

# Registratie aan de bron

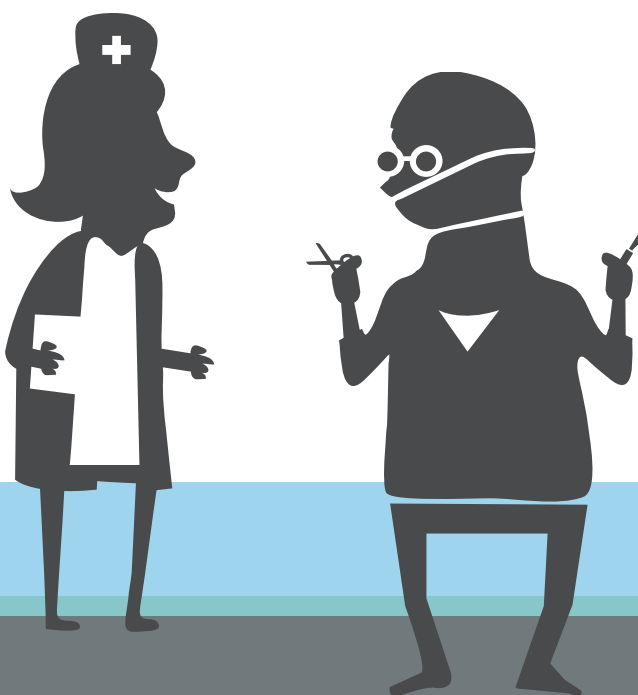
Zorginformatie delen en optimaliseren

## Architectuur vol.2

Technische implementatie van zib's

De praktische aspecten van de implementatie van zib's op het niveau van de applicaties en de gegevensuitwisseling

April 2017  
Versie 1.0



**Documenthistorie**

Versie	Datum	Omschrijving
1.0	20 april 2017	Versie vastgesteld door het Programmateam

**Nota bene:**

Dit document is gebaseerd op de zorginformatiebouwstenen Release 2016.

In de loop van 2017 zal een aantal bouwstenen aangepast worden. Zo zal van een aantal bouwstenen het voorvoegsel “Overdracht” verdwijnen. En ook zal er voor een aantal gegevenselementen waaronder “Adres” een aparte subbouwsteen komen. Dit soort aanpassingen is in dit architectuurdocument nog niet meegenomen. Zodra de aanpassingen gepubliceerd zijn zal van dit document een nieuwe versie uitgebracht worden waarin die aanpassingen zijn meegenomen.

Het document is tot stand gekomen met inbreng van

Naam	Organisatie / rol
Ernst de Bel	Radboudumc
Lindsay Chang	AMC/VUmc
Sjaak Gondelach	UMC Utrecht
Heleen Hoogvliet	LUMC
Gé Klein Wolterink	Adviseur
Igor Schoonbrood	Maastricht UMC+
Fred Smeele	Programmamanagement Registratie aan de bron
Michiel Sprenger	Nictiz
Albert-Jan Spruyt	Nictiz
Ben van der Stigchel	Erasmus MC
Daniël Woning	Isala
Michael van der Zel	UMC Groningen

Contact: [info@registratieaandebron.nl](mailto:info@registratieaandebron.nl)

## Inhoud

1.	Inleiding .....	6
1.1	Achtergrond .....	6
1.2	Doel van dit document .....	6
1.3	Inhoud van dit document .....	6
2.	Implementatiemodel .....	7
2.1	Architectuurmodel .....	7
2.2	Implementatie in de praktijk; een model .....	7
2.3	Implementatiethema's .....	9
3	Zib's en de inrichting van zorginformatiesystemen .....	10
3.1	Inleiding .....	10
3.2	Zorginformatiebouwstenen als gegevensmodellen .....	10
3.3	Vulling en instantiaties .....	13
3.4	Niet voor alle gegevens bestaan zib's .....	13
4	Gegevenssets bestaande uit zib-instantiaties .....	14
4.1	Inleiding .....	14
4.2	Selectie van gegevens .....	14
4.3	De praktijk .....	16
5	Uitwisseling op basis van HL7 CDA en FHIR .....	18
5.1	Inleiding .....	18
5.2	HL7 CDA en FHIR .....	18
5.2.1	HL7 CDA (Clinical Document Architecture) .....	18
5.2.2	HL7 FHIR (Fast Healthcare Interoperability Resources) .....	19
5.2.3	Discussie verschil CDA en FHIR .....	19
5.3	Uitwisseling op basis van HL7 CDA .....	20
5.3.1	Mapping van zib's en CDA entries .....	20
5.3.2	Uitwisseling op basis van CDA documenten .....	21
5.4	Uitwisseling op basis van HL7 FHIR .....	22
5.4.1	Mapping van zib's en FHIR resources .....	22
5.4.2	Uitwisseling op basis van FHIR documents .....	23
5.4.3	Uitwisseling op basis van (enkelvoudige) FHIR resources .....	24
6	Praktische oplossingen voor de infrastructuur .....	26
6.1	Inleiding .....	26
6.2	Diverse point-to-point oplossingen. ....	27
6.3	Oplossingen gebaseerd op het IHE XDS profiel .....	27

6.3.1	Oplossingen gebaseerd op IHE MHD.....	28
6.4	Client-server oplossingen gebaseerd op (FHIR) RESTful services .....	29
6.5	AORTA (het LSP) .....	30
7	Bijlage A - Lijst van afkortingen en begrippen .....	33
8	Bijlage B - Voorbeeld instantiatie gegevensset .....	35
9	Bijlage C - HL7 CDA.....	38
9.1	De structuur van een CDA document.....	38
9.2	CDA templates.....	39
10	Bijlage D - HL7 FHIR .....	40
10.1	FHIR resources .....	40
10.2	Uitwisselingsformaten die worden ondersteund door FHIR .....	41
10.3	FHIR RESTful services.....	42
11	Bijlage E – Mapping van zibs en CDA entries; een voorbeeld.....	44
12	Bijlage F – Mapping van zibs en FHIR resources; een voorbeeld .....	46

## Overzicht van figuren

Figuur 1 - Het vijflagen architectuurmodel en de positionering van zib's .....	7
Figuur 2 – Implementatiemodel .....	8
Figuur 3 - Implementatiethema's .....	9
Figuur 4 - Zib als gegevensmodel voor de inrichting van zorginformatiesystemen .....	10
Figuur 5 - Voorbeeld van de implementatie van (een deel van) de zib Patiënt in een databasetabel .....	11
Figuur 6 – Relatie zib-database: van fysiek model naar logisch model .....	12
Figuur 7 - Instantiaties van bouwstenen met concrete gegevens op basis van onderliggende modellen .....	13
Figuur 8 - Gegevenssets gebaseerd op zib-instantiaties .....	14
Figuur 9 - Samenstellen van een gegevensset .....	15
Figuur 10 - Variant met een DWH .....	17
Figuur 11 - Uitwisseling op basis van HL7 CDA of FHIR .....	18
Figuur 12 - Uitwisseling op basis van HL7 CDA.....	21
Figuur 13 - Uitwisseling op basis van HL7 FHIR documenten .....	23
Figuur 14 – Het beschikbaar maken van gegevens op basis van enkelvoudige FHIR resources .....	25
Figuur 15 - Oplossingen voor de IT infrastructuur.....	26
Figuur 16 - Uitwisseling van documenten op basis van secure email etc. ....	27
Figuur 17 - De werking van een XDS omgeving .....	28
Figuur 18 - IHE MHD en XDS .....	29
Figuur 19 – Client-server gebaseerde uitwisseling op basis van FHIR RESTful services.....	30
Figuur 20 - Het LSP voor PULL verkeer met signaal en opvraag functie .....	30
Figuur 21 - Het LSP voor PUSH verkeer (versturen) .....	32
Figuur 22 - Voorbeeld systeemopzet .....	35
Figuur 23 - Voorbeeld instantiatie gegevensset A .....	37
Figuur 24 - De structuur van een CDA document .....	38
Figuur 25 - Voorbeeld CDA document .....	38
Figuur 26 - CDA document – XML basisstructuur.....	39
Figuur 27 – Voorbeelden van relaties tussen FHIR resources .....	40
Figuur 28 – Voorbeeld resource, de eenheid van uitwisseling die RESTful omgeving gebruikt.....	41
Figuur 29 – Voorbeeld van een FHIR message met resources en references .....	41
Figuur 30 – Voorbeeld van een FHIR document met resources en references.....	42
Figuur 31 - CDA document (deel)voorbeeld met referentie naar de zib Contact.....	45
Figuur 32 - StructureDefinition (deel) voor de relatie tussen de zib en de resource Patient .....	48

## 1. Inleiding

---

### 1.1 Achtergrond

Het programma [Registratie aan de bron](#) (Radb) heeft als doel het structureel verbeteren van de registratie en het hergebruik van patiëntgegevens. In dat kader worden verschillende producten ontwikkeld die bedoeld zijn om brede toepassing te vinden in de zorg in Nederland. In dit architectuurdokument wordt ingegaan op verschillende thema's met als doel om achtergrond en inhoud van de ontwikkelde producten duidelijk te maken, de context waarin ze in de zorg in Nederland toegepast kunnen worden en wat er moet gebeuren om de implementatie in de praktijk mogelijk te maken. Het architectuurdokument zal bestaan uit meerdere volumes.

Volume 1 van het architectuurdokument is het basisdocument met daarin beschreven de definitie en basisbeginselen van de zorginformatiebouwstenen (zib's) en hoe die in de praktijk toegepast kunnen worden.

Dit document, volume 2 van het architectuurdokument, gaat in op de technische implementatie van zib's in de praktijk. De nadruk ligt op de praktische aspecten van de implementatie van zib's op het niveau van de applicaties en de gegevensuitwisseling.

Volume 3 van het architectuurdokument heeft als focus, continuïteit van zorg: gegevensuitwisseling in de zorg – het proces. Er wordt ingegaan op de manier waarop zibs in de praktijk gebruikt kunnen worden voor de uitwisseling van gegevens in de zorg als onderdeel van het zorgproces.

Volume 4 van het architectuurdokument heeft als ondertitel "Hergebruik van gegevens voor registratie en onderzoek" en gaat in op de manier waarop zib's in de praktijk gebruikt kunnen worden voor de aanlevering aan registraties en onderzoek

### 1.2 Doel van dit document

Het doel van dit document is:

- Het beschrijven van de technische aspecten van de implementatie van zorginformatiebouwstenen
- Het bevorderen van inzicht in de mogelijkheden en de beperkingen van de implementatie van zorginformatiebouwstenen
- Het bevorderen van het gebruik van zorginformatiebouwstenen

Dit document is in eerste instantie bestemd voor projectleiders, inhoudelijk deskundigen, ontwikkelaars, leveranciers en anderen die betrokken zijn bij het programma. Daarnaast zijn delen van het document ook bruikbaar voor bestuurders, managers en anderen die geïnteresseerd zijn in de implementatie van zorginformatiebouwstenen in de praktijk. Veronderstelde voorkennis is de inhoud van volume 1 van het architectuurdokument.

### 1.3 Inhoud van dit document

Om de inhoud van het document te structureren wordt gebruik gemaakt van het vijflagen architectuurmodel. Dat wordt toegelicht in hoofdstuk 2. Aan de hand daarvan wordt een viertal implementatiethema's gedefinieerd die worden uitgewerkt in de daarop volgende hoofdstukken:

- Hoofdstuk 3 – Zib als gegevensmodel voor de inrichting van zorginformatiesystemen
- Hoofdstuk 4 – Gegevenssets door selectie van zib-instantiaties
- Hoofdstuk 5 – Uitwisseling op basis van HL7 CDA en FHIR
- Hoofdstuk 6 – Praktische oplossingen voor de infrastructuur

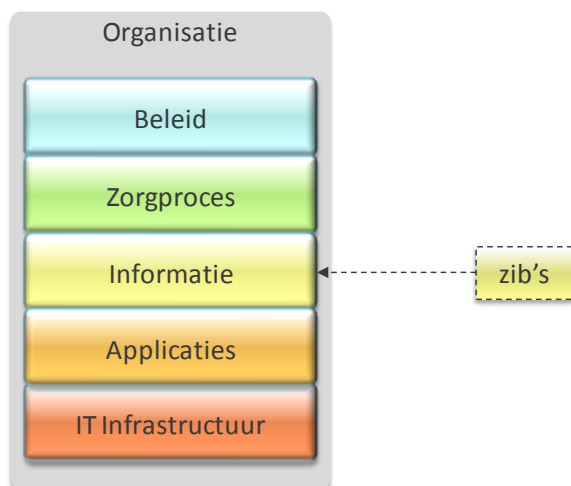
In een aantal bijlagen worden details uitgewerkt.

## 2. Implementatiemodel

### 2.1 Architectuurmodel

We gebruiken het vijflagen architectuurmodel dat is weergegeven in Figuur 1 om de uitwerking van de technische implementatie van zib's te structureren. Het model beschrijft de samenhang waarbinnen informatieoplossingen in de praktijk worden gerealiseerd binnen een organisatie. Het model is zowel toe te passen in een grote organisatie als een algemeen ziekenhuis of universitair medisch centrum (umc), als in een gezondheidscentrum of huisartsenpraktijk.

De kern van het model wordt gevormd door het zorgproces dat wordt ondersteund door zorginformatie (informatielaag). Die informatie wordt verwerkt in software systemen en applicaties (applicatielaag) en een onderliggende IT infrastructuur. En dat alles moet passen binnen het beleid van de organisatie (beleidslaag).

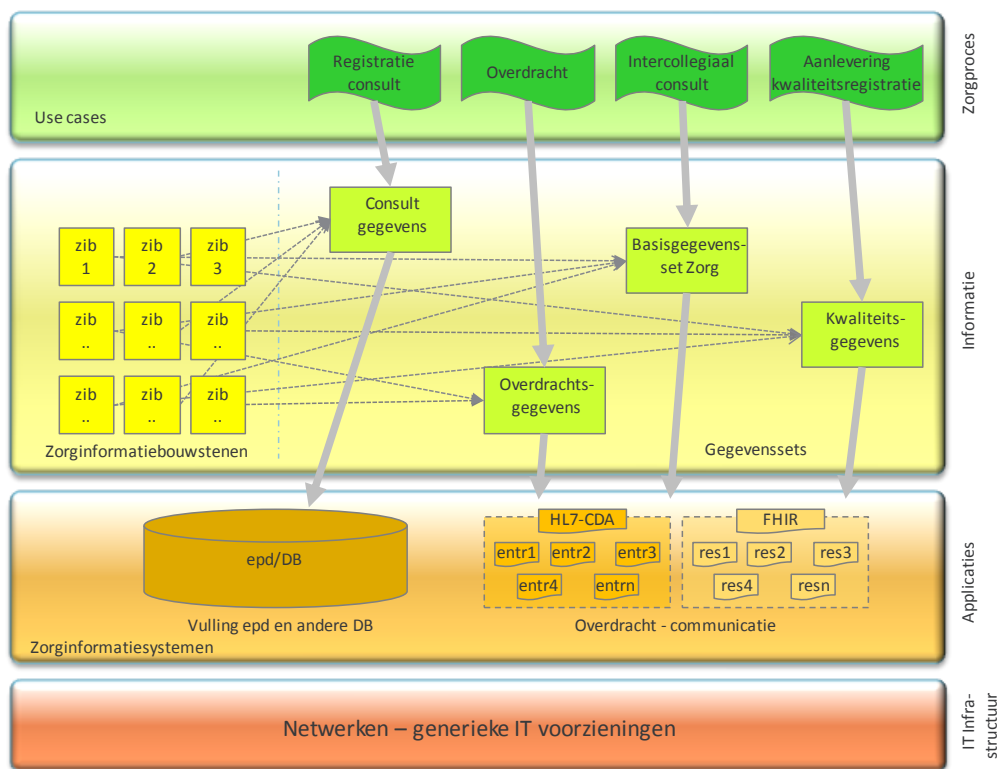


**Figuur 1 - Het vijflagen architectuurmodel en de positionering van zib's**

Zorggegevens ontstaan tijdens het zorgproces en worden m.b.v. de applicaties vastgelegd op basis van gegevensdefinities beschreven in modellen in de informatielaag. Door bij het vastleggen gebruik te maken van de gegevensdefinities in zib's, wordt eenduidigheid van de gegevens en eenheid van taal bereikt over applicaties heen, een noodzakelijke voorwaarde voor interoperabiliteit en meervoudig gebruik van die gegevens. De zib's zijn neutraal ten opzichte van het zorgproces en daarbij behorende use cases; dat wil zeggen dat ze in alle zorgprocessen toepasbaar zijn en niet specifiek voor één bepaald proces. Tegelijkertijd zijn zib's ook neutraal ten opzichte van software systemen, applicaties en de IT infrastructuur; leveranciers moeten zelf keuzes maken hoe ze de zib's technisch in hun systemen willen implementeren.

### 2.2 Implementatie in de praktijk; een model

Bij de implementatie van de zib's spelen diverse zaken op de verschillende lagen een rol. Om die verder te duiden en uit te werken, maken we gebruik van het implementatiemodel in Figuur 2.



**Figuur 2 – Implementatiemodel**

Het model is niet bedoeld als een gedetailleerde weergave van een praktijkimplementatie maar als een denkmodel dat gebruikt kan worden om een aantal implementatie aspecten te duiden en ten opzichte van elkaar neer te zetten.

Hieronder volgt een korte beschrijving van het model, alleen bedoeld om de hoofdlijnen neer te zetten. Vervolgens wordt uitgewerkt op welke manier het model leidend is voor de verdere detailuitwerking in de volgende hoofdstukken. Een korte beschrijving van het model:

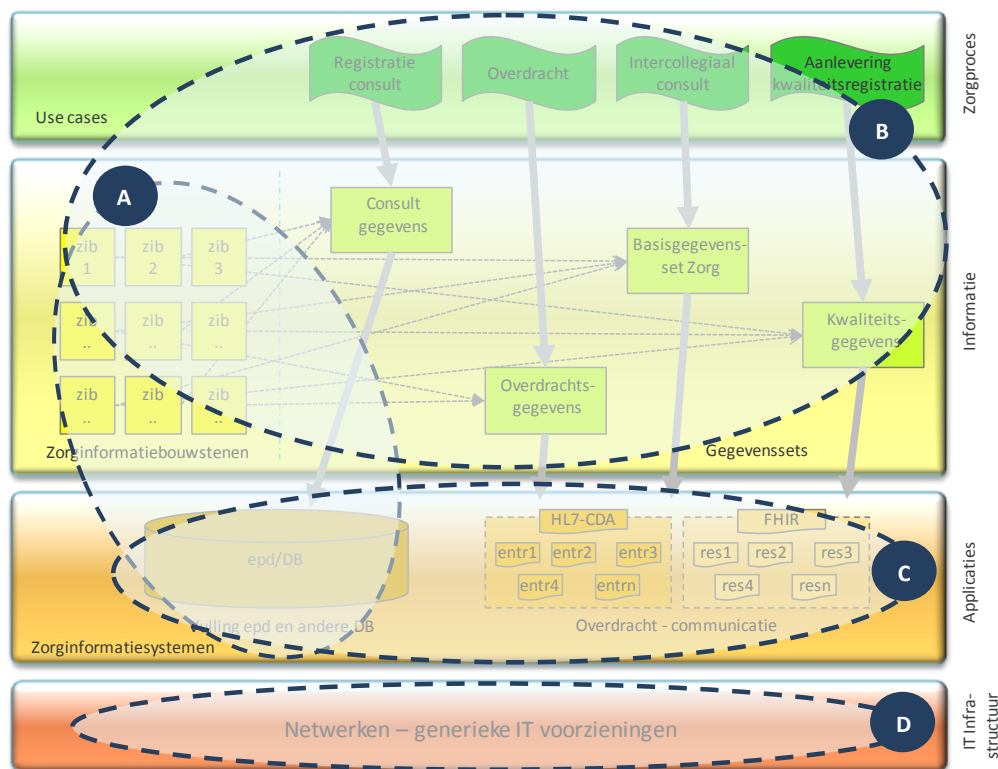
- Uitgangspunt zijn de zorgprocessen die zich op de laag 'zorgproces' van het 5-lagenmodel afspelen. (N.B.: de beleidslaag laten we in dit geval buiten beschouwing). [proceslaag]
- Binnen de zorgprocessen zijn specifieke gebruikssituaties (use cases) te onderkennen zoals 'Registratie consult', 'Overdracht', 'Intercollegiaal consult', 'Aanlevering kwaliteitsregistratie' etc. [proceslaag]
- Voor elk van de use cases bestaat een goed gedefinieerde gegevensset die in die use case wordt 'verwerkt': vastgelegd, geraadpleegd, gewijzigd of verwijderd of wordt verzonden naar- of ontvangen van een andere actor. Die gegevensset bestaat uit een selectie van specifieke zorggerelateerde gegevens. Denk bijvoorbeeld aan: de gegevens die worden vastgelegd tijdens een consult, de overdrachtsgegevens die worden meegestuurd bij een verwijzing van de patiënt of de kwaliteitsgegevens voor aanlevering aan een kwaliteitsregistratie. [informatielaag]
- De gegevensset bestaat meestal uit één of meer gegevens van dezelfde of van verschillende gegevenssoorten (objecten of entiteiten). Bijvoorbeeld: de medicatie van de patiënt en/of zijn allergieën, uitslagen, afspraken etc. Bij iedere use case hoort een selectie van specifieke instanties van die objecten uit het dossier van de patiënt, die samen de gegevensset vormen. Bijvoorbeeld: de actuele medicatie, de laatste lab-uitslagen van de klinische chemie en/of de toekomstige afspraken. [informatielaag]



- Om eenheid van taal te creëren, voldoen de gegevens aan de specificaties zoals vastgelegd in de zorginformatiebouwstenen (zib's), dus: dezelfde items, itemtypes, codelijsten en waardebereik. Hierdoor ontstaat een eenduidige definitie en betekenis van de gegevens die onafhankelijk is van het "onderliggende fysieke" systeem of database. [informatielaag]  
Nota bene: niet voor alle gegevens zullen bouwstenen bestaan of ontwikkeld worden, maar alleen voor die gegevens waarvoor dat van belang is in het kader van meervoudig gebruik.
- In het geval gegevens binnen één systeem vastgelegd en gedeeld worden door zorgverleners is het model hiermee compleet. Wat de ene zorgverlener vastlegt in het (epd) systeem, kan de andere zorgverlener, die hetzelfde systeem gebruikt, direct zien (mits bevoegd). [applicatielaag]
- In het geval gegevens uitgewisseld worden tussen meerdere systemen is het van belang om afspraken te maken over de uitwisselingsstandaard die hiervoor wordt gebruikt zoals bv. HL7 CDA of HL7 FHIR. [applicatielaag]
- Bij uitwisseling moet ook een netwerk oplossing afgesproken worden zoals bv. een op IHE XDS gebaseerde oplossing of het LSP. [IT infrastructuurlaag]

## 2.3 Implementatiethema's

Voor de verdere detailuitwerking is een viertal implementatiethema's gedefinieerd volgens de indeling die is weergegeven in Figuur 3.



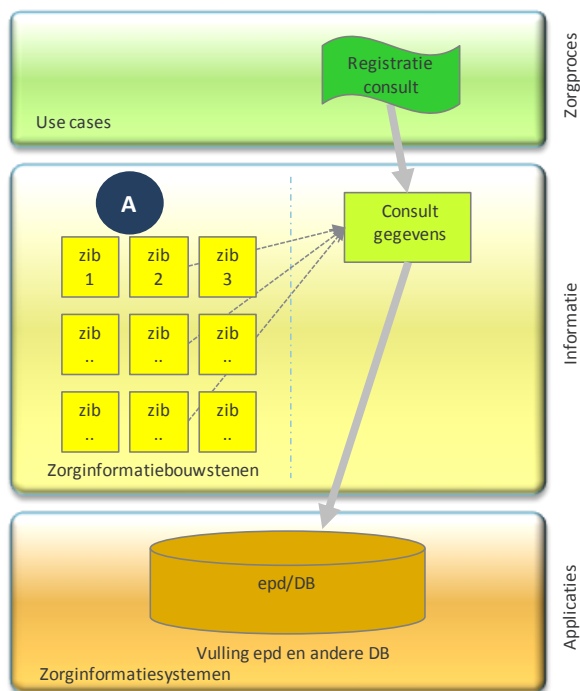
Figuur 3 - Implementatiethema's

#	Thema	Hoofdstuk
A	Zib's en de inrichting van zorginformatiesystemen	3
B	Gegevenssets bestaande uit zib-instantiaties	4
C	Uitwisseling m.b.v. uitwisselingsstandaarden zoals HL7 CDA of HL7 FHIR	5
D	Praktische oplossingen voor de infrastructuur	6

### 3 Zib's en de inrichting van zorginformatiesystemen

#### 3.1 Inleiding

De basisprincipes van de zib's staan beschreven in het architectuurdokument volume 1<sup>1</sup>. In dit hoofdstuk wordt thema A van het implementatiemodel beschreven: de manier waarop zib's zich verhouden tot de inrichting van zorginformatiesystemen. Meer specifiek: de relatie tussen zib's (logische gegevensmodellen) en gegevens in databases (de fysieke structuur om gegevens in op te slaan). Dit betreft m.n. de informatielaag en de applicatielaag zoals weergegeven in Figuur 4.



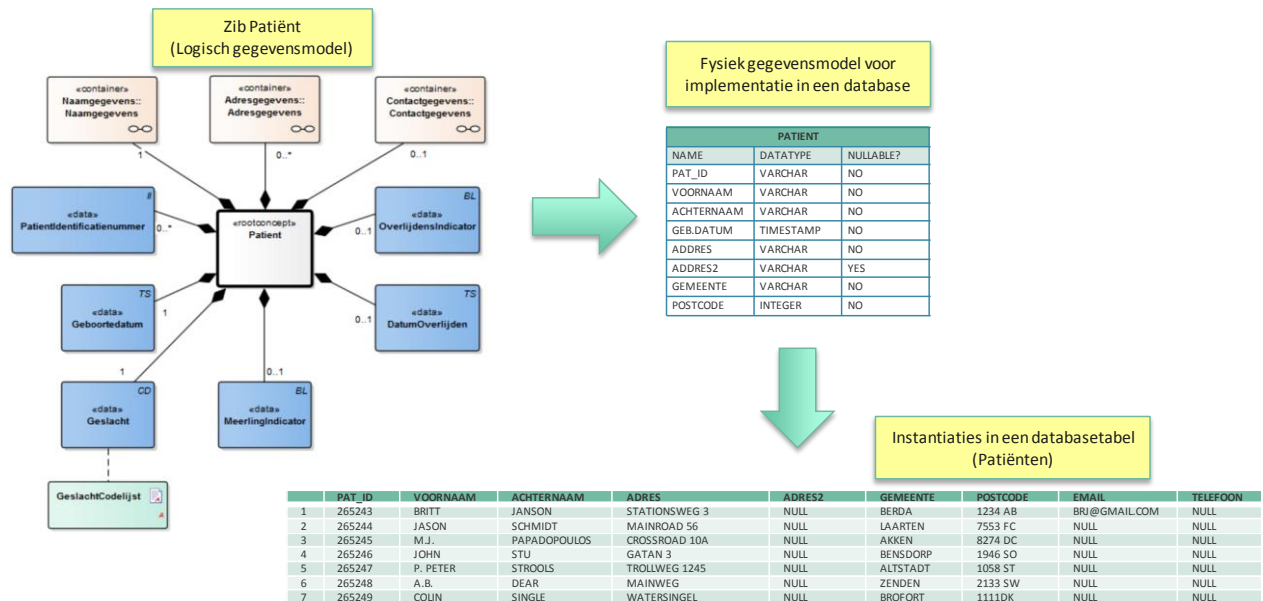
**Figuur 4 – Zib's en de inrichting van zorginformatiesystemen**

#### 3.2 Zorginformatiebouwstenen als gegevensmodellen

Een zorginformatiebouwsteen is een gegevensmodel bestaande uit één of meer gegevenselementen, inclusief hun betekenis, datatypes en onderlinge relaties, die als geheel een relevant zorginhoudelijk concept beschrijven. Een zib heeft de vorm van een Detailed Clinical Model (DCM). Dit is een document waarin de definitie van het zorginhoudelijke concept is vastgelegd. Deze gegevensmodellen, ook wel 'logische gegevensmodellen' genoemd, worden door informatieanalisten samen met gebruikers (artsen, verpleegkundigen) gemaakt om objecten of concepten uit de werkelijkheid te beschrijven. De gegevensmodellen kunnen vervolgens door software ontwikkelaars gebruikt worden om de objecten of concepten in software te implementeren.

Een zib kan dus gebruikt worden door software ontwikkelaars als een 'bouwtekening' om hun software- en databaseontwerp vorm te geven. Afhankelijk van de implementatie kunnen zij de specificatie van de zib en de bijbehorende gegevenselementen bijvoorbeeld vertalen naar classes met hun eigenschappen in de applicatiesoftware en tabellen en items (kolommen) in databases. Dit worden ook wel 'fysieke gegevensmodellen' genoemd. Een voorbeeld daarvan is weergegeven in Figuur 5.

<sup>1</sup> [https://www.registratieaandebron.nl/pdf/Architectuurdokument\\_Registratie\\_aan\\_de\\_bron\\_-\\_Volume\\_1\\_v1.0.pdf](https://www.registratieaandebron.nl/pdf/Architectuurdokument_Registratie_aan_de_bron_-_Volume_1_v1.0.pdf)



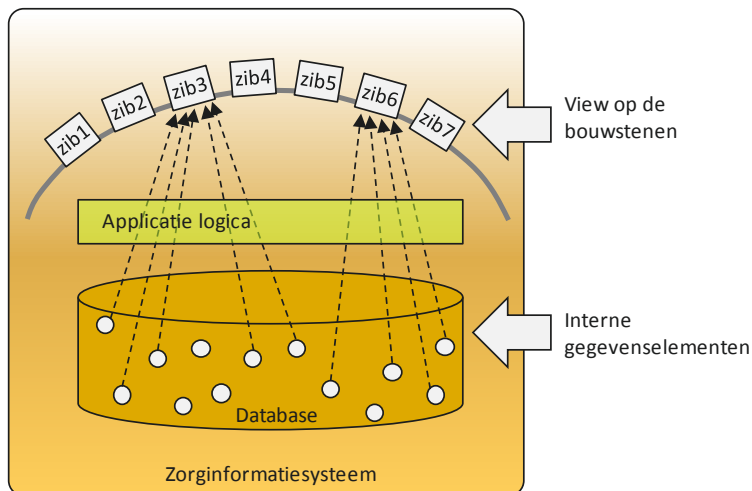
**Figuur 5 - Voorbeeld van de implementatie in een databasetabel**

Gegevensmodellen vormen de tussenstap die tijdens de ontwerpfase van (zorg)informatiesystemen gebruikt wordt om de werkelijke wereld (de gegevens die in operationele processen verwerkt worden), functioneel te beschrijven, zodat die vervolgens vertaald kan worden naar applicatiesoftware en opslagstructuren in databases. Precies dus wat het vijflagen architectuurmodel weergeeft.

Er zijn veel verschillende zorginformatiesystemen van verschillende leveranciers en de databasestructuur van al die systemen is verschillend: andere tabellen, andere items en itemnamen en andere datatypes. Dat betekent dat er in al die verschillende databases vrijwel zeker geen (1-op-1 implementatie van een) zib te vinden zal zijn. De meeste van die systemen bestonden ook al lang voordat er zib's gedefinieerd werden.

Hoe de gegevens in een database opgeslagen zijn, is ook niet relevant. Het gaat erom dat er een mechanisme is (bijvoorbeeld door mapping in de applicatielogica) dat het mogelijk maakt om een view op die gegevens te realiseren zodanig dat die het object dat de zib beschrijft, oplevert met de juiste items in de juiste vorm (datatypes en codelijsten). Dit is gevisualiseerd in Figuur 6.

Andersom moet het ook mogelijk zijn een object dat aan de specificaties van een zib voldoet, aan het zorginformatiesysteem aan te bieden, waarna de applicatie het in het interne database format kan opslaan.



**Figuur 6 – Relatie zib-database: van logisch model naar fysiek model**

Het systeem is in staat om de gegevens in de database, die daarin zijn opgeslagen op basis van het fysieke gegevensmodel, om te zetten in een view en naar de buitenwereld te presenteren in een vorm die in lijn is met het logisch gegevensmodel van de zib's. Dit betekent:

- Dat de gegevens-elementen in de database, die behoren bij een specifieke bouwsteen, in samenhang met elkaar weergegeven kunnen worden conform het model van de zib. (De relatie tussen de gegevens-elementen in de database, het fysieke model, en de gegevens-elementen van de zib, het logische model, moet dus in het systeem bekend zijn).
- Dat de datatypes en waardelijsten in de view overeenkomen met de definitie van de bouwsteen. Dat kan (toevallig) overeenkomen met de manier waarop de datatypes en waardelijsten in de database zijn geïmplementeerd maar het kan ook zijn dat er op een eenduidige manier een vertaling moet plaatsvinden bijvoorbeeld via mapping.
- Dat de relatie tussen bouwstenen onderling (bv een indicatie gerelateerd aan een verrichting), ook fysiek in de onderliggende database is vastgelegd of daaruit afgeleid kan worden.

Die view of kijk op de bouwstenen kan op een abstracte/logische manier ingevuld worden doordat er in het zorginformatiesysteem ergens een "laag" is waar de informatie op die manier gemodelleerd terug te vinden is maar het kan ook een meer concrete vorm aannemen doordat er daadwerkelijk een externe interface op het systeem is waar die view terug te vinden is.

Een zorginformatiesysteem dat zib compliant is moet dus in staat zijn om een eenduidige vertaling te maken van de interne fysieke gegevensmodellen naar de logische gegevensmodellen van de bouwstenen en vice versa, inclusief de datatypes, waardelijsten en de onderlinge relaties tussen de bouwstenen.

Meer over het onderwerp zib compliance is te vinden in het document<sup>2</sup> "Zib compliance, een raamwerk en aanpak voor toetsing" en de bijbehorende spreadsheet<sup>3</sup> "Template zib compliance toets Interne Inrichting", allebei te downloaden op de website van het programma Registratie aan de bron.

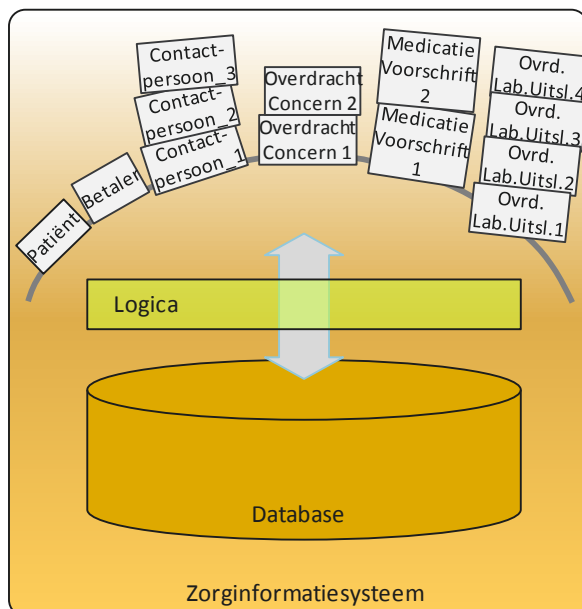
<sup>2</sup> <https://www.registratieaandebron.nl/pdf/Zib-compliance%E2%80%93v1.pdf>

<sup>3</sup> [https://www.registratieaandebron.nl/pdf/Template\\_Zib-compliancetoets\\_Interne\\_Inrichting%E2%80%93v1.xlsx](https://www.registratieaandebron.nl/pdf/Template_Zib-compliancetoets_Interne_Inrichting%E2%80%93v1.xlsx)

### 3.3 Vulling en instantiaties

De discussie ging tot dusver over het gegevensmodel versus de database implementatie. Een volgende stap is de vulling van de database met concrete gegevens.

Na implementatie van de database, is het mogelijk om gegevens conform de specificaties van een zib, dat wil zeggen als een zib-instantiatie, aan te bieden, waarna die in het interne formaat van de database worden weggeschreven. Andersom geldt ook dat de gegevens die opgeslagen zijn in het interne formaat van de database, daar weer uitgelezen kunnen worden in de vorm die overeenkomt met de specificaties van de zib, als een zib-instantiatie. Dit schrijven en lezen van zib's in en uit de database kan bijvoorbeeld gebeuren door speciaal voor dat doel ontwikkelde interfaces (bv. API's of webservices) of m.b.v. een invulscherm. Met behulp van dergelijke interfaces is het dan mogelijk één of meer (of een selectie van) instantiaties van een bouwsteen in de database te schrijven of daaruit te lezen. Bijvoorbeeld het schrijven van labuitslagen in de database of het lezen van de actuele medicatie van een patiënt uit de database. Dat is gevisualiseerd in Figuur 7.



**Figuur 7 - Instantiaties van bouwstenen met concrete gegevens op basis van onderliggende modellen**

Een zorginformatiesysteem dat zib compliant is moet dus in staat zijn om aan zib gerelateerde gegevenselementen die in de database aanwezig zijn, in de vorm van zib-instantiaties beschikbaar te stellen aan de buitenwereld. En vice versa: om zib-instantiaties die door de omgeving aan het systeem worden aangeboden, in de database te verwerken (te schrijven).

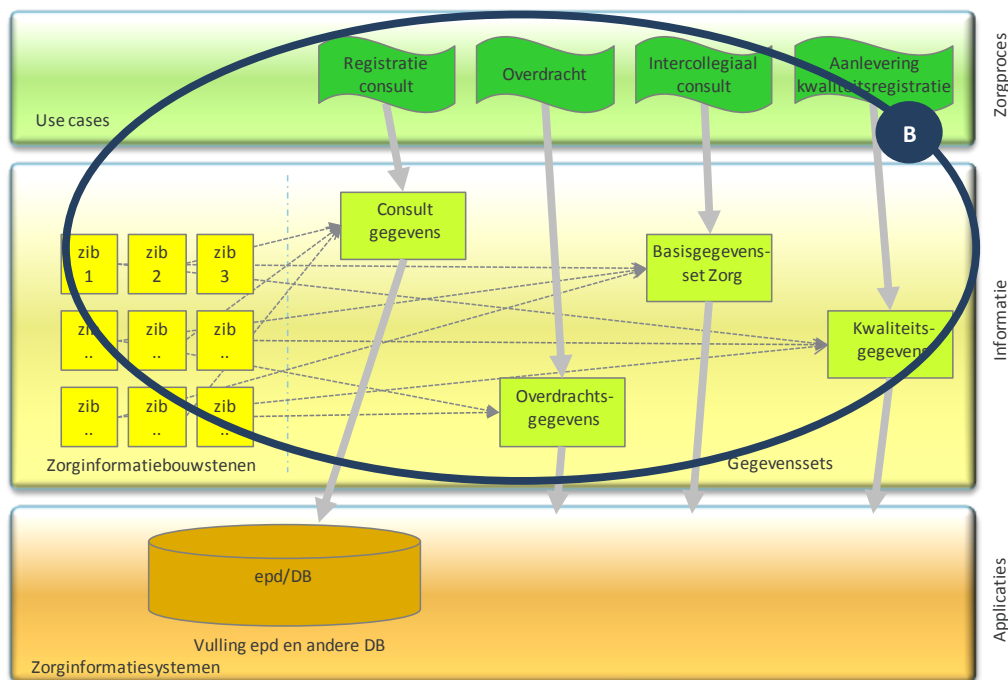
### 3.4 Niet voor alle gegevens bestaan zib's

Zoals al in paragraaf 2.2 is aangegeven, zullen niet voor alle gegevens bouwstenen bestaan of ontwikkeld worden, maar alleen voor die gegevens waarvoor dat van belang is in het kader van meervoudig gebruik. Een zorginformatiesysteem zal in de meeste gevallen dus ook gegevens bevatten waarvoor geen zib's bestaan. Voor de verwerking in en door het systeem maakt dat geen verschil; het betekent enkel dat er een ander logisch gegevensmodel gebruikt wordt dan het gegevensmodel van de zib standaard.

## 4 Gegevenssets bestaande uit zib-instantiaties

### 4.1 Inleiding

In het vorige hoofdstuk is beschreven wat de relatie is tussen zib's en de inrichting van zorginformatiesystemen en hun databases. In dit hoofdstuk wordt de relatie gelegd tussen de gegevens in de zorginformatiesystemen en het zorgproces en de rol die zib's daarbij spelen. Het mechanisme dat daaraan ten grondslag ligt is weergegeven in Figuur 8.



**Figuur 8 - Gegevenssets bestaande uit zib-instantiaties**

Uitgangspunt vormen de zorgprocessen. Daarbinnen zijn use cases (activiteiten) te onderkennen zoals in het voorbeeld van Figuur 8 'Registratie consult', 'Overdracht', 'Intercollegiaal Consult' en 'Aanlevering kwaliteitsregistraties'. Onderdeel van de use cases zijn transacties waarbij goed gedefinieerde sets van gegevens door zorgverleners worden vastgelegd, ingezien of uitgewisseld. Deze gegevenssets bestaan uit een selectie van specifieke aan het zorgproces en de use case gerelateerde gegevens. Elke use case en daarbij behorende transactie (er kunnen meerdere transacties per use case zijn) kent dus een eigen specifieke gegevensset. Voor de use cases die zijn weergegeven in Figuur 8 zijn dat bijvoorbeeld: 'Consultgegevens', 'Overdrachtsgegevens', 'Basisgegevensset Zorg' (= Medische Samenvatting) en 'Kwaliteitsgegevens'.

De gegevenssets behorende bij specifieke use cases worden op basis van de zorginformatiebouwstenen gedefinieerd als een set van bouwstenen, mogelijk aangevuld met gegevens waarvoor geen bouwstenen bestaan. Dat mechanisme wordt in dit hoofdstuk verder toegelicht.

### 4.2 Selectie van gegevens

Een zorginformatiesysteem dat zib-compliant is, is in staat om gegevens die worden aangeboden conform de definitie van de zib's, op te slaan en ook weer in lijn met die definitie op te leveren in de vorm van "zib-instantiaties"; zie paragraaf 3.3.

Het is ook mogelijk om meer specifieke “vragen” te stellen in de vorm van queries en op die manier een selectie te maken van gegevens uit de database die vervolgens in de vorm van zib-instantiaties worden weergegeven. Een aantal voorbeelden van zo’n selectie van gegevens:

- Alle instantiaties van de zib Patiënt van patiënten ouder dan 80 jaar
- Alle instantiaties van de zib OverdrachtVerrichting van het type orthopedische verrichting
- Alle instantiaties van de zib OverdrachtLaboratoriumUitslag van het huidige jaar voor patiënt X

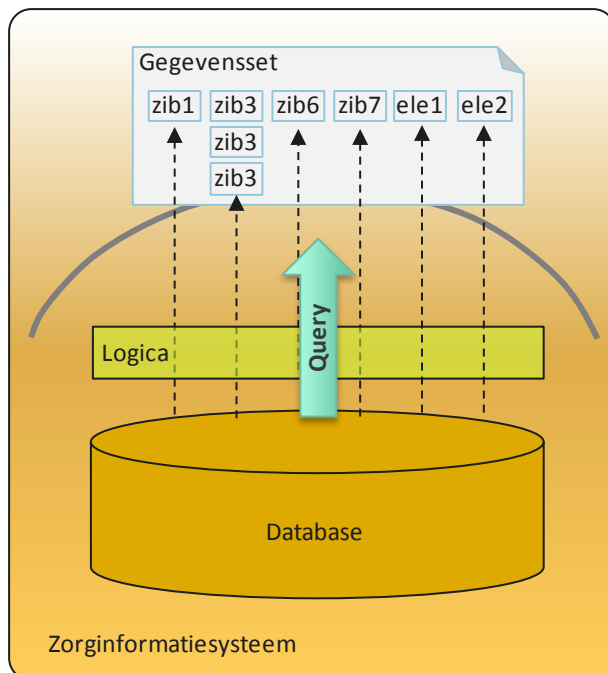
Een gegevensset voor een bepaalde use case kan op die manier worden samengesteld. Zo zou een (hypothetische) gegevensset A als volgt gedefinieerd kunnen worden.

Gegevensset A bestaat uit:

- Zib Patiënt; één instantiatie voor patiënt X
- Zib OverdrachtVerrichting; alle instantiaties voor patiënt X van type orthopedische verrichting
- Zib Lichaamsgewicht voor patiënt X; één instantiatie, het laatst gemeten gewicht
- Zib Lichaamslengte voor patiënt X, één instantiatie, het laatst gemeten lengte

Dit voorbeeld is verder in detail uitgewerkt in Bijlage B - Voorbeeld instantiatie gegevensset.

Figuur 9 geeft het proces weer voor het geval er niet voor alle gegevens, die onderdeel zijn van de gegevensset, bouwstenen zijn. Een gegevensset wordt in dat geval samengesteld door middel van een query op de database waarbij de resultaten vervolgens in een aantal zib-instantiaties worden weergegeven (zib1, zib3, zib6, zib7) en een aantal instantiaties vangegevenselementen waarvoor geen zib's bestaan (ele1, ele2).



**Figuur 9 - Samenstellen van een gegevensset**

Een gegevensset die onderdeel vormt van een bepaalde use case wordt samengesteld uit een selectie van gegevens uit de database op basis van specifieke selectieregels waarvan de resultaten worden weergegeven als instantiaties van zib's, mogelijk aangevuld met instantiaties van gegevens waarvoor (nog) geen bouwstenen bestaan. Die selectieregels zijn gebaseerd op de informatiebehoefte van zorgprofessionals om hun taak uit te kunnen voeren in een bepaald zorgproces.

### Mogelijke aanvullende bewerking

Naast de beschreven methodiek van selectie kan er in principe voor het samenstellen van een gegevensset ook nog een bewerking op de geselecteerde gegevens plaatsvinden. Een voorbeeld van een bewerking is: bepaal de hoogst gemeten bloeddruk tijdens een bepaalde tijdsperiode.

Als in de context van een concrete use case de hoogst gemeten bloeddruk tijdens een bepaalde tijdsperiode de gewenste informatie is, kan er voor gekozen worden om alle bloeddrukken die tijdens die bepaalde periode gemeten zijn in de vorm van meerdere zib-instantiaties onderdeel te laten zijn van de gegevensset. Aan de ontvangende kant kan dan de hoogst gemeten bloeddruk bepaald worden. Het is ook mogelijk om aan de zendende kant, de hoogst gemeten bloeddruk te bepalen, en die in de vorm van een enkele zib-instantiatie onderdeel te laten zijn van een gegevensset. In het gegeven voorbeeld kan de bewerking, het bepalen van de hoogst gemeten bloeddruk, dus zowel aan de zendende als aan de ontvangende kant plaatsvinden. Deze keuzes worden gemaakt tijdens het systeemontwerp. Keuzes die gemaakt worden kunnen b.v. ingegeven zijn door privacy eisen die aan de oplossing worden gesteld (niet meer gegevens oversturen dan nodig voor het doel).

### **4.3 De praktijk**

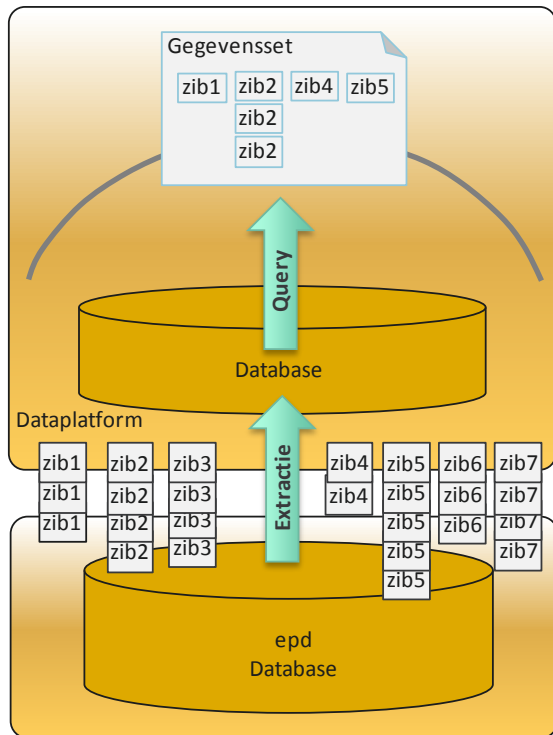
In het voorgaande is een mechanisme beschreven dat in feite bestaat uit twee stappen:

- Gegevens die opgeslagen zijn in de database kunnen door een zib compliant systeem aan de buitenwereld beschikbaar worden gesteld in de vorm van instantiaties van bouwstenen.
- Voor een specifieke use case kan een gegevensset worden gedefinieerd die bestaat uit een selectie van gegevens uit de database die in de vorm van instantiaties van bouwstenen worden weergegeven.

Er zijn verschillende manieren om dit mechanisme in de praktijk te implementeren. Bij de systeem variant die is weergegeven in Figuur 9 spelen de genoemde stappen zich volledig in één systeem af en komt het resultaat, de gegevensset, via een interface (een user interface of een webservice) voor de buitenwereld beschikbaar. Het systeem in figuur 9 zou een elektronisch patiëntendossier (epd) kunnen zijn dat steeds de actuele situatie van een patiënt realtime kan weergeven.

Een andere variant zou kunnen bestaan uit een epd met daarnaast een dataplatform (bv. een data warehouse) voor managementinformatie of research. Die variant is weergegeven in Figuur 10. Het dataplatform kan dienen als bron voor de aanlevering aan kwaliteitsregistraties. Hierbij gaat het niet om realtime informatie (dat is voor die processen niet nodig), maar bijvoorbeeld om geaggregeerde en geanonimiseerde data. Op deze manier zijn er in de praktijk veel verschillende varianten mogelijk, afgestemd op de eisen die het proces stelt, maar de principestappen blijven gelijk.





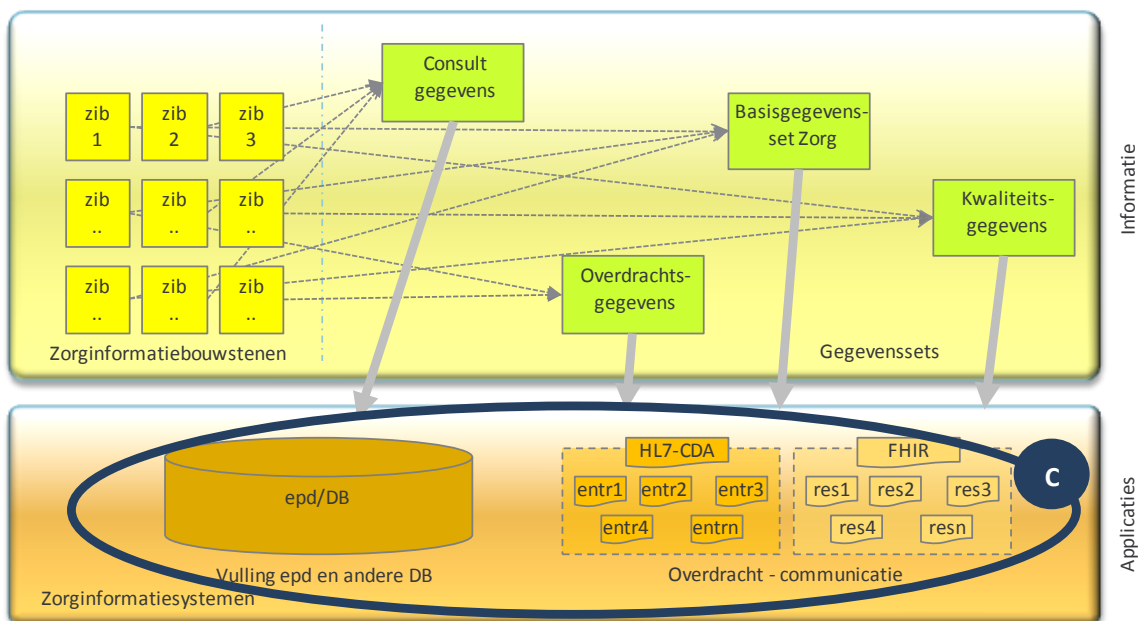
**Figuur 10 - Variant met een DWH**

## 5 Uitwisseling op basis van HL7 CDA en FHIR

### 5.1 Inleiding

In de vorige hoofdstukken is beschreven op welke manier op zib gebaseerde instantiaties van gegevens(sets) kunnen worden samengesteld op basis van informatie in onderliggende databases van systemen die zib compliant zijn. Deze gegevenssets zijn specifiek voor bepaalde zorgprocessen en daarbij behorende use cases. In de context van deze use cases kan het zijn dat deze gegevenssets (of subsets daarvan) gedeeld worden doordat ze via schermen ingezien kunnen worden, maar het kan ook zijn dat de gegevens uitgewisseld worden tussen systemen. In dit hoofdstuk wordt beschreven hoe zo'n uitwisseling in de praktijk werkt op basis van HL7 CDA en HL7 FHIR. Andere oplossingen zijn ook mogelijk maar worden in dit document niet beschreven.

De stappen die in de vorige hoofdstukken beschreven zijn, staan los van de manier waarop de uitwisseling plaatsvindt. Uitgangspunt voor de uitwisseling vormen gegevenssets zoals beschreven in het vorige hoofdstuk. Het formaat van uitwisseling is met name terug te vinden op de applicatielaag.



Figuur 11 - Uitwisseling op basis van HL7 CDA of FHIR

### 5.2 HL7 CDA en FHIR

Beide zijn standaarden die zijn ontwikkeld door HL7 (Health Level 7)<sup>4</sup>, een internationale standaardisatieorganisatie op het gebied van interoperabiliteit van zorginformatie.

#### 5.2.1 HL7 CDA (Clinical Document Architecture)

HL7 CDA is een onderdeel van de HL7v3 standaard. De laatste versie (Release 2.0) stamt uit 2005. Een kerncomponent van die standaard is het HL7 Referentie Informatie Model (RIM). In het RIM zijn de structuur van de gegevens, de concepten, datatypes en vocabulaire vastgelegd waardoor interoperabiliteit gewaarborgd kan worden. HL7 CDA documenten zijn op XML gebaseerd en specificeren de codering, structuur en semantiek van klinische documenten die bedoeld zijn voor het opslaan en uitwisselen van

<sup>4</sup> Nationaal: <http://www.hl7.nl/>

Internationaal: <http://www.hl7.org/>

medische gegevens. Een belangrijk kenmerk van een CDA document is de menselijke leesbaarheid. Het document bevat alle informatie die nodig is om de inhoud te kunnen interpreteren. Zowel gestructureerde als ongestructureerde tekst kan worden opgenomen.

Een meer gedetailleerde beschrijving van HL7 CDA is te vinden in de Bijlage C - HL7 CDA. Daar is beschreven dat de klinische informatie in het document vastgelegd wordt in de body van het document die bestaat uit zogeheten “sections” en daarbinnen “entries”. Bouwstenen komen terug op entry-level niveau. Door de zib-modellen te mappen naar entry-level templates is het mogelijk om op een eenduidige en reproduceerbare manier gegevens die gebaseerd zijn op de bouwstenen onder te brengen in CDA documenten.

De CDA standaard specificeert niet hoe de documenten worden getransporteerd. Meer over mogelijke transportmechanismen in hoofdstuk 6..

### 5.2.2 HL7 FHIR (Fast Healthcare Interoperability Resources)

De FHIR standaard is een latere ontwikkeling van HL7 waarin de ervaring van eerdere standaarden is meegenomen. FHIR is volledig op basis van internetstandaarden ontwikkeld, waaronder XML, JSON en HTTP. Een belangrijke toepassing is bijvoorbeeld het ontsluiten van epd systemen voor mobiele toepassingen.

De kern van FHIR wordt gevormd door herbruikbare bouwstenen, de zogenaamde resources, waarmee oplossingen voor uitwisseling van zowel administratieve als zorginhoudelijke gegevens kunnen worden gebouwd. Een lijst van beschikbare resources is te vinden op <http://www.hl7.org/fhir/resourcelist.html>. D.m.v. profiling (het maken van specificaties van aanscherpingen in de vorm van structure definitions en extensions) en onderlinge verwijzingen is het mogelijk een specifieke set van resources voor een bepaalde use case te definiëren.

Door de gegevenselementen van de zib-modellen te mappen op FHIR resource-elementen is het mogelijk om op een eenduidige en interoperabele manier gegevens die gebaseerd zijn op de bouwstenen onder te brengen in FHIR resources.

Een meer gedetailleerde beschrijving van HL7 FHIR is te vinden in Bijlage D - HL7 FHIR.

Daar wordt ook uitgelegd dat FHIR meerdere uitwisselingsmethoden ondersteunt:

- Op basis van resources
- Op basis van messages (qua methodiek vergelijkbaar met HL7V2 messaging)
- Op basis van documenten (qua methodiek vergelijkbaar met HL7 CDA)

### 5.2.3 Discussie verschil CDA en FHIR

Zoals uit de paragrafen hiervoor blijkt zijn er overeenkomsten maar ook behoorlijk verschillen tussen CDA en FHIR.

De CDA standaard bestaat langer en beperkt zich tot de definitie van een klinisch document, d.w.z. de structuur en de manier waarop de inhoud van het document moet worden vastgelegd. Dat klinisch document kan gebruikt worden om gegevenssets voor uitwisseling te “verpakken” zodat de gegevens via een transportmechanisme gedeeld en vervolgens weer “uitgepakt” kunnen worden.

De ervaring leert dat de definitie van een HL7 CDA template voor een gegevensset voor een bepaalde use case de nodige specialistische kennis behoeft en dus betrokkenheid van HL7 CDA experts. Zodra een template gedefinieerd is, kan het op eenduidige en reproduceerbare wijze door leveranciers geïmplementeerd worden in de software van de systemen.

De FHIR standaard bestaat veel korter en is bedoeld om in te spelen op ontwikkelingen in de markt die meer en meer verschuiven richting internettechnologieën en mobiele toepassingen. Daarbij hoort ook de wens om snellere en meer iteratieve softwareontwikkelmethodieken toe te passen. Het idee is dat de

kenmerken van FHIR het mogelijk maken dat relatief snel oplossingen ontwikkeld kunnen worden die in de praktijk kunnen worden gebruikt. Waar bij CDA vaak specifieke HL7 kennis vereist is, is FHIR voor ontwikkelaars gewoon “another API”.

Het feit dat FHIR gebaseerd is op zogeheten resources is een groot verschil met CDA. De resources worden binnen FHIR gedefinieerd als herbruikbare bouwstenen en zijn daarmee conceptueel vergelijkbaar met zib's (zorginformatiebouwstenen). In de voorgaande hoofdstukken is al uitgelegd dat bij systemen die zib-compliant zijn de inrichting van de databases is gebaseerd op de definitie van de zib's (Hoofdstuk 3) en dat vervolgens de definitie van de gegevenssets gebaseerd is op een selectie van zib-instantiaties (Hoofdstuk 4). Dat betekent dat in het geval gekozen wordt voor FHIR als formaat voor uitwisseling, er op model niveau een mapping of transformatie moet plaatsvinden van zib's → FHIR resources en omgekeerd van FHIR resources → zib's. Meer over mapping in paragraaf 5.4.1.

Ook het feit dat de uitwisseling bij FHIR ook kan plaatsvinden op het niveau van de resources in plaats van in de vorm van een document zoals bij CDA, is een verschil. Een ander verschil is dat de FHIR standaard, anders dan de CDA standaard ook een uitwisselingsmechanisme specificeert, de FHIR RESTful services. Meer daarover in par. 6.4.

In de volgende paragrafen wordt in meer detail beschreven hoe uitwisseling van gegevens in zib-compliant systemen werkt voor respectievelijk HL7 CDA en HL7 FHIR. Beide standaarden vinden in de praktijk hun toepassing. Het zal afhangen van de specifieke situatie, de mogelijkheden die de applicaties bieden en mogelijk factoren als kennis en ervaring van betrokken partijen welke keuzes in praktische oplossingen gemaakt worden.

### 5.3 Uitwisseling op basis van HL7 CDA

#### 5.3.1 Mapping van zib's en CDA entries

Bij de mapping van gegevenselementen tussen zib's en CDA entries wordt uitgegaan van de volgende regels:

- zib's worden gemapt op Entry Level Templates<sup>5</sup>
- alle gegevenselementen van zib's moeten volledig en ondubbelzinnig gemapt kunnen worden op CDA elementen en weer terug wat betreft:
  - de DCM::DefinitieCode c.q. de definitie (in het code XML element)
  - het datatype
  - de waardelijst
  - de structuur en de onderlinge relaties met andere gegevenselementen
- de mapping is vastgelegd door
  - een id (DCM::ConceptId) over te nemen (in het templateId XML element)
  - een losse spreadsheet met mappings
- de mapping is bij voorkeur computable (door de computer verwerkbaar)
- als waardelijsten niet overeenkomen moet er een mapping definitie zijn die openbaar is en bij voorkeur computable
- in het CDA template kunnen meer elementen voorkomen dan er zib gegevenselementen zijn. Mits alle zib gegevenselementen volledig en ondubbelzinnig gemapt kunnen worden is dat geen probleem

---

<sup>5</sup> De zib's zijn vergelijkbaar met de IHE XDS Entry Content Modules. Zie bijvoorbeeld entry level template voor Allergy Intolerance:

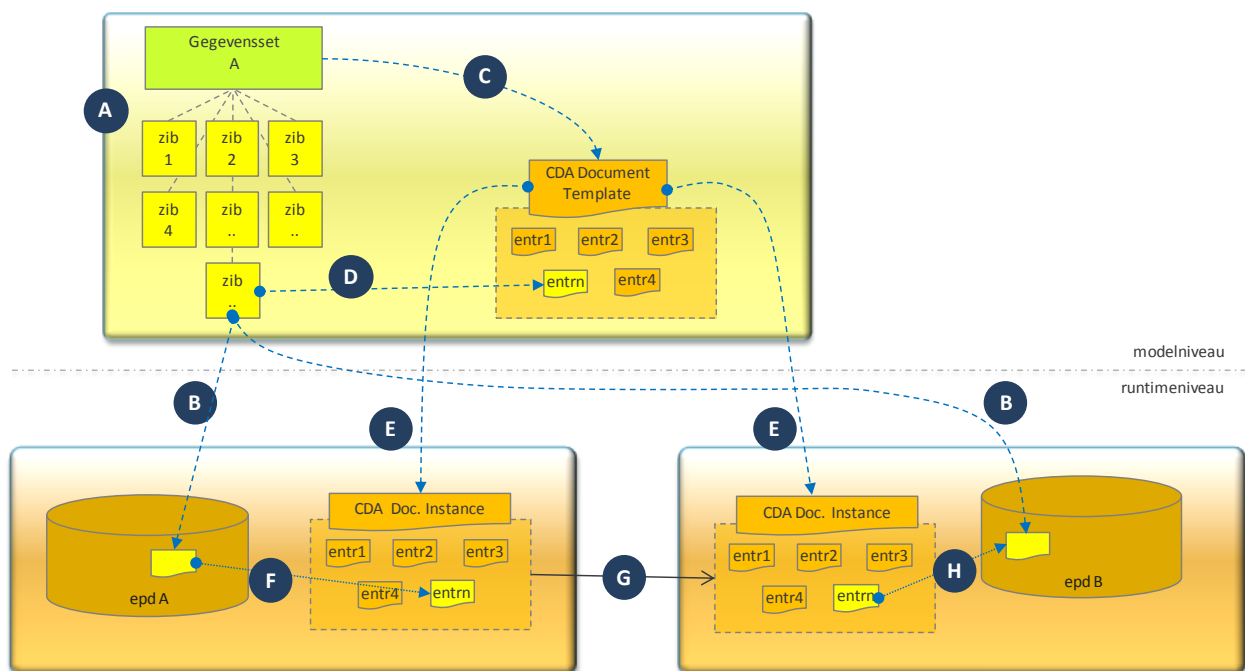
[http://wiki.ihe.net/index.php/CDA\\_Entry\\_Content\\_Modules#Allergy\\_and\\_Intolerance\\_Concern\\_1.3.6.1.4.1.1\\_9376.1.5.3.1.4.5.3](http://wiki.ihe.net/index.php/CDA_Entry_Content_Modules#Allergy_and_Intolerance_Concern_1.3.6.1.4.1.1_9376.1.5.3.1.4.5.3)

- het is aan te raden om een zo passend mogelijke doel CDA class te kiezen, bijvoorbeeld voor de zib Contact, de CDA class Encounter

Zie: Bijlage E – Mapping van zibs en CDA entries; een voorbeeld.

## 5.3.2 Uitwisseling op basis van CDA documenten

Het principe van uitwisseling van gegevens op basis van HL7 CDA wordt beschreven aan de hand van Figuur 12. In de praktijk is deze uitwisseling onderdeel van een zorgproces en gaan er menselijke handelingen aan vooraf. In deze paragraaf wordt echter alleen ingegaan op de specifieke aspecten die te maken hebben met de manier waarop zo'n uitwisseling plaatsvindt op basis van CDA documenten. In het kader van een bepaalde use case is een gegevensset A gedefinieerd zoals beschreven in de voorgaande hoofdstukken. Die gegevensset wordt uitgewisseld tussen epd A en epd B.



**Figuur 12 - Uitwisseling op basis van HL7 CDA**

Toelichting bij Figuur 12:

- Een gegevensset A wordt als onderdeel van een use case gedefinieerd op basis van zib's als model (zie hoofdstuk 4)
- De epd's en bijbehorende databases zijn zib compliant ingericht (zie hoofdstuk 3)
- Op basis van de definitie van de gegevensset wordt een CDA Document Template gedefinieerd met sectie en entry level templates (zie Bijlage C - HL7 CDA)
- De zib modellen zijn gemapt op de entry-level templates
- Als gegevens uitgewisseld worden tussen epd A en epd B wordt een CDA instantiatie gegenereerd op basis van de CDA Document Template;..
- Die instantiatie wordt gevuld met de juiste gegevens uit epd A op basis van de specificatie d.m.v. mapping of transformatie (fysiek model → zib)
- De CDA instantiatie wordt als een document verstuurd (zie hoofdstuk 6)
- De inhoud van het ontvangen document wordt d.m.v. mapping of transformatie (zib → fysiek model), feitelijk via de omgekeerde stappen van het begin, verwerkt (of getoond, of overgenomen etc.) in epd B

Het hierboven beschreven mechanisme vormt de basis voor het ontwerp van de software in de applicaties die de CDA instantiaties genereert en verwerkt. Runtime, d.w.z. bij de daadwerkelijke uitwisseling van gegevens, vindt de mapping van de (fysieke) gegevens van de database naar de CDA instantiatie en vice versa door de software direct plaats. In Figuur 12 is dat weergegeven door onderscheid te maken tussen modelniveau en runtimeniveau.

Het mechanisme waarmee de uitwisseling plaatsvindt (bolletje G in de figuur) wordt toegelicht in hoofdstuk 6.

#### 5.4 Uitwisseling op basis van HL7 FHIR

Voor de uitwisseling van gegevens op basis van FHIR tussen zib compliant systemen, is de mapping van zib's en FHIR resources van belang. Dat wordt verder toegelicht in par. 5.4.1. Vervolgens wordt in par. 5.4.2 de uitwisseling op basis van FHIR documenten (document exchange format) beschreven en in par. 5.4.3 de uitwisseling op basis van enkelvoudige resources (resource exchange format). Op uitwisseling op basis van FHIR messages wordt in dit document niet verder ingegaan omdat dat in de praktijk (nog) niet vaak aan de orde is.

Meer informatie over FHIR in Bijlage D - HL7 FHIR, van dit document.

##### 5.4.1 Mapping van zib's en FHIR resources

FHIR resources zijn in de basis generiek en worden met behulp van profielen uitgebreid en specifiek gemaakt voor een specifieke toepassing. In een profiel wordt bijvoorbeeld beschreven:

- Welke resource elementen worden gebruikt en welke niet en welke additionele elementen worden toegevoegd die geen onderdeel zijn van de basisspecificatie
- Welke terminologieën worden gebruikt in bepaalde elementen
- Hoe de resource elementen mappen naar lokale eisen en/of implementaties

Door de manier waarop profiling wordt toegepast binnen FHIR kunnen er voor een bepaalde basis resource een groot aantal verschillende profielen bestaan, bijvoorbeeld afhankelijk van zorgdomein, land, instelling of leverancier. Om interoperabiliteit te borgen is het van belang dat binnen een bepaalde use case dezelfde profielen gebruikt worden. Een register met FHIR profielen is te vinden op [www.simplifier.net](http://www.simplifier.net) (in samenwerking met HL7 Nederland).

De kern van de relatie tussen bouwstenen op basis van zib's en FHIR is dat de bouwstenen op modelniveau via mapping "omgezet" moeten worden in FHIR resources en vice versa. Daarvoor zijn dus specifieke profielen nodig. Die profielen worden door Nictiz en andere marktpartijen ontwikkeld en in beheer gegeven bij HL7 Nederland. De profielen zijn terug te vinden op <https://www.simplifier.net/NL-BasicComponents> (het project NL-BasicComponents). Het onderhanden werk bij Nictiz is terug te vinden op <https://www.simplifier.net/nictiz>.

Bij de mapping van gegevenselementen tussen zib's en FHIR resources wordt uitgegaan van de volgende regels:

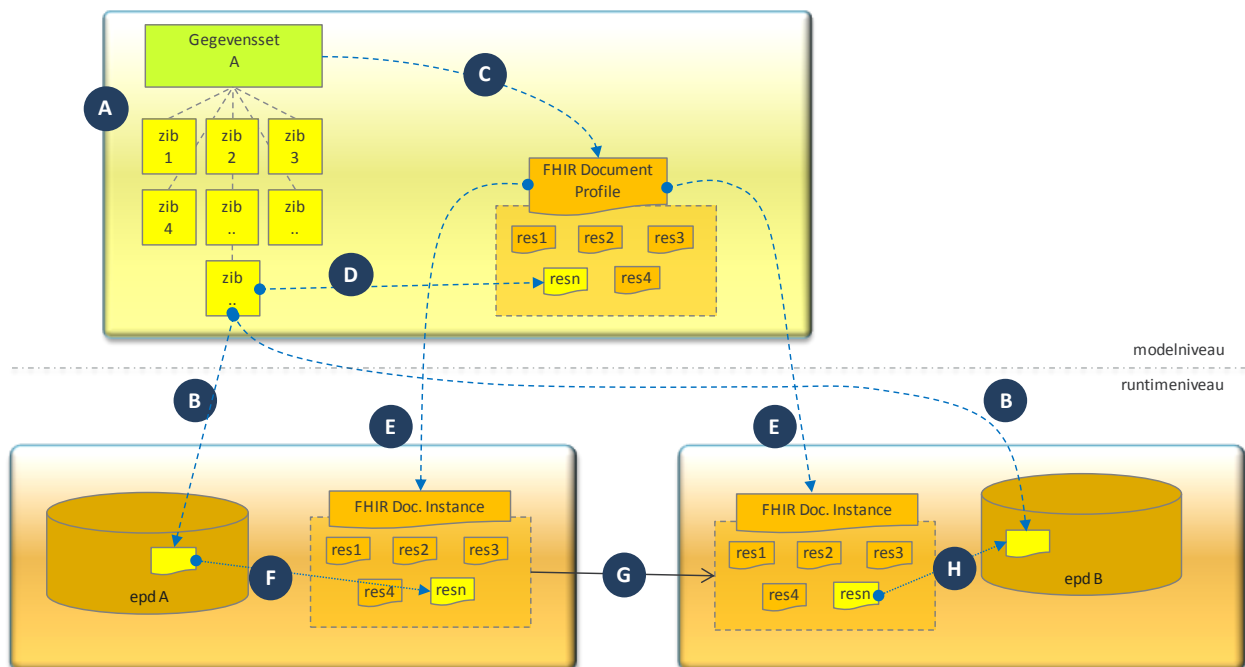
- alle gegevenselementen van zib's moeten volledig en ondubbelzinnig gemapt kunnen worden op FHIR resource-elementen en weer terug wat betreft:
  - de DCM::DefinitieCode c.q. de definitie
  - het datatype
  - de waardelijst
  - de structuur en de onderlinge relaties met andere gegevenselementen

- de mapping is vastgelegd in de vorm van het mapping onderdeel van een FHIR StructureDefinition<sup>6</sup>/ ElementDefinition<sup>7</sup> profile. de mapping is bij voorkeur computable (door de computer verwerkbaar)
- als waardelijsten niet overeenkomen moet er een mapping definitie zijn
- in het kader van maximale interoperabiliteit heeft het de voorkeur de waardelijsten van de doel FHIR resource gebruiken
- in het FHIR profiel kunnen meer resource-elementen voorkomen dan er zib gegevenselementen zijn. Mits alle zib gegevenselementen volledig en ondubbelzinnig gemapt kunnen worden, is dat geen probleem.
- Voor sommige zib's zijn er meer FHIR resources die bruikbaar zijn om de zib op te mappen. Het is dan aan te raden om een zo passend mogelijk doel resource te kiezen, bijvoorbeeld voor de zib Contact de FHIR resource Encounter (en niet Observation die ook bruikbaar zou zijn).

Een voorbeeld van een uitvoerige mapping is te vinden in Bijlage F – Mapping van zibs en FHIR resources; een voorbeeld.

## 5.4.2 Uitwisseling op basis van FHIR documents

Het principe van uitwisseling van gegevens op basis van HL7 FHIR document format wordt beschreven aan de hand van Figuur 13. In de praktijk is deze uitwisseling onderdeel van een zorgproces en gaan er menselijke handelingen aan vooraf. In deze paragraaf wordt echter alleen ingegaan op de specifieke aspecten die te maken hebben met de manier waarop zo'n uitwisseling plaatsvindt op basis van FHIR documenten. In het kader van een bepaalde use case is een gegevensset A gedefinieerd zoals beschreven in de voorgaande hoofdstukken. Die gegevensset wordt uitgewisseld tussen epd A en epd B. Het mechanisme waarmee de uitwisseling plaatsvindt wordt hier niet toegelicht maar besproken in hoofdstuk 6.



Figuur 13 - Uitwisseling op basis van HL7 FHIR documenten

<sup>6</sup> <https://www.hl7.org/fhir/structuredefinition.html>

<sup>7</sup> <https://www.hl7.org/fhir/elementdefinition.html>

Toelichting bij Figuur 13:

- A. Een gegevensset A wordt als onderdeel van een use case gedefinieerd op basis van zib's als model (zie hoofdstuk 4)
- B. De epd's en bijbehorende data bases zijn ingericht op basis van het gegevensmodel van de zib's (zie hoofdstuk 3)
- C. Op basis van de definitie van de gegevensset wordt een FHIR Document Profile gedefinieerd op basis van FHIR resources
- D. De FHIR resources in het document zijn gebaseerd op een mapping van zib's naar resources.(zie par. 5.4.1)
- E. Als gegevens uitgewisseld worden tussen epd A en epd B wordt een FHIR document instance gegenereerd op basis van de FHIR Document Profile
- F. Die instantiatie wordt gevuld met de juiste gegevens uit epd A op basis van de specificatie d.m.v. mapping of transformatie (fysiek model → FHIR resource)
- G. Het FHIR document wordt als een document verstuurd (zie hoofdstuk E)
- H. De inhoud van het ontvangen document wordt d.m.v. mapping of transformatie (FHIR resource → fysiek model), feitelijk via de omgekeerde stappen van het begin, verwerkt (of getoond, of overgenomen etc.) in epd B

Het mechanisme van FHIR documenten is vergelijkbaar met de methodiek van uitwisseling op basis van CDA gebaseerde documenten.

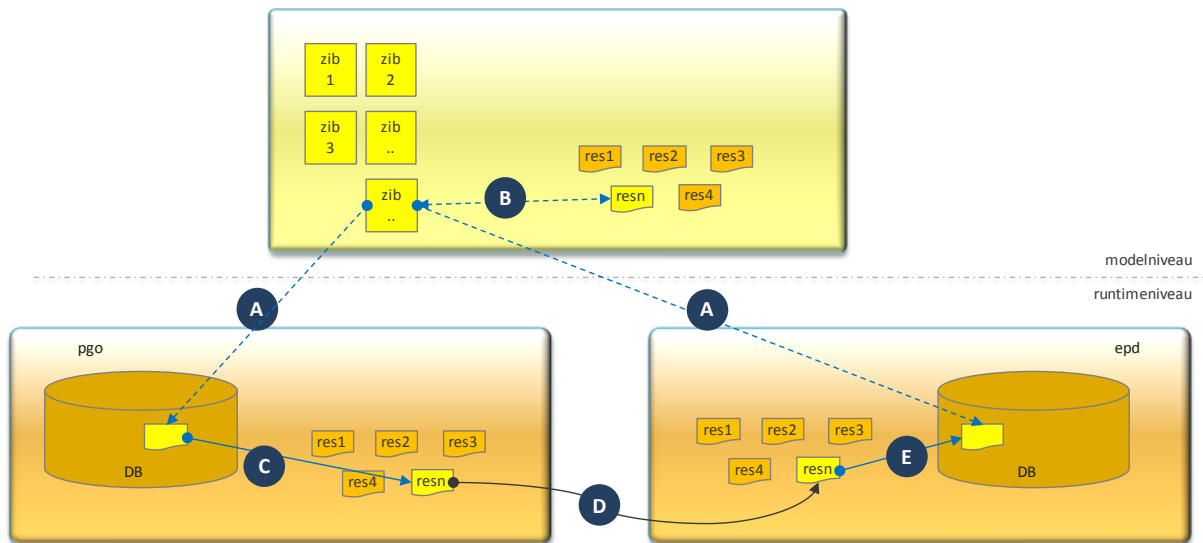
Ook in dit geval vormen de zib's het gegevensmodel op basis waarvan de epd's en bijbehorende databases zijn ingericht. Verder vindt er op model niveau een mapping of transformatie plaats van zib's → FHIR resources en omgekeerd (par. 5.4.1). Dat vormt de basis waarop de inhoud, d.w.z. de resources en hun onderlinge samenhang, van het FHIR Document Profile zijn gebaseerd. En dat vormt weer de basis voor het ontwerp van de software in de applicaties die de FHIR resources en document instantiaties genereert en verwerkt. Tijdens runtime, d.w.z. de daadwerkelijke uitwisseling van gegevens, vindt de mapping van de (fysieke) gegevens van de database naar de FHIR resources en document instantiatie en vice versa door de software rechtstreeks plaats. In Figuur 13 is dat weergegeven door onderscheid te maken tussen modelniveau en runtimeniveau.

Het mechanisme waarmee de uitwisseling plaatsvindt (bolletje G in de figuur) wordt toegelicht in hoofdstuk 6.

#### 5.4.3 Uitwisseling op basis van (enkelvoudige) FHIR resources

Naast de uitwisseling van FHIR documents en FHIR messages ondersteunt de FHIR standaard ook mechanismes voor het uitwisselen van enkelvoudige resources. Als uitwisselingsmechanisme wordt daarbij gebruik gemaakt van RESTful services (zie paragraaf 6.4). Deze mogelijkheid van uitwisseling zou bijvoorbeeld toepassing kunnen vinden bij het koppelen van een pgo (persoonlijke gezondheidsomgeving) met een epd van een instelling. Daarbij gaat het niet zozeer om de uitwisseling van specifieke gegevenssets maar bijvoorbeeld om het actualiseren van bepaalde gegevens. Zo'n opzet is weergegeven in Figuur 14.





**Figuur 14 – Het beschikbaar maken van gegevens op basis van eenvoudige FHIR resources**

Er zijn verschillende interacties mogelijk in deze opzet. In het voorbeeld van de figuur gaan we er van uit dat er op initiatief van de pgo een update van een bepaald gegevenselement plaatsvindt in het epd. Het mechanisme dat daarvoor wordt gebruikt is gebaseerd op RESTful services hetgeen wordt toegelicht in paragraaf 6.4.

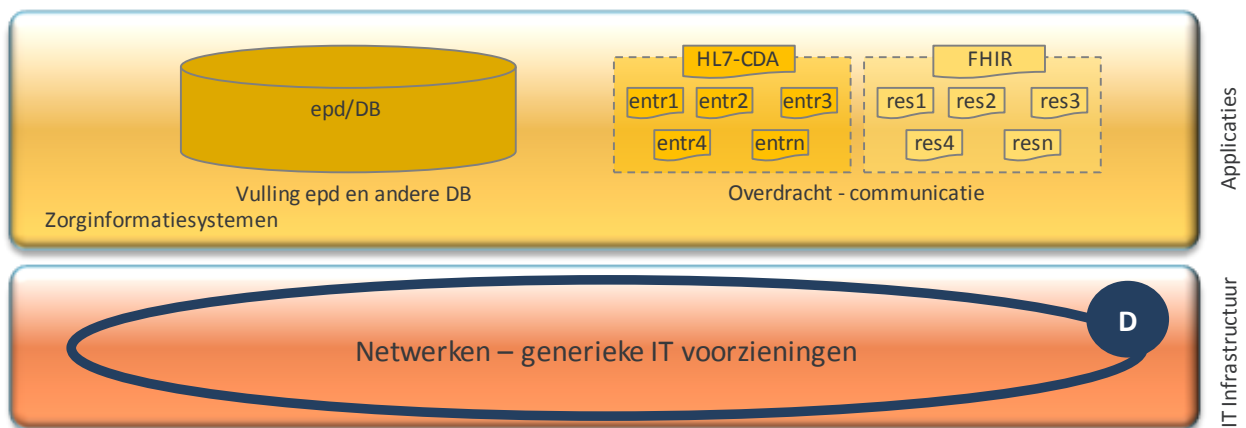
Toelichting bij Figuur 14:

- A. De pgo en het epd en bijbehorende data bases zijn zib compliant ingericht (zie hoofdstuk 3)
- B. Voor elke zib is er een mapping gedefinieerd naar FHIR resources (zib → FHIR resource) en omgekeerd (FHIR resource → zib) (zie par. 5.4.1)
- C. Op initiatief van de pgo vindt een schrijf of update actie plaats van bepaalde gegevens (bv. een thuismeting) in het epd. Daartoe wordt een instantiatie van de resource gegenereerd op basis van de (fysieke) gegevens in de database van de pgo
- D. Die resource instantiatie wordt via een mechanisme gebaseerd op FHIR RESTful services overgestuurd naar het epd (zie paragraaf 6.4)
- E. De inhoud van de ontvangen resource instantiatie wordt d.m.v. mapping of transformatie (FHIR resource → fysiek model), feitelijk via de omgekeerde stappen van het begin, verwerkt (of getoond, of overgenomen etc.) in het epd.

## 6 Praktische oplossingen voor de infrastructuur

### 6.1 Inleiding

Voor de daadwerkelijke uitwisseling van gegevens (zie vorige hoofdstuk) is een uitwisselingsmechanisme nodig als onderdeel van de IT infrastructuur. Dat is weergegeven in Figuur 15.



**Figuur 15 - Oplossingen voor de IT infrastructuur**

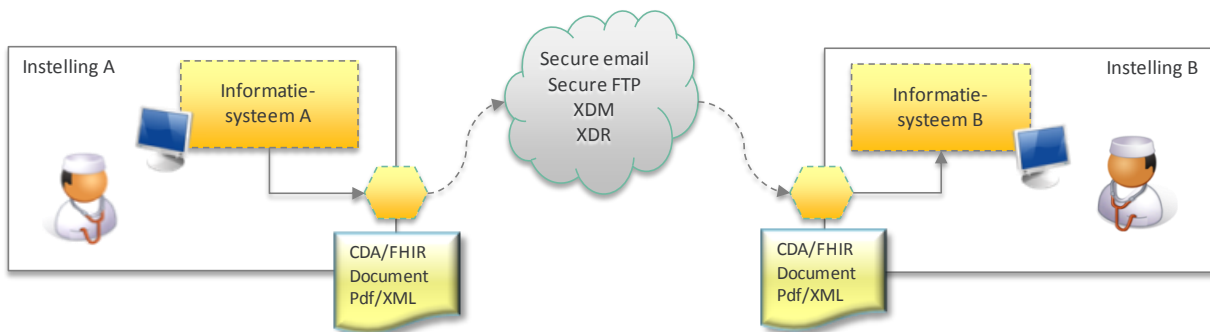
In dit hoofdstuk wordt een aantal transportmechanismes beschreven voor de uitwisselingsformaten van het vorige hoofdstuk. Een overzicht is weergegeven in Tabel 1. De opsomming is niet perse compleet, maar het is de bedoeling om een aantal oplossingen die in de dagelijkse praktijk het meest voor de hand liggen kort te beschrijven.

#	Praktijkoplossingen NB. Dit overzicht is niet compleet maar geeft een aantal oplossingen weer die in de praktijk het meest voor de hand liggen.	Type	Wat kan het formaat van de content zijn?		
			HL7 CDA Documents	HL7 FHIR Documents	HL7 FHIR Resources
1	Point-to-point oplossingen gebaseerd op <ul style="list-style-type: none"> <li>Secure email met attachment</li> <li>Secure FTP</li> <li>IHE XDM en IHE XDR profielen</li> </ul>	Push	V	V	-
2	Oplossingen gebaseerd op het IHE XDS profiel (zoals o.a. geïmplementeerd in regionale samenwerkingsverbanden)	Publish & pull	V	V	-
3	Client-server oplossingen gebaseerd op FHIR RESTful services	Push, pull	-	V	V
4	AORTA (LSP)	Pull, push	V	V	V

**Tabel 1 - Praktijkoplossingen voor de IT Infrastructuur**

## 6.2 Diverse point-to-point oplossingen.

Er is een aantal basismechanismen zoals attachment bij secure email, secure FTP en oplossingen gebaseerd op IHE XDM<sup>8</sup> (Cross-enterprise Document Media Interchange) en IHE XDR<sup>9</sup> (Cross-enterprise Document Reliable Interchange) dat voor CDA en FHIR document exchange gebruikt kan worden. Dat kan zowel in de vorm van een pdf als een xml bericht dat verstuurd wordt.



**Figuur 16 - Uitwisseling van documenten op basis van secure email etc.**

Deze oplossing is geschikt voor point-to-point uitwisseling van documenten op basis van push verkeer: de zendende partij neemt het initiatief om een document te versturen naar de ontvangende partij die bij de zendende partij bekend moet zijn. Dit is een relatief simpele oplossing voor één-op-één verkeer.

## 6.3 Oplossingen gebaseerd op het IHE XDS profiel

XDS<sup>10</sup> staat voor Cross-enterprise Document Sharing. XDS is een van de technische profielen van IHE. Het IHE-XDS profiel bestaat sinds 2003 en heeft als doel om medische documenten en beelden tussen zorginstellingen te delen op een gestandaardiseerde infrastructuur. Meerdere leveranciers kunnen op basis van het XDS profiel XDS infrastructuren bouwen, die informatie kunnen uitwisselen met XDS infrastructuren van andere leveranciers. Een XDS infrastructuur kan worden gezien als een generieke oplossing om documenten en beelden uit te wisselen. Het IHE-XDS profiel kan veel bestandsformaten uitwisselen zoals PDF, Word, JPG, DICOM en ook HL7 CDA en HL7 FHIR documenten.

Een XDS infrastructuur bestaat in principe uit vier samenwerkende systeemdelen (zie Figuur 17). De werking van XDS kan als volgt kort worden beschreven.

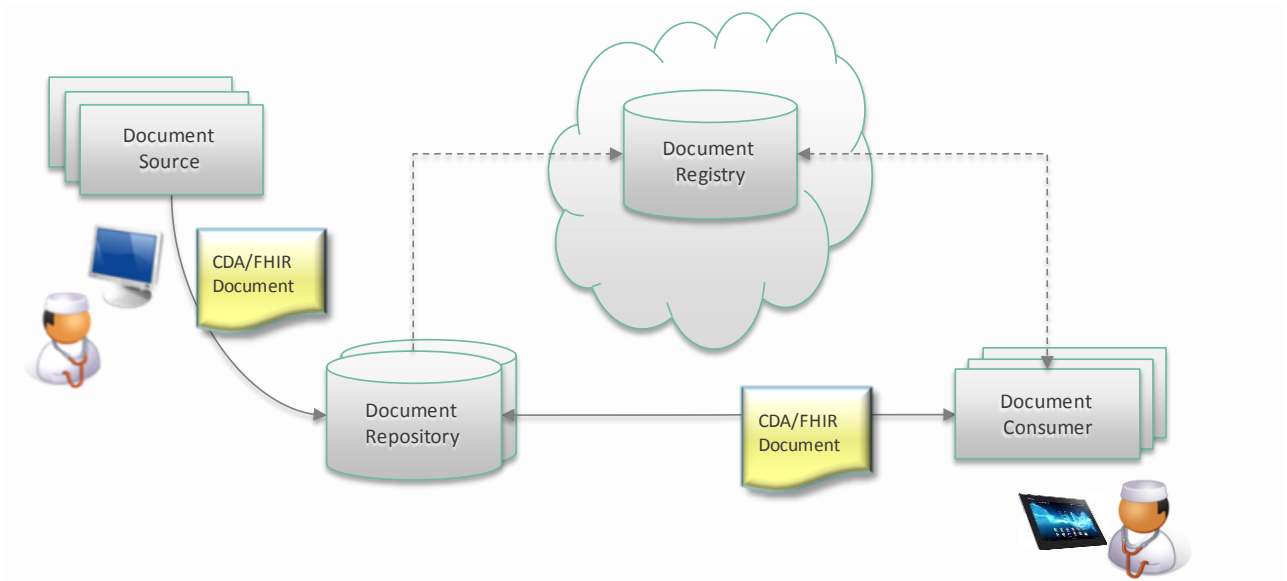
**Documenten opslaan:** een Document Source (zoals een epd, een lab systeem, een PACS) stuurt een document samen met de daarbij behorende metadata, naar een Document Repository (opslagomgeving voor beelden en/of documenten). De Document Repository slaat het bestand op, en stuurt vervolgens de metadata (aangevuld met de eigen metadata over hoe het bestand kan worden opgehaald) naar de Document Registry. De Document Registry bestaat uit een centrale index waarin deze metadata worden opgeslagen.

**Documenten opvragen:** de Document Consumer (zoals een zorginformatiesysteem, een epd, een viewer) vraagt aan de Document Registry welke bestanden er van een bepaalde patiënt beschikbaar zijn. De Document Registry stuurt een lijst terug met de beschikbare documenten (van alle aangesloten locaties). De eindgebruiker selecteert een van de bestanden uit de lijst, waarna de Document Consumer het gekozen document opvraagt aan de Document Repository en deze vervolgens toont aan de eindgebruiker.

<sup>8</sup> [http://wiki.ihe.net/index.php/Cross-enterprise\\_Document\\_Media\\_Interchange](http://wiki.ihe.net/index.php/Cross-enterprise_Document_Media_Interchange)

<sup>9</sup> [http://wiki.ihe.net/index.php/Cross-enterprise\\_Document\\_Reliable\\_Interchange](http://wiki.ihe.net/index.php/Cross-enterprise_Document_Reliable_Interchange)

<sup>10</sup> Bron: <https://www.nictiz.nl/standaarden/xds>



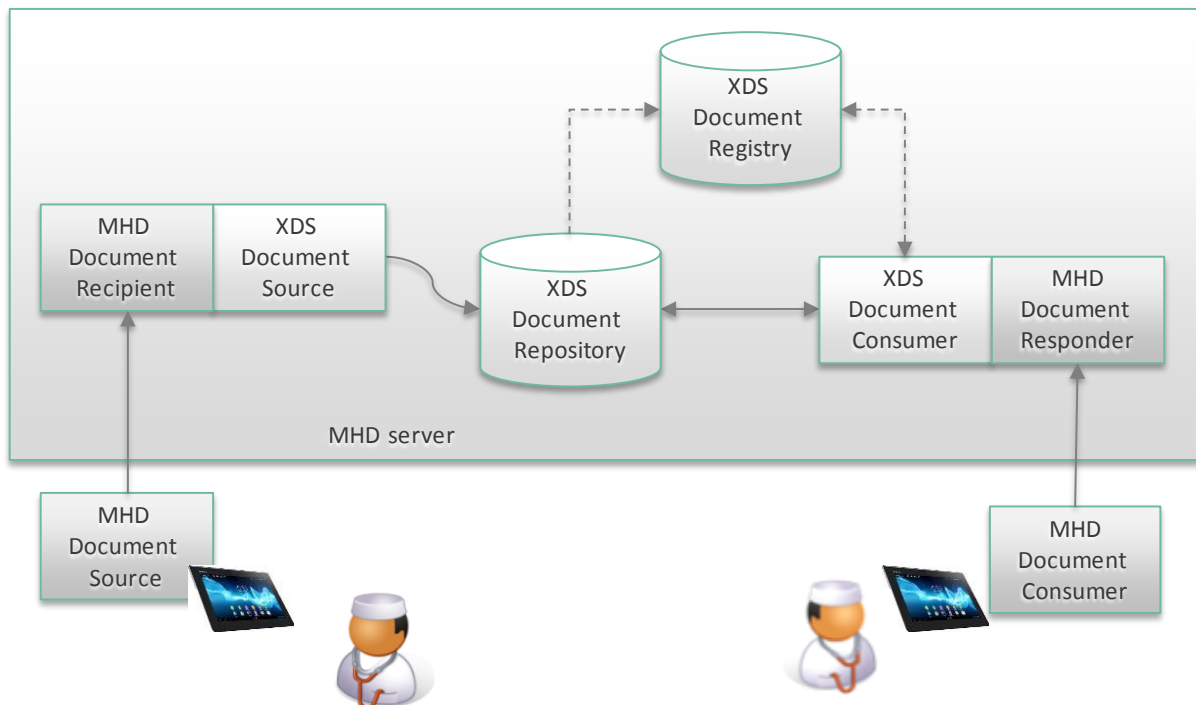
**Figuur 17 - De werking van een XDS omgeving**

In een XDS infrastructuur is er altijd maar één Document Registry. Alle andere systeemdelen kunnen in meervoud voorkomen. Opslag kan dus bijvoorbeeld op meerdere locaties en servers plaatsvinden, dit maakt het systeem ook zo flexibel. Document Consumers kunnen in verschillende systemen worden geïnstalleerd. XDS is content-agnostisch, dat wil zeggen dat er in principe elk type document mee kan worden opgeslagen, onafhankelijk van het functionele of technische type, het bestandsformaat, de inhoud van het document, of het doel. Deze oplossing kan dus zowel gebruikt worden voor CDA documenten als voor FHIR documenten.

Samenwerkende instellingen die afspraken maken over de uitwisseling van medische gegevens via een XDS-netwerk worden ook wel Affinity Domain genoemd. Het XDS profiel beschrijft vooral de technische afspraken die nodig zijn voor de XDS software, maar daarnaast moeten ook op andere interoperabiliteitsniveaus afspraken worden gemaakt. XDS maakt gebruik van SOAP gebaseerde webservices.

### 6.3.1 Oplossingen gebaseerd op IHE MHD

Het Mobile access to Health Data (MHD) profiel is oorspronkelijk in 2012 door IHE ontwikkeld als een RESTful API op een IHE document sharing omgeving zoals XDS, XCA of XDR. Het MHD profiel definieert een standaard interface, een Application Programming Interface (API), voor mobiele devices zodat het gebruik van mobiele applicaties meer consistent en herbruikbaar is. De transacties die voor de interface zijn gedefinieerd bouwen voort op de concepten van XDS, maar maken het eenvoudiger voor toegang door omgevingen met beperkte mogelijkheden zoals het geval is bij mobiele devices. Het MHD profiel vervangt niet XDS. MHD kan worden gebruikt om mobiele devices en andere systemen die beperkte mogelijkheden hebben toegang te verlenen tot een XDS Repository. In Figuur 18 is een mogelijke manier weergegeven om MHD te implementeren in een document sharing omgeving die op XDS gebaseerd kan, maar niet hoeft te zijn.



**Figuur 18 - IHE MHD en XDS**

Het MHD profiel definieert twee actoren (MHD Document Source en MHD Document Recipient) en een transactie om nieuwe documenten van een mobiel device aan te bieden aan een ontvangend systeem. Een tweede set van actoren (MHD Document Consumer en MHD Document Responder) en transacties worden gebruikt om een lijst van document met specifieke metadata op te vragen, en om een kopie van een specifiek document op te halen. Vanuit het perspectief van de MHD Document Source en de MHD Document Consumer vormen de andere MHD en XDS actoren een FHIR server die een FHIR dialect gebruikt dat wordt gespecificeerd door het MHD profiel.

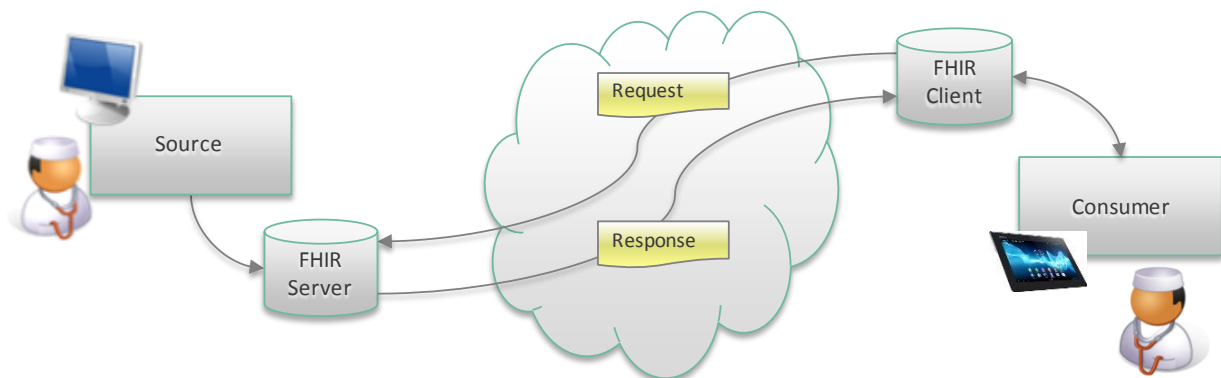
## 6.4 Client-server oplossingen gebaseerd op (FHIR) RESTful services

Client-server oplossingen zijn in principe ook point-to-point oplossingen met als basis de samenwerking tussen twee software applicaties, die zich op verschillende computers bevinden. Het is een asymmetrisch model:

- de server is permanent beschikbaar en is reactief
- een client is bij gelegenheid actief en neemt het initiatief tot communicatie met de server

Client-server oplossingen maken in het algemeen gebruik van webservices. Een specifieke variant daarvan zijn RESTful services waarvan ook FHIR gebruik maakt: zie <http://www.hl7.org/fhir/http.html>. RESTful services maken gebruik van het HTTP-protocol. Meer details betreffende FHIR RESTful services zijn te vinden in paragraaf 10.3.

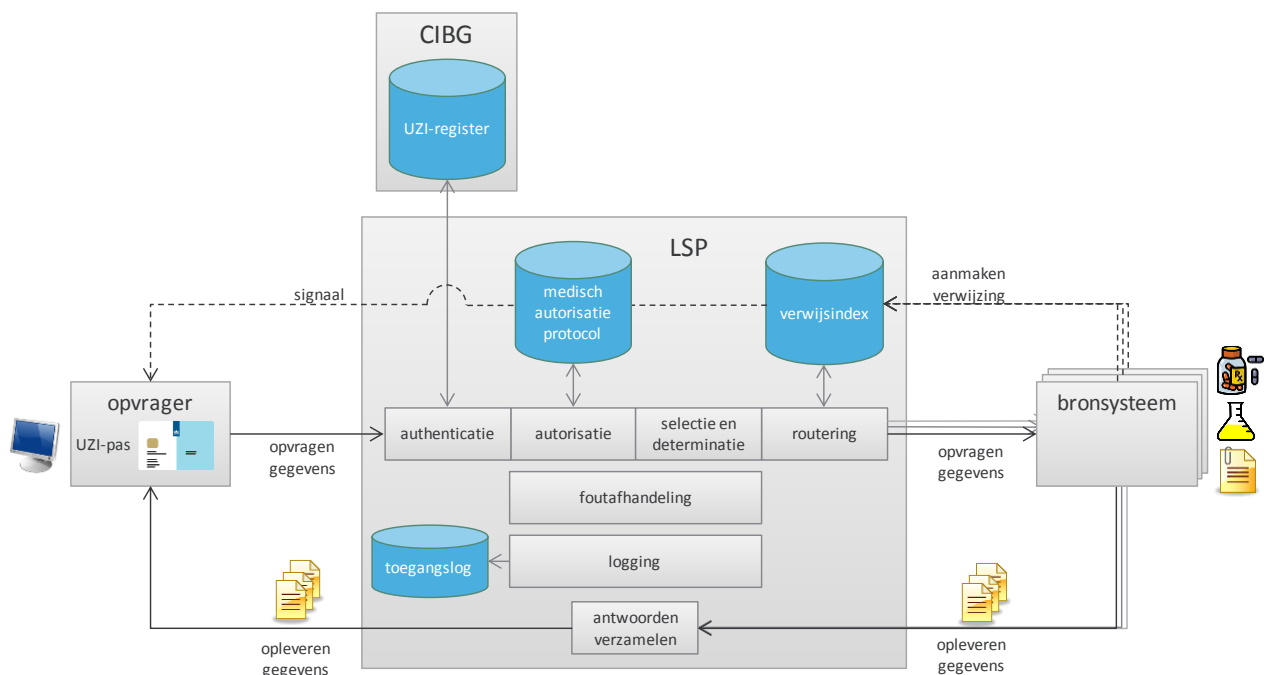
Een systeemvoorbeeld is weergegeven in Figuur 19. Een praktische implementatie kan bijvoorbeeld bestaan uit een FHIR client die onderdeel is van een persoonlijke gezondheidsomgeving (pgo) van een patiënt terwijl de FHIR server onderdeel vormt van het informatiesysteem van een ziekenhuis. De patiënt kan op die manier, bijvoorbeeld via een mobiel apparaat, toegang krijgen tot zijn/haar gegevens in het ziekenhuisinformatiesysteem.



**Figuur 19 – Client-server gebaseerde uitwisseling op basis van FHIR RESTful services**

## 6.5 AORTA (het LSP)

AORTA is de naam voor de landelijke infrastructuur die in eerste instantie vanuit het ministerie van VWS is opgezet en later in productie is genomen door de Vereniging van Zorgaanbieders voor Zorgcommunicatie (VZVZ), waarin o.a. huisartsen, huisartsenposten, apotheken en ziekenhuizen samenwerken om gegevens met elkaar uit te wisselen. De centrale component van AORTA is het Landelijk SchakelPunt (LSP), dat volledig operationeel is sinds eind 2014. Vrijwel alle apotheken en de meeste huisartsen(posten) en ziekenhuizen zijn op het LSP aangesloten en gebruiken het voor de uitwisseling van medicatiegegevens (verstrekkingen) of huisartswaarneemgegevens (professionele samenvatting).



**Figuur 20 - Het LSP voor PULL verkeer met signaal en opvraag functie**

Het LSP slaat zelf geen medische gegevens op, maar legt verwijzingen vast naar gegevens in aangesloten systemen. De uitwisseling van gegevens vindt plaats op basis van HL7v3 berichten. De aard van de 'payload' waarin de medische gegevens worden uitgewisseld is feitelijk vrij invulbaar, aangezien het LSP zelf niet naar de payload kijkt. In de release AORTA 8 wordt gebruik gemaakt van een CDA-compatible formaat, conform standaardtemplates die door Nictiz worden beheerd, maar in principe zou het LSP ook

bijv. PDF documenten of FHIR resources als payload kunnen gebruiken. De toegevoegde waarde ligt in de infrastructurele functies die het LSP biedt. Op hoofdlijnen betreft dit:

**Verwijsindex:** Elk bronsysteem dat gegevens van een bepaald type beschikbaar heeft voor een patiënt, meldt deze aan bij de verwijsindex op het LSP. Ook alle mutaties op onderliggende gegevens worden binnen 15 minuten gemeld. Op deze manier weet het LSP altijd waar relevante gegevens zijn opgeslagen, maar ook wanneer deze voor het laatst zijn bijgewerkt. De gegevens zelf blijven in het bronsysteem, totdat een vragend systeem een query stelt aan het LSP. Daarop bepaalt het LSP de relevante bronsystemen en stuurt hen een vraag voor de gewenste gegevens. De bronsystemen leveren deze op en het LSP verzamelt ze (zonder verdere bewerking) en stuurt ze terug als query response.

**Authenticatie:** Het LSP ondersteunt alleen beveiligde verbindingen op basis van een UZI-servercertificaat, dat is verstrekt aan aangesloten zorgaanbieders. Aangesloten applicaties die gebruik maken van zo'n certificaat moeten eerst een acceptatieproces doorlopen om aan te tonen dat ze voldoen aan alle technisch eisen (nog los van de kwalificatie voor de gestandaardiseerde medische payload). Daarnaast is het opvragen van gegevens via het LSP alleen mogelijk met gebruik van een persoonsgebonden UZI-pas, waarmee een token wordt gegenereerd bij elke bevraging. Deze UZI-pas kan zijn van een zorgverlener zelf of van een door de zorgverlener gemandateerde medewerker.

**Autorisatie:** Op basis van de UZI-rolcode van de verantwoordelijk zorgverlener en het autorisatieprotocol dat de medische beroepsgroepen met elkaar hebben afgesproken, bepaalt het LSP welke gegevens beschikbaar zijn voor inzage.

**Selectie- en determinatieservice (SDS):** Vanaf AORTA 8 bevat het LSP de intelligentie om een contextspecifieke vraag vanuit een systeem (bijv. de vraag om het 'diabetesdossier t.b.v. ketenzorg') automatisch om te zetten in een set queries voor generieke bouwstenen. Op basis van alle relevante autorisatiecriteria (zie hieronder) wordt dan bepaald welke set aan bouwstenen nodig is en welk filter er per bouwsteentype moet worden toegepast op de instantiaties (bijv. alleen uit een bepaalde periode of met een bepaalde bepalingscode). Groot voordeel is dat vragende systemen een zeer diverse en flexibel aanpasbare informatiebehoefte kunnen laten vervullen, terwijl de bronsystemen alleen een beperkte set stabiele bouwsteenqueries hoeft te ondersteunen. Dit principe leent zich goed voor implementatie met FHIR en het LSP zal op korte termijn gereed gemaakt worden om via een FHIR-interface benaderd te worden.

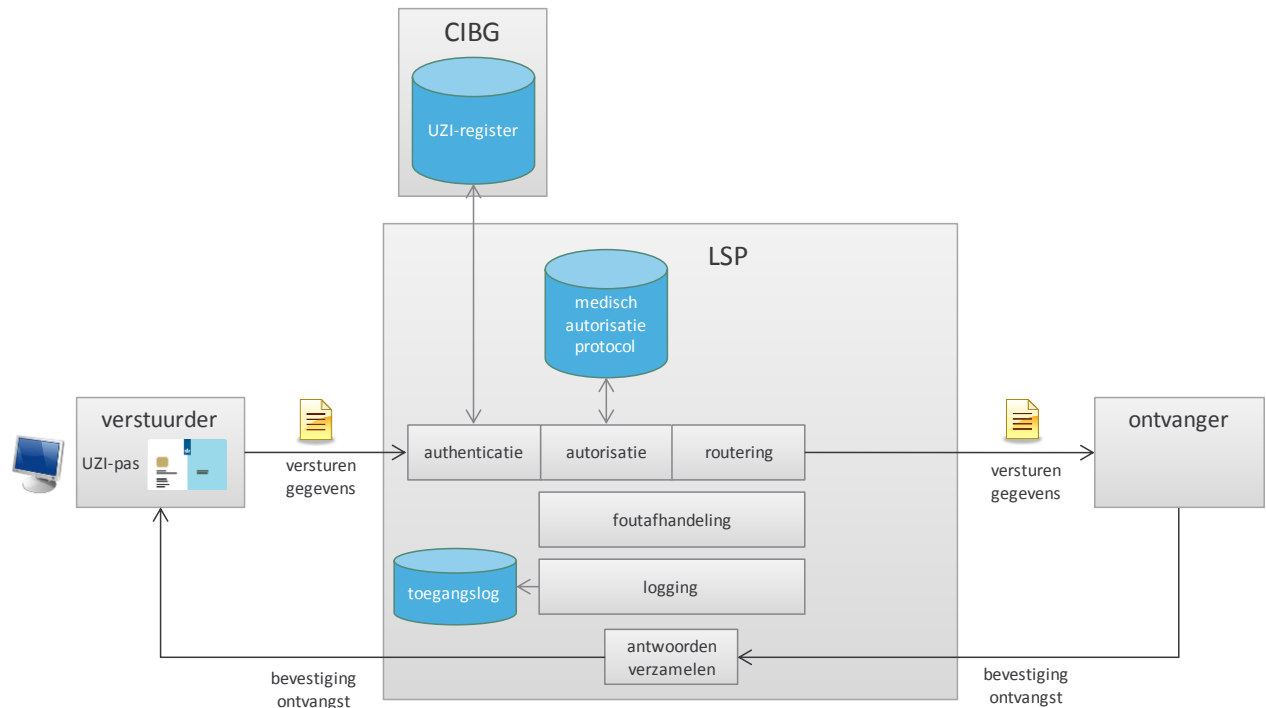
**Routing:** Naast bovenstaande functie voor het faciliteren van 'PULL'-verkeer, kan het LSP ook als router fungeren voor 'PUSH'-verkeer. Zie ook Figuur 21. Daarbij kan een verzendend systeem elke medische payload waarover afspraken zijn gemaakt adresseren aan een ontvangend systeem en door laten sturen via het LSP. Hierdoor zijn authenticatie, autorisatie en logging gegarandeerd en hoeft de verzender alleen het zogenaamde applicatie ID van de ontvanger te kennen.

**Logging:** Elke uitwisseling van medische gegevens via het LSP wordt gelogd, zodanig dat inzage zonder behandelrelatie tussen zorgverlener en patiënt achteraf traceerbaar is. De medische inhoud zelf wordt echter niet op het LSP gelogd.

**Signaalfunctie:** Geïnteresseerde systemen kunnen een abonnement nemen op de gegevens van de meest relevante patiënten (bijv. een apotheek voor diens patiënten met chronische medicatie). Iedere keer als de verwijsindex wordt bijgewerkt door een bronsysteem, wordt er dan een signaal verzonden aan het geabonneerde systeem. Het signaal zelf bevat geen medische gegevens, maar die kunnen vervolgens volgens de normale methode (zie 'verwijsindex') worden opgehaald. Hierdoor kan een geabonneerd systeem proactief up-to-date blijven voor deze patiënten.

Het LSP is momenteel landelijk operationeel voor medicatieverstrekkingen, huisartswaarneming en dossieruitwisseling in de JGZ, maar is sinds AORTA 8 snel configureerbaar (zowel via PUSH als PULL).

De volgende zorgtoepassingen die zullen worden uitgerold zijn Ketenzorg, Laboratoriumuitslagen en Medicatieproces (volledig medicatieoverzicht).



**Figuur 21 - Het LSP voor PUSH verkeer (versturen)**



## 7 Bijlage A - Lijst van afkortingen en begrippen

Afkorting	Betekenis	Toelichting
API	Application Programming Interface	<a href="#">Wikipedia</a>
ANY	Generiek Datatype	<a href="#">Datatypes</a>
BgZ	Basisgegevensset Zorg	<a href="#">BgZ</a>
BL	Boolean	<a href="#">Datatypes</a>
CD	Concept Descriptor	<a href="#">Datatypes</a>
CDA	Clinical Document Architecture	<a href="#">HL7 CDA standaard</a>
DB of db	Database	<a href="#">Wikipedia</a>
dbc	Diagnose Behandel Combinatie	<a href="#">Wikipedia</a>
DCM	Detailed Clinical Model	<a href="#">Wikipedia</a>
ecd	Elektronisch Cliënten Dossier	
ecg	Electrocardiogram	
ED	Encapsulated Data	<a href="#">Datatypes</a>
epd	Elektronisch Patiënten Dossier	
FHIR	Fast Healthcare Interoperability Resources	<a href="#">HL7 FHIR standaard</a>
FTP	File Transfer Protocol	<a href="#">Wikipedia</a>
his	Huisartseninformatiesysteem	
HL7	Health Level Seven	<a href="#">HL7 Nederland</a>
HTTP	Hypertext Transfer Protocol	<a href="#">Wikipedia</a>
IHE	Integrating the Healthcare Enterprise	<a href="#">IHE Nederland</a>
II	Instance Identifier	<a href="#">Datatypes</a>
INT	Integer Number	<a href="#">Datatypes</a>
JSON	JavaScript Object Notation	<a href="#">Wikipedia</a>
lis	Laboratorium informatiesysteem	
LROI	Landelijke Registratie Orthopedische Implantaten	<a href="#">LROI</a>
MHD	Mobile acces to Health Documents	<a href="#">IHE Wiki</a>
NICE	Nationale Intensive Care Evaluatie	<a href="#">NICE</a>
pgd	Persoonlijk Gezondheidsdossier	
pgo	Persoonlijke Gezondheidsomgeving	
PQ	Physical Quantity	<a href="#">Datatypes</a>
ST	String	<a href="#">Datatypes</a>
TS	Timestamp	<a href="#">Datatypes</a>

umc	Universitair Medisch Centrum	
VPN	Virtual Private Network	<a href="#">Wikipedia</a>
vvt	Verpleeg-, Verzorgingshuizen en Thuiszorg	
XDM	Cross-Enterprise Document Media Interchange	<a href="#">IHE Wiki</a>
XDR	Cross-Enterprise Document Reliable Intherchange	<a href="#">IHE Wiki</a>
XDS	Cross-Enterprise Document Sharing	<a href="#">IHE Wiki</a>
XML	EXtensible Markup Language	<a href="#">Wikipedia</a>
zib	Zorginformatiebouwsteen	
zis	Ziekenhuisinformatiesysteem	

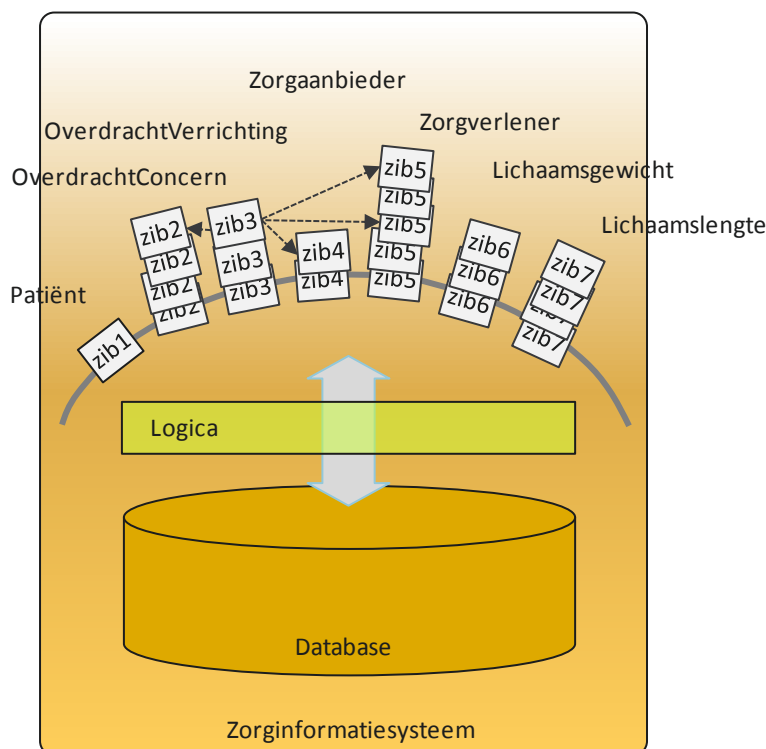
## 8 Bijlage B - Voorbeeld instantiatie gegevensset

Het voorbeeld van de gegevensset die in paragraaf 4.2 is gegeven wordt hier verder uitgewerkt. Voor de volledigheid zij nog vermeld dat in veel praktische gevallen een gegevensset ook gegevenselementen bevat waarvoor geen bouwstenen gedefinieerd zijn. Dat wordt toegelicht in de paragrafen 3.4 en 4.2. In dit voorbeeld wordt ervan uitgegaan dat de gegevensset alleen gegevenselementen bevat waarvoor zib's bestaan.

Voor een bepaalde use case is de volgende gegevensset A gedefinieerd:

- Zib Patiënt; één instantiatie voor patiënt X
- Zib OverdrachtVerrichting; alle instantiaties voor patiënt X van type orthopedische verrichtingen
- Zib Lichaamsgewicht voor patiënt X; één instantiatie, laatst gemeten gewicht
- Zib Lichaamslengte voor patiënt X, één instantiatie, laatst gemeten lengte

We gaan uit van de systeemopzet in onderstaande figuur.



**Figuur 22 - Voorbeeld systeemopzet**

Alle relevante gegevens zijn opgeslagen in de database van het epd; patiëntgegevens, gegevens over de uitgevoerde verrichtingen, meetgegevens etc. Die gegevens die volgens een eigen intern datamodel opgeslagen zijn in het systeem kunnen "gezien" worden via een view op de zib's. Het systeem heeft alle informatie om die view te genereren.

Op de "zib-laag" kunnenwe dan bijvoorbeeld de volgende view zien:

Zib1	Patiënt	Een instantiatie van de zib Patiënt voor patiënt X.
Zib2	OverdrachtConcern	Alle instantiaties van de zib OverdrachtConcern voor alle problemen inclusief diagnoses die in het systeem geregistreerd zijn voor patiënt X

Zib3	OverdrachtVerrichting	Alle instantiaties van de zib OverdrachtVerrichting voor alle verrichtingen die in het systeem geregistreerd zijn voor patiënt X
Zib4	Zorgaanbieder	Alle instantiaties van de zib Zorgaanbieder voor alle zorgaanbieders die in het systeem geregistreerd zijn waarbij patiënt X onder behandeling is
Zib5	Zorgverlener	Alle instantiaties van de zib Zorgverlener voor alle zorgverleners die in het systeem geregistreerd zijn waarbij patiënt X onder behandeling is
Zib6	Lichaamsgewicht	Alle instantiaties van de zib Lichaamsgewicht voor alle lichaamsgewichten die in het systeem geregistreerd zijn voor patiënt X
Zib7	Lichaamslengte	Alle instantiaties van de zib Lichaamslengte voor alle lichaamslengtes die in het systeem geregistreerd zijn voor patiënt X

De zib OverdrachtVerrichting kent een aantal verwijzingen naar andere zib's zoals:

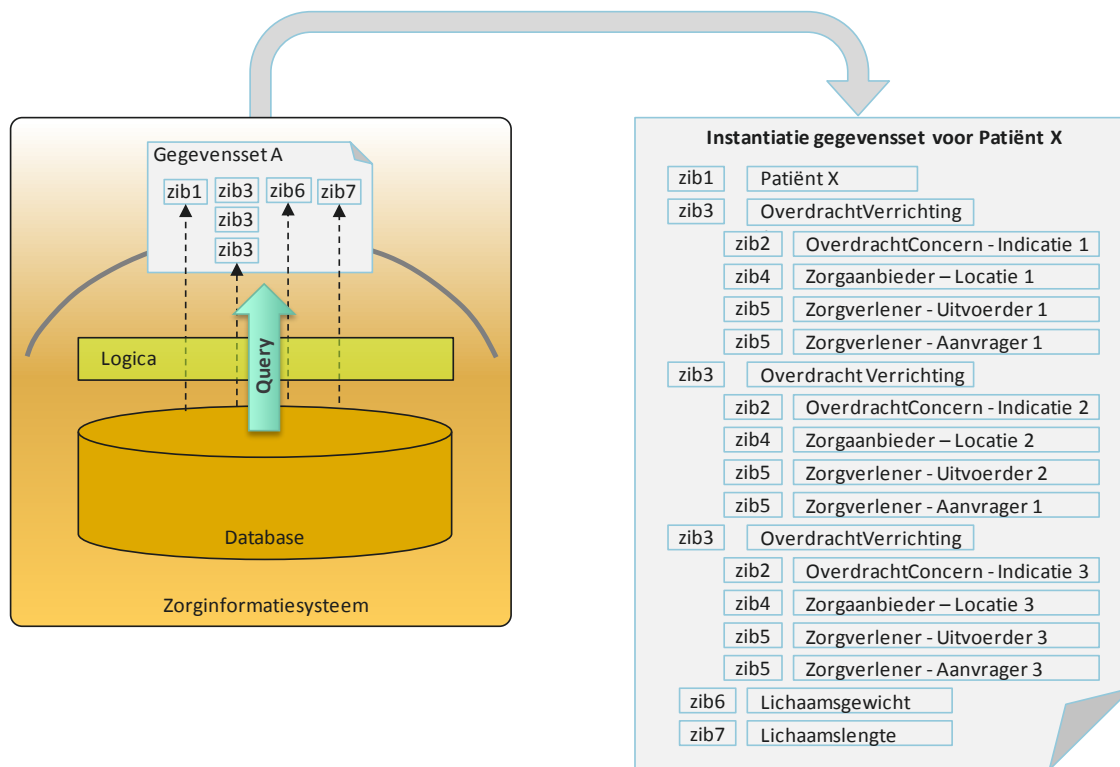
- Indicatie::Probleem[OverdrachtConcern] 0..\*
- Product::Product[MedischHulpmiddel] 0..\*
- Locatie::Zorgaanbieder[Zorgaanbieder] 1..1
- UitgevoerdDoor::Zorgverlener[Zorgverlener] 1..\*
- AangevraagdDoor::Zorgverlener[Zorgverlener] 0..\*

Dat betekent in het gegeven voorbeeld dat er een relatie is tussen een instantiatie van de zib OverdrachtVerrichting (zib3) en de instantiatie van

- de zib OverdrachtConcern (zib2) die de indicatie bevat die hoort bij de verrichting
- de zib Zorgaanbieder (zib4) die de zorgaanbieder bevat die hoort bij de locatie waar de verrichting plaatsvond
- de zib Zorgverlener (zib5) die de zorgverlener bevat die de verrichting heeft uitgevoerd
- de zib Zorgverlener (zib5) die de zorgverlener bevat die de verrichting heeft aangevraagd

Deze relaties zijn in Figuur 22 weergegeven met behulp van de stippellijnen. Hier geldt dat de relaties alleen bestaan mits afgesproken is dat de informatie tijdens het zorgproces is vastgelegd en dit ook daadwerkelijk is gebeurd (want voor al deze gegevenselementen geldt kardinaliteit 0..1) en mits het systeem deze relaties ondersteunt.

Op basis van de beschikbare gegevens is voor de casus van dit voorbeeld een gegevensset samen te stellen volgens de regels die aan het begin van deze bijlage beschreven zijn. Dat is weergegeven in Figuur 23.



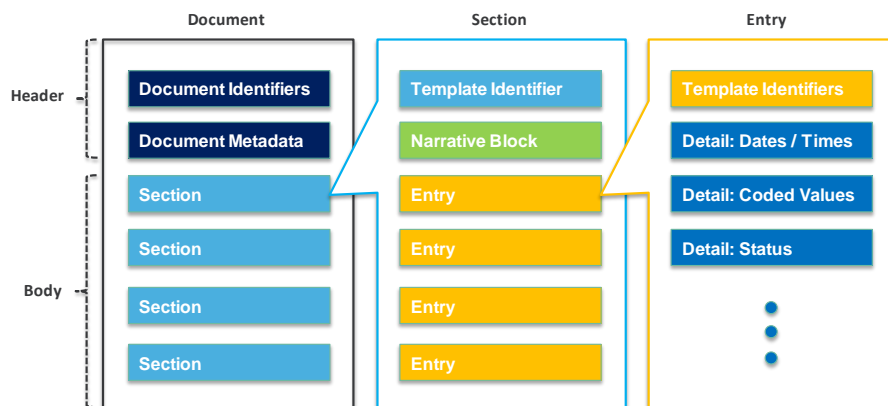
Figuur 23 - Voorbeeld instantiatie gegevensset A

## 9 Bijlage C - HL7 CDA

Bron: <https://www.linkedin.com/pulse/connecting-ios-your-emr-using-healthkit-cda-part-three-eric-whitley>

### 9.1 De structuur van een CDA document

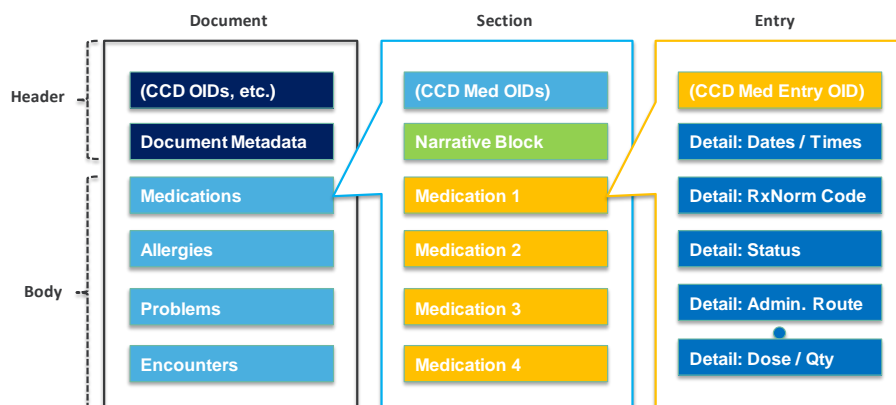
De HL7 Clinical Document Architecture (CDA) is een op XML gebaseerde document markup standaard die de structuur en semantiek van klinische documenten ten behoeve van uitwisseling specificeert. De structuur van een CDA document is weergegeven in Figuur 24. Een CDA document is opgebouwd uit een document header en een document body.



Figuur 24 - De structuur van een CDA document

In de header zijn gegevens over de patiënt en over het document zelf vastgelegd, sterk gestructureerd en ook de semantiek is er in vastgelegd. Daarnaast bevat de header informatie over betrokken personen en organisaties en episodes waarover gedocumenteerd wordt. De informatie in de header ondersteunt de uitwisseling van klinische documenten over organisatiegrenzen heen..

De echte klinische gegevens worden in de body vastgelegd die bestaat uit sections en entries. Elke section bevat in ieder geval een onderdeel leesbare (verhalende) tekst, wat een verplicht deel van CDA release 2 documenten is. Dit garandeert de interoperabiliteit tussen twee menselijke communicatiepartners. Een section kan worden gezien als een groep van een bepaald soort gegevens zoals "medicijnen". Een entry is een onderdeel van een section en bevat een clinical statement, dat is bv. informatie als "Enalaprilmaleaat 10 mg / hydrochloorthiazide 25 mg oraal tablet start 2009-09-25". Een section kan ook eisen of beperkingen definiëren rond het aantal entries in een section template. De afzonderlijke entries bevatten de kritische detailinformatie en verschillen per template. Al deze entries kunnen lineair of recursief hiërarchisch met elkaar verbonden zijn.

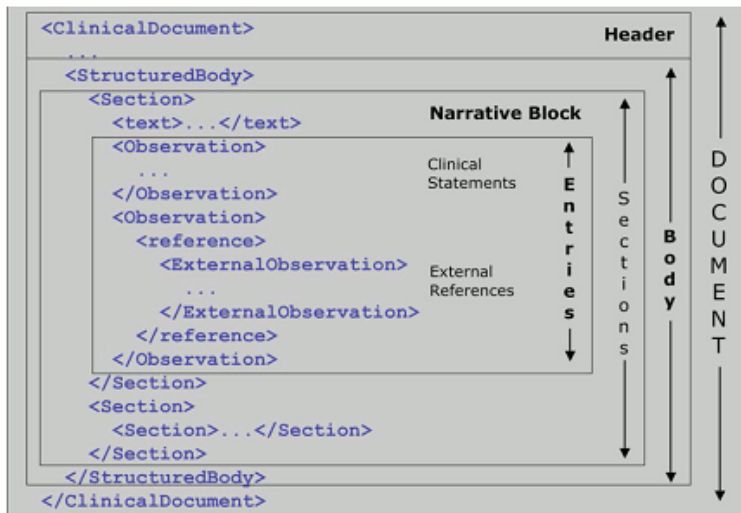


Figuur 25 - Voorbeeld CDA document

## 9.2 CDA templates

Bron: <http://healthstandards.com/blog/2012/03/22/consolidated-cda/>

De XML basisstructuur van een CDA document is weergegeven in Figuur 26.



Figuur 26 - CDA document – XML basisstructuur

De HL7 Consolidated CDA (C-CDA) is een implementatie handleiding voor CDA release 2.0. die een bibliotheek van herbruikbare CDA templates definieert en het gebruik ervan voorschrijft voor een aantal specifieke documenttypen.

Templates zijn van belang voor CDA vanwege de volgende redenen:

- Het leidt tot snellere ontwikkeling door het gebruik van herbruikbare bouwblokken
- Het leidt tot snellere implementatie omdat gebruikers eenmaal ontwikkelen, meervoudig gebruiken
- Omdat ze modulair zijn kunnen templates (zoals bloeddruk, gewicht) samengevoegd worden met andere templates in verschillende implementatie handleidingen.
- Templates maken incrementele ontwikkeling mogelijk doordat men kan beginnen met een simpele CDA en vervolgens templates kan toevoegen

Templates kunnen binnen een CDA-document gebruikt worden op meerdere niveaus:

1. Document-level templates, zoals CCD, kunnen worden gebruikt om een template te definiëren voor het gehele document.
2. Section-level templates, zoals allergieën of medicijnen, kunnen worden gebruikt om te bepalen welke specifieke informatie zal worden opgenomen in elke section.
3. Entry-level templates, zoals specifieke waarnemingen of verrichtingen, kunnen worden gebruikt om te bepalen hoe de informatie wordt gecodeerd binnen elke entry

Zorginformatiebouwstenen komen terug op entry-level niveau. Door de zib-modellen te mappen naar entry-level templates is het mogelijk om op een eenduidige en reproduceerbare manier gegevens die gebaseerd zijn op de bouwstenen onder te brengen in CDA documenten.

## 10 Bijlage D - HL7 FHIR

Bron: [http://www.ringholm.com/persist/FHIR\\_for\\_executives.pdf](http://www.ringholm.com/persist/FHIR_for_executives.pdf)

Zie ook: <http://www.hl7.org/fhir/>

Zie ook: <http://wiki.hl7.org/index.php?title=FHIR>

De belangrijkste bouwblokken van FHIR zijn Resources, References en Profiles.

### 10.1 FHIR resources

Resources zijn kleine, afzonderlijke, logische eenheden voor uitwisseling met een gedefinieerd gedrag en betekenis. Ze zijn de kleinste eenheid van transactie. Voorbeelden zijn Patient, Practitioner, Allergy Intolerance, Family History en Care Plan maar ook Encounter en PaymentNotice. De resources zijn onderverdeeld in de volgende categorieën (FHIR versie DSTU2): Clinical, Identification, Workflow, Infrastructure, Conformance, Financial.

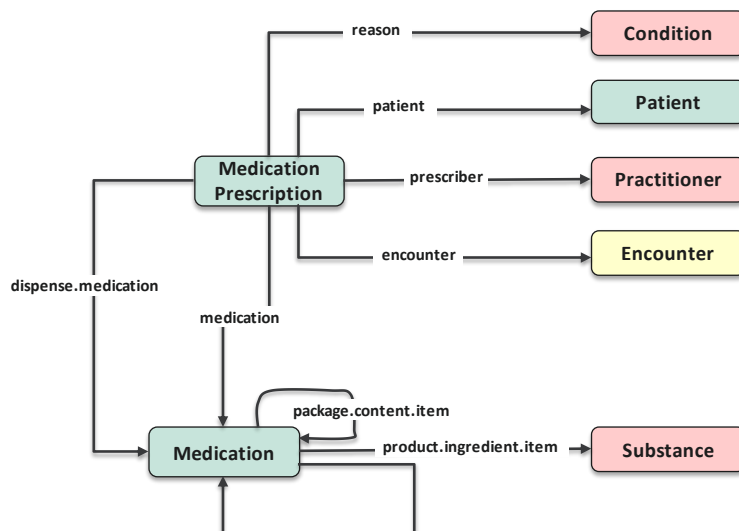
Een lijst met beschikbare resources is te vinden op: <http://www.hl7.org/fhir/resourcelist.html>.

Een resource bestaat uit drie delen:

1. Structured data – bedoeld om 80% van de vaak voorkomende use cases te ondersteunen.  
“We nemen alleen gegevens-elementen mee als we zijn ervan overtuigd zijn dat 80% van de implementaties er gebruik van zal maken”. Andere inhoud gaat naar extensions.
2. Narrative – tekstuele samenvatting van de inhoud van een resource
3. Extensions – bedoeld om niet vaak voorkomende use cases te ondersteunen.

### References

References zijn links van de ene resource naar een andere. Hiermee kunnen resources samengevoegd worden om een netwerk (of web) van gegevens te vormen die een gezondheidsdossier vormen (of in ieder geval een deel daarvan).



Figuur 27 – Voorbeelden van relaties tussen FHIR resources

### Profiles

Zie: <http://www.hl7.org/fhir/profiling.html>.

Resources hebben geen ingebouwde beperkingen op hoe ze worden gebruikt – dat gebeurt d.m.v. Profiles. Degenen die gegevens uitwisselen definiëren de specifieke manier waarop ze de resources en hun relaties willen gebruiken door gebruik te maken Profiles.



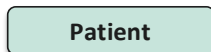
Profielen worden gedefinieerd door HL7, een land, een regio, een organisatie of een project. Bijvoorbeeld: Een bepaald ziekenhuis kan regels hebben voor de inhoud van een verwijzing voor pediatrische chirurgie aan dat ziekenhuis. Het kan informatie vereisen, zoals huisarts, namen van de ouders, de leeftijd van het kind, geslacht, etc. Als één van deze elementen ontbreekt, zou dat een niet geldige verwijzing zijn (voor dat type verwijzing naar dat ziekenhuis). Een profile definieert regels over de standaard resources en legt zaken vast als: welke elementen zijn nodig, welke terminologie codes worden gebruikt door een bepaald systeem en welke additionele extensies zijn gedefinieerd.

## 10.2 Uitwisselingsformaten die worden ondersteund door FHIR

FHIR maakt de uitwisseling mogelijk van één en dezelfde set van resources, ongeacht of die worden uitgewisseld als individuele resources, als onderdeel van berichten (messages) of documenten (documents). Het verschil tussen FHIR en andere message en document gebaseerde oplossingen is dat het de mogelijkheid toevoegt dat individuele resources worden opgehaald, geactualiseerd en opgevraagd. In bericht of document gebaseerde systemen zoals HL7v2 en CDA is men gedwongen om een verzameling van gegevens op te halen, ongeacht of men slechts een deel daarvan nodig heeft.

### Resource uitwisselingsformaat

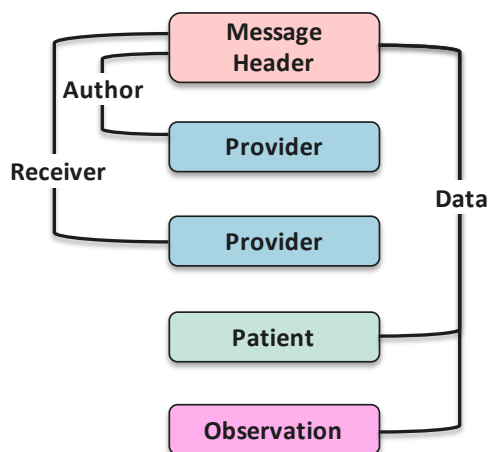
De enkelvoudige resource is de eenheid van uitwisseling voor dit formaat. Het uitwisselingsmechanisme dat wordt gebruikt is de FHIR RESTful benadering (par. 10.3).



**Figuur 28 – Voorbeeld resource, de eenheid van uitwisseling die RESTful omgeving gebruikt**

### Message uitwisselingsformaat

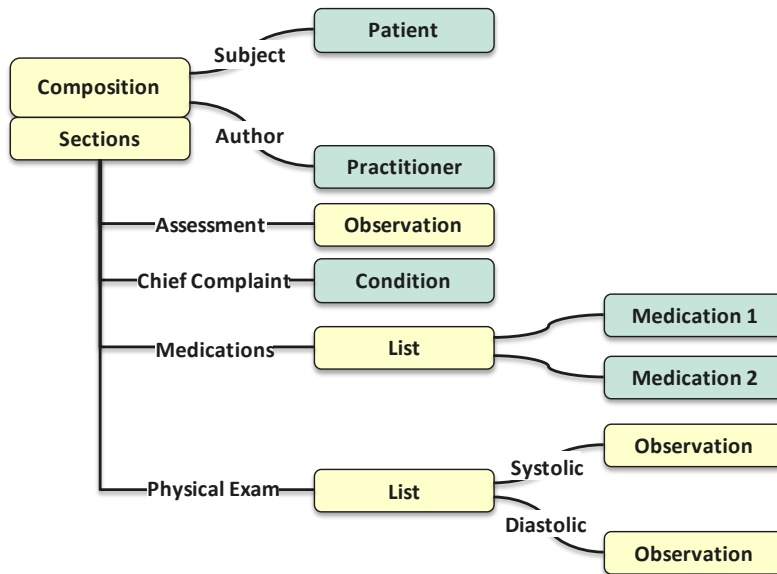
FHIR biedt ook de mogelijkheid om het message uitwisselingsformaat te gebruiken. In principe dekken FHIR messages dezelfde functionaliteit af als HL7V2. Een FHIR message bestaat uit een lijst van resources (vergelijkbaar met HL7V2 message segmenten) zoals is weergegeven in Figuur 29.



**Figuur 29 – Voorbeeld van een FHIR message met resources en references**

### Document uitwisselingsformaat

FHIR ondersteunt ook het document uitwisselingsformaat. FHIR documenten dekken in principe dezelfde functionaliteit af als CDA. Resources (vergelijkbaar met sections in CDA) bevatten zowel tekst als gegevens die door de computer verwerkt kunnen worden. Een FHIR document is een samenstel van section-level resources zoals is weergegeven in Figuur 30.



**Figuur 30 – Voorbeeld van een FHIR document met resources en references**

## 10.3 FHIR RESTful services

In tegenstelling tot de HL7 CDA standaard die geen uitwisselingsmechanisme bevat, bevat de HL7 FHIR standaard een uitwisselingsmechanisme dat FHIR RESTFUL services heet. Het ondersteunt FHIR resources, messages en documents en is speciaal toegesneden op de FHIR omgeving. Alhoewel FHIR messages en documenten ook met behulp van andere mechanismen uitgewisseld kunnen worden, is uitwisseling op resource niveau altijd gebaseerd op een RESTful omgeving.

RESTful<sup>11</sup> services zijn web services gebaseerd op het REST architectuurprincipe. Een web service is een verzameling van open protocollen en standaarden gebruikt voor het uitwisselen van gegevens tussen applicaties of systemen. In de REST-architectuur draait alles om “resources”. RESTful services zijn relatief licht, goed schaalbaar en onderhoudbaar en worden vaak gebruikt om API's<sup>12</sup> voor webgebaseerde applicaties te maken.

REST staat voor REpresentational State Transfer. REST is een op webstandaarden gebaseerde architectuur en maakt gebruik van het HTTP-protocol voor datacommunicatie. Het draait om resources waarbij elke component een resource is en een resource toegankelijk is via een gemeenschappelijke interface met HTTP standaardmethoden. In de REST architectuur biedt een REST server eenvoudig toegang tot resources en de REST-client krijgt toegang en presenteert de resources. Elke resource wordt geïdentificeerd door een URI<sup>13</sup>. REST maakt gebruik van verschillende representaties van een resource, zoals tekst, JSON en XML.

Een kenmerk van het HTTP-protocol en dus van de FHIR RESTful services is dat ze gebaseerd zijn op zogeheten “stateless” protocollen, d.w.z. dat elke request gezien wordt als een onafhankelijk transactie die niet gerelateerd is aan enig vorig request zodat de communicatie bestaat uit onafhankelijke paren van request en response. De server hoeft dus geen sessie informatie bij te houden als er meerdere requests zijn. Door gebruikmaking van een stateless protocol en standaardbewerkingen, streven REST gebaseerde

<sup>11</sup> Bron: [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)

<sup>12</sup> API: (Application programming interface): een verzameling definities op basis waarvan een computerprogramma kan communiceren met een ander programma of onderdeel (meestal in de vorm van bibliotheken).

<sup>13</sup> URI (Uniform resource identifier): [https://nl.wikipedia.org/wiki/Uniform\\_resource\\_identifier](https://nl.wikipedia.org/wiki/Uniform_resource_identifier)

systemen naar snelle prestaties, betrouwbaarheid, en het vermogen om te groeien, met hergebruikte componenten die kunnen worden beheerd en bijgewerkt zonder invloed te hebben op het systeem als geheel, ook als het actief is.

In HTTP is vastgelegd welke vragen (de Engelse term hiervoor is requests) een cliënt aan een server kan stellen en welke antwoorden (de Engelse term is responses) een webserver daarop kan teruggeven. Elke vraag bevat een URL die naar een resource verwijst. De volgende HTTP requestmethoden worden meestal gebruikt in een REST gebaseerde architectuur:

- GET Biedt een alleen-lezen toegang tot een resource
- PUT Om een nieuwe resource te maken
- DELETE Om een resource te verwijderen
- POST Om een bestaande resource bij te werken of nieuwe resource te creëren
- OPTIONS Om ondersteunde bewerkingen op een resource te initiëren

FHIR RESTful services vormen een specifieke variant waarbij een aantal zaken concreet gedefinieerd worden in de FHIR standaard: zie <http://www.hl7.org/fhir/http.html>. Zo kent de FHIR RESTful API requests in de vorm van interacties die zijn gebaseerd op de hierboven genoemde http “verbs” zoals GET, PUT, DELETE etc.

Interacties zijn bijvoorbeeld:

Interactie	Betekenis	Gebruikt HTTP verb
<a href="#">read</a>	Read the current state of the resource	GET
<a href="#">vread</a>	Read the state of a specific version of the resource	GET
<a href="#">update</a>	Update an existing resource by its id (or create it if it is new)	PUT
<a href="#">delete</a>	Delete a resource	DELETE
<a href="#">history</a>	Retrieve the update history for a particular resource	GET

Elke interactie is als volgt gedefinieerd:

```
VERB [base]/[type]/[id] {?_format=[mime-type]}".
```

Het verplichte deel daarvan is:

```
VERB [base]/[type]/[id]
```

Daarbij is:

VERB	De http “verb”
base	Service Root URL: The address where all of the resources defined by this interface are found. Vorm: http(s)://server{/path}
type	The name of a resource type (e.g. "Patient")
id	The <a href="#">Logical Id</a> of a resource

Een read interactie kan er als volgt uitzien:

```
GET https://fhirserver/resources/patient/347
```

## 11 Bijlage E – Mapping van zibs en CDA entries; een voorbeeld

Het principe van de mapping van zib's en CDA entries is beschreven in paragraaf 5.3.1 inclusief de regels die daarbij gehanteerd worden. In deze bijlage wordt een voorbeeld gegeven.

In Figuur 31 is een deel van een CDA document te zien met section level en entry level entries en daarin een verwijzing naar de zib Contact (via de DCM:Id van de zib Contact).

```
<?xml version="1.0" encoding="utf-8"?>
<ClinicalDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:hl7-
org:v3">
  <realmCode code="NL" />
  <typeId root="2.16.840.1.113883.1.3" extension="POCD_HD000040" />
  <templateId root="2.16.840.1.113883.2.4.6.10.100001" />
  <id assigningAuthorityName="Maastricht Universitair Medisch Centrum+" />
  <code codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" code="34133-9"
displayName="SUMMARIZATION OF EPISODE NOTE" />
  <title>SUMMARIZATION OF EPISODE NOTE</title>
  <effectiveTime value="20161227143330+0000" />

  <!-- NL-CM: 2.16.840.1.113883.2.4.3.11.60.40.3 -->
  ...
  <component>
    <structuredBody>
      <component>
        <section>
          <title>Encounters</title>
          <entry>
            <!-- ZIB Contact als Encounter -->
            <encounter moodCode="EVN" classCode="ENC">
              <templateId root="NL-CM:15.1.1" />
              <effectiveTime xsi:type="IVL_TS">
                <low value="201410281222+0500" /><!-- NL-CM:15.1.3 BeginDatumTijd -->
                <high value="201410311504+0500" /> <!-- NL-CM:15.1.4 EindDatumTijd -->
              </effectiveTime>

              <performer typeCode="PRF">
                <assignedEntity classCode="ASSIGNED">
                  <templateId root="NL-CM:15.1.7" />
                  ... reference ...
                </assignedEntity>
              </performer>

              <participant typeCode="LOC">
                <!-- Locatie::Zorgaanbieder -->
                <participantRole classCode="SDLOC">
                  <templateId root="NL-CM:15.1.8" />
                  ... reference ...
                </participantRole>
              </participant>

              <entryRelationship typeCode="SUBJ">
                <act moodCode="EVN" classCode="ACT">
                  <!-- Reden Contact -->
                  <templateId root="NL-CM:15.1.13" />
                </entryRelationship typeCode="SUBJ">

```



```
        inversionInd="false">
        <!-- Probleem::ProbleemNaam -->
        <observation classCode="OBS" moodCode="EVN">
            <templateId root="NL-CM:15.1.6" />
            .....
        </observation>
    </entryRelationship>
</act>
</entryRelationship>

</encounter>
</entry>
</section>

</component>
</structuredBody>
</component>
</ClinicalDocument>
```


Figuur 31 - CDA document (deel)voorbeeld met referentie naar de zib Contact

## 12 Bijlage F – Mapping van zibs en FHIR resources; een voorbeeld

In paragraaf 5.4.1 is het principe van de mapping van zib's en FHIR resources beschreven en de regels die daarbij gehanteerd worden. In deze bijlage wordt een voorbeeld gegeven.

### Voorbeeld: Profiel resource Patient gekoppeld aan zib Patiënt.

Op Simplifier.net staat het profiel van de patient resource gebaseerd op de zib Patiënt met de naam nl-core-patient (zie <https://www.simplifier.net/NL-BasicComponents/nl-core-patient/rendered>). Het overzicht daarvan is in de tabel hieronder weergegeven. Het profiel bevat verwijzingen naar andere subelementen zoals nl-core-humanname en nl-core-address. Het resource profiel bevat ook een aantal elementen die niet terugkomen in de zib Patiënt zoals Contact met een relatie naar de zib Contactpersoon en careProvider/GeneralPractitioner met een referentie naar de zib's Zorgaanbieder en Zorgverlener.

Name	Type	Description & Constraints ?
 Patient	Patient	Patient
 <a href="#">identifier</a>		PatientIdentificationNumber <b>Slice:</b> Unordered, Open At End, by system
 <a href="#">identifier (BSN)</a>		
 <a href="#">system</a>	uri	<b>Fixed Value:</b> <a href="http://fhir.nl/fhir/NamingSystem/bsn">http://fhir.nl/fhir/NamingSystem/bsn</a>
 <a href="#">value</a>		BSN (Burgerservicenummer (Dutch person identification number)) <b>Example:</b> 123456782
 <a href="#">name</a>	nl-core-humanname	NameInformation
 <a href="#">telecom</a>		ContactDetails <b>Slice:</b> Unordered, Open At End, by system
 <a href="#">telecom (TelephoneNumbers)</a>		TelephoneNumbers
 <a href="#">system</a>	code	<b>Fixed Value:</b> phone
 <a href="#">value (TelephoneNumber)</a>		TelephoneNumber
 <a href="#">use (NumberType)</a>		NumberType
 <a href="#">telecom (EmailAddresses)</a>		EmailAddresses
 <a href="#">system</a>	code	<b>Fixed Value:</b> email
 <a href="#">value (EmailAddress)</a>		EmailAddress
 <a href="#">use (EmailType)</a>		EmailType
 <a href="#">gender (Gender)</a>		Gender
 <a href="#">birthDate (DateOfBirth)</a>		Patient's date of birth. The date of birth is mandatory for a patient. A vague date (such as only the year) is permitted.
 <a href="#">deceased[x]</a>		DeathIndicator/DateOfDeath
 <a href="#">address</a>	nl-core-address	AddressInformation
 <a href="#">multipleBirth[x]</a>		MultipleBirthIndicator
 <a href="#">contact</a>		ContactPerson
 <a href="#">relationship</a>		Role or Relationship <b>Slice:</b> Unordered, Open At End, by coding.system

 <a href="#">relationship (Relationship)</a>		Relationship <b>Binding:</b> <a href="#">RelatieCodelijst</a> (extensible)
 <a href="#">relationship (Role)</a>		Role <b>Binding:</b> <a href="#">RolCodelijst</a> (extensible)
 <a href="#">name</a>	<a href="#">nl-core-humanname</a>	NameInformation
 <a href="#">telecom</a>		ContactDetails <b>Slice:</b> Unordered, Open At End, by system
 <a href="#">telecom</a> (TelephoneNumbers)		TelephoneNumbers
 <a href="#">system</a>	<a href="#">code</a>	<b>Fixed Value:</b> phone
 <a href="#">value</a> (TelephoneNumber)		TelephoneNumber
 <a href="#">use (NumberType)</a>		NumberType
 <a href="#">telecom (EmailAddresses)</a>		EmailAddresses
 <a href="#">system</a>	<a href="#">code</a>	<b>Fixed Value:</b> email
 <a href="#">value (EmailAddress)</a>		EmailAddress
 <a href="#">use (EmailType)</a>		EmailType
 <a href="#">address</a>	<a href="#">nl-core-address</a>	AddressInformation
 <a href="#">careProvider</a>	<a href="#">Reference(nl-core-organization   nl-core-practitioner)</a>	GeneralPractitioner

De relatie tussen alle gegevenselementen van de zib Patient en de overeenkomstige elementen van de resource Patient is vastgelegd in een resource van het type StructureDefinition (zie: <https://www.simplifier.net/NL-BasicComponents/nl-core-patient/xmlview>). Een deel van die StructureDefinition is weergegeven in Figuur 32. Daar is te zien hoe het FHIR resource-element "Patient.identifier" is gekoppeld aan het concept "Patiëntidentificatienummer" met Id NL-CM:0.1.7 van de zib Patient.

<pre> &lt;StructureDefinition&gt;    &lt;id value="nl-core-patient" /&gt;   &lt;meta&gt;     &lt;versionId value="15" /&gt;     &lt;lastUpdated value="2017-01-23T13:46:49.591+00:00" /&gt;   &lt;/meta&gt;   &lt;url value="http://fhir.nl/fhir/StructureDefinition/nl-core-patient" /&gt;   &lt;version value="1.0" /&gt;   &lt;name value="nl-core-patient" /&gt;   &lt;status value="draft" /&gt;   &lt;publisher value="HL7 Netherlands" /&gt;   &lt;description value="A Patient resource as defined ..... " /&gt;   &lt;requirements value="Patient..... Functional requirements taken from Clinical Building Block (Zorginformatiebouwsteen) Patient v2.0.1 (release 2015). Names and addresses are also in compliance with HL7 V3 Basic Components." /&gt;   &lt;copyright value="CC0" /&gt;   &lt;fhirVersion value="1.0.2" /&gt;   &lt;mapping&gt;     &lt;identity value="zib-patient" /&gt;     &lt;uri value="https://zibs.nl/wiki/Patient(NL)" /&gt;     &lt;name value="Zorginformatiebouwsteen Patient" /&gt;   &lt;/mapping&gt; </pre>	<p>Resource nl-core-patient</p> <p>Mapping naar zib Patient</p>
--	---



Figuur 32 - StructureDefinition (deel) voor de relatie tussen de zib en de resource Patient