de minimale foutafhandeling is gelijk aan de motivatie van de Initiële vulling runtime, zie 4.3.3.2.1.
	Gebruikshandleiding	Zie Technisch Ontwerp Initiële Vulling[4].

## 4.3.3.6.7 ProtocolleringService

	Interface identiteit	ProtocolleringService
	<i>Resources</i>	
	<i>Resource 1</i>	
	Syntax	<b>ProtocolleringAntwoordBericht</b> verwerkBericht( <b>ProtocolleringBericht</b> )
	Semantiek	Converteert de protocollerings gegevens van Lo3 formaat naar BRP formaat en slaat deze op in de BRP database.
	Foutafhandeling	N.v.t.

Data types/constanten	Class ProtocolleringAntwoordBericht Class ProtocolleringBericht
Foutafhandeling	In het geval van technische fouten worden deze gelogd in een logbestand.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp Initiële Vulling[4].

#### 4.3.3.6.8 BRP database

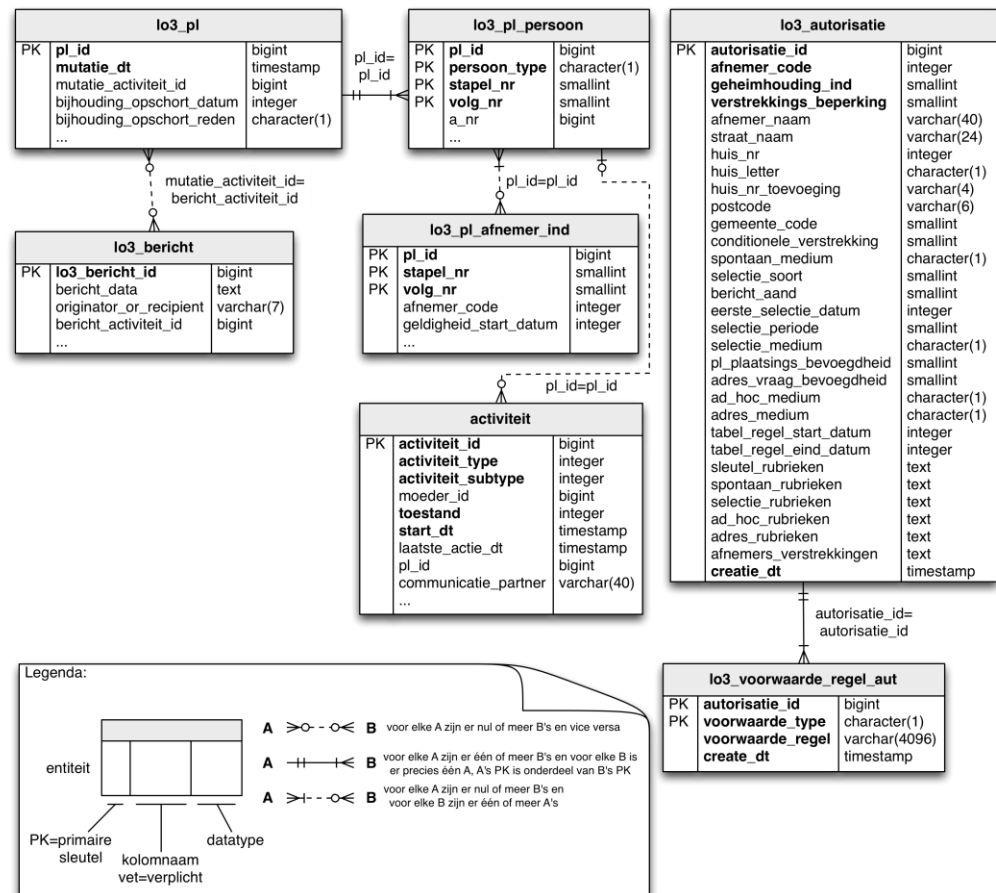
De initiële vulling vult de BRP database met gegevens zoals personen, autorisaties, protocollering en afnemerindicaties. Hiervoor worden de volgende database schema's gebruikt:

- autaut;
- conv;
- ist;
- kern;
- prot;
- verconv.

Meer informatie over het BRP gegevensmodel kan in het Gegevenswoordenboek BRP[19] gevonden worden.

#### 4.3.3.6.9 GBA-V database

Voor het initieel vullen van de centrale BRP voorziening wordt gebruik gemaakt van de GBA-V database. Voor de extractie van relevante informatie wordt slechts een subset van de in GBA-V beschikbare tabellen gebruikt. Deze zijn hieronder weergegeven.

**Afbeelding 11 Entiteit relatie diagram van GBA-V database (gebruikt deel)**

Er zijn een viertal database queries gedefinieerd die gegevens van personen, autorisaties, afnemerindicaties en protocollering uit de GBA-V database extraheren. Deze queries vullen de initiële vulling tabellen, zie hiervoor paragraaf 4.3.3.6.10.

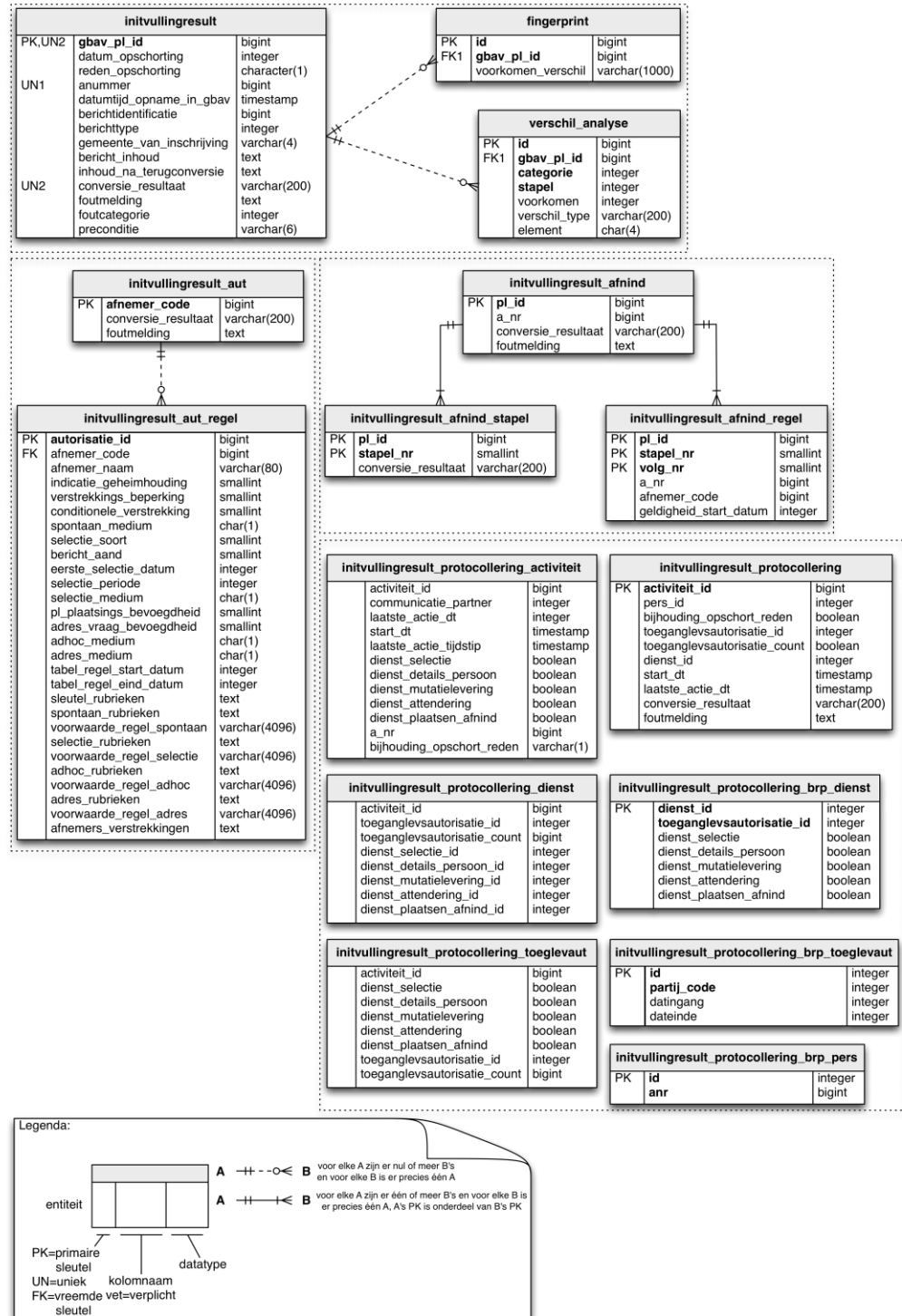
#### 4.3.3.6.10 Initiële vulling database

De logging m.b.t. de voortgang van het initieel vullen van de centrale BRP voorziening en verschillen in geval van een terug-conversie vindt plaats in de initiële vulling database. Hierbij worden in de volgende tabellen gegevens gelogd:

1. initvullingresult, fingerprint en verschil\_analyse: lo3 gegevens van personen die uit de GBA-V database zijn geëxtraheerd, het resultaat van de initiële vulling, terug-conversie en verschillen analyse;
2. initvullingresult\_aut, initvullingresult\_aut\_regel: lo3 gegevens van autorisaties die uit de GBA-V database zijn geëxtraheerd en het resultaat van de initiële vulling;
3. initvullingresult\_afnind, initvullingresult\_afnind\_stapel en initvullingresult\_afnind\_regel: lo3 gegevens van afnemer indicaties die uit de GBA-V database zijn geëxtraheerd en het resultaat van de initiële vulling;
4. initvullingresult\_protocollering\_activiteit, initvullingresult\_protocollering\_brp\_dienst, initvullingresult\_protocollering\_brp\_pers, initvullingresult\_protocollering\_brp\_toeglevaut, initvullingresult\_protocollering\_dienst,



initvullingresult\_protocollering\_toeglevaut en initvullingresult\_protocollering: bevatten de protocolleringsgegevens die uit de GBA-V database zijn geëxtraheerd en het resultaat van de initiële vulling.

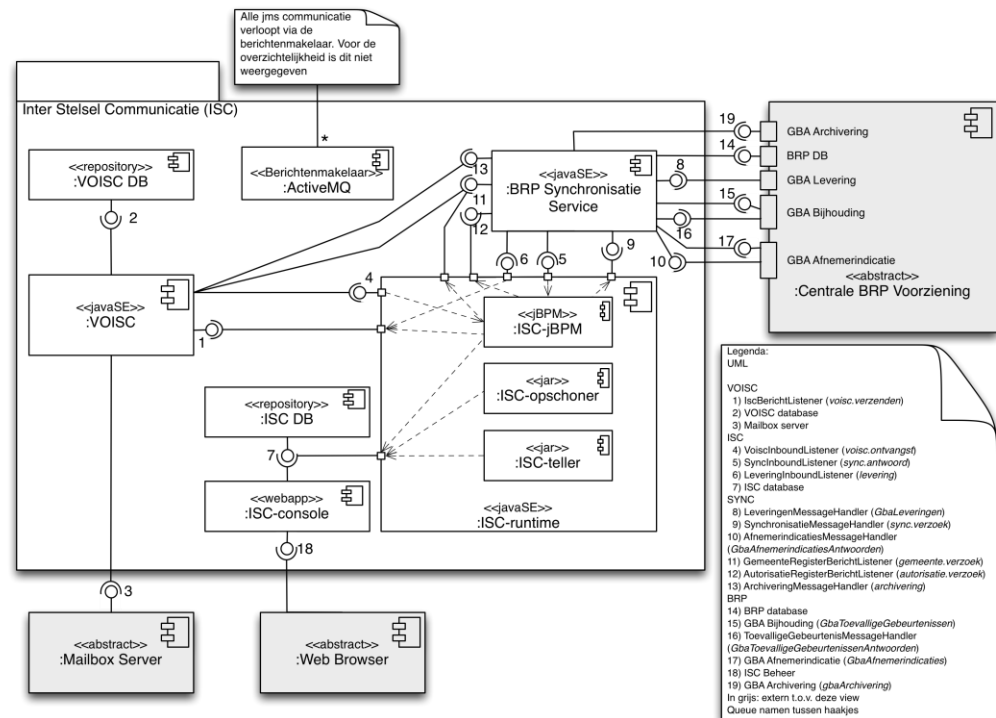


Afbeelding 12 Entiteit relatie diagram van de IV database

#### 4.3.4 View-deel 2: Interstelsel Communicatie (ISC) & Synchronisatie

Dit deel van de proces view beschrijft de componenten van de migratiesoftware die het mogelijk maken om tijdens de duale periode met zowel GBA systemen als BRP systemen bijhoudingen en leveringen te kunnen doen in het BRP stelsel.

##### 4.3.4.1 Primaire presentatie



**Afbeelding 13 Component & Connector View ISC & Synchronisatie**

##### 4.3.4.2 Element catalogus

###### 4.3.4.2.1 VOISC

Eigenschappen	Functionaliteit
	<p>Deze component biedt functionaliteit voor het overbruggen van het 'gat' tussen de wijze waarop met de mailbox server wordt gecommuniceerd en de wijze waarop de migratie componenten berichten uitwisselen. Ook fungeert deze component als "middle-tier" berichten makelaar. Ontvangen berichten worden eerst opgeslagen in de database om vervolgens door een ander proces te worden gelezen en doorgestuurd. Deze functionaliteit kan worden samengevat in de volgende operaties:</p> <ul style="list-style-type: none"> <li>• verzenden van lo3 berichten naar de mailbox server;</li> <li>• ontvangen van lo3 berichten van de mailbox server;</li> <li>• verzenden van isc berichten naar een JMS queue;</li> <li>• ontvangen van isc berichten van een JMS queue;</li> <li>• opslaan van berichten in de VOISC database;</li> <li>• opschonen van oude berichten in de VOISC database.</li> </ul>

	<p><i>Betrouwbaarheid</i></p> <p>De communicatie met de mailbox server ondersteunt geen transacties, in tegenstelling tot de database en JMS berichtenmakelaar. De berichtenverwerking kan in de volgende vier stappen worden verdeeld:</p> <ol style="list-style-type: none"> <li>1. ontvangen van ISC bericht van JMS queue en opslaan in de VOISC database: kan als één atomaire actie worden uitgevoerd d.m.v. JTA;</li> <li>2. lezen van bericht uit de VOISC database, versturen van ISC bericht naar JMS queue en bijwerken van de bericht status in de VOISC database: kan als één atomaire actie worden uitgevoerd d.m.v. JTA;</li> <li>3. ontvangen van LO3 bericht van mailbox server en opslaan in de VOISC database: kan niet als één atomaire actie worden uitgevoerd;</li> <li>4. lezen van bericht uit VOISC database, versturen van LO3 bericht naar de mailbox server: kan niet als één atomaire actie worden uitgevoerd.</li> </ol> <p>Het TO VOISC[8] beschrijft welke maatregelen in de software zijn genomen zodat mogelijk foutsituaties zo goed mogelijk worden afgehandeld.</p> <p><i>Prestaties</i></p> <p><i>Resource benodigdheden</i></p> <p><i>Beveiliging</i></p> <p>Voor wat betreft de communicatie met de mailbox server dient voldaan te worden aan te beveiligingsvoorschriften uit het LO3.X.</p> <p>Het VOISC component maakt gebruik van X.509-certificaten voor het beveiligen van de transport-laag. Om in te loggen in de mailbox server is een wachtwoord nodig. Deze wachtwoorden zijn niet versleuteld opgeslagen in de 'lo3_mailbox' tabel in de VOISC database. Deze tabel staat beschreven in paragraaf 4.3.4.6.2.</p> <p><i>Concurrency</i></p> <p>Deze component is zo ontworpen dat het horizontaal en verticaal schaalbaar is. Deze component kan parallel en verdeeld over meerdere nodes berichten verwerken. Daarnaast is het mogelijk om het aantal verwerkende threads binnen een proces aan te passen. Het voordeel van parallelle threads is dat deze geheugen kunnen delen zoals bijvoorbeeld de hibernate cache. Dit kan schelen in het geheugen gebruik. Het voordeel van parallelle processen is de mogelijkheid om de processen over meerdere nodes te verdelen.</p>
Relaties	Zie Afbeelding 13
Interfaces	<p><i>Gebruikt de volgende interfaces</i></p> <ul style="list-style-type: none"> <li>• VOISC Database, zie 4.3.4.6.2</li> <li>• Mailbox Server, zie 4.3.4.6.3</li> </ul> <p><i>Levert de volgende interfaces</i></p> <ul style="list-style-type: none"> <li>• VoiscGbaBerichtService, zie 4.3.4.6.1</li> </ul>
Gedrag	N.v.t.

## 4.3.4.2.2 VOISC Database

Eigenschappen	<p><i>Functionaliteit</i> De opslag van gegevens in een PostgreSQL DBMS ter ondersteuning van de bericht verwerkende taken van het VOISC component zodat deze taken onafhankelijk van elkaar kunnen worden uitgevoerd.</p> <p><i>Betrouwbaarheid</i> Zie hiervoor de PostgreSQL documentatie.</p> <p><i>Prestaties</i> Zie hiervoor de PostgreSQL documentatie.</p> <p><i>Resource benodigdheden</i> Zie hiervoor de PostgreSQL documentatie.</p> <p><i>Beveiliging</i> <i>Concurrency</i> Zie hiervoor de PostgreSQL documentatie.</p>
Relaties	Zie Afbeelding 13
Interfaces	<p><i>Levert de volgende interfaces</i></p> <ul style="list-style-type: none"> <li>• VOISC Database, zie 4.3.4.6.2</li> </ul>
Gedrag	N.v.t.

## 4.3.4.2.3 ISC-runtime

Eigenschappen	<p><i>Functionaliteit</i> Deze component is het 'zenuwcentrum' van ISC &amp; Synchronisatie. Deze component orkestreert de berichtenuitwisseling die nodig is voor de interstelselcommunicatie en synchronisatie. Deze functionaliteit bestaat uit:</p> <ol style="list-style-type: none"> <li>1. het uitvoeren van langlopende processen waarbij menselijke interactie mogelijk is;</li> <li>2. ondersteunen van berichtenuitwisseling tussen de verschillende ISC componenten;</li> <li>3. rapportages;</li> <li>4. beheerfunctionaliteit.</li> </ol> <p>Hierbij wordt gebruikt gemaakt van de functionaliteit voor het executeren van langlopende bedrijfsprocessen:</p> <ol style="list-style-type: none"> <li>1. jBPM voor het uitvoeren van processen die gedefinieerd zijn in JPDL;</li> <li>2. jBPM Console voor beheerfunctionaliteit en rapportages.</li> </ol> <p><i>Betrouwbaarheid</i> De ISC module voert processen uit door gebruik te maken van de jBPM service. Om te zorgen dat de ISC modules foutsituaties correct kan afhandelen dienen statusovergangen binnen een proces in een transactie te worden uitgevoerd zodat de overgang van een status alleen succesvol is als alle taken die voor deze statusovergang zijn gedefinieerd geslaagd zijn. Binnen jBPM betekent dit concreet dat een proces start door een "signal" (meestal een binnenkomend bericht of actie van een beheerder) en eindigt als de volgende "wait-state" is bereikt of als het proces is afgerond. Gedurende deze uitvoer is er sprake van communicatie met andere</p>
---------------	---

	<p>modules, wordt er informatie in de database opgeslagen en vinden er berichttransformaties plaats. Deze uitvoer dient in zijn geheel te slagen of falen. De ISC module dient een "systeemcrash" te kunnen overleven het systeem daarbij in een correcte status achter te laten zodat deze bij een herstart verder gaat waar het is gebleven.</p> <p><i>Prestaties</i>  <i>Resource benodigdheden</i>  <i>Beveiliging</i>  <i>Concurrency</i></p> <p>Er kunnen meerdere ISC-runtime processen tegelijk draaien, deze verschillende processen dienen wel dezelfde ISC database te gebruiken en berichten makelaar te gebruiken.</p>
Relaties	Zie Afbeelding 13
Interfaces	<p><i>Gebruikt de volgende interfaces</i></p> <ul style="list-style-type: none"> <li>• VoiscIscBerichtListener, zie 4.3.4.6.1</li> <li>• ISC Database, zie 4.3.4.6.8</li> <li>• SynchroniseerNaarBrpService, zie 4.3.3.6.1</li> <li>• LeesUitBrpService, zie 4.3.3.6.2</li> <li>• GemeenteRegisterService, zie 4.3.4.6.10</li> <li>• BlokkeringVerzoekService, zie 4.3.4.6.14</li> <li>• BlokkeringInfoVerzoekService, zie 4.3.4.6.11</li> <li>• DeblokkeringVerzoekService, zie 4.3.4.6.12</li> <li>• AutorisatieRegisterService, zie 4.3.4.6.15</li> <li>• PlaatsAfnemersindicatieService, zie 4.3.4.6.16</li> <li>• VerwijderAfnemersindicatieService, zie 4.3.4.6.17</li> <li>• VerwerkToevalligeGebeurtenisService, zie 4.3.4.6.18</li> <li>• ZoekPersoonOpActueleGegevensService, zie 4.3.4.6.19</li> </ul> <p><i>Levert de volgende interfaces</i></p> <ul style="list-style-type: none"> <li>• IscVoiscInbound, zie 4.3.4.6.5</li> <li>• IscVoiscControlInbound, zie 4.3.4.6.6</li> <li>• IscSyncInbound, zie 4.3.4.6.4</li> <li>• IscLeveringInbound, zie 4.3.4.6.7</li> <li>• IscBeheer, zie 4.3.4.6.13</li> </ul>
Gedrag	N.v.t.

#### 4.3.4.2.4 ISC Database

Eigenschappen	<p><i>Functionaliteit</i></p> <p>De opslag van berichten, management informatie en locking informatie in een PostgreSQL DBMS zodat binnenkomende berichten aan elkaar en aan processen kunnen worden gecorreleerd, processen robuust kunnen worden uitgevoerd en rapportage aan een beheerder kan worden getoond.</p> <p><i>Betrouwbaarheid</i></p> <p>Zie hiervoor de PostgreSQL documentatie.</p> <p><i>Prestaties</i></p> <p>Zie hiervoor de PostgreSQL documentatie.</p> <p><i>Resource benodigdheden</i></p> <p>Zie hiervoor de PostgreSQL documentatie.</p>
---------------	---

	<i>Beveiliging</i> <i>Concurrency</i> Zie hiervoor de PostgreSQL documentatie.
Relaties	Zie Afbeelding 13
Interfaces	<i>Levert de volgende interfaces</i> <ul style="list-style-type: none"> <li>ISC Database, zie 4.3.4.6.8</li> </ul>
Gedrag	N.v.t.

#### 4.3.4.2.5 BRP Synchronisatie Service

Eigenschappen	<i>Functionaliteit</i> Deze component ondersteund ISC bij het uitvoeren van operaties die database intensief zijn. Deze functionaliteit bestaat uit het kunnen lezen en schrijven van personen uit de BRP database en het converteren van LO3 persoonslijsten naar BRP personen en vice versa. <i>Betrouwbaarheid</i> Zie hiervoor de beschrijving van deze component bij initiële vulling in paragraaf 4.3.3.2.2. <i>Prestaties</i> <i>Resource benodigdheden</i> <i>Beveiliging</i> <i>Concurrency</i> Zie hiervoor de beschrijving van deze component bij initiële vulling in paragraaf 4.3.3.2.2.
Relaties	Zie Afbeelding 13
Interfaces	<i>Gebruikt de volgende interfaces</i> <ul style="list-style-type: none"> <li>BRP Database, zie 4.3.4.6.9</li> </ul> <i>Levert de volgende interfaces</i> <ul style="list-style-type: none"> <li>SynchroniseerNaarBrpService, zie 4.3.3.6.1</li> <li>LeesUitBrpService, zie 4.3.3.6.2</li> <li>GemeenteRegisterService, zie 4.3.4.6.10</li> <li>BlokkringVerzoekService, zie 4.3.4.6.14</li> <li>BlokkringInfoVerzoekService, zie 4.3.4.6.11</li> <li>DeblokkeringVerzoekService, zie 4.3.4.6.12</li> <li>AutorisatieRegisterService, zie 4.3.4.6.15</li> <li>PlaatsAfnemersindicatieService, zie 4.3.4.6.16</li> <li>VerwijderAfnemersindicatieService, zie 4.3.4.6.17</li> <li>VerwerkToevalligeGebeurtenisService, zie 4.3.4.6.18</li> <li>ZoekPersoonOpActueleGegevensService, zie 4.3.4.6.19</li> </ul>
Gedrag	N.v.t.

#### 4.3.4.3 Context diagram

Het systeemoverzicht beschrijft de migratiesoftware in relatie tot haar omgeving. Dit is schematisch weergegeven in Afbeelding 1.

#### 4.3.4.4 Variabiliteit

Van de componenten die hierboven zijn beschreven is alleen de BRP Synchronisatie Server op verschillende manieren inzetbaar. Deze component wordt ook gebruikt als onderdeel van de initiële vulling configuratie, zie paragraaf 4.3.3. De BRP Synchronisatie Service bestaat uit meerdere services waarvan bij de ISC alleen gebruik gemaakt wordt van de volgende services:

- SynchroniseerNaarBrpService

- LeesUitBrpService
- GemeenteRegisterService
- BlokkeringVerzoekService
- BlokkeringInfoVerzoekService
- DeblokkeringVerzoekService
- AutorisatieRegisterService
- PlaatsAfnemersindicatieService
- VerwijderAfnemersindicatieService
- VerwerkToevalligeGebeurtenisService
- ZoekPersoonOpActueleGegevensService

Welke services beschikbaar zijn kan worden geconfigureerd in een Spring configuratie bestand. Daarnaast dient bij het uitvoeren van deze component te worden opgegeven of deze in de initiële vulling of ISC modus moet draaien.

#### 4.3.4.5

##### Motivatie

Het ISC en Synchronisatie proces is ontworpen conform de SOA stijl. Een service georiënteerde architectuur bestaat uit een verzameling van gedistribueerde componenten die services leveren en componenten die services gebruiken. Deze componenten zijn redelijk onafhankelijk van elkaar en kunnen door verschillende teams worden ontwikkeld en verschillende technologie gebruiken. ISC bestaat uit de volgende soorten componenten:

1. componenten die services leveren en gebruiken, zoals de mailbox server (extern), centrale BRP voorziening (extern), VOISC en BRP Synchronisatie Service;
2. ISC-runtime voor de routing van berichten tussen service gebruikers en service leveranciers, daarnaast biedt deze component functionaliteit voor het correleren van berichten, logging en integratie met jBPM;
3. orkestratie in de vorm van jBPM die procesdefinities uitvoert wanneer bepaalde gebeurtenissen optreden zoals binnenkomende berichten of periodieke gebeurtenissen;
4. point-to-point asynchrone berichtenverwerkingssysteem, wat het versturen van JMS berichten via queues ondersteund.

Er is gekozen voor de SOA architectuurstijl voor ISC omdat deze de volgende voordelen biedt:

1. het is uitermate geschikt voor de integratie van verschillende systemen;
2. de orkestratie server ondersteund bij langlopende berichtencycli tussen GBA en BRP systemen, inclusief interactie met beheerders;
3. asynchrone berichtenverwerking biedt een grote mate van betrouwbaarheid en schaalbaarheid;
4. verschillende componenten zijn eenvoudig door verschillende teams te ontwikkelen en onderhouden vanwege de lage mate van koppeling tussen de componenten;
5. er is veel software beschikbaar die deze stijl ondersteund waardoor gebruik gemaakt kan worden van bestaande software wat de ontwikkelinspanning ten goede komt.

## 4.3.4.6 Interfaces

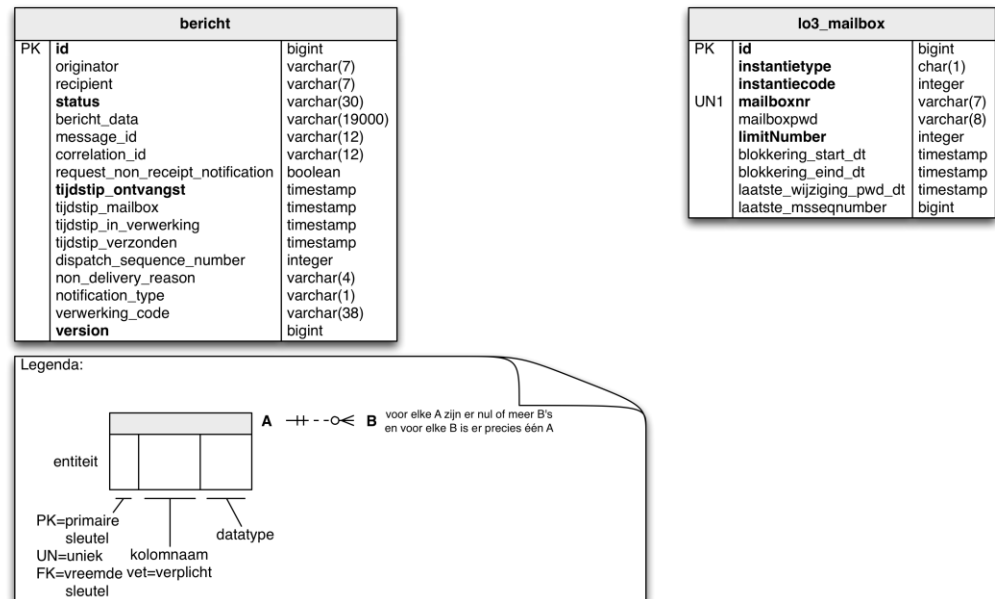
## 4.3.4.6.1 VoiscIscBerichtListener

Interface identiteit	VoiscIscBerichtListener
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>void</b> onMessage( <b>javax.jms.TextMessage</b> )
Semantiek	<p>Verwerkt het binnengekomen JMS TextMessage bericht door deze op te slaan in de bericht tabel in de VOISC database. Hierbij dient het TextMessage de volgende JMS properties te hebben:</p> <ul style="list-style-type: none"> <li>• iscBerichtReferentie, een unieke id die dit bericht identificeert;</li> <li>• iscCorrelatieReferentie, een verwijzing naar de unieke id van een bericht waar dit bericht een antwoord op is;</li> <li>• voaOriginator, de instantiecode van de verzendende partij;</li> <li>• voaRecipient, de instantiecode van de ontvangende partij.</li> </ul> <p>De inhoud van het TextMessage (TextMessage.getText()) is het te versturen LO3 bericht.</p> <p>Het bericht dat wordt opgeslagen in de VOISC database wordt door een ander proces gelezen en verstuurd naar de mailbox server.</p>
Foutafhandeling	N.v.t.
Data types/constanten	javax.jms.TextMessage
Foutafhandeling	Wanneer er een fout optreedt wordt er ook een bericht in de VOISC database opgeslagen maar dan met als inhoud de foutmelding en een andere status wat tot gevolg heeft dat het proces dat normaal gesproken dit bericht doorstuurt naar de mailbox server dit bericht nu terugstuurt naar ISC waar dit foutbericht verder wordt afgehandeld.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp VOISC[8].

## 4.3.4.6.2 VOISC Database

In de VOISC database staan alle gegevens opgeslagen die nodig zijn voor de communicatie tussen de mailbox server en ISC. Hieronder staan de verschillende tabellen weergegeven.





Afbeelding 14 Entiteit relatie diagram van de VOISC database

## 4.3.4.6.3 Mailbox Server

Meer informatie over de GBA mailbox server staat beschreven in het Logisch Ontwerp GBA[15].

## 4.3.4.6.4 IscSyncInbound

Interface identiteit	IscSyncInbound
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>Message process(Message)</b>
Semantiek	Ontvangt antwoord berichten van de BRP Synchronisatie Service. Het binnenkomend JMS Message bericht dient in de body het betreffende antwoordbericht te bevatten in XML. Bij ontvangst van het antwoordbericht door de ESB zal dit XML bericht worden geparsed naar het corresponderen antwoord type. Zie hiervoor de datatypes. Nadat het bericht succesvol is geparsed wordt het proces opgezocht waarin dit antwoordbericht gecorreleerd is. Vervolgens wordt dit bericht als parameter aan dit proces doorgegeven en wordt het proces hervat.
Foutafhandeling	N.v.t.
Data types/constantsen	Class BlokkeringAntwoordBericht Class BlokkeringInfoAntwoordBericht Class DeblokkeringAntwoordBericht Class SynchroniseerNaarBrpAntwoordBericht Class LeesUitBrpAntwoordBericht Class ConverteerNaarBrpAntwoordBericht Class ConverteerNaarLo3AntwoordBericht Class ZoekPersoonAntwoordBericht Class LeesGemeenteRegisterAntwoordBericht Class LeesAutorisatieRegisterAntwoordBericht

	Class VerwerkAfnemersindicatieAntwoordBericht Class VerwerkToevalligeGebeurtenisAntwoordBericht Class OngeldigBericht Class OnbekendBericht
Foutafhandeling	De volgende fouten kunnen optreden: <ul style="list-style-type: none"> <li>- binnenkomend bericht is ongeldig: er wordt een OngeldigBericht doorgestuurd naar het proces en deze voorziet in de verdere foutafhandeling;</li> <li>- binnenkomend bericht is onbekend: er wordt een OnbekendBericht doorgestuurd naar het proces en deze voorziet in de verdere foutafhandeling;</li> <li>- binnenkomend bericht kan niet aan een proces worden gecorreleerd: er wordt een nieuw foutmelding proces gestart en daaraan wordt dit binnenkomend bericht doorgegeven;</li> <li>- technische fout: alle tot nu toe uitgevoerde wijzigingen of verstuurde berichten worden teruggedraaid (rollback) en na een bepaalde periode wordt geprobeerd het binnenkomende bericht opnieuw te verwerken. Na een aantal pogingen zal het bericht worden geparkeerd in een zogenaamde Dead Letter Queue (DLQ) en zal zonder inmengen van een beheerder het bericht niet meer worden verwerkt.</li> </ul>
Variabiliteit	De functionaliteit behorend bij de verwerking van het specifieke bericht is afhankelijk van de definitie van het bijbehorende jBPM proces.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp ISC[7].

## 4.3.4.6.5 IscVoiscInbound

Interface identiteit	IscVoiscInbound
Resources	
Resource 1	
Syntax	<b>Message process(Message)</b>
Semantiek	Deze service ontvangt JMS berichten van het VOISC component, dit zijn berichten die verstuurd zijn door LO3 partijen via de mailbox server of foutberichten als antwoord op een LO3 bericht wat door de VOISC niet verstuurd kan worden. Bij ontvangst van het bericht door de ESB zal dit LO3 bericht worden geparsed naar het corresponderende LO3 type. Zie hiervoor de datatypes. Nadat het bericht succesvol is geparsed wordt het proces opgezocht waaraan dit bericht gecorreleerd is. Als er een bestaande proces instantie wordt gevonden dan wordt dit bericht als parameter aan dit proces doorgegeven en wordt het proces hervat. Anders wordt er een nieuw proces gestart dat hoort bij de afhandeling van het soort binnenkomen LO3 bericht.
Foutafhandeling	N.v.t.
Data types/constanten	Class Ag11Bericht Class Ag21Bericht

	Class Ag31Bericht Class Gv01Bericht Class Gv02Bericht Class Ib01Bericht Class If01Bericht Class If21Bericht Class If31Bericht Class If31Bericht Class Ii01Bericht Class Iv01Bericht Class La01Bericht Class Lf01Bericht Class Lg01Bericht Class Lq01Bericht Class Ng01Bericht Class Pf01Bericht Class Pf02Bericht Class Pf03Bericht Class Tb01Bericht Class Tb02Bericht Class Tf01Bericht Class Tf11Bericht Class Tv01Bericht Class Vb01Bericht Class Wa01Bericht Class Wa11Bericht Class Wf01Bericht
Foutafhandeling	De volgende fouten kunnen optreden: <ul style="list-style-type: none"> <li>- binnenkomend bericht is ongeldig: er wordt een OngeldigBericht doorgestuurd naar het proces en deze voorziet in de verdere foutafhandeling;</li> <li>- binnenkomend bericht is onbekend: er wordt een OnbekendBericht doorgestuurd naar het proces en deze voorziet in de verdere foutafhandeling;</li> <li>- binnenkomend bericht kan niet aan een proces worden gecorreleerd: er wordt een nieuw foutmelding proces gestart en daaraan wordt dit binnenkomend bericht doorgegeven;</li> <li>- technische fout: alle tot nu toe uitgevoerde wijzigingen of verstuurd berichten worden teruggedraaid (rollback) en na een bepaalde periode wordt geprobeerd het binnenkomende bericht opnieuw te verwerken. Na een aantal pogingen zal het bericht worden geparkeerd in een zogenaamde Dead Letter Queue (DLQ) en zal zonder inmengen van een beheerder het bericht niet meer worden verwerkt.</li> </ul>
Variabiliteit	De functionaliteit behorend bij de verwerking van het specifieke bericht is afhankelijk van de definitie van het bijbehorende jBPM proces.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp ISC[7].

## 4.3.4.6.6 IscVoiscControlInbound

Interface identiteit	IscVoiscControlInbound
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>Message</b> process( <b>Message</b> )
Semantiek	Deze service verwacht een JMS bericht met in de body een VoiscBericht met daarin een XML bericht wat geparsed kan worden in een LeesGemeenteRegisterVerzoekBericht. Bij het binnenkomen van het VoiscBericht wordt een nieuw "voisc-control" proces gestart wat het bericht verder afhandelt.
Foutafhandeling	N.v.t.
Data types/constanten	Class LeesGemeenteRegisterVerzoekBericht
Foutafhandeling	De volgende fouten kunnen optreden: <ul style="list-style-type: none"> <li>- binnenkomend bericht is ongeldig: er wordt een OngeldigBericht doorgestuurd naar het proces en deze voorziet in de verdere foutafhandeling;</li> <li>- binnenkomend bericht is onbekend: er wordt een OnbekendBericht doorgestuurd naar het proces en deze voorziet in de verdere foutafhandeling;</li> <li>- binnenkomend bericht kan niet aan een proces worden gecorreleerd: er wordt een nieuw foutmelding proces gestart en daaraan wordt dit binnenkomend bericht doorgegeven;</li> <li>- technische fout: alle tot nu toe uitgevoerde wijzigingen of verstuurd berichten worden teruggedraaid (rollback) en na een bepaalde periode wordt geprobeerd het binnenkomende bericht opnieuw te verwerken. Na een aantal pogingen zal het bericht worden geparkeerd in een zogenaamde Dead Letter Queue (DLQ) en zal zonder inmengen van een beheerder het bericht niet meer worden verwerkt.</li> </ul>
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp ISC[7].

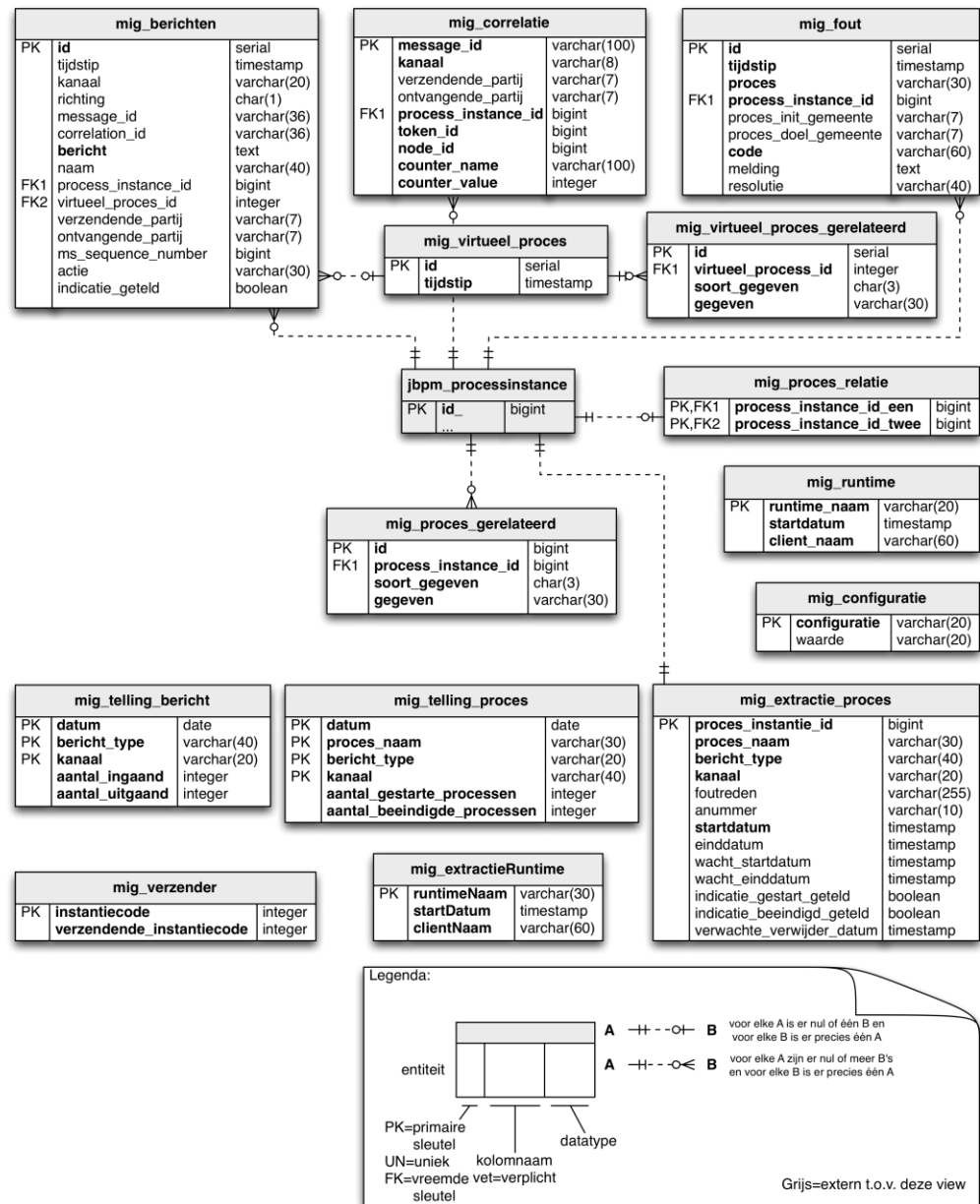
## 4.3.4.6.7 IscLeveringInbound

Interface identiteit	IscLeveringsBerichtService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>Message</b> process( <b>Message</b> )
Semantiek	Deze service verwacht een JMS bericht met in de body een LO3 leveringsbericht. Dit bericht zal door deze service worden doorgestuurd naar het VOISC component die het bericht vervolgens bij de juiste LO3 partij aflevert. In tegenstelling tot de andere ISC services wordt voor dit bericht geen jBPM proces gestart.
Foutafhandeling	N.v.t.
Data types/constanten	N.v.t.
Foutafhandeling	In het geval van een fout bij de verwerking van een

	bericht worden alle tot nu toe uitgevoerde teruggedraaid (rollback) en na een bepaalde periode wordt geprobeerd het binnenkomende bericht opnieuw te verwerken. Na een aantal pogingen zal het bericht worden geparkeerd in een zogenaamde Dead Letter Queue (DLQ) en zal zonder inmengen van een beheerder het bericht niet meer worden verwerkt.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp ISC[7].

#### 4.3.4.6.8 ISC Database

De ISC database bestaat uit tabellen die ondersteunen bij de bericht verwerkende taken van ISC. Dit zijn tabellen die gebruikt worden door maatwerk software van de verschillende ISC sub-modules. Deze tabellen bestaan naast de standaard JBoss tabellen die als onderdeel van het JBoss Enterprise SOA Platform installatie worden aangemaakt. Meer informatie m.b.t. de JBoss tabellen kan men vinden in de JBoss documentatie.



Afbeelding 15 Entiteit relatie diagram van de ISC database

## 4.3.4.6.9 BRP Database

De BRP Synchronisatie Service wijzigt personen in de BRP database. Hiervoor worden de volgende database schema's gebruikt:

- conv;
- ist;
- kern;
- verconv.

Meer informatie over het BRP gegevensmodel kan in het Gegevenswoordenboek BRP[19] gevonden worden.

## 4.3.4.6.10 GemeenteRegisterService

Interface identiteit	GemeenteRegisterService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>LeesGemeenteRegisterAntwoordBericht</b> verwerkBericht( <b>LeesGemeenteRegisterVerzoekBericht</b> )
Semantiek	Deze service levert op verzoek een lijst met gemeentecode, partijcode en datum overgang BRP van alle gemeentes uit de centrale BRP voorziening.
Foutafhandeling	N.v.t.
Data types/constanten	Class LeesGemeenteRegisterVerzoekBericht Class LeesGemeenteRegisterAntwoordBericht
Foutafhandeling	Wanneer er een technische fout optreedt dan wordt een antwoord bericht gestuurd van het type: SynchronisatieFoutBericht.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

## 4.3.4.6.11 BlokkeringInfoVerzoekService

Interface identiteit	BlokkeringInfoVerzoekService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>BlokkeringInfoAntwoordBericht</b> verwerkBericht( <b>BlokkeringInfoVerzoekBericht</b> )
Semantiek	Deze service bepaald of een a-nummer in de centrale BRP voorziening geblokkeerd is. Het resultaat hiervan wordt in het antwoordbericht teruggestuurd.
Foutafhandeling	N.v.t.
Data types/constanten	Class BlokkeringInfoVerzoekBericht Class BlokkeringInfoAntwoordBericht
Foutafhandeling	Wanneer er een technische fout optreedt dan wordt een antwoord bericht gestuurd van het type: SynchronisatieFoutBericht.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

## 4.3.4.6.12 DeBlokkeringVerzoekService

Interface identiteit	DeBlokkeringVerzoekService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>DeBlokkeringAntwoordBericht</b> verwerkBericht( <b>DeBlokkeringVerzoekBericht</b> )
Semantiek	Deze service zal voor het a-nummer uit het verzoekbericht proberen de blokkering in de centrale BRP voorziening op te heffen.
Foutafhandeling	N.v.t.
Data types/constanten	Class DeBlokkeringVerzoekBericht

	Class DeBlokkingAntwoordBericht
Foutafhandeling	<ul style="list-style-type: none"> <li>Als het a-nummer niet geblokkeerd is zal er een SynchronisatieFoutBericht worden teruggestuurd.</li> <li>Als een ander proces het a-nummer probeert te deblokken dan het proces dat het a-nummer heeft geblokkeerd dan wordt er een SynchronisatieFoutBericht teruggestuurd.</li> <li>Als de gemeente waarvoor het a-nummer geprobeerd wordt te deblokken anders is dan de gemeente waarvoor het a-nummer is geblokkeerd dan wordt er een SynchronisatieFoutBericht teruggestuurd.</li> <li>Wanneer er een technische fout optreedt dan wordt een antwoord bericht gestuurd van het type: SynchronisatieFoutBericht.</li> </ul>
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

#### 4.3.4.6.13 IscBeheer

Deze gebruikersinterface stelt een beheerder in staat om de JBPM processen te beheren. Dit beheer bestaat uit het inzien van rapportages m.b.t. de verwerking van processen, het inzien van de status van individuele processen en het invoeren van gegevens voor processen die wachten op een beheerderskeuze. Meer informatie hierover staat in het technisch ontwerp van ISC [7].

#### 4.3.4.6.14 BlokkeringVerzoekService

Interface identiteit	BlokkeringVerzoekService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>BlokkeringAntwoordBericht</b> verwerkBericht( <b>BlokkeringVerzoekBericht</b> )
Semantiek	Deze service zal voor het a-nummer uit het verzoekbericht proberen de persoon met dit a-nummer te blokkeren en stuurt een antwoordbericht met daarin het resultaat en mogelijk blokkeringsinformatie.
Foutafhandeling	N.v.t.
Data types/constanten	Class BlokkeringVerzoekBericht Class BlokkeringAntwoordBericht
Foutafhandeling	Er zal eerst gecontroleerd worden of er reeds een blokkering voor dit a-nummer bestaat, wanneer dit het geval is zal een antwoordbericht worden gestuurd met de status 'GEBLOKKEERD' met daarin de informatie wie en waarom dit a-nummer geblokkeerd is.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].



## 4.3.4.6.15 AutorisatieRegisterService

Interface identiteit	AutorisatieRegisterService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>LeesAutorisatieRegisterAntwoordBericht</b> verwerkBericht( <b>LeesAutorisatieRegisterVerzoekBericht</b> )
Semantiek	Deze service leest uit de BRP alle leveringsautorisatie voor het GBA stelsel en geeft dit terug in het antwoordbericht.
Foutafhandeling	N.v.t.
Data types/constanten	Class LeesAutorisatieRegisterVerzoekBericht Class LeesAutorisatieRegisterAntwoordBericht
Foutafhandeling	N.v.t.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

## 4.3.4.6.16 PlaatsAfnemersindicatieService

Interface identiteit	PlaatsAfnemersindicatieService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>void</b> verwerkBericht( <b>PlaatsAfnemersindicatieVerzoekBericht</b> )
Semantiek	Deze service stuurt een verzoekbericht naar de BRP voor het plaatsen van een afnemer indicatie voor een persoon. Er wordt geen antwoord terug gestuurd omdat de daadwerkelijke afhandeling in de BRP gebeurt en het antwoord asynchroon wordt afgehandeld.
Foutafhandeling	N.v.t.
Data types/constanten	Class PlaatsAfnemersindicatieVerzoekBericht
Foutafhandeling	Fouten tijdens de het doorsturen van het afnemer indicatie bericht worden gelogd.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

## 4.3.4.6.17 VerwijderAfnemersindicatieService

Interface identiteit	VerwijderAfnemersindicatieService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>void</b> verwerkBericht( <b>VerwijderAfnemersindicatieVerzoekBericht</b> )
Semantiek	Deze service stuurt een verzoekbericht naar de BRP voor het verwijderen van een afnemer indicatie voor een persoon. Er wordt geen antwoord terug gestuurd omdat de daadwerkelijke afhandeling in de BRP gebeurt en het antwoord asynchroon wordt afgehandeld.
Foutafhandeling	N.v.t.
Data	Class VerwijderAfnemersindicatieVerzoekBericht

types/constanten	
Foutafhandeling	Fouten tijdens de het doorsturen van het afnemer indicatie bericht worden gelogd.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

## 4.3.4.6.18 VerwerkToevalligeGebeurtenisService

Interface identiteit	VerwerkToevalligeGebeurtenisService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>VerwerkToevalligeGebeurtenisAntwoordBericht</b> verwerkBericht( <b>VerwerkToevalligeGebeurtenisVerzoekBericht</b> )
Semantiek	Deze service transformeert een GBA toevallige gebeurtenis bericht in een BRP bijhoudingsbericht en stuurt dit bericht naar het GBA koppelvlak van de BRP Bijhouding. Het antwoord van de BRP Bijhouding (resultaat) wordt asynchroon ontvangen niet als onderdeel van deze verwerking geretourneerd. Alleen in het geval van een fout wordt een antwoordbericht gestuurd.
Foutafhandeling	N.v.t.
Data types/constanten	Class VerwerkToevalligeGebeurtenisVerzoekBericht Class VerwerkToevalligeGebeurtenisAntwoordBericht
Foutafhandeling	Wanneer er fouten optreden bij het versturen van het bijhoudingsverzoek naar de BRP dan wordt er een VerwerkToevalligeGebeurtenisAntwoordBericht gemaakt met daarin de foutmelding en wordt dit antwoord geretourneerd.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.
Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

## 4.3.4.6.19 ZoekPersoonOpActueleGegevensService

Interface identiteit	ZoekPersoonOpActueleGegevensService
<i>Resources</i>	
<i>Resource 1</i>	
Syntax	<b>ZoekPersoonAntwoordBericht</b> verwerkBericht( <b>ZoekPersoonOpActueleGegevensVerzoekBericht</b> )
Semantiek	Deze service biedt een zoek mogelijkheid naar <i>ingeschreven</i> personen in de BRP op basis van a-nummer, bsn, geslachtsnaam of postcode. De lijst met gevonden personen wordt in het antwoordbericht geretourneerd.
Foutafhandeling	N.v.t.
Data types/constanten	Class ZoekPersoonOpActueleGegevensVerzoekBericht Class ZoekPersoonAntwoordBericht
Foutafhandeling	Er moet minimaal 1 zoekcriteria zijn opgegeven. Als aan deze voorwaarde niet is voldaan wordt een exceptie gelogd en wordt het verzoekbericht genegeerd.
Variabiliteit	N.v.t.
Kwaliteitsattributen	N.v.t.

Motivatie	N.v.t.
Gebruikshandleiding	Zie Technisch Ontwerp BRP Synchronisatie Service[6].

## **4.4 Deployment View**

### **4.4.1 Omschrijving**

De deployment view maakt inzichtelijk aan welke hardware elementen de componenten uit de proces view zijn toegewezen. Deze view bestaat uit software artefacten, hardware nodes en execution nodes.

Deze view kan worden gebruikt om:

- analyses uit te voeren m.b.t. prestaties, beschikbaarheid, betrouwbaarheid en veiligheid;
- integratietesten vorm te geven;
- technisch beheerders te informeren over het gebruik van de hardware;
- kostenramingen te maken voor de aanschaf van hardware.

### **4.4.2 View-delen overzicht**

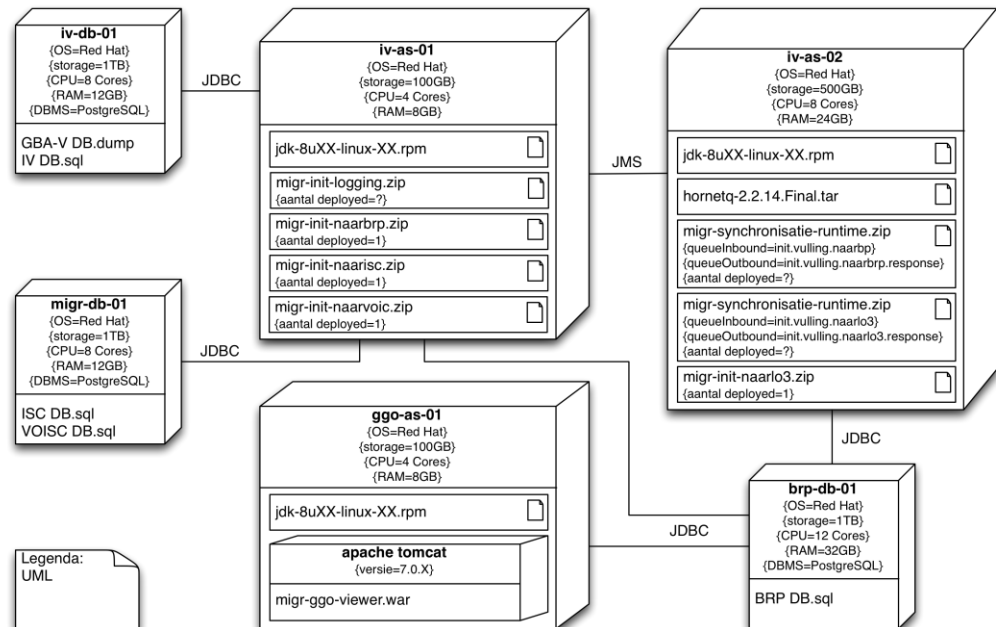
De migratiesoftware bestaat uit twee delen die elk hun eigen toepassing en karakteristieken hebben:

1. initiële vulling;
2. interstelselcommunicatie (ISC) & synchronisatie.

Omdat de deployment van beide delen wezenlijk anders is zijn deze elk in een eigen view-deel beschreven.

#### 4.4.3 View-deel 1: Initiële vulling

##### 4.4.3.1 Primaire presentatie



**Afbeelding 16 Deployment view van de initiële vulling**

##### 4.4.3.2 Element catalogus

###### 4.4.3.2.1 Initiële vulling database server 1 (iv-db-01)

Eigenschappen	<p><i>Omschrijving</i></p> <p>Deze server dient als bron voor de initiële vulling en daarnaast wordt hier logging informatie opgeslagen voor analyse doeleinden. Voordat de initiële vulling gestart kan worden dient hier eerst een dump van de GBA-V productieomgeving te worden geladen. Deze database wordt dus gebruikt om te lezen uit de GBA-V tabellen en te schrijven naar de initiële vulling tabellen.</p>
Relaties	Zie Afbeelding 16.
Interfaces	N.v.t.
Gedrag	N.v.t.

###### 4.4.3.2.2 BRP database server 1 (brp-db-01)

Eigenschappen	<p><i>Omschrijving</i></p> <p>Deze database is geoptimaliseerd voor het initiële vulling proces. Uiteindelijk zal de vulling van deze database gebruikt worden om de database van de centrale BRP voorziening te vullen. De specificaties van deze database zijn dus niet noodzakelijkerwijs gelijk aan die van de BRP productie omgeving. Deze database wordt voornamelijk gebruikt om te schrijven maar ook om te lezen.</p>
Relaties	Zie Afbeelding 16.
Interfaces	N.v.t.

Gedrag	N.v.t.
--------	--------

## 4.4.3.2.3 Initiële vulling applicatie server 1 (iv-as-01)

Eigenschappen	<i>Omschrijving</i> Deze applicatie server wordt belast met het extractieproces (init-naarbrp, init-naarisc, init-naarvoisc) en het logging proces (init-logging). Met name het logging proces zal deze node zwaar belasten omdat dit veel rekenkracht vergt i.v.m. de verschillenanalyse en bovendien parallel kan worden uitgevoerd om zo maximaal gebruik te kunnen maken van de beschikbare resources op deze server.
Relaties	Zie Afbeelding 16.
Interfaces	N.v.t.
Gedrag	N.v.t.

## 4.4.3.2.4 Initiële vulling applicatie server 2 (iv-as-02)

Eigenschappen	<i>Omschrijving</i> Op deze server vindt de conversie plaats van LO3- naar BRP-formaat en vice versa. Daarnaast draaien er meerdere van deze processen parallel en draait op deze server de berichtenmakelaar die de berichten die op de queue staan op het filesysteem opslaat.
Relaties	Zie Afbeelding 16.
Interfaces	N.v.t.
Gedrag	N.v.t.

## 4.4.3.2.5 GGO applicatie server 1 (ggo-as-01)

Eigenschappen	<i>Omschrijving</i> Deze server host de webserver voor de ggo-viewer. De ggo-viewer wordt als web applicatie in apache tomcat gedeployed.
Relaties	Zie Afbeelding 16.
Interfaces	N.v.t.
Gedrag	N.v.t.

## 4.4.3.2.6 Migratie database server 1 (migr-db-01)

Eigenschappen	<i>Omschrijving</i> Deze database wordt gevuld met mailbox gegevens uit de GBA-V database die nodig zijn voor de migratievoorziening om te kunnen communiceren met partijen via de mailbox server.
Relaties	Zie Afbeelding 16.
Interfaces	N.v.t.
Gedrag	N.v.t.

4.4.3.3 Context diagram  
N.v.t.4.4.3.4 Variabiliteit  
N.v.t.

#### 4.4.3.5 Motivatie

De initiële vulling wordt uitgevoerd in een 'laboratorium' opstelling. Dit betekent dat de omstandigheden waaronder de initiële vulling draait voorspelbaar zijn. Bovendien is het hele initiële vulling proces herhaalbaar. Dus als een fout optreedt in een proces of een node uitvalt dan kan het hele proces opnieuw gestart worden. Dit heeft wel gevolgen voor de doorlooptijd van een initiële vulling en moet dus beperkt blijven maar er hoeven hierdoor dus geen kostbare oplossingen te worden ingezet om met deze foutsituaties om te gaan.

Het is daarom niet nodig om voor de initiële vulling gebruik te maken van zogenaamde 'high-availability clusters' om de beschikbaarheid zo hoog mogelijk te houden. Op basis van deze redenering is de volgende ontwerpbeslissing genomen:

*Initiële vulling nodes zijn niet meervoudig uitgevoerd om de beschikbaarheid te vergroten.*

De GBA-V en IV database zijn op een andere node gedeployed dan de BRP database. Dit is gedaan om de prestaties te verhogen en omdat er geen relaties bestaan tussen de GBA-V, IV en BRP database leidt dit niet tot concessies m.b.t. de gegevenskwaliteit. Dit leidt tot de volgende ontwerpbeslissing:

*Er zijn twee nodes die dienen als database server. Eén node waarop de GBA-V en IV databases zijn gedeployed en één database waarop de BRP database is gedeployed.*

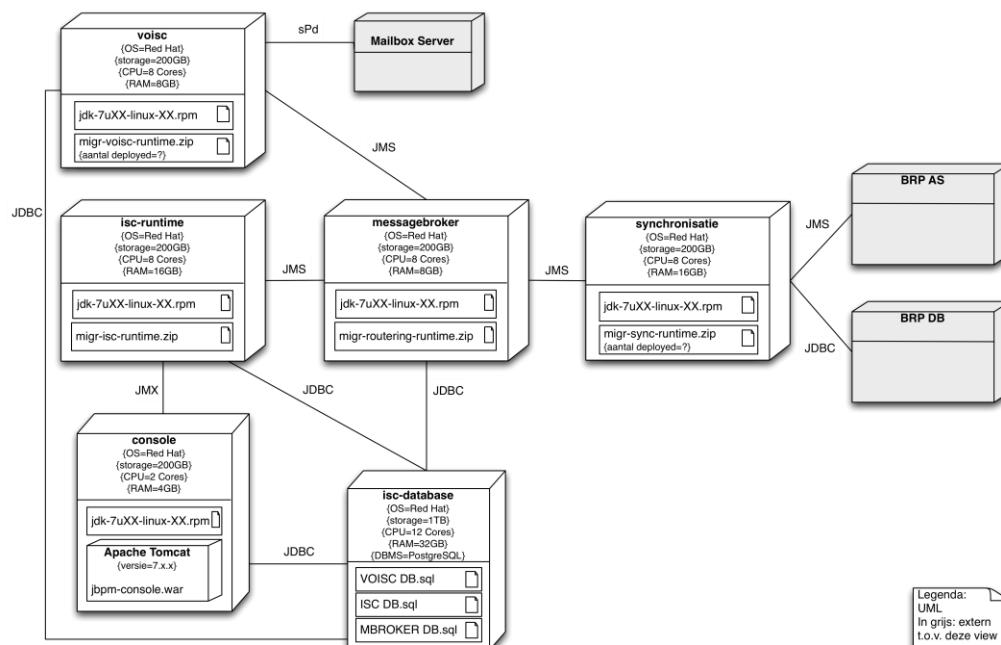
De heen-conversie wordt volledig afgerond voordat de terug-conversie wordt gestart. Onderdeel van het de terug-conversie is het loggen van de verschillen analyse. Beide processen vereisen een grote capaciteit en het is daarom gunstig voor de prestaties om deze te verdelen over twee verschillende nodes. Vandaar de volgende ontwerpbeslissing:

*De processen voor heen-conversie en terug-conversie zijn gedeployed op dezelfde node. Het proces voor logging is op een andere node gedeployed dan het terug-conversie proces.*

#### 4.4.3.6 Interfaces N.v.t.

#### 4.4.4 View-deel 2: Interstelsel communicatie (ISC) & Synchronisatie

##### 4.4.4.1 Primaire presentatie



**Afbeelding 17 Deployment view van ISC & Synchronisatie**

##### 4.4.4.2 Element catalogus

###### 4.4.4.2.1 ISC database server (isc-database)

Eigenschappen	<i>Omschrijving</i> Op deze server zijn alle databases gedeployed van de migratiesoftware.
Relaties	Zie Afbeelding 17.
Interfaces	N.v.t.
Gedrag	N.v.t.

###### 4.4.4.2.2 VOISC applicatie server (voisc)

Eigenschappen	<i>Omschrijving</i> Deze server ontvangt LO3 berichten van de mailbox server en verstuurd deze door naar ISC & Synchronisatie en vice versa. Deze server is dubbel uitgevoerd, zie voisc-as-02.
Relaties	Zie Afbeelding 17.
Interfaces	N.v.t.
Gedrag	N.v.t.

###### 4.4.4.2.3 ISC applicatie server (isc-runtime)

Eigenschappen	<i>Omschrijving</i> Deze server orkestreert de berichten cycli tussen GBA en BRP partijen en verwerkt synchronisatie berichten van
---------------	---



	GBA partijen in de BRP.
Relaties	Zie Afbeelding 17.
Interfaces	N.v.t.
Gedrag	N.v.t.

#### 4.4.4.2.4 BRP Synchronisatie applicatie server (synchronisatie)

Eigenschappen	<i>Omschrijving</i> Op deze server is de BRP Synchronisatie Service gedeployed die de BRP database via een migratie specifieke service ontsluit. Deze server is dubbel uitgevoerd, zie sync-as-02.
Relaties	Zie Afbeelding 17.
Interfaces	N.v.t.
Gedrag	N.v.t.

#### 4.4.4.2.5 Messagebroker

Eigenschappen	<i>Omschrijving</i> Deze server host de activemq messagebroker, dit proces verwerkt alle berichten en zijn informatie op in de database.
Relaties	Zie Afbeelding 16.
Interfaces	N.v.t.
Gedrag	N.v.t.

#### 4.4.4.2.6 Console

Eigenschappen	<i>Omschrijving</i> Deze server host de webserver voor het beheer van de jBPM processen. Hiervoor wordt gebruik gemaakt van de jBPM console.
Relaties	Zie Afbeelding 16.
Interfaces	N.v.t.
Gedrag	N.v.t.

#### 4.4.4.3 Context diagram N.v.t.

#### 4.4.4.4 Variabiliteit N.v.t.

#### 4.4.4.5 Motivatie Er wordt geen kant en klare failover omgeving opgeleverd. De technisch beheerder zal deze voorzieningen zelf moeten inrichten.

*De database wordt niet als cluster opgeleverd en zonder replicatie voorziening.*

Om de beheerder mogelijkheden te bieden m.b.t. het inrichten van een zogenaamde 'high-availability cluster' is het mogelijk dat alle applicatie servers dubbel worden uitgevoerd. Daarnaast zijn de hoge capaciteitseisen een reden om horizontaal schalen mogelijk te maken. Dit leidt tot de volgende ontwerpbeslissing:

*Alleen ISC en Synchronisatie applicatie servers zijn meervoudig uitgevoerd.*

Vanuit veiligheidsoptiek moeten de privileges van servers zoveel mogelijk ingeperkt worden tot het niveau dat noodzakelijk is om correct te functioneren. Door servers een zo'n specifiek mogelijke rol te geven wordt het eenvoudiger om de privileges van een server te managen. Dit heeft geleid tot de volgende ontwerpbeslissing:

*De verschillende soorten componenten worden op een eigen server gedeployed.*

4.4.4.6 Interfaces  
N.v.t.

## 5 Relatie tussen views

Omdat alle views van de softwarearchitectuur de migratiesoftware beschrijven zullen er veel overeenkomsten zijn tussen de views. Dit hoofdstuk geeft inzicht in de relatie tussen de verschillende views zodat de lezer begrijpt hoe de architectuur in zijn geheel werkt. Het beschrijven van relaties (mappings) tussen die views draagt hieraan bij.

Binnen alle views is de naamgeving van elementen consistent doorgevoerd. Op basis van de naam kan daarom de relatie tussen elementen uit views gelegd worden. Toch is het voor sommige views waardevol om dit expliciet te maken. Deze zijn hieronder beschreven.

### 5.1 Proces View ↔ Logical View

De logical view toont modules en de proces view toont componenten. Uit welke componenten een module is opgebouwd is in onderstaande tabel weergegeven.

Component	Module	Sub-module
ggo-viewer	ggo-viewer	n.v.t.
init-naarbrp	init	runtime
init-logging	init	logging
init-naarlo3	init	naarlo3
Initiële vulling DB	-	-
HornetQ	-	-
Berichtenmakelaar	-	-
BRP Synchronisatie Service	conversie	model
		regels
	synchronisatie	runtime
		dal
voisc DB	-	-
voisc	voisc	db
		runtime
		spd
isc DB	-	-
isc-runtime	isc	jbpm
		Jbpm-algemeen
		jbpm-command
		jbpm-uc202
		jbpm-uc301
		jbpm-uc309
		jbpm-uc311
		jbpm-uc811
		jbpm-uc812
		jbpm-uc1003
		opschoner
		telling
isc-console	isc	console-webapp
		console-mig4jsf

**Tabel 1 Relatie tussen componenten en modules**

**5.2****Proces View ↔ Deployment View**

De volgende tabel toont de relatie tussen componenten en de node waarop een component is gedeployed.

Component	Node
ggo-viewer	ggo-as-01
init-naarbrp	iv-as-01
init-logging	iv-as-01
init-naarlo3	iv-as-02
Initiële vulling DB	iv-db-01
HornetQ	iv-as-02
Berichtenmakelaar	
BRP Synchronisatie Service	iv-as-02
	synchronisatie
voisc DB	isc-database
voisc	voisc
isc DB	iv-db-02
	isc-database
isc-runtime	isc-runtime
isc-console	console

**Tabel 2 Relatie tussen componenten en nodes**

## 6 Motivering

In dit hoofdstuk staan alle ontwerpbeslissingen die van toepassing zijn op meerdere views. Daarnaast worden een aantal – voor de architectuur relevante – use cases beschreven en hoe deze van invloed zijn op de architectuur.

### 6.1 Ontwerpbeslissingen

#	Omschrijving
OB.01	De migratiesoftware wordt ontwikkelt m.b.v. Java technologie.
OB.02	De migratiesoftware uit zich in twee systemen: ISC & Synchronisatie en Initiële Vulling.
OB.03	De migratiesoftware dient na de duale periode uitgeschakeld kunnen worden zonder dat de centrale BRP voorziening hierdoor faalt.
OB.04	De migratiesoftware communiceert met GBA partijen via de GBA-mailbox server, daarbij gebruik makend van het sPd protocol.
OB.05	De migratiesoftware communiceert met de centrale BRP voorziening (en vice versa) d.m.v. point-to-point berichten, hierbij gebruik makend van JMS. De uitzondering hierop is het synchroniseren van een Lo3 persoonslijst in de BRP database. Hierbij wordt direct met de BRP database gekoppeld, gebruik makend van JDBC.
OB.06	Persoonsgegevens die vallen onder het LO3 stelsel zijn tenminste van LO3 kwaliteit wanneer deze worden opgeslagen in de BRP database tijdens synchronisatie door de migratiesoftware.
OB.07	De migratiesoftware t.b.v. synchronisatie wijzigen alleen gegevens in de BRP database die onder het LO3 stelsel vallen. <ul style="list-style-type: none"> <li>Bij update van LO3 persoon in BRP en daarbij een update van een relatie tussen de LO3 persoon en BRP persoon, mag de relatie-view van BRP zijde mogelijk ook aangepast worden. Regels hiervoor moeten nog vastgelegd worden.</li> </ul>
OB.08	De BRP werkt intern met de tijdzone '+0000', of te wel: met de tijdzone UTC. Alle tijdstippen vastgelegd in de BRP dienen als UTC tijdstippen te worden geïnterpreteerd. Dus de BRP Synchronisatie Service slaat tijdstippen in de BRP op met tijdzone '+0000'
OB.09	Bij conversie van een datum veld naar een datum/tijd veld zal de datum aangevuld worden met de tijd 01:00 +0000. Dit heeft onder andere invloed op de vertaling van datum opneming (LO3) naar datum/tijd registratie (BRP).
OB.10	Interactie met de BRP database vanuit de migratiesoftware mag uitsluitend geschieden via de BRP Synchronisatie Service.
OB.11	Interactie met de sPd mailbox server vanuit de migratiesoftware mag uitsluitend geschieden via de VOISC.
OB.12	Workflow of proces logica dient geprogrammeerd te worden m.b.v. JPDl voor ISC & Synchronisatie.
OB.13	De software artefacten die als onderdeel van de migratiesoftware worden opgeleverd dienen geschikt te zijn voor een Linux besturingssysteem.
OB.14	Om gegevenskwaliteit te waarborgen dienen de volgende lock regels te worden nageleefd: <ol style="list-style-type: none"> <li>een persoonslijst waarvoor een verhuisproces is gestart door ISC maar nog niet is afgerond, mag niet worden bewerkt door BRP. Dit noemen we <b>blokkeren</b> binnen migratie en de persoonsaanduiding binnen BRP.</li> </ol>

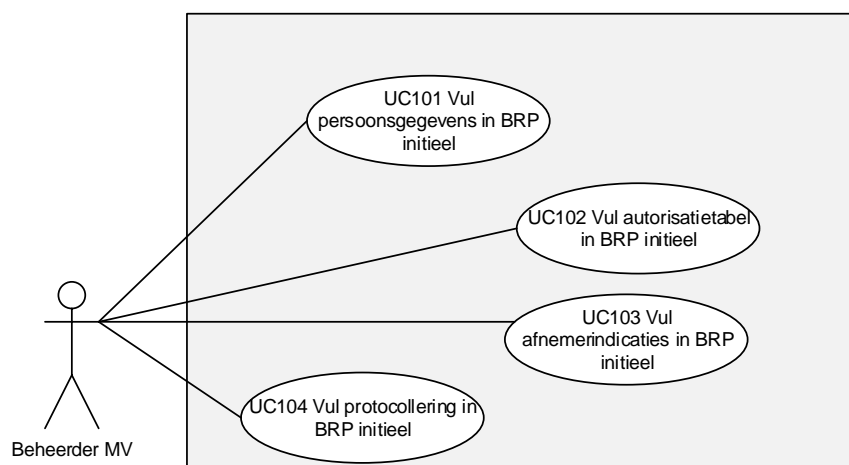
#	Omschrijving
	<ol style="list-style-type: none"> <li>2. binnen ISC mogen er niet meerdere synchronisatie processen lopen die betrekking hebben op dezelfde persoonslijst. Dit noemen we <b>proces locking</b>.</li> <li>3. in de BRP database moeten lees- en schrijfacties op de database die berichten te weeg brengen op een bepaalde manier uit elkaar gehouden worden. Binnen de BRP is hiervoor het <b>logisch locken</b> mechanisme geïntroduceerd. De BRP Synchronisatie Service dient aan te sluiten op dit mechanisme.</li> </ol>
OB.15	Componenten binnen de migratievoorziening slaan wachtwoorden altijd versleuteld op tenzij het wachtwoorden betreft om resources voor services te kunnen benaderen (zoals databases, berichtenmakelaars en mailservers). De opslag van wachtwoorden dient zodanig te zijn gedocumenteerd dat adequate beveiligingsmaatregelen genomen kunnen worden.
OB.16	Alle componenten binnen de migratievoorziening dienen te loggen conform de uitgangspunten zoals deze zijn beschreven in Appendix C: Loggen in de Migratievoorziening.

**Tabel 3 Ontwerpbeslissingen relevant voor alle views****6.2****Relevante Use Cases**

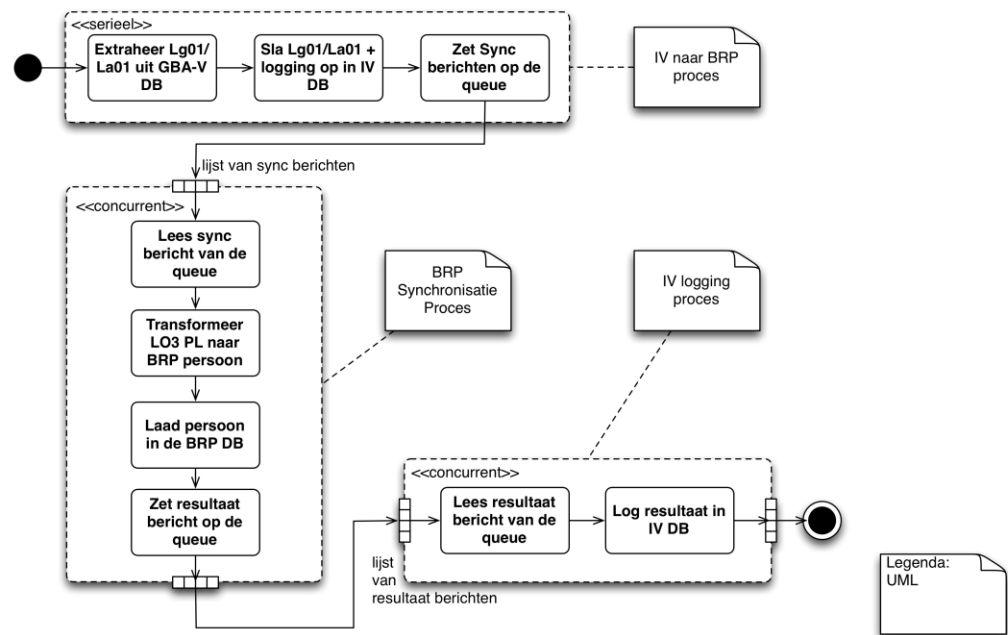
Ter onderbouwing van de gekozen architectuur zijn een aantal belangrijke use cases uitgewerkt. Deze use cases beschrijven het gedrag van het systeem en zijn gebruikt in de analyse voor de eisen aan de architectuur. Met behulp van UML use case diagrammen wordt een overzicht van de verschillende use cases inzichtelijk gemaakt. Voor een gedetailleerde beschrijving van de use cases wordt verwezen naar de corresponderende use case documentatie.

**6.2.1***Use Case 101: Vul persoonsgegevens in BRP initieel*

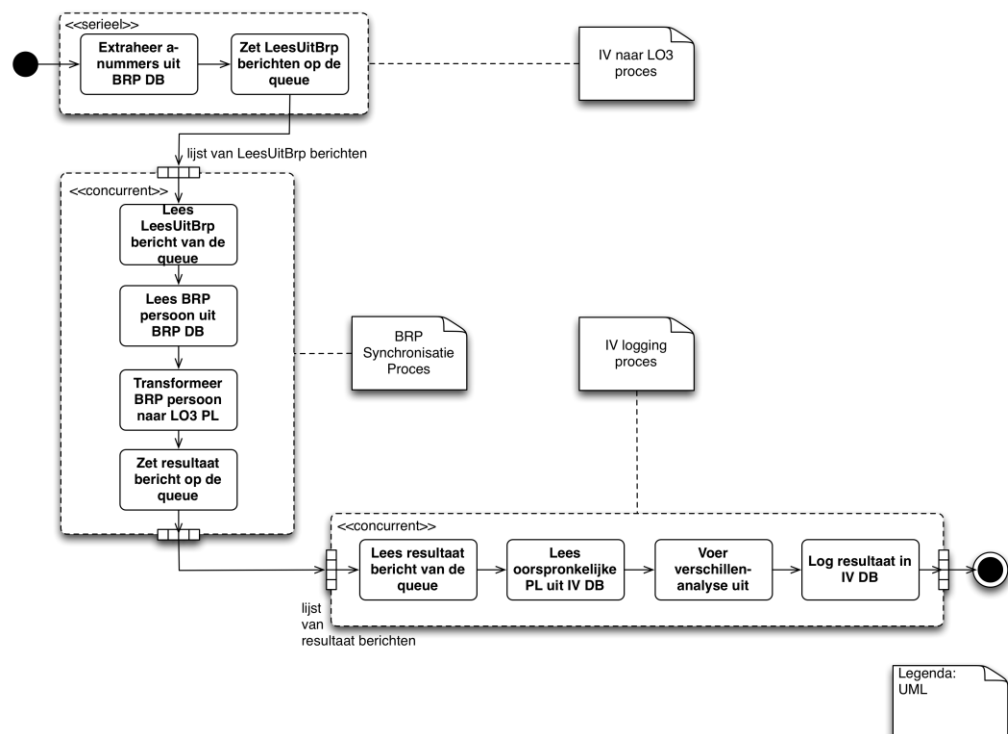
De use case voor het initieel vullen van de centrale BRP voorziening is onderdeel van het GGO use case model. Het model m.b.t. initieel vullen is in Afbeelding 18 weergegeven.

**Afbeelding 18 Use Case Model Initieel vullen BRP**

De initiële vulling bestaat uit twee processen, namelijk de heen-conversie en de terug-conversie. De terug-conversie is vooral bedoeld voor analyse doeleinden en stelt de gebruiker in staat om vergelijkingen uit te voeren tussen de oorspronkelijke LO3 persoonslijsten en het resultaat van een conversie naar BRP en weer terug. Hierbij kan worden onderzocht of er gegevensverlies optreedt tijdens de conversie. Hieronder staat voor elk van deze twee processen een activiteiten diagram die inzicht geeft wijze waarop het proces wordt uitgevoerd.



**Afbeelding 19 Activiteiten diagram van heen-conversie in initiële vulling**

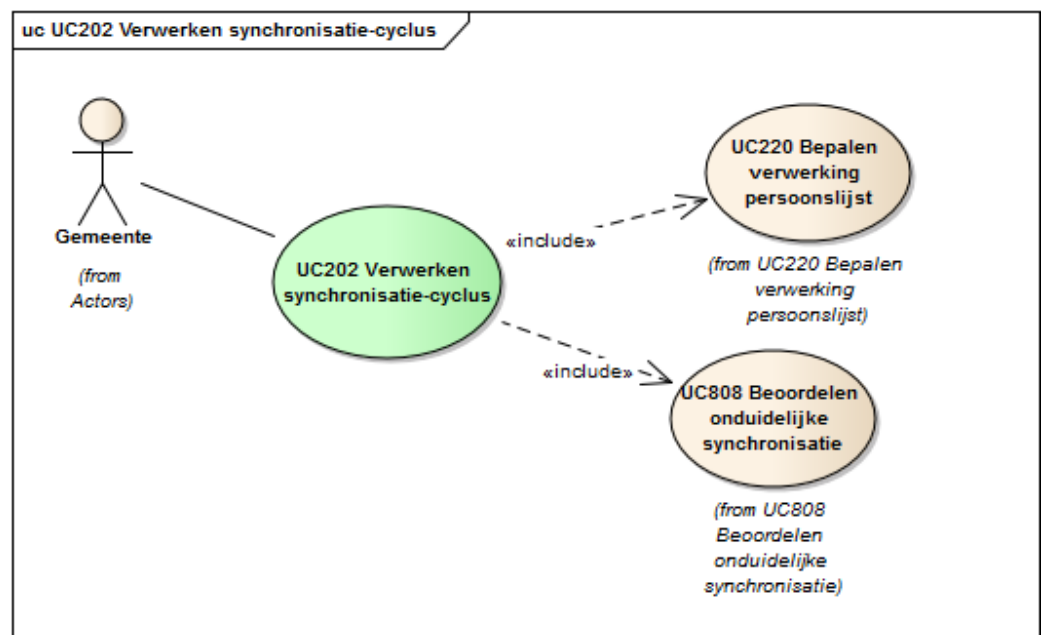


**Afbeelding 20 Activiteiten diagram van terug-conversie in initiële vulling**

## 6.2.2

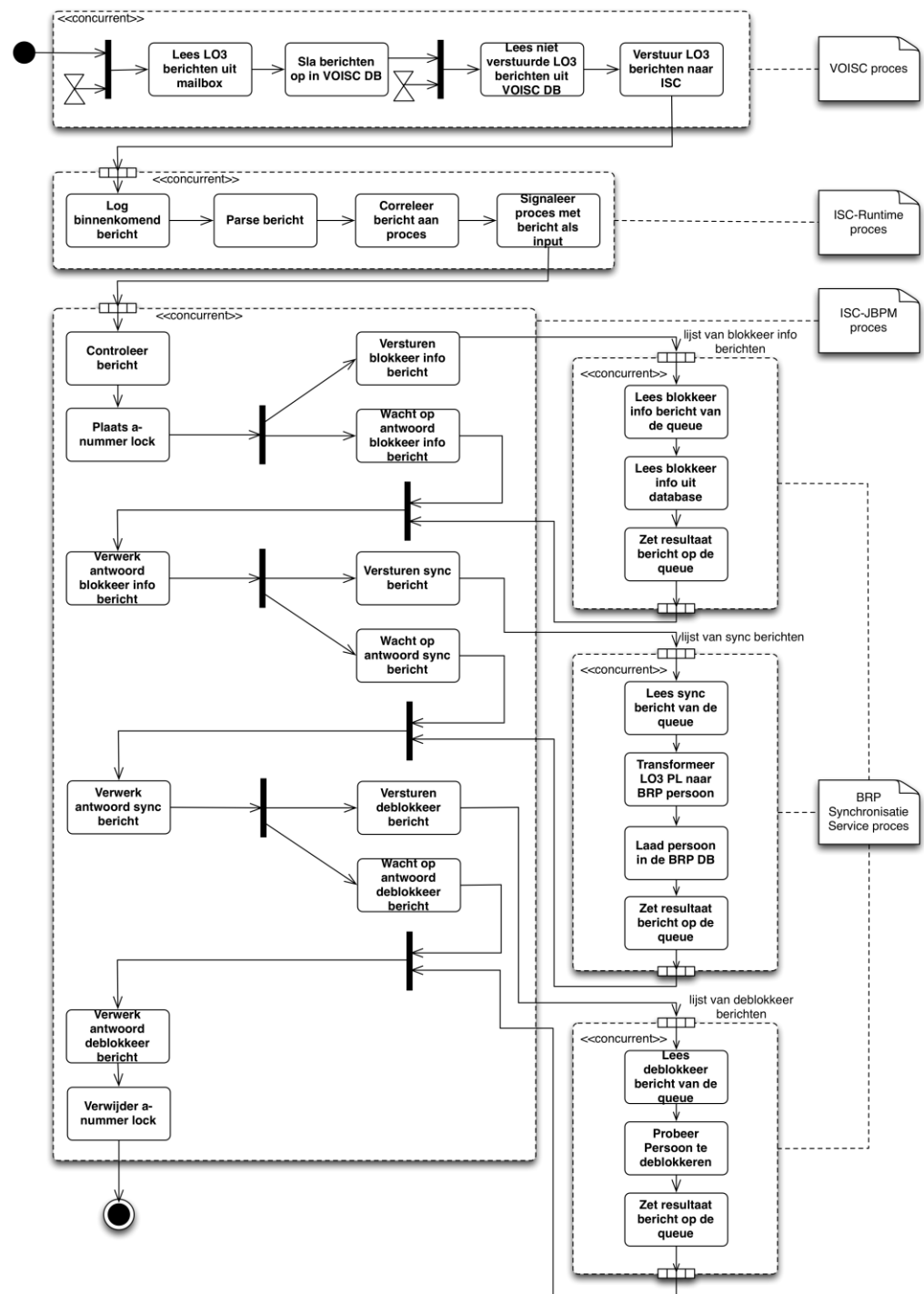
*Use Case 202: Verwerken synchronisatie-cyclus*

Voor stap 2.2 van het BRP Opleveringsplan zal naast het initieel vullen van de centrale BRP voorziening ook het synchroon houden van de centrale BRP voorziening worden opgeleverd. Om inzichtelijk te maken hoe de componenten van ISC & Synchronisatie qua gedrag functioneren is hier use case 202 in een activiteiten diagram uitgewerkt.

**Afbeelding 21 Use Case diagram van verwerken synchronisatie-cyclus**

Het synchronisatie proces start met het ontvangen van een LO3 synchronisatiebericht. Het VOISC component ontvangt het synchronisatiebericht en slaat dit op in de VOISC database. Een ander VOISC proces leest de ontvangen berichten uit de VOISC database en stuurt deze door naar het ISC component. Hier wordt het bericht gelogd en vervolgens gecorreleerd aan een jBPM proces als er sprake is van een vervolg bericht, anders wordt een nieuw jBPM proces gestart. Dit proces wordt uitgevoerd en handelt het bericht verder af. Tijdens het uitvoeren van dit jBPM proces vindt er berichtenuitwisseling plaats met de BRP Synchronisatie Service enerzijds om te bepalen hoe het synchronisatiebericht moet worden verwerkt en anderzijds voor het converteren van de LO3 persoonslijst uit het synchronisatiebericht naar een BRP persoon en de uiteindelijke opslag van deze persoon in de centrale BRP voorziening. Dit proces is schematisch weergegeven in onderstaand activiteiten diagram. Om het diagram overzichtelijk te houden zijn de alternatieve paden hier niet weergegeven.





### Afbeelding 22 Activiteiten diagram van ISC en Synchronisatie

Om de gegevenskwaliteit te waarborgen tijdens het uitvoeren van deze processen dient de locking aanpak voldoende garanties te bieden. Voor stap 2.2 van het BRP opleveringsplan is gekomen tot de aanpak zoals beschreven in Appendix A: Locking aanpak in versie 2.2 van de BRP.

## 7 Woordenlijst

Een lijst met definities van termen die in dit document worden gebruikt.

Term	Omschrijving
Component	Een software element met runtime gedrag. Componenten worden gebruikt om het gedrag van het systeem te beschrijven.
GBA-V	GBA Verstrekkingen (GBA-V) bevat een kopie van alle persoonslijsten uit de GBA. Uit GBA-V worden namens het college van burgemeester en wethouders gegevens verstrekt aan afnemers. [15]
Kwaliteitsattribuut	Een meetbare of testbare eigenschap van een systeem dat wordt gebruikt om aan te duiden hoe goed een systeem de behoeften van de belanghebbenden vervult. Door samen met de belanghebbenden van een systeem een aantal kwaliteitsattributen af te spreken wordt voor de belanghebbenden de kwaliteit van een systeem meetbaar.
La01	Een Synchronisatieantwoord bericht als antwoord op een verzoek (Lq01) bericht. [15]
Lg01	Een Synchronisatiebericht dat wordt verstuurd door een gemeente of de RNI om persoonsgegevens te synchroniseren met de GBA-V. [15]
Module	Een module is een aanduiding voor een groepering van implementatie artefacten die verantwoordelijk is voor een specifiek stuk functionaliteit. Modules worden gebruikt om een decompositie van de functionaliteit van een systeem te kunnen maken, daarnaast vormen ze de basis voor het verdelen van werk tussen ontwikkelteams.
Softwarearchitectuur	De softwarearchitectuur van een systeem is de structuur van dat systeem, bestaande uit software elementen, de extern zichtbare eigenschappen van deze elementen, en de relaties daartussen.

## 8 Afkortingen

Een lijst met belangrijke afkortingen die in dit document worden gebruikt.

Afkorting	Omschrijving
BOP	BRP Opleveringsplan
BPM	Business Process Management
BPMN	Business Process Model and Notation
BRP	Basisregistratie Personen
CRUD	Create Read Update Delete
DBMS	Database Management System
ETL	Extract Transform Load
GGO	Gemeentelijke gegevensoverdracht
ISC	Interstelselcommunicatie
JBM	JBoss Messaging
jBPM	Java for Business Process Management
JMS	Java Message Service
JPDL	jBPM Process Definition Language
JSF	JavaServer Faces
JTA	Java Transaction API
RUP	Rational Unified Process
SAD	Software Architectuur Document
SOA	Service Oriented Architecture
sPd	sPd staat voor "Simple Pd". Het kan beschouwd worden als een tussenvorm tussen de echte Pd (P7) Protocol Data Unit en de binnen de applicatie gehanteerde formaten.
UML	Unified Modeling Language
VOISC	Verzenden & Ontvangen ISC
XML	eXtensible Markup Language

## 9 Referenties

Een opsomming van referenties naar materiaal waarnaar in dit document verwezen wordt.

#	Omschrijving	Organisatie	Versie	Datum
1	Clements, et al., Documenting Software Architecture: Views and Beyond	Addison Wesley		2002
2	Kruchten, P. B. The 4+1 view model of architecture. <i>Software, IEEE</i> , 12(6), 42-50.			1995
3	Bass, L., Clements, P., & Kazman, R. <i>Software architecture in practice</i> .	Addison-Wesley Professional		2003
4	Technisch Ontwerp Initiële Vulling	OperatieBRP		
5	Technisch Ontwerp Conversie	OperatieBRP		
6	Technisch Ontwerp BRP Synchronisatie Service	OperatieBRP		
7	Technisch Ontwerp ISC	OperatieBRP		
8	Technisch Ontwerp VOISC	OperatieBRP		
9	UC202 Verwerken synchronisatie-cyclus	OperatieBRP		
10	Wiki pagina over defensief programmeren <a href="https://www.modernodam.nl/confluence/x/DgBg">https://www.modernodam.nl/confluence/x/DgBg</a>	OperatieBRP		17-05-2014
11	Wiki pagina met ontwikkelaarshandleiding migratie <a href="https://www.modernodam.nl/confluence/x/CQCY">https://www.modernodam.nl/confluence/x/CQCY</a>	OperatieBRP		17-05-2014
13	BRP Opleveringsplan (No-sync) T.b.v. Gartner	OperatieBRP	1.0	19-08-2013
14	Logisch Ontwerp GBA, versie 3.9	BPR	3.9	
15	Logisch Ontwerp BRP	OperatieBRP	0.6	25-04-2014
16	MV Use Case Model ISC	OperatieBRP		
17	Use Case Model Gemeentelijke gegevensoverdracht	OperatieBRP		
18	Gegevenswoordenboek BRP	OperatieBRP		
19	Requirements Prestatie-efficiëntie	OperatieBRP	2.2	
20	Requirements Overdraagbaarheid	OperatieBRP	1.2	
21	Requirements Onderhoudbaarheid	OperatieBRP	1.3	
22	Requirements Documentatie	OperatieBRP	1.3	
23	Requirements Beveiligbaarheid	OperatieBRP	2.5	
24	Requirements Betrouwbaarheid	OperatieBRP	1.1	
25	Technisch Ontwerp Routing	OperatieBRP		
26	Technisch Ontwerp Deltabepaling	OperatieBRP		

## 10 Appendix A: Locking aanpak in versie 2.2 van de BRP

### 10.1 Inleiding

Aan de kwaliteit van de basisregistraties worden hoge eisen gesteld. Het verbeteren van de gegevenskwaliteit van de GBA is daarom één van de doelstellingen van het programma. Deze hoge mate van gegevenskwaliteit uit zich in de functionaliteit van de BRP en in de achterliggende technische componenten. Deze technische componenten gebruiken een locking mechanisme om de gegevenskwaliteit in de database te waarborgen.

De BRP zal in meerdere stappen worden opgeleverd zodat bij iedere volgende stap nieuwe functionaliteit wordt toegevoegd aan de BRP. Dit heeft als voordeel dat men al delen van het nieuwe systeem in productie kan nemen terwijl de ontwikkeling van de BRP nog doorgaat. Deze stappen staan beschreven in het Brp Opleveringsplan (BOP)[14]. Omdat de functionaliteit per stap verschilt is het de vraag of voor elke stap dezelfde locking aanpak nodig is. De beschrijving van de huidige situatie is een momentopname van de migratiesoftware die op het moment van schrijven nog in ontwikkeling is.

### 10.2 Doelstelling

Het beschrijven van een locking aanpak die moet worden toegepast in de software van de migratiesoftware zodat de BRP functionaliteit - die als onderdeel van stap 2.2 van het BOP wordt opgeleverd - de gegevenskwaliteit van persoonsgegevens in de BRP database voldoende kan waarborgen.

### 10.3 Vragen en afbakening

Om te komen tot een locking aanpak dienen de volgende vragen te worden beantwoord:

1. Wat is de achtergrond van de locking problematiek?
2. Is de functionele afbakening o.b.v. BOP stap 2.2 relevant voor de locking aanpak?

### 10.4 Locking problematiek

Aan de BRP worden hoge eisen gesteld m.b.t. gegevenskwaliteit. Dit is van invloed op de functionaliteit, de achterliggende programmatuur en databases. De locking problematiek heeft betrekking op de wijze waarop de BRP software (inclusief migratiecomponenten) gegevens leest en persisteert in de database. Dit dient op een zodanige wijze te gebeuren dat gegevens correct geïnterpreteerd worden en dat gegevens worden opgeslagen zoals dit bedoeld is.

Door het gelijktijdig verwerken van grote hoeveelheden berichten bestaat de mogelijkheid dat het systeem dezelfde gegevens gelijktijdig wil lezen en schrijven en hierdoor inconsistenties optreden. Het voornaamste hulpmiddel hierin zijn database transacties.

Omdat database transacties zo kort mogelijk dienen te zijn (denk aan milliseconden) en een geautomatiseerd bedrijfsproces in verhouding erg lang duurt (denk aan seconden, minuten of dagen!), bestaat een geautomatiseerd bedrijfsproces vaak uit meerdere database transacties. Ondanks dat zijn hierop vaak dezelfde ACID (atomicity, consistency, isolation, durability) eigenschappen gewenst als voor database transacties. Dit is een zogenaamde bedrijfstransactie. Als dit het geval is

dan heeft men te maken met zogenaamde "offline concurrency" en dient men speciaal hiervoor voorzieningen te treffen om de ACID eigenschappen te leveren. Binnen migratie herkennen we deze bedrijfstransacties als jBPM processen, bijvoorbeeld "UC202 Synchroniseer naar BRP". Een belangrijke vraag die dus moet worden beantwoord is:

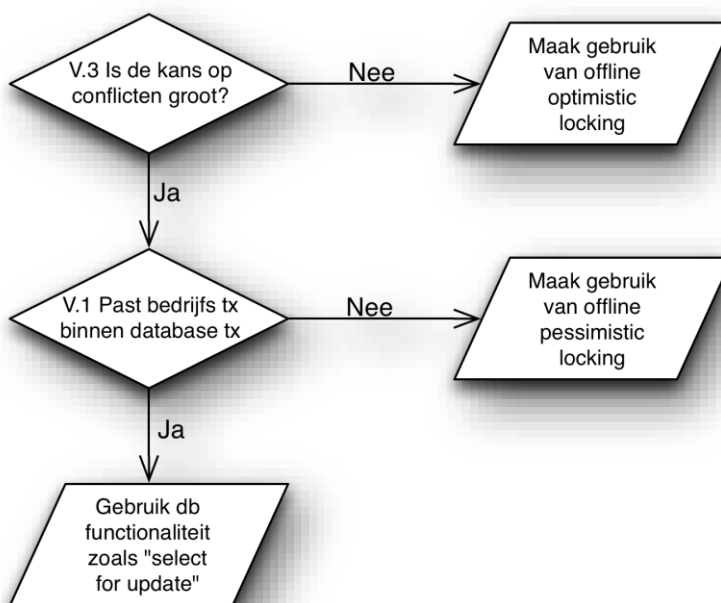
#### *V.1 Passen de bedrijfstransacties binnen een database transactie?*

Als een bedrijfstransactie binnen een database transactie past kan gebruik gemaakt worden van de hulpmiddelen die een database hiervoor biedt. Een voorbeeld hiervan is het "SELECT FOR UPDATE" commando. Daarnaast is het relevant te weten welk isolatie niveau wordt gebruikt:

#### *V.2 Wat is het gebruikte isolatie niveau? (Read Uncommitted, Read Committed, Repeatable Read, Serializable)*

Er zijn twee bekende strategieën om het concurrency probleem op te lossen: optimistisch locken en pessimistisch locken. Omdat optimistisch locken veel eenvoudiger is en hierbij de kans op runtime fouten kleiner is, heeft optimistisch locken de voorkeur, tenzij de kans op conflicten erg groot is. In dit geval is de pessimistische strategie aan te raden. De tweede vraag die dus moet worden beantwoord is:

#### *V.3 Is de kans op conflicten tussen meerdere bedrijfstransacties groot of klein?*



### **Afbeelding 23 Beslisboom voor locking methode**

Wijzigingen op een persoon in het GBA stelsel worden dagelijks gesynchroniseerd. In een synchronisatie bericht worden alle wijzigingen op die persoon gebundeld. De kans dat meerdere processen tegelijkertijd een wijzigingen voor dezelfde persoonslijst verwerken is daarom klein. In geval van een a-nummer wijziging is er kans op een conflict maar deze kans is ook klein gezien het kleine aantal a-nummer

wijzigingen. Tenslotte zal het synchronisatie proces als onderdeel van BOP stap 2.2 geen personen in de BRP relateren en vinden er geen bijhoudingen in de BRP plaats. In latere BOP stappen zal dit gaan veranderen. In BOP stap 3.1 wordt door de BRP geleverd en wordt er dus gelezen door BRP software componenten uit de BRP database waarin door migratiecomponenten worden geschreven. In BOP stap 4.3 zal er ook door BRP software componenten worden geschreven in de BRP database. Kortom, na BOP stap 2.2 veranderen de uitgangspunten en zal een andere locking aanpak nodig zijn.

## 10.5 BOP stap 2.2

Stap 2.2 in het BOP heet "BRP database schaduwdraaien" en deze stap volgt op stap 2.1 "Initiële vulling. De software die als onderdeel van stap 2.2 wordt opgeleverd bestaat dus uit een BRP database met migratie applicatie componenten. Omdat er nog geen BRP applicatiecomponenten worden opgeleverd bestaat er in stap 2.2 geen kans op conflicten op wijzigingen van gegevens in de BRP database. Het enige proces dat als onderdeel van BOP stap 2.2 wordt opgeleverd en gegevens uit de BRP database schrijft of leest is "UC202 Synchroniseer naar BRP". Dit proces valt uiteen in de volgende taken:

- controleer Ig01 of Ia01 bericht -> leest stamgegevens uit de BRP;
- controleer op blokkering ivm verhuizing -> leest gegevens gerelateerd aan persoon (!);
- sync naar brp -> leest en schrijft persoonsgegevens (!);
- blokkering ivm verhuizing aanpassen -> leest en schrijft gegevens gerelateerd aan persoon (!).

Taak 2, 3 en 4 zijn afhankelijk van elkaar en dienen dus binnen een bedrijfstransactie te worden afgehandeld. Omdat de database transacties per taak worden gemanaged en niet over het gehele proces liggen past deze bedrijfstransactie niet in een database transactie.

## 10.6 Conclusie

Omdat de kans op conflicten tussen UC202 processen klein is heeft optimistisch locking de voorkeur.

Echter, de huidige locking implementatie voor UC202 is pessimistisch. Deze variant is complexer dan optimistisch locking en heeft niet de voorkeur. Maar omdat deze optie reeds geïmplementeerd is, is het de vraag of men dit zou moeten wijzigen. Dit hangt af van de gekozen implementatie en of deze goed werkt. Zouden er wijzigingen nodig zijn in de huidige implementatie dan kan het rendabel zijn om deze te veranderen in een optimistisch locking implementatie.

In de huidige implementatie heeft men gekozen voor pessimistisch locking, als dit het uitgangspunt zou zijn dan is de offline variant (zie Afbeelding 23) de juiste omdat de bedrijfstransactie niet binnen een database transactie valt.

## 10.7 Aanbevelingen

Het is aan te bevelen om de huidige implementatie goed te analyseren en te testen. Blijkt de pessimistisch locking implementatie fouten te bevatten dan moet men de afweging maken of de keuze voor optimistisch locking niet de voorkeur heeft boven het bugfixen van de foutgevoelige en complexere pessimistisch variant. Hieronder staat kort beschreven hoe de "pessimistisch offline lock" geïmplementeerd dient te worden. Dit kan gebruikt worden in de analyse van huidige implementatie. De optimistisch lock variant is hier verder niet beschreven.

Voor de huidige implementatie is gekozen met de kennis dat de locking strategie in de BRP software componenten nog niet vast staat. Bovendien zijn nog niet alle eisen m.b.t. locking voor de BOP stappen na 2.2 geïdentificeerd. Daarom zal er voor de BOP stappen die volgen op 2.2 afstemming met BRP moeten plaatsvinden om de juist locking aanpak voor komende BOP stappen te bepalen.

De UC202 pessimistic lock implementatie dient te werken volgens het “pessimistic offline lock” design pattern en maakt gebruik van een *exclusive read lock* om zo te garanderen dat altijd gewerkt wordt met de meest recente persoonsgegevens horende bij een a-nummer. De lock op a-nummer dient verkregen te worden voordat de persoonsgegevens horende bij dit a-nummer worden geladen. Aan het einde van het proces dient de lock te worden los gelaten.

Er dient een persistent lock te worden gebruikt en wanneer deze wordt opgeslagen in de database dient dit binnen een database transactie te geschieden. Een unique constraint op de a-nummer kolom garandeert dat een lock niet tweemaal wordt uitgegeven. Houdt er echter wel rekening mee dat wanneer een isolatie niveau lager dan “serializable” wordt gebruikt, sprake kan zijn van phantom reads. Zou een implementatie - bij dit isolatie niveau - eerst controleren op een lock d.m.v. een select dan kan het zijn dat onterecht geconcludeerd wordt dat er geen lock aanwezig is, dit kan uiteindelijk leiden tot een unique constraint violation bij het aanmaken van de lock. Een simpele en effectieve implementatie probeert altijd een lock aan te maken d.m.v. een insert zonder te controleren of deze aanwezig is en concludeert bij een SQL exception dat de lock niet aan te maken is.

Als laatste dient er timeout functionaliteit te bestaan die locks opruimt van processen die halverwege zijn gecrashed en niet meer de kans hebben gehad om de lock los te laten.



## 11 Appendix B: NFR kruistabel

In deze appendix wordt per non-functional requirement (NFR) beschreven welke consequentie deze NFR heeft voor de software architectuur. Een NFR kan een ontwerpbeslissing tot gevolg hebben maar dat hoeft niet. Het kan zijn dat een NFR niet relevant is voor de software of dat de gestelde eis geen invloed heeft op de bestaande software architectuur. Deze NFR's zijn onderverdeeld in een aantal categorieën. Per categorie is een kruistabel opgenomen die beschrijft hoe een specifieke NFR tot uitdrukking komt.

### 11.1 Eisen ten aanzien van prestatie-efficiëntie

De onderstaande tabel beschrijft hoe de lijst met eisen ten aanzien van prestatie-efficiëntie [20] tot uitdrukking komt in de software architectuur.

Code	Ontwerpbeslissing
RD-PRE-001	N.v.t.
RD-PRE-002	De migratievoorziening dient horizontaal schaalbaar te zijn. De verschillende migratie componenten kunnen parallel en op meerdere nodes naast elkaar draaien. De VOISC service is hierop een uitzondering omdat het sPd protocol niet toestaat dat meerdere processen tegelijk één mailbox uitlezen. Omdat er meerdere mailboxen gebruikt worden kunnen er wel meerdere VOISC processen gestart worden. Dus is hier sprake van beperkte horizontale schaalbaarheid voor dit specifieke component.
RD-PRE-003	De migratievoorziening dient minimaal 25.000 synchronisatieberichten per uur te kunnen verwerken.
RD-PRE-004	N.v.t.
RD-PRE-005	Geen impact.
RD-PRE-006	N.v.t.
RD-PRE-007	N.v.t.
RD-PRE-008	Migratiecomponenten die resources van de systeemomgeving cachen die niet meer nodig zijn dienen functionaliteit te hebben voor het legen van de cache zonder dat hiervoor het component moet worden herstart.
RD-PRE-009	De migratievoorziening dient binnen 10 minuten opgestart te zijn.

### 11.2 Eisen ten aanzien van overdraagbaarheid

De onderstaande tabel beschrijft hoe de lijst met eisen ten aanzien van overdraagbaarheid [21] tot uitdrukking komt in de software architectuur.

Code	Ontwerpbeslissing
RD-OVD-001	N.v.t.
RD-OVD-002	N.v.t.
RD-OVD-003	N.v.t.

### 11.3 Eisen ten aanzien van onderhoudbaarheid

De onderstaande tabel beschrijft hoe de lijst met eisen ten aanzien van onderhoudbaarheid [22] tot uitdrukking komt in de software architectuur.

Code	Ontwerpbeslissing
RD-OH-001	Deze eis is verwerkt in de implementatie view, zie § 4.2.6.1.
RD-OH-002	Deze eis is verwerkt in de implementatie view, zie § 4.2.6.1.
RD-OH-003	De migratievoorziening bestaat uit componenten met elk een duidelijke verantwoordelijkheid, zie ook de logische view 4.1.
RD-OH-004	Deze eis is verwerkt in de implementatie view, zie § 4.2.6.1.
RD-OH-005	Deze eis is verwerkt in de implementatie view, zie § 4.2.6.1.
RD-OH-006	Deze eis is verwerkt in de implementatie view, zie § 4.2.6.1.
RD-OH-007	Deze eis is verwerkt in de implementatie view, zie § 4.2.6.1.
RD-OH-008	Deze eis is verwerkt in de implementatie view, zie § 4.2.6.1. Deze eis is ook zichtbaar in het ontwerp van het conversiemodel en entiteiten model die bewust gescheiden zijn om de impact van wijzigingen in het fysieke datamodel op de conversie software te minimaliseren.
RD-OH-009	Deze eis is verwerkt in de implementatie view, zie § 4.2.6.1.

#### 11.4 Eisen ten aanzien van documentatie

De onderstaande tabel beschrijft hoe de lijst met eisen ten aanzien van documentatie [23] tot uitdrukking komt in de software architectuur.

Code	Ontwerpbeslissing
RD-DOC-001	SAD en TO's zijn in Nederlands geschreven.
RD-DOC-002	N.v.t.
RD-DOC-003	N.v.t.
RD-DOC-004	N.v.t.
RD-DOC-005	N.v.t.
RD-DOC-006	SAD en TO's gebruiken standaard UML, ERD en BPMN notaties
RD-DOC-007	Geen impact.
RD-DOC-008	SAD en TO's beschrijven ontwerpbeslissingen.
RD-DOC-009	Geen impact.
RD-DOC-010	Geen impact.
RD-DOC-011	Voor elk component in de proces view (4.3.3.2 en 4.3.4.2) is beschreven hoe dit component schaalbaar is. Daarnaast is voor RD-PRE-002 ook een korte toelichting geschreven.
RD-DOC-012	Geen impact omdat de migratievoorziening in haar koppelvlakken gebruik maakt van open standaarden of aansluit op koppelvlakken zoals deze zijn beschreven in het LO GBA [14].
RD-DOC-013	Geen impact omdat bij de broncode zogenaamde pom.xml bestanden worden opgeleverd met daarin precies beschreven welke externe software en welke versie daarvan wordt gebruikt door de migratievoorziening.
RD-DOC-014	De verschillende modules, hun verantwoordelijkheden en onderlinge afhankelijkheden staan beschreven in de Logical View (4.1). Daarnaast zijn de verschillende componenten en de wijze waarop ze onderling informatie uitwisselen beschreven in de Proces View (4.3).
RD-DOC-016	N.v.t.
RD-DOC-017	N.v.t.
RD-DOC-018	N.v.t.
RD-DOC-019	N.v.t.
RD-DOC-020	N.v.t.
RD-DOC-021	N.v.t.

Code	Ontwerpbeslissing
RD-DOC-023	N.v.t.
RD-DOC-024	Geen impact.
RD-DOC-025	Geen impact omdat de software van de migratievoorziening platform onafhankelijk is.
RD-DOC-026	De gebruikte software bibliotheken staan vermeld in zogenaamde pom.xml bestanden die worden meegeleverd met de broncode van de migratievoorziening.

### 11.5 Eisen ten aanzien van beveiligbaarheid

De onderstaande tabel beschrijft hoe de lijst met eisen ten aanzien van beveiligbaarheid [24] tot uitdrukking komt in de software architectuur.

Code	Ontwerpbeslissing
RD-BEV-004	Deze eis is verwerkt in de richtlijnen voor ontwikkelaars zoals gedocumenteerd in de implementatie view, zie § 4.2.6.1.
RD-BEV-006	N.v.t.
RD-BEV-013	N.v.t.
RD-BEV-018	Geen impact.
RD-BEV-019	Deze eis is verwerkt in de implementatie view, zie § 4.2.6.1.
RD-BEV-020	Geen impact.
RD-BEV-021	Geen impact.
RD-BEV-022	Geen impact.
RD-BEV-023	Componenten binnen de migratievoorziening dienen te loggen zoals beschreven is in Appendix C: Loggen in de Migratievoorziening.
RD-BEV-024	Geen impact.
RD-BEV-025	Geen impact.
RD-BEV-026	Geen impact.
RD-BEV-027	N.v.t.
RD-BEV-028	N.v.t.
RD-BEV-029	Geen impact.
RD-BEV-030	In Appendix E: Toelichting bij OWASP top tien, staat beschreven welke maatregelen zijn genomen per OWASP punt.
RD-BEV-031	Geen impact.
RD-BEV-032	Zie hiervoor ontwerpbeslissing OB.15 die beschreven is in paragraaf 6.1.
RD-BEV-033	Geen impact.
RD-BEV-034	Geen impact.
RD-BEV-035	Geen impact.
RD-BEV-036	Geen impact.
RD-BEV-037	N.v.t.
RD-BEV-039	1.1.1.1 N.v.t.
RD-BEV-040	1.1.1.2 N.v.t.
RD-BEV-041	Deze eis is verwerkt in de betrouwbaarheid van initiële vulling (4.3.3) en ISC (4.3.4).
RD-BEV-042	Geen impact.
RD-BEV-043	N.v.t.
RD-BEV-044	Geen impact.
RD-BEV-045	Geen impact.
RD-BEV-046	Geen impact.
RD-BEV-047	De wijze waarop de migratievoorziening kan loggen op externe systemen staat beschreven in paragraaf 12.3 (Loggen op extern systeem).
RD-BEV-048	De wijze waarop de log meldingen binnen de migratievoorziening actief kunnen worden gemeld aan een beheerder staat beschreven in paragraaf 12.4 (Actieve meldingen).
RD-BEV-049	Dit is beschreven in het SAD, zie deployment view van initiële vulling (4.4.3) en ISC (4.4.4).

Code	Ontwerpbeslissing
RD-BEV-050	N.v.t.
RD-BEV-052	Geen impact.

**11.6 Eisen ten aanzien van betrouwbaarheid**

De onderstaande tabel beschrijft hoe de lijst met eisen ten aanzien van betrouwbaarheid [25] tot uitdrukking komt in de software architectuur.

Code	Ontwerpbeslissing
RD-BET-001	N.v.t.
RD-BET-002	Deze eis is verwerkt in de betrouwbaarheid van initiële vulling (4.3.3) en ISC (4.3.4).
RD-BET-003	N.v.t.
RD-BET-004	Deze eis is verwerkt in de concurrency van initiële vulling (4.3.3) en ISC (4.3.4).
RD-BET-005	N.v.t.

## 12 Appendix C: Loggen in de Migratievoorziening

Binnen het BRP landschap is logging een term die breed gebruikt wordt. Voor de implementatie wordt de logging opgedeeld in verschillende specifieke begrippen.

### 12.1 Gestructureerde logging

Met gestructureerde logging wordt het opslaan van gestructureerde gegevens in een permanente opslag aangeduid (bijvoorbeeld gegevens in een database tabel). De structuur van deze gegevens en wanneer deze worden opgeslagen dient uitputtend te zijn beschreven in het functioneel ontwerp en zullen specifiek worden geïmplementeerd.

### 12.2 Event logging

Met event logging wordt het opslaan van ongestructureerde meldingen aangeduid (bijvoorbeeld meldingen in een logbestand). Deze event logging is tweeledig: enerzijds kunnen functioneel meldingen worden beschreven anderzijds worden vanuit de techniek logmeldingen opgeslagen ter ondersteuning van het oplossen van problemen met de applicatie.

#### 12.2.1 Functionele event logging

De functionele event logging dient in het functioneel ontwerp te worden beschreven waarbij een melding code en melding tekst worden beschreven. Hiernaast kan tijdens de functionele beschrijving een 'context' worden opgebouwd met gestructureerde gegevens die met de logmelding worden meegestuurd (denk hierbij bijvoorbeeld aan het a-nummer van een persoon tijdens het verwerken van een bijhouding).

#### 12.2.2 Technische event logging

De technische event logging wordt geïmplementeerd tijdens het ontwikkelen en onderhouden van de applicatie en dient ter ondersteuning van (onverwachte) problemen in de applicatie of de omgeving.

Minimaal dient het volgende te worden gelogd:

- elke exceptie die niet wordt afgehandeld door het systeem.

Minimaal dient een uniek ID (UUID) te worden bepaald voor elke 'unit of work' (een binnenkomend bericht, een actie door een eindgebruiker of een actie naar aanleiding van een timer, etc.) die buiten het systeem wordt geïnitieerd. Dit 'verwerking-id' dient vervolgens met interne communicatie binnen het systeem landschap te worden meegestuurd en opgepakt zodat de verwerking van een actie over verschillende componenten gevolgd kan worden.

#### 12.2.3 Logging niveaus

Binnen de logging worden verschillende niveaus gebruikt. De volgende tabel benoemt de te gebruiken niveaus en beschrijft wanneer deze gebruikt dienen te worden.

Niveau	Omschrijving
DEBUG	Technische event logging op een zeer specifiek en gedetailleerd niveau.
INFO	Het niveau waarop functionele event logging wordt gemeld en bevat technische event logging over gebeurtenissen en

Niveau	Omschrijving
	beslissingen in het systeem.
WARNING	Waarschuwingen voor de functioneel of technisch beheerder dat het systeem een fout heeft gedetecteerd, maar geen directe actie benodigd is, omdat de fout is gemeld aan een extern systeem, eindgebruiker of op dat moment kon worden genegeerd.
ERROR	Waarschuwingen dat het systeem een fout heeft gedetecteerd waardoor de verwerking is gestopt en een beheerder direct actie op dient te ondernemen.

### 12.3 Loggen op extern systeem

De migratievoorziening gebruikt SLF4J met Log4j om te loggen. Deze frameworks bieden standaard ondersteuning voor syslog. Dit maakt het mogelijk om de migratievoorziening componenten zo te configureren dat de log bestanden op een andere server worden opgeslagen.

### 12.4 Actieve meldingen

Door het gebruik van Log4j kan een beheerder naar syslog loggen als dat nodig is. Er bestaan standaard oplossingen om log files te scannen voor bepaalde log meldingen om vervolgens daarop actie te kunnen ondernemen. Een voorbeeld van een tool die dit kan is Swatch (<https://sourceforge.net/projects/swatch/>). De migratievoorziening levert dus zelf geen functionaliteit om logmeldingen actief te melden aan een beheerder maar maakt het wel mogelijk voor een beheerder om dit zelf in te richten m.b.v. standaard tooling.



## 13 Appendix D: Monitoring

Binnen het ICT landschap is monitoring een term die breed gebruikt wordt. Binnen BRP wordt hiermee de aansluiting bedoeld die wordt geboden om de status van het systeem te kunnen inzien. Expliciet wordt hiermee de tooling uitgesloten waarmee de gegevens worden uitgelezen. Deze tooling wordt gezien als het domein van de technisch (applicatie) beheerder. De enige beperking is dat de tooling om kan gaan met het hierna beschreven protocol.

### 13.1 Java Management Extensions

Binnen Java is de de facto standaard hiervoor Java Management Extensions (JMX). Binnen JMX wordt een API beschreven waarmee op een gestandaardiseerde manier informatie kan worden opgehaald en eventueel functionaliteit kan worden gestart. De specifieke (object)namen waarmee informatie opgehaald kan worden verschilt per system en wordt in de specifieke technische ontwerpen beschreven.

### 13.2 Communicatie protocol

JMX maakt standaard gebruik van RMI als communicatie protocol. Echter dit protocol voldoet niet aan de technische eisen die BRP hier aan stelt. Specifiek wordt er een callback mechanisme gebruikt waardoor het server-component (de applicatie) een port en IP-adres aan het client-component (de monitoring tool) aanbiedt waarop verbinding gemaakt dient te worden. Binnen een container omgeving (Docker) heeft het server-component geen besef van het IP-adres waarop verbinding wordt gemaakt en geeft een ongeldig IP-adres (het adres binnen het Docker netwerk) op. De dynamische poort aanwijzing zorgt ook voor firewall problemen maar kan eventueel verholpen worden door deze poort op een vaste poort in te stellen.

Hierdoor is gekozen een alternatief communicatie protocol voor JMX te introduceren. Dit communicatie protocol (genoemd 'serialize' of 'serialize+ssl') maakt gebruik van een simpele directe verbinding waardoor er geen probleem met een callback-mechanisme meer bestaat. De implementatie van dit protocol kan worden gevonden in het 'alg-util-jmx' project en is gebaseerd op de standaard java voorzieningen voor communicatie (Socket, ServerSocket, SSLSocket) en serializatie (ObjectInputStream en ObjectOutputStream).

Om gebruik te maken van een alternatief communicatie kanaal dient een monitoring tool de bibliotheek (jar) toe te voegen aan het klasse-pad en het protocol op te geven in de JMX connectie (bijvoorbeeld:  
service:jmx:serialize+ssl://localhost:3481).

## 14 Appendix E: Toelichting bij OWASP top tien

De migratievoorziening dient (voor zover van toepassing) te zijn beveiligd tegen de risico's zoals benoemd in de OWASP top-10 lijst. Hieronder volgt een opsomming van de OWASP top 10 (ref...) met daarbij de toelichting of een punt van toepassing is en zo ja, welke maatregel zijn genomen.

Opmerking: Onderstaande toelichting heeft alleen betrekking op de Initiële Vulling (IV) componenten.

- A1-Injection
  - Een specifieke manipulatie techniek. Het verwerken van onvertrouwde data zonder controle of het uitvoeren van functies zonder autorisatie.
  - Van toepassing: Ja
  - Maatregelen: 1) Zeer strikte input validatie van berichten conform de LO3 specificatie. 2) Het gebruik van API's voor SQL waarmee met geparameteriseerde queries wordt gewerkt.
- A2-Broken Authentication and Session Management
  - Het onbedoeld verkrijgen van toegang tot (delen) van functies, het verkrijgen van sleutels, wachtwoorden, etc.
  - Van toepassing: Nee
- A3-Cross-Site Scripting (XSS)
  - Validatiefouten in webapplicaties in combinatie met een webbrowser bij de gebruiker. Niet van toepassing.
  - Van toepassing: Nee
- A4-Insecure Direct Object References
  - Een specifieke manipulatie techniek. Het verwerken van onvertrouwde data zonder controle of het uitvoeren van functies zonder autorisatie.
  - Van toepassing: Nee
- A5-Security Misconfiguration
  - Foutieve configuratie van de applicatie of van de onderliggende lagen in de gebruikte software stack.
  - Van toepassing: Nee
- A6-Sensitive Data Exposure
  - Het lekken van gevoelige informatie.
  - Van toepassing: Nee
- A7-Missing Function Level Access Control
  - Het onbedoeld verkrijgen van toegang tot (delen) van functies.
  - Van toepassing: Nee
- A8-Cross-Site Request Forgery (CSRF)
  - Een aanval gericht op de webbrowser van een gebruiker waarbij een web-applicatie dit kan voorkomen.
  - Van toepassing: Nee
- A9-Using Components with Known Vulnerabilities
  - Bekende kwetsbaarheden in gebruikte standaard software componenten.
  - Van toepassing: Ja
  - Maatregelen: Het periodiek controleren of er kwetsbaarheden zijn gemeld voor de externe software bibliotheken die worden gebruikt voor de migratievoorziening.

- A10-Unvalidated Redirects and Forwards
  - Validatie problemen op websites.
  - Van toepassing: Nee