

Conceptual Design Koppeltaal 1.3.x naar 2.0



Inhoudsopgave

[**Versiebeheer**](#_yyukhqk1p22y) **3**

[**Introductie**](#_azvz73yny2g3) **4**

[Versie definitie](#_sb1f3x23fp7w) 4

[**Einddoel: Koppeltaal 2.0**](#_afu7ido5k76u) **5**

[Toelichting verschillen](#_180s3u4c3kke) 5

[FHIR DSTU1 vs FHIR STU3](#_sn54iotowse3) 5

[Messaging vs REST API](#_qimdjk7k7q57) 5

[Versie op focal resource vs versie op alle resources](#_nt7xulajm49) 6

[Opslag per bericht vs opslag per resource](#_5geaa5qqfg0z) 6

[Geen CareTeam vs een CareTeam](#_gxvq5tbxdn5y) 6

[Activity vs task](#_nqrpa6pkzbvz) 7

[Domeinscheiding](#_kcfavioqx4u1) 7

[**Uitgangspunten op weg naar Koppeltaal 2.0**](#_n2uk4659z27g) **8**

[Algemene besluiten en uitgangspunten](#_40ui591aabuh) 8

[Software inhoudelijke besluiten](#_kmkl6t3ai1tf) 8

[**Compatibiliteit tussen 1.3.x en 2.0**](#_ld4kibnvqqc0) **9**

[Conversie](#_nupta4g5mxeq) 10

[Interactie tussen twee API’s](#_ah7fw6po12hk) 10

[**Benodigde stappen mogelijkheid van backwards compatibiliteit te creëren**](#_5udoyogy1ozh) **11**

[Beschrijving wijzigingen](#_mkrn2gqep65e) 11

[Versie per resource](#_9l61eifu06to) 11

[CareTeam](#_uwwavzmdpm3j) 12

[Belangrijke notes voor de CareTeam toevoeging](#_bm4pw26bw4jb) 13

[**Koppeltaal Server: Migraties**](#_w880d9z92sm1) **14**

[1.3.x naar 1.3.5](#_hll6955z9syf) 14

[Applicatieversie introduceren](#_n2gxs86f69nx) 14

[Tussenversie naar 2.0](#_wie10b6jeic1) 14

[Functioneel beheer overzetten naar nieuwe server](#_6wd7d6gazs5f) 14

[Berichten migreren naar resources](#_zb9062mwnl52) 14

# Versiebeheer

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Datum** | **Auteur** | **Wijzigingen** |
| 0.1 | 19-02-2018 | J. Horst | Set up document   * Einddoel (hfst. 1) * Uitgangspunten (hfst. 2) * Compatibiliteit (hfst.3) * Benodigde stappen (hfst. 4) |
| 0.2 | 20-02-2018 | J. Horst | Uitbreiding document   * Benodigde stappen (hfst. 4) uitwerking voorbeelden * Migratie (hfst. 5) * Samenvatting en conclusie (hfst. 6) |
| 0.3 | 23-02-2018 | J. Horst | * Versieoverzicht toegevoegd aan de introductie * Versies doorgevoerd * Impactuitbreiding toegevoegd voor 1.3.5 |
|  |  |  |  |

# 

# Introductie

In 2017 en begin 2018 is er binnen de Koppeltaal community veel gesproken en geschreven over de komende versie(s) van Koppeltaal Server.

Koppeltaal 2.0 met een nieuwe API op basis van FHIR STU3 is als het voorlopige einddoel gedefinieerd, maar de weg er naar toe was onduidelijk. Ondertussen is er een versie 1.3.5 en 1.4 genoemd om zo de stap en backwards compatibiliteit te kunnen garanderen.

Dit document is geschreven om duidelijk te maken en samen te vatten welke richting gekozen gaat worden. Hieraan vooraf is een Koppeltaal community meeting geweest met technische insteek. Ook is er een IT Deelnemersraad geweest met een functionelere insteek zodat de partijen van elkaar weten welke richting Koppeltaal op gaat en welke richting van elkaar verwacht wordt.

Om de lezer van dit document een helder inzicht te geven in de ontstane versies is het document opgebouwd vanuit het einddoel.

* Het eerste hoofdstuk bevat daarom een overzicht van Koppeltaal 2.0 en een beschrijving van de verschillen tussen 2.0 en 1.3.x.
* Het tweede hoofdstuk vat samen welke besluiten er zijn genomen door de Koppeltaal Community waaruit volgt welke versies nodig zijn om de eindsituatie te bereiken.
* Het derde hoofdstuk beschrijft het compatibiliteitsvraagstuk tussen 2.0 en 1.3.x. en het vierde hoofdstuk gaat dan in op de versie die er voor moet zorgen dat er compatibiliteit kan zijn en wat er voor nodig is om dit te bereiken.
* Als laatste wordt de migratie beschreven die nodig is om tot de 2.0 situatie te komen.

## Versie definitie

Onderstaande tabel geeft weer welke versies in het document gebruikt worden.

|  |  |  |
| --- | --- | --- |
| **Versienummer** | **Versie uitleg** | **Opmerking** |
| 1.3.x | Huidige codebase |  |
| 1.3.x-n | Gebruik voor patches | Optioneel |
| 1.3.4 | Ruimte voor tussenrelease | Optioneel |
| 1.3.5 | STU1, compatible met STU3 |  |
| ... |  | Optioneel. Ruimte voor extra releases tussen 1.3.5 en 2.0 |
| 2.0 | STU3, compatible met STU1 | Nieuwe server |

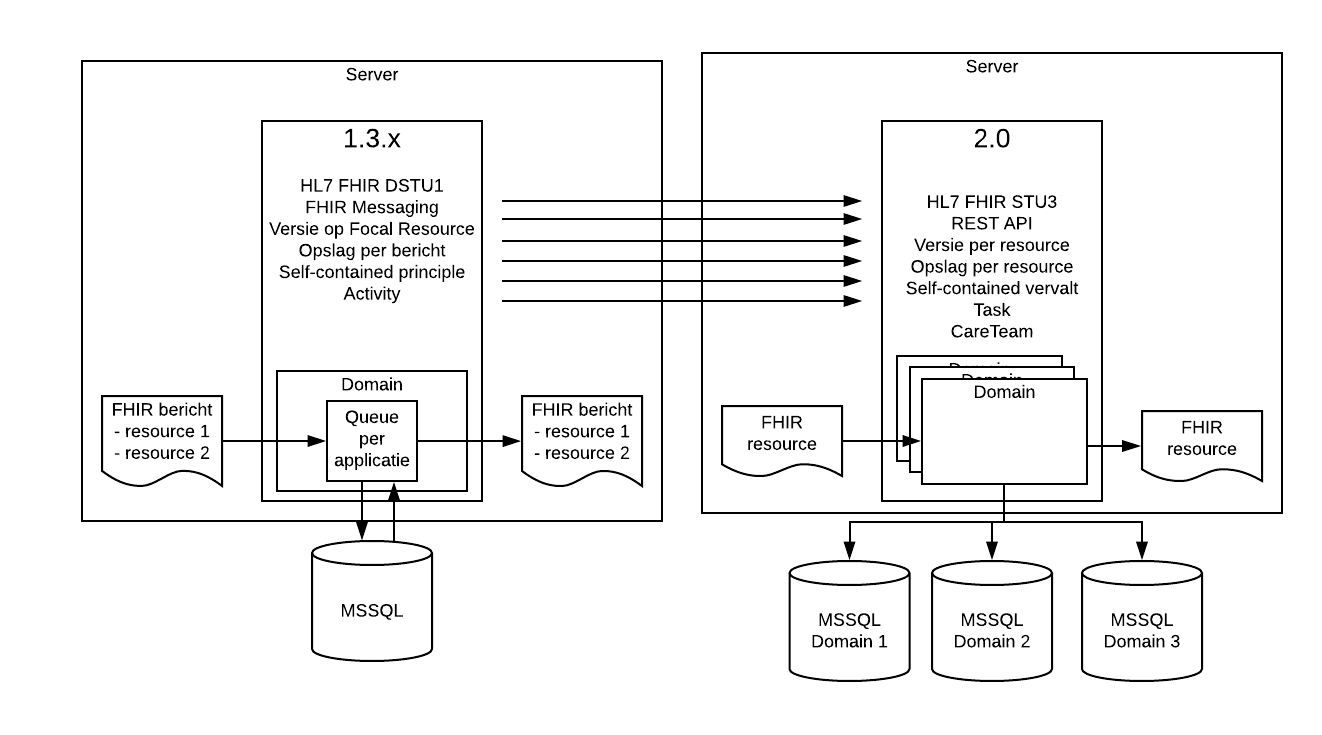
# Einddoel: Koppeltaal 2.0

Om de weg naar Koppeltaal 2.0 te schetsen is van belang dat eerst duidelijk beschreven wat Koppeltaal 2.0 is.

Hieronder staat een figuur die duidelijk maakt wat de belangrijke verschillen zijn tussen de huidige situatie en Koppeltaal 2.0

## Toelichting verschillen

### FHIR DSTU1 vs FHIR STU3

In de huidige Koppeltaal versie is de uitwisselingsstandaard gebaseerd op de eerste HL7 FHIR versie, de Draft Standard For Trial Use 1(DSTU1). Deze versie is in 2014 gereleased en is met één tussenversie gegroeid naar een Standard For Trial Use 3 (STU3) (dus geen draft meer). In deze versie zijn diverse resources volwassener geworden en nieuwe resources toegevoegd. Daarnaast is FHIR STU3 in Nederland gebruikt voor het vertalen van de zorginformatiebouwstenen.

### Messaging vs REST API

Koppeltaal is gestart met het FHIR messaging concept. Om dit te realiseren zijn er diverse berichten gedefinieerd die de taal van Koppeltaal bepalen. Deze berichten bevatten de resources die nodig zijn om dit bericht te verwerken.

Een voorbeeld van het bericht is het CreateOrUpdateCarePlan bericht. Dit bericht bevat een CarePlan met activities, een Patient en één of meerdere Participants die een rol hebben in het CarePlan. Een participant wordt meestal ingevuld met een Practitioner. Al deze resources zijn in het bericht (Bundle) toegevoegd en worden middels het bericht uitgewisseld.

De reden om dit concept te wijzigen ligt voornamelijk in de flexibiliteit. Als een activiteit of zorgteam wijzigt resulteert dat in een update van alle CarePlans die ooit uitgestuurd zijn. De oplossing ligt dus in het gescheiden opsturen van resources. Dit kan nog steeds via FHIR Messaging, maar ligt niet voor de hand, omdat losse resources veel sneller en makkelijker via een REST API uitgewisseld kunnen worden.

### Versie op focal resource vs versie op alle resources

Zoals in het plaatje hierboven weergegeven en beschreven is worden in de 1.3.x situatie de resources gezamenlijk in een bericht uitgewisseld. De resources worden ook binnen het bericht opgeslagen op de server. Hierom is destijds gekozen voor versionering per bericht in plaats van versionering per resources.

Een voorbeeld is het hierboven beschreven bericht, de CreateOrUpdateCarePlan. Alleen het CarePlan (focal resource) binnen het bericht heeft een versie.

In de 2.0 situatie worden resources los opgeslagen en daarom is een versie per resource noodzakelijk. Daarnaast is het sowieso ook gewenst om elke resource te versioneren omdat alleen dan gegarandeerd kan worden dat altijd de juiste versie van een resource in de omloop is.

### Opslag per bericht vs opslag per resource

In Koppeltaal Server 1.3.x worden inkomende berichten alleen als volledig bericht opgeslagen. Er is geen noodzaak (geweest) om dit per resource op te slaan. In Koppeltaal 2.0 is dit een basisvereiste omdat we met de REST API resources los uitwisselen in plaats van via een bericht.

### Geen CareTeam vs een CareTeam

Al eerder is benoemd dat bij het wijzigen van een zorgteam in de situatie van Koppeltaal 1.3.x een update nodig is van alle CarePlans die beïnvloed worden door deze wijziging. Er zijn zelfs applicaties waar het zorgteam uit de gehele afdeling bestaat omdat er in de applicatie zelf geen mogelijkheid is om hiervoor een striktere definitie vast te leggen. Dit zorgt berichten in Koppeltaal van een paar MB, iets wat onder andere performanceproblemen oplevert.

Om dit op te lossen wordt in 2.0 een resource CareTeam geïntroduceerd. Deze resource is in DSTU1 nog niet bekend, maar is toegevoegd in STU3. Een CareTeam bevat de lijst met participanten van het zorgteam. Vanuit een CarePlan kan verwezen worden naar het bijbehorende CareTeam(s).

### Activity vs task

Een CarePlan kan diverse onderdelen bevatten die een deelnemer (vaak de patiënt) moet uitvoeren in de vorm van een taak. Dit is in DSTU1 onderdeel van het CarePlan onder de naam ‘activity’ en is geen losse resource, maar is ingebouwd in het CarePlan.

Het nadeel van dit concept is dat bij een update van een taak een heel CarePlan verstuurd moet worden. Een extra bericht in de berichtenset van Koppeltaal 1.x heeft er voor gezorgd dat los status updates op een taak gestuurd kunnen worden.

Om hier flexibeler in te zijn wordt in Koppeltaal 2.0 de activity in het CarePlan leeggelaten en wordt de resource ‘Task’ gebruikt. Dit is een losse resource die apart verstuurd kan worden en veel lijkt op de Activity.

### Domeinscheiding

Iets wat niet in de lijst staat in het figuur aan het begin van het hoofdstuk is concept domeinscheiding. Wel is in het figuur duidelijk dat dit toegepast gaat worden. Dit houdt kort beschreven in dat de resources van verschillende domeinen in fysiek gescheiden tabellen opgeslagen worden. Dit concept wordt in een los project uitgewerkt en wordt hier verder niet meer genoemd.

# Uitgangspunten op weg naar Koppeltaal 2.0

Nadat de technische uitgangspunten van Koppeltaal 2.0 helder gemaakt waren is er door de Koppeltaal architecten en ontwikkelaars gekeken naar de tussenstappen om te komen van de huidige situatie naar de 2.0 situatie. Eerst werd de conclusie getrokken dat er maar één mogelijkheid was, namelijk dat alle partijen in één keer over zouden stappen naar 2.0. Dit werd door de community afgewezen vanwege de financiële consequenties hiervan voor de aangesloten partijen.

Terug op de tekentafel is alles nog een keer in kaart gebracht en het resultaat hiervan is opnieuw aan de community voorgelegd en ditmaal goedgekeurd.

Hieronder volgt een opsomming van uitgangspunten/besluiten die genomen zijn ten aanzien van de migratie naar Koppeltaal 2.0.

## Algemene besluiten en uitgangspunten

* 1.3.x blijft nog 3 jaar bruikbaar
* 2.0 moet tegelijkertijd bruikbaar zijn voor nieuw aan te sluiten partijen
* 2.0 moet met minimale inspanning van aangesloten partijen kunnen communiceren met 1.3.x
* Het gebruik van STU3 levert geen scopeverbreding op.
  + Alle verplichte velden worden indien mogelijk gevuld met bestaande velden die te mappen zijn uit DSTU1 velden
  + Alle andere verplichte velden worden met default waarden gevuld
  + Alle niet verplichte velden worden indien mogelijk gevuld met bestaande velden die te mappen zijn uit DSTU1 velden
  + Alle overige niet verplichte velden worden niet gebruikt en worden daarom niet geïmplementeerd in Koppeltaal 2.0.
  + Bovenstaande geldt ook voor valuesets. Indien mogelijk wordt er een mapping gedaan, waar dit niet mogelijk is vervallen de nieuwe STU3 waarden.
* Het gebruik van velden op resources en de interpretatie ervan moet gelijk getrokken worden over alle applicaties heen.  
  *Dit punt wordt in het document verder nergens toegelicht of uitgewerkt. Koppeltaal zal de lead nemen hierin en zal vragen aan de aangesloten partijen hoe zij de huidige resources op veldniveau interpreteren en in de software gebruiken.*

## Software inhoudelijke besluiten

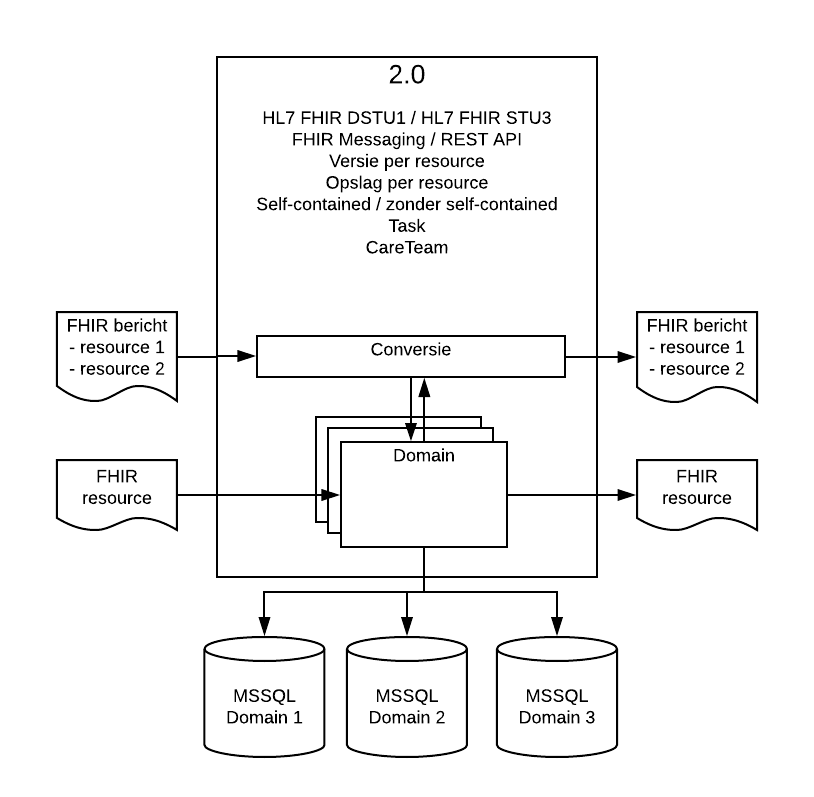
* Referenties van een resource naar een andere resource worden gelegd zonder versie
* Migratie en conversie van berichten is een taak van Koppeltaal. De aangesloten partijen moeten hier zo weinig mogelijk werk aan hebben.

# Compatibiliteit tussen 1.3.x en 2.0

In hoofdstuk 2 staat beschreven dat 1.3.x gebruikers gebruik moeten kunnen maken van aangesloten partijen die 2.0 compatible zijn. De oplossing daarvan ligt in een Koppeltaal Server die én STU3/REST API/etc. ondersteunt én DSTU1/FHIR Messaging ondersteunt.

In de documenten die hiervoor gedeeld zijn is deze versie 1.4 genoemd, maar daar wordt van afgezien omdat er een reële kans is dat er voor deze versie al eerdere versies aan het nummer 1.4 toe zijn. Daarnaast is Koppeltaal 2.0 met backwards compatibiliteit nog steeds één versie, alleen heeft deze twee API versies.

Het plaatje wat ontstaan is met deze versie is dus een samenvoeging van het figuur aan het begin van hoofdstuk 1:



## Conversie

Het belangrijkste in bovenstaand figuur is dat er een module ‘Conversie’ is toegevoegd. De verantwoordelijkheden van deze module zijn

* Converteren van DSTU1 bericht naar STU3 resources
  + Inclusief default vullen van verplicht velden  
    BELANGRIJK: Dit was eerder genoemd als onderdeel van de adapters, maar dat is hiermee vervallen. Koppeltaal Server neemt dit voor zijn rekening.
* Bij ophalen van een DSTU1 bericht op basis van een query (bijv. GetNextNewAndClaim) een bericht samenstellen op basis van losse resources

## Interactie tussen twee API’s

Naast de conversie moet een trigger geïmplementeerd worden tussen de nieuwe en oude API

* Een inkomend STU3 resource moet, indien binnen het domein nodig, een bericht in de queue klaarzetten voor de DSTU1 applicaties (zodat ook dat een push notificatie triggert)

# Benodigde stappen mogelijkheid van backwards compatibiliteit te creëren

In het vorige hoofdstuk is geschetst wat er nodig is in de Koppeltaal 2.0 server om te komen tot een versie die backwards compatibel is met 1.3.x. Maar niet alles wat nodig is, is hier genoemd omdat er nog gaten zijn die opgevuld moeten worden in een eerdere versie.

Deze versie is in het introductie 1.3.5 genoemd.

De inhoud van deze versie is bedoeld om een minimale functionele scopeverbreding te doen die het mogelijk maakt voor de conversietool om berichten te converteren tussen 1.3.x en 2.0.

## Beschrijving wijzigingen

### Versie per resource

Om berichten vanuit STU1 te converteren naar STU3 resources zullen alle resources binnen een bericht (niet alleen de focalresource) een versie moeten hebben. Op dit moment ligt de versie vast in de resourceID door deze erachter te plakken.

Vanaf de tussenversie zal Koppeltaal Server iedere resource voorzien van een versie. Koppeltaal Server zorgt er voor dat 1.3.x applicaties geen versie binnen krijgen zodat verwijzingen/references binnen de bundle blijven werken.

*De impact voor de applicaties is*

* *Op 1.3.5 het resolven van references rekening moet houden met een versie (die onderdeel van de resourceid).*
* *Door de Koppeltaal Server de versie gecheckt als de applicatie zegt 1.3.5 te ondersteunen.*
* *Een applicatie moet goed checken op welke manier resources geupdate worden. Een Patient kan vanaf 1.3.5 via meerdere berichten een nieuwere versie krijgen. Dit moet elke partij dus wel verwerken  
  Bijvoorbeeld: Een update van een Patient die binnenkomt via een CreateOrUpdateCarePlan heeft invloed op de versie van die Patient. Als een applicatie dus bijv. wel CreateOrUpdatePatient geimplementeerd heeft terwijl deze applicatie niet de CreateOrUpdateCarePlan verwerkt, kan het zijn dat het onmogelijk wordt om CreateOrUpdatePatients te gaan sturen. Immers, de partij mist soms nieuwe versies.*

Het volgende voorbeeld illustreert de impact van de meekomende versie:

*Applicatie 1 :gebruikt versie 1.3.x (DSTU1)*

*Applicatie 2 : gebruikt versie tussen 1.3.5 (DSTU1)*

* *Applicatie 1* stuurt een CreateOrUpdateCarePlan. Hierin staat een o.a. de Patient resource zonder versie.
* Koppeltaal Server bepaalt de versie.
* *Applicatie 2* ontvangt het CreateOrUpdateCarePlan met daarin de Patient resource met een versie
* *Applicatie 1* stuurt nogmaals een CreateOrUpdateCarePlan met daarin dezelfde Patient met nieuwe informatie over de patient
* Koppeltaal Server update de patient versie
* *Applicatie 2* verwerkt deze CreateOrUpdateCarePlan met de nieuwe Patient versie
* *Applicatie 2* update de voorgenoemde Patient resource in een CreateOrUpdatePatient bericht.   
  [[1]](#footnote-1)Belangrijk: Nu moet *Applicatie 2* een resource version meesturen en deze wordt ook gecheckt tegen de Koppeltaal Server Patient resource version. Als deze verouderd is, zal de update niet uitgevoerd kunnen worden.

### CareTeam

Een CareTeam is een nieuwe resource in Koppeltaal 2.0 die geïntroduceerd zal moeten worden in 1.3.5 om de conversie van de berichten te kunnen doen tussen 2.0 en 1.3.5.

Hieronder volgen de uitgangspunten van de CareTeam implementatie.

* Op elke Participant wordt een referentie naar de CareTeam resource (op basis van Other resource) geïntroduceerd.
* Alle CareTeam(s) waarnaar wordt verwezen vanuit de participants moeten worden opgenomen in het bericht.
* Koppeltaal Server checkt of een applicatie die zegt 1.3.5 te ondersteunen CareTeams verstuurt binnen een CarePlan.
* Per participant wordt een extra property toegevoegd, namelijk een reference naar het CareTeam

<participant>

<role>

<coding>

<system value="http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlanParticipantRole" />

<code value="Requester" />

<display value="Requester" />

</coding>

</role>

<member>

<reference value="https://demo.koppeltaal.nl/Caregiver/1" />

</member>

**<careteam>**

**<reference value="https://demo.koppeltaal.nl/CareTeam/1" />**

**</careteam>**

</participant>

* Onderin het bericht worden alle unieke CareTeams als Other resource opgenomen.
* Een 1.3.5 applicatie ontvangt een CarePlan bericht zonder CareTeam als een oudere versie een bericht stuurt. Koppeltaal verwacht dat 1.3.5 applicaties hiermee om kunnen gaan.
* Een 1.3.2. applicatie ontvangt een bericht zonder aanpassingen, dus met CareTeam.

#### Belangrijke notes voor de CareTeam toevoeging

* De toevoeging van dit CareTeam kan impact hebben op adapters, dat is door het Koppeltaal team niet te voorzien.
* Er kunnen situaties zijn waarin het geen functionele impact heeft, de adapters zouden dit ook zelf kunnen oplossen door voor het versturen een default CareTeam toe te voegen. Let wel, dit wordt wel gemigreerd naar de 2.0 situatie en daarom is wenselijk om de implementatie nu al zo dicht mogelijk bij die situatie te doen.

# 

# Koppeltaal Server: Migraties

Tot dusver is in vorige hoofdstukken gefocust op functionaliteit van de Koppeltaal Server en het gevolg en de impact voor aangesloten partijen.

Er zijn twee Koppeltaal versies besproken en er is een huidige versie. Van Koppeltaal 1.3.x naar 2.0 gaat dus in twee stappen. In dit hoofdstuk is beschreven welke migraties er nodig zijn op de Koppeltaal Server om te kunnen bieden wat hiervoor beschreven is.

## 1.3.x naar 1.3.5

De migratie naar de eerste tussenversie bestaat voorzover op moment van schrijven in te schatten is uit één stap die hieronder wordt beschreven.

### Applicatieversie introduceren

Elke applicatie krijgt een versienummer van Koppeltaal Server waar de betreffende applicatie compatibel mee is. Het is dus een Koppeltaal Server versie.

Bij de update naar tussenversie wordt dat voor iedereen dezelfde versie, namelijk 1.3.x waarbij de x nog onbekend is.

Dit versienummer wordt geupdate op het moment dat een applicatie aangeeft te willen werken in de tussenversie. Op dat moment kan Koppeltaal de CareTeam- en versiechecks uitvoeren.

## Tussenversie naar 2.0

Die migratie van tussenversie naar 2.0 bestaat uit de volgende stappen.

### Functioneel beheer overzetten naar nieuwe server

Alle domeinen, applicaties, gebruikers en bijbehorende configuratie zal overgezet moeten worden naar de nieuwe server.

### Berichten migreren naar resources

Als de eerste stap afgerond is kunnen alle berichten per domein overgezet worden. Deze moeten via de conversietool opgeslagen worden zodat ze direct in de juiste structuur opgeslagen staan.

Het volgende moet nog verder uitgewerkt worden als we deze migratiestap uitvoeren:

* Overzetten van berichten genereert nieuwe resource versions. Dit is niet gewenst
* Overzetten van berichten genereert push notificaties. Dit is mogelijk niet gewenst

1. [↑](#footnote-ref-1)