



Koppeltaal Architectuur 1.3.5

postadres: Postbus 30920, 2500 GX Den Haag
bezoekadres: Oude Middenweg 55, 2491 AC Den Haag
telefoon: (070) 317 34 50; fax: (070) 320 74 37; e-mail: info@vzvz.nl
www.vzvz.nl

Het is niet toegestaan om zonder voorafgaande toestemming van VZVZ Servicecentrum de gegevens te kopiëren, te wijzigen, te distribueren, te verspreiden, of op andere wijze te gebruiken en / of te exploiteren.

Datum: [23-11-2018]
Versie: [1.3]
Status: [Review]
Classificatie: [Koppeltaal]
Contact: [deijmann@vzvz.nl]

Inhoudsopgave

1	Doel en aanleiding	6
1.1	Doel en scope	6
1.2	Leeswijzer	6
1.3	Eenheid van taal	8
2	Koppeltaal	8
2.1	Business architectuur.....	8
2.1.1	Primaire interacties: behandelplan, activiteiten, en workflow.....	8
2.1.2	Positie van Koppeltaal in het GGZ referentiedomeinen model	10
2.1.3	Interoperabiliteit: domein en applicaties.....	11
2.1.4	Beheer interacties:	12
2.1.5	Specificatie van informatiemodel applicaties	12
2.1.6	Ondersteunende interacties: scenario's en usecases	14
2.2	Juridisch kader.....	19
3	Interoperabiliteit: Domein en Applicaties	21
3.1	Context.....	21
3.2	Koppeltaal componenten	23
3.2.1	Koppeltaal Domeinen	24
3.2.2	Applicatie model.....	25
4	Informatiemodel	29
4.1	Architectuur in het kort	29
4.2	Uitgangspunten.....	29
4.3	Bericht – Conceptueel Model.....	29
4.4	Resource versioning.....	30
4.5	Response	31
4.6	FHIR Resources.....	31
5	Interactie tussen Informatiesystemen.....	34
5.1	Informatie zoeken en lezen	35
5.2	Routering	36
5.2.1	Berichten versturen	36
5.2.1.1	Berichten Routeren	37
5.3	Berichten Ophalen.....	38
5.4	Notificaties	39
5.5	Autorisatie flow – SSO	39
5.6	OAuth2 Specs, Web en Mobile launch	40
5.6.1	OAuth2 Specs gedistribueerde authenticatie	40
5.6.2	Web Launch.....	41
5.6.3	Mobile Launch	42
5.6.4	Gebruik van Refresh Token	42
5.7	Storage Service Proxy.....	43
6	Primaire interacties: gegevensuitwisseling	44
6.1	Exchange Informatie model	44
6.2	Activity Definities lezen en updaten	45
6.3	Care Plan uitwisseling	46
6.4	Patiënt en behandelaar uitwisseling	47
6.5	Het starten van een Applicatie	48
6.6	Activity Status en Result updates	49
6.7	User berichten	51
7	Woordenlijst.....	52
8	Appendix: Voorbeeld resource versioning	54
9	Appendix: Response voorbeelden.....	54

10	Appendix: FHIR Resources	57
10.1	(Message) Bundle	57
10.2	MessageHeader.....	58
10.3	ActivityDefinition.....	60
10.4	CarePlan	63
10.5	CareTeam	66
10.5.1	Definitie	67
10.5.2	CareTeam elementen	67
	Careteam Valuesets.....	69
10.5.3	CareTeamStatus (nieuw)	69
10.5.4	Concepts	69
10.5.5	OtherResourceUsage (update).....	69
10.5.6	Toevoeging CarePlan definitie	70
10.5.7	CarePlan.participant.careTeam	70
10.5.8	CarePlan.activity.participant.careTeam	70
10.6	CarePlanActivityStatus	71
10.7	CarePlanActivityResult	72
10.8	Questionnaire	73
10.9	Observation.....	75
10.10	UserMessage	75
10.11	StorageItem.....	76
11	Appendix: KoppelTaal server Datadiagram	78

Wijzigingenblad

Versie	Datum	Omschrijving	Auteur
0.1	20-03-2018	Draft	Reli Todea
0.11	6 april 2018	Review draft voor feedback aan Reli en toets door Wouter Tesink op passendheid vorm	Sergej van Middendorp
0.17	25 Mei 2018	Review Tom Deijman en Wouter Tesink	Reli Todea
0.19	30 mei 2018	Aanvullingen business laag	Sergej van Middendorp
0.19a	3 juni 2018	Review Tom Deijman en Johan verwerkt	Reli Todea
0.20	3 juni03 Juni 2018	Aanvullingen business laag n.a.v. overleg Tom – Sergej en samenvoeging 0.19a Reli	Sergej van Middendorp
0.20 feedback PBK	9 juni	Schriftelijke feedback als vervolg op de reviewsessie 8 juni.	PBK
0.21	14 juni	Opmerkingen aangevuld met reacties en plan voor verwerking door Tom, Reli, Sergej	Sergej van Middendorp
0.5	22 juni	Review comments volgens plan voor verwerking verwerkt.	Reli Todea, Sergej van Middendorp
0.6	29 juni	Laatste opmerkingen van Tom verwerkt	Sergej van Middendorp
1.0	6 juli 2018	Kleine zaken verduidelijkt. Diagrammen in eenheid van taal (Engels of Nederlands) voor versie 1.1 Overdracht naar documentatie beheer.	Tom Deijmann
1.1	16 juli 2018	Klein tekstuele wijzigingen. Datadiagram in appendix toegevoegd. MaximumMessageAgeInDays vervangt de 56 dagen maximale bewaartijd van berichten.	Tom Deijmann
1.2		Update a.d.h.v. TO release 1.3.5.	Tom Deijmann
1.3	23-11-2018	Kleine tekstuele aanpassingen	Tom Deijmann

Figuren:

Figuur 1. Koppeltaal architectuur	7
Figuur 2. Businessrollen waar Koppeltaal een functie heeft.	10
Figuur 3. Koppeltaal domeinen, applicaties en beheerfuncties	11
Figuur 4. Blended Care proces.....	18
Figuur 5. Applicatie rollen in het Koppeltaal Systeem	19
Figuur 6. Registratie op berichttype	21
Figuur 7. Domein en Applicatie model	21
Figuur 8. Applicatie-instantie datamodel	24
Figuur 9. Koppeltaal Domein Management	25
Figuur 10. Applicatie typen	25
Figuur 11. Applicatie Domein model	26
Figuur 12. Applicatieregister	27
Figuur 13. Koppeltaal gebruikers	27
Figuur 14. Koppeltaal database tabelstructuur.....	28
Figuur 15. Bericht samenstelling.....	30
Figuur 16. Koppeltaal FHIR Resources	32
Figuur 17. Koppeltaal Bundle	33
Figuur 18. Informatie zoeken en lezen	35
Figuur 19. Routing	36
Figuur 20. Bericht Routing	37
Figuur 21. Ophalen berichten	38
Figuur 22. Bericht status	39
Figuur 23. Voorbeeld Game Launch via SSO	41
Figuur 24. Storage Service Proxy gebruik	43
Figuur 25. Proxying door het KTS Storage Service	44
Figuur 26. Koppeltaal berichten	45
Figuur 27. Activity Definitie.....	45
Figuur 28. Update activity definitie	45
Figuur 29. Activity Definitie opvragen.....	46
Figuur 30. Sync activity definities	46
Figuur 31. CarePlanstructuur	47
Figuur 32. CarePlan bericht.....	47
Figuur 33. Routeren CreateOrUpdatePatient.....	48
Figuur 34. Patient synchronisatie	48
Figuur 35. Launch sequence.....	49
Figuur 36. Activity Status	50
Figuur 37. Interventie en update behandelplan.....	50
Figuur 38. Voorbeeld van een "Condition" respons bericht.	56
Figuur 39. KoppelTaal server Datadiagram	78

1 Doel en aanleiding

1.1 Doel en scope

Het doel van dit document is om de architectuur van Koppeltaal GGZ te beschrijven, zoals deze is in versie 1.3.5. Met deze beschrijving kan VZVZ verantwoordelijkheid nemen voor de architectuur van Koppeltaal en kunnen toekomstige plannen voor Koppeltaal onder architectuur doorgevoerd worden. De scope is een beschrijving van de architectuur op 'Enterprise Architectuur' niveau, waarbij het TOGAF raamwerk, en de Archimate methode zoveel mogelijk gebruikt worden als taal voor vastlegging van de architectuur.

De architectuurbeschrijving beperkt zich tot het wat en hoe van Koppeltaal 'zelf'. Daar waar relaties met de organisatie en de processen van Koppeltaal en VZVZ relevant zijn worden deze relaties wel benoemd, maar niet verder uitgewerkt. In dit document vind je bijvoorbeeld geen beschrijving van het proces van ketenbeheer, maar wel een beschrijving van voor de architectuur belangrijke rollen en functionaliteiten voor Ketenbeheer. Voor de zaken buiten scope wordt op een nader te bepalen plaats de beschrijving ontwikkeld en beheerd. Waar deze beschikbaar is proberen we daarnaar te verwijzen.

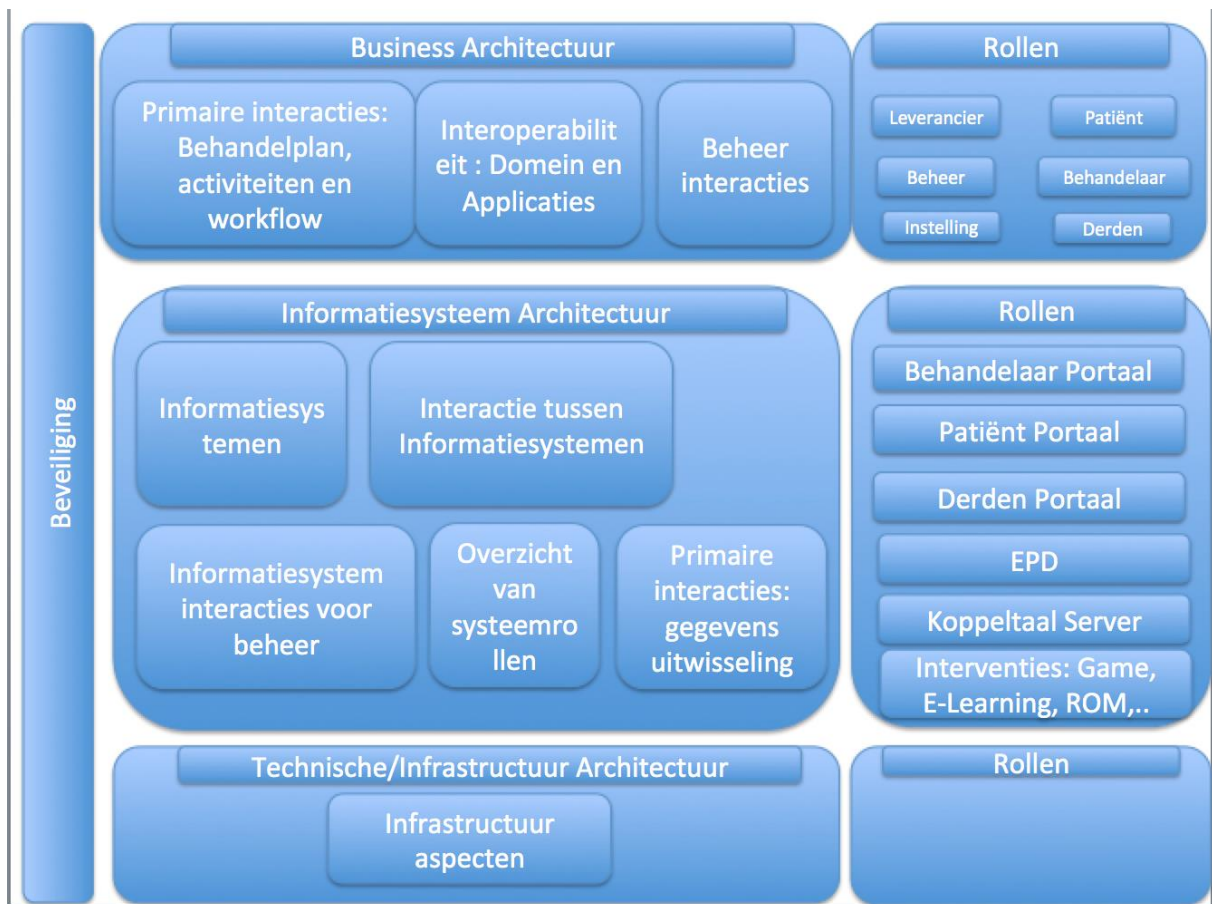
Tenslotte beperkt deze architectuur zich in deze versie tot de beschrijving die is vastgesteld aan de hand van de beoordeling van de architectuur zoals door VZVZ uitgevoerd in het najaar van 2017. Daarin is een aantal aspecten van architectuur benoemd die in de documentatie van Koppeltaal ontbraken, of niet op het juiste kwaliteitsniveau waren. Deze zijn met dit document geadresseerd.

Buiten scope ontstaat daarbij een lijst met eisen en wensen voor verdere ontwikkeling van de documentatie na deze mijlpaal. Deze eisen en wensen worden na afronding van dit document opgepakt onder coördinatie van de architecten van VZVZ servicecentrum.

1.2 Leeswijzer

Dit architectuurdokument is bedoeld voor VZVZ om vragen te kunnen beantwoorden over huidige mogelijkheden en beperkingen van de Koppeltaal infrastructuur en dient als basis voor eventuele uitbreiding van functionaliteiten. Het document bevat eveneens (verwijzingen naar) standaarden die door GGZ instellingen en ICT leveranciers voor die instellingen voor behandelprocessen en de daarbij behorende gegevensuitwisseling.

De architectuurbeschrijving in dit document is globaal onder te verdelen in vier secties, namelijk de 'business' architectuur, de informatiesysteemarchitectuur, de technische of infrastructuurarchitectuur en een aparte bespreking van de beveiligingsaspecten.



Figuur 1. Koppeltaal architectuur

De samenhang van Koppeltaal is gecompliceerd. De keuze voor beschrijving in lagen zorgt ervoor dat aspecten die eerst op hoofdlijnen zijn besproken, in een later hoofdstuk in detail worden uitgewerkt. Ook gebeurt het in dit document dat een aspect van de architectuur eerst vanuit een bepaald gezichtspunt wordt beschreven (bijvoorbeeld de gebruiker) en dat datzelfde aspect in een later hoofdstuk opnieuw wordt beschreven, maar dan vanuit een ander gezichtspunt (bijvoorbeeld de beheerder).

Als je na het lezen van een sectie dus meer detail nodig hebt om het goed te begrijpen, dan volgt dat detail meestal later, in een ander hoofdstuk. Ook als je na het lezen van een sectie denkt dat een beschrijving beperkt is, dan is de kans groot dat het even later vanuit een ander gezichtspunt verder wordt uitgewerkt of aangevuld.

Het leesadvies is dus wees geduldig en lees het geheel door. Maak eventueel een aantekening van je vraag en check of de vragen die je hebt na het lezen van het document beantwoord zijn. Zo niet laat het ons weten.

1.3 Eenheid van taal

De architectuur van Koppeltaal GGZ heeft zich over de afgelopen vier jaar ontwikkeld op basis van de visie op architectuur die in 2014 ontwikkeld is aan de start van het project. Gedurende die vier jaar is er veel gebeurd en zijn er steeds meer leveranciers aangesloten uit de complexe en multidisciplinaire 'sector' die de GGZ is. Om ons heen is in diezelfde tijd ook veel gebeurd. Denk aan het programma Medmij, de nieuwe GGZ, Positieve gezondheid, het informatieberaad, en de wereldwijde adoptie van FHIR. De invloed van deze factoren, en het voortschrijdend inzicht hebben ironisch genoeg, geleid tot een diversiteit van gebruik van begrippen binnen Koppeltaal GGZ.

Met dit document proberen we ook daar een integratie te bewerkstelligen. Dit doen we door in de businessarchitectuur zoveel mogelijk van de kernbegrippen in Koppeltaal een definitie te geven. Daarnaast hebben we een begrippenlijst ontwikkeld die aan het eind van het document als bijlage kan helpen om de definities te overzien en tevens synoniemen of relaties tussen begrippen te duiden.

Ondanks al deze maatregelen, kan het nog voorkomen dat in deze versie imperfecties op dit gebied te vinden zijn. Ook is het soms bewust zo dat een term in het gebruik binnen Koppeltaal verschilt. Ter illustratie. We kiezen ervoor om het begrip cliënt te hanteren op de 'business architectuur' laag. Terwijl het in de tweedelijns GGZ en ook bij de huisarts in het algemeen over een patiënt gaat. Op de informatiesysteem architectuur laag, spreekt Koppeltaal van Patient, omdat deze term in de FHIR standaard weer de norm is. Waar mogelijk verklaren we deze wisseling bij het eerste gebruik in de tekst, en anders in de begrippenlijst.

In Koppeltaal worden ook Nederlands en Engels door elkaar gebruikt. Over het algemeen wordt in de technische context van Koppeltaal Engels gesproken. Met name omdat veel software ontwikkelaars, ook in Koppeltaal, niet Nederlands zijn. Waar mogelijk wordt in dit document een Nederlandse term voor een architectuur artefact gedefinieerd. En waar relevant worden de Engelse varianten daarvan benoemd en eventuele synoniemen gedefinieerd. Ook in de begrippenlijst proberen we deze te bundelen.

2 Koppeltaal

2.1 Business architectuur

2.1.1 Primaire interacties: behandelplan, activiteiten, en workflow

De stichting Koppeltaal heeft als doel een **integratiestandaard** te ontwikkelen, beheren, en implementeren, waarmee interne **gegevensuitwisseling** en **applicatie-integratie** tussen **ICT-systemen** en **eHealth-platformen** wordt geregeld bij **zorginstellingen** die diensten voor gedragsverandering en daarmee direct samenhangende diensten verlenen in het algemeen en instellingen voor **Geestelijke Gezondheidszorg** (GGZ) in het bijzonder¹.

¹ Zie: <https://www.koppeltaal.nl/over-ons/over-stichting-koppeltaal>

Gegevensuitwisseling tussen ICT-systemen en eHealth-platformen via de integratie standaard van de stichting Koppeltaal (Koppeltaal) is relevant in de context van **blended care in de GGZ**. Bij blended care in de GGZ worden reguliere face-to-face gesprekken gecombineerd met **online interventies** zoals bijvoorbeeld chat, beeldbellen, **online behandelmodules** en **online inzage in het eigen gezondheidsdossier**². Hierdoor kan een cliënt niet alleen tijd, maar ook plaats-onafhankelijk zorg gebruiken via een tablet of smartphone³.

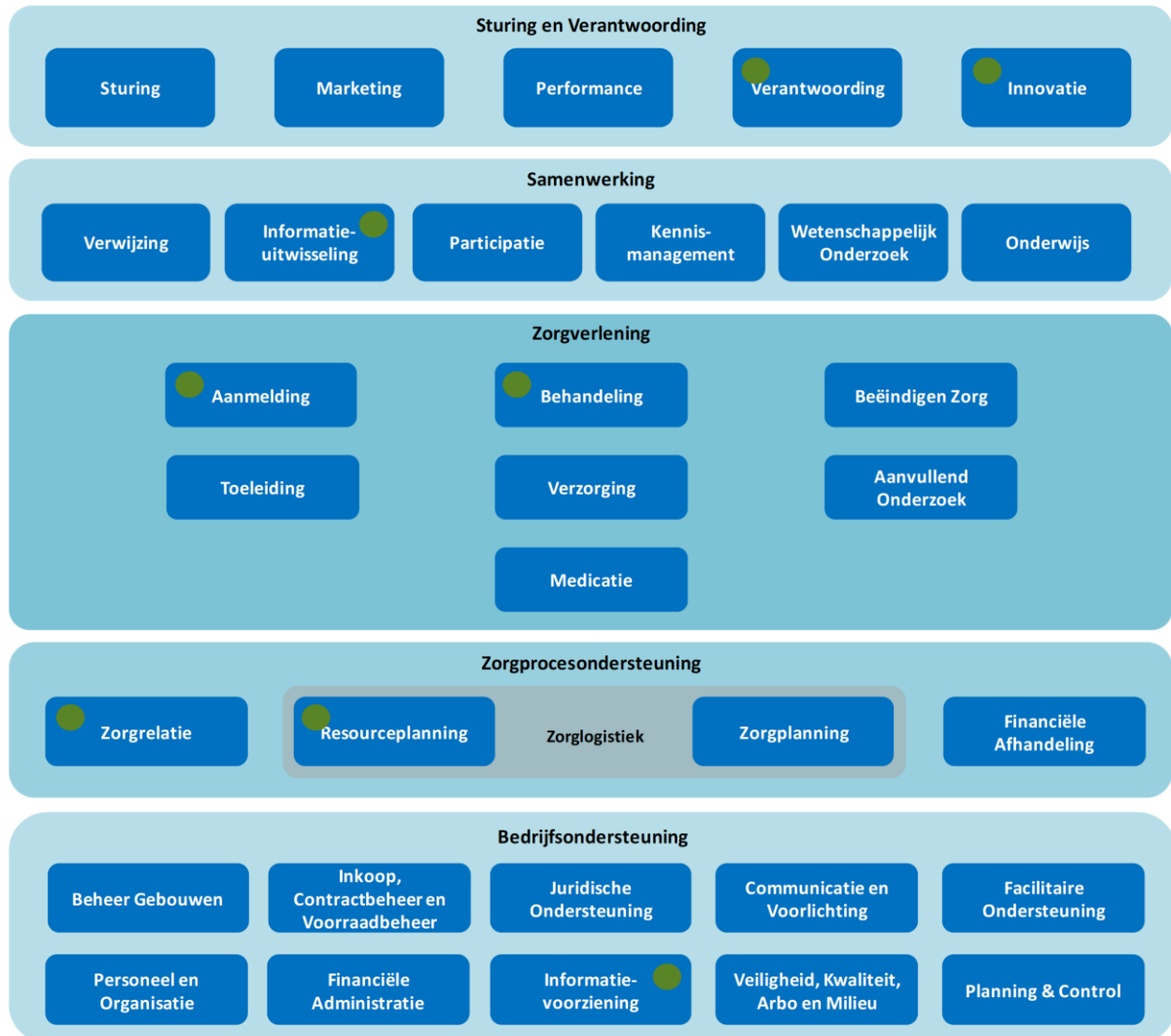
Een blended care behandeling vormt een geheel met een begin en een eind, in Koppeltaal noemen we dat een **behandelplan**. In dat plan worden verschillende **activiteiten** uitgevoerd, eventueel ondersteund door een bepaalde, logische, volgorde (**workflow**). Deze activiteiten kunnen zijn, het bepalen van de doelen van een behandeling, het maken van een afspraak, het uitvoeren van een (online) interventie, het bespreken of bekijken van **voortgang, status, resultaten**, en het **evalueren** van de vooruitgang van de conditie van de Cliënt ten opzichte van de behandeldoelen. Voor zover deze activiteiten door een ICT-systeem of een eHealth platform worden ondersteund, is gegevensuitwisseling via Koppeltaal mogelijk.

Bij een blended care behandeling zijn tenminste een **cliënt** en een **behandelaar** betrokken. En steeds vaker ook '**derden**', zoals vrienden, familie, lotgenoten, en ervaringsdeskundigen.

² Koppeltaal biedt deze functie niet. Deze functie wordt in het Medmij afsprakenstelsel voorzien. Zie <http://www.medmij.nl>

³ Zie: <http://www.ggznederland.nl/themas/e-health>

2.1.2 Positie van Koppeltaal in het GGZ referentiedomeinen model⁴



Figuur 2. Businessrollen waar Koppeltaal een functie heeft.

In het door GGZ Nederland en Nictiz opgestelde GGZ Domein referentiemodel speelt Koppeltaal een mogelijke rol in de met een groene cirkel aangeduide subdomeinen.

Primair helpt Koppeltaal om behandeling te ondersteunen. Specifiek eHealth in blended care processen⁵. Secundair kan het ook aanmeldingsprocessen helpen ondersteunen via het synchroniseren van patiëntgegevens over verschillende applicaties. Daarnaast kan Koppeltaal ook een rol spelen in de zorgrelatie en de resourceplanning, omdat er rond

⁴ Zie <https://www.nictiz.nl/standaardisatie/referentiedomeinenmodellen/referentiedomeinmodel-ggz/>

⁵ We beperken de beschrijving van Koppeltaal in de context van de GGZ instelling tot de hoofdlijnen. Gedetailleerde beschrijvingen van Koppeltaal volgen hieronder.

een behandelplan met Koppeltaal relaties gelegd kunnen worden tussen cliënt, behandelaar, en derden (zoals familie, vrienden, ervaringsdeskundigen, etc.).

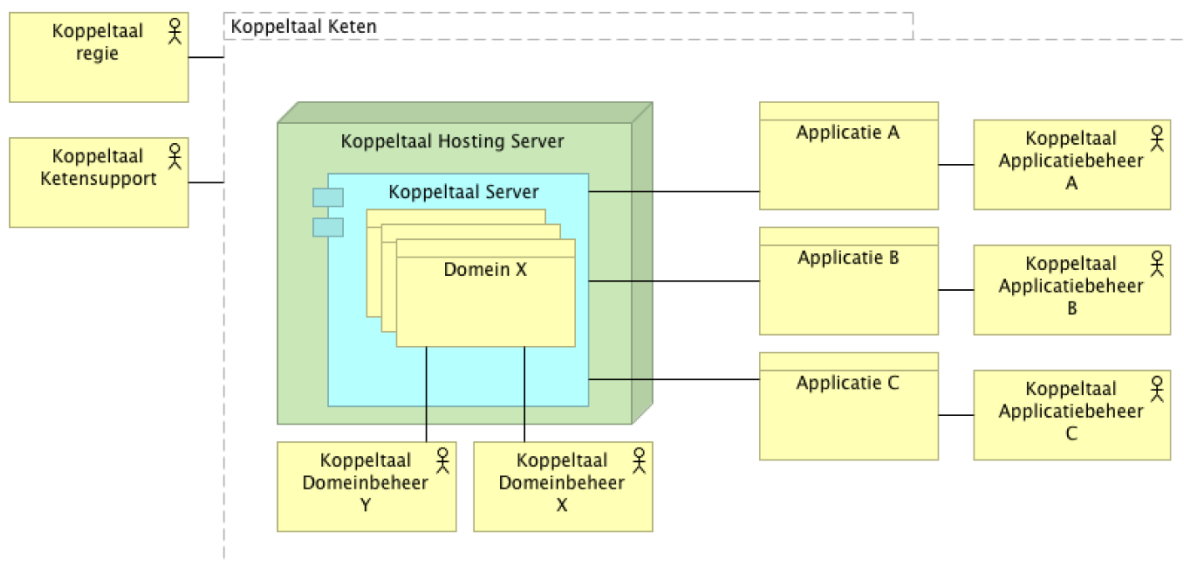
Koppeltaal kan ook een rol spelen in de informatievoorziening, en de informatie uitwisseling. Zolang de scope daarvan is voorziening en uitwisseling binnen de verantwoordelijkheid van de aan Koppeltaal deelnemende zorgaanbieder.

Ten slotte kan Koppeltaal een rol spelen in de verantwoording en de innovatie. Via Koppeltaal kan informatie voor besturing verkregen worden over de inzet van eHealth in blended care processen geïntegreerd over verschillende applicaties heen. Daarnaast kan met Koppeltaal de adoptie van eHealth onder cliënten en behandelaren versneld worden via de door Koppeltaal veroorloofde keuzevrijheid, flexibiliteit, en gebruikersgemak.

2.1.3 Interoperabiliteit: domein en applicaties

In de doelstelling van stichting Koppeltaal is middels het woord 'interne' een beperking voor de gegevensuitwisseling opgenomen. Met deze beperking wordt bedoeld dat de gegevensuitwisseling altijd plaatsvindt onder de verantwoordelijkheid van één **zorgaanbieder**. In de koppeltaal architectuur is daarom gekozen voor het concept **domein**. Een domein, in Koppeltaal, is synoniem aan een zorgaanbieder, en beperkt de gegevensuitwisseling tot *interne gegevensuitwisseling* onder verantwoordelijkheid van de betreffende zorgaanbieder.

Gegevens worden uitgewisseld tussen verschillende **applicaties**. In Koppeltaal staat het begrip applicaties voor alle vormen van ICT-systemen en eHealthplatforms die voor een zorgaanbieder relevant zijn om gegevens tussen uit te wisselen in de context van blended care behandelingen in de **instelling**. Applicaties worden geleverd door verschillende **leveranciers**. Deze leveranciers gebruiken Koppeltaal om gegevens uit te wisselen onder verantwoordelijkheid van de zorgaanbieder in een domein. Domeinen zijn voor applicaties beschikbaar via de **Koppeltaal Server**, een applicatie die via internet beschikbaar is voor bij Koppeltaal aangesloten leveranciers.



Figuur 3. Koppeltaal domeinen, applicaties en beheerfuncties

2.1.4 Beheer interacties:

Om Koppeltaal als dienst aan het hierboven beschreven netwerk ter beschikking te stellen is er interactie tussen de beheer rollen van zorgaanbieders, leveranciers, en stichting Koppeltaal. **Koppeltaal regie** zorgt voor optimale beschikbaarheid van informatie in de **Koppeltaal keten**. Koppeltaal regie wordt gevoerd door VZVZ servicecentrum. De rol **Koppeltaal ketensupport**⁶ zorgt voor de dienstverlening aan alle domeinen en applicaties die aangesloten zijn bij Koppeltaal. Koppeltaal ketensupport wordt op dit moment uitgevoerd door VitalHealth Solutions. De rol **Koppeltaal domeinbeheer**⁷ is de gedelegeerd verantwoordelijke namens een zorgaanbieder voor het domein van de zorgaanbieder. Namens de zorgaanbieder bepaalt de Koppeltaal domeinbeheerder welke applicaties met elkaar in een domein uitwisselen op basis van overeenkomsten tussen de zorgaanbieder en de applicatie-leveranciers. De rol Koppeltaal **domeinbeheer** is daarnaast verantwoordelijkheid voor de beschikbaarheid van dienstverlening via Koppeltaal voor behandelaars en cliënten in het betreffende domein. De rol van Koppeltaal domeinbeheer wordt uitgevoerd door VitalHealth Solutions onder verantwoordelijkheid van de deelnemende zorgaanbieder. En per applicatie zorgt de rol **Koppeltaal applicatiebeheer**⁸ voor de beschikbaarheid van de betreffende applicatie voor de verschillende domeinen waarop deze applicatie is aangesloten. De rol van Koppeltaal applicatiebeheer wordt door VitalHealth Solutions uitgevoerd onder verantwoordelijkheid van de betreffende Koppeltaal IT deelnemer.

Lokale verstoringen, ketenverstoringen, en veranderingen van belang voor het functioneren van Koppeltaal worden tussen deze rollen gecoördineerd afgehandeld volgens het proces ketensamenwerking in de productiefase.

2.1.5 Specificatie van informatiemodel applicaties

In Koppeltaal worden veel verschillende termen gebruikt voor het begrip applicatie. Het is voor de coherentie van de architectuur belangrijk het gebruik van deze verschillende termen nader te specificeren. Hiervoor dient het onderstaande informatiemodel.

eHealth is volgens GGZ Nederland "elektronische gezondheid, ofwel zorg via internet"⁹. Volgens het KNMG is eHealth: "het gebruik van informatie- en communicatietechnologie ter ondersteuning of verbetering van de gezondheid en de gezondheidszorg"¹⁰, en volgens een analyse in zorgvisie: "[is] de definitie van eHealth [...] breed en beperkt zich niet tot één specifieke technologie, al speelt internettechnologie een prominente rol. EHealth is evenmin gericht op specifieke gebruikersgroepen."

Voor Koppeltaal hanteren we de eHealth definitie van de website ggzstandaarden.nl die zegt:

"Het gebruik van informatie- en communicatietechnologie om mensen met psychische klachten en/of aandoeningen te informeren en/of te ondersteunen met betrekking tot

⁶ In architectuur ook aangeduid als Koppeltaal Admin(istrator)

⁷ In architectuur ook aangeduid als Domein Admin(istrator) of domain admin

⁸ In architectuur ook aangeduid als Applicatie Admin(istrator) of application admin

⁹ Zie: <http://www.ggznederland.nl/themas/e-health>

¹⁰ Zie: <https://www.knmg.nl/advies-richtlijnen/dossiers/ehealth.htm>

hun psychische gezondheid om zo het herstelproces en daarmee hun kwaliteit van leven te verbeteren.¹¹

Daarnaast hanteren we vanaf hier ook de schrijfwijze van deze standaard en die is: **eHealth**.

We maken onderscheid tussen twee typen **eHealth applicaties: eHealthplatforms** en **eHealth modules**. eHealthplatforms bieden zowel interactie 'over' een behandeling als modules voor interventie 'in' een behandeling. Interactie 'over' een behandeling gaat bijvoorbeeld over diagnose, behandelplan, activiteiten, voortgang en resultaten van een behandeling. eHealth modules zijn specifieke interventies voor gebruik 'in' een behandeling. Het VIPP GGZ programma definieert een eHealth module als "Een eHealth module is onderdeel van de behandeling en/of de data die verzameld wordt met de module wordt gebruikt in het zorgproces (het kan bijvoorbeeld gaan om (een combinatie van) instructies, vragenlijsten en bewegend beeld in interactie met de cliënt".

In een behandeling kunnen cliënten bijvoorbeeld een level van een game spelen waarmee ze werken aan een verbetering van hun conditie. eHealth modules worden 'los' van het eHealthplatform aangeboden, en een eHealthplatform omvat altijd één of meerdere modules. In een platform zijn de modules over het algemeen ook sterk geïntegreerd in de overige functionaliteiten. Naast eHealth applicaties spelen ook ROM en EPD systemen een rol in Koppeltaal. Deze typen systemen worden later beschreven.

eHealth wordt ook ontsloten via **portalen**. Volgens Wikipedia is een portaal: "een webpagina die dienstdoet als "toegangspoort" tot een reeks andere websites, die over hetzelfde onderwerp gaan". Via portalen kunnen Cliënten, behandelaars, en derden toegang krijgen tot functionaliteiten 'over' een behandeling en interventies voor gebruik 'in' een behandeling.

Naast eHealthmodules, eHealthplatforms en portalen spelen **Routine Outcome Measurement (ROM)** en **Elektronische Cliënt Dossier (EPD)** systemen een rol in Koppeltaal. ROM systemen worden ingezet om voor, tijdens, en na de behandeling te meten wat de conditie van de cliënt is. Het doel van deze metingen is om te kunnen sturen in de behandeling, om te leren wat werkt en wat niet werkt, om te verantwoorden, en voor onderzoek¹².

EPD's worden ingezet voor veel verschillende functies. In de context van Koppeltaal zijn besturing op eHealth en de behandelomgeving voor de behandelaar relevant.

In de eerste lijns GGZ, bij de huisarts, wordt blended care aangeboden door de praktijkondersteuner huisarts voor de GGZ (POH-GGZ). Het EPD voor de POH is het Keteninformatiesysteem (**KIS**). Ook deze applicaties worden via Koppeltaal geïntegreerd met eHealthplatforms en eHealthmodules.

Naast de beschreven applicatie gebruiken GGZ instellingen nog een veelheid aan applicaties, waaronder Elektronische Voorschrijfsystemen (EVS), Kantoorautomatisering, etc. Deze applicaties vallen buiten de scope van Koppeltaal.

¹¹ Zie: <https://www.ggzstandaarden.nl/generieke-modules/ehealth/inleiding>

¹² Zie: <https://www.ggzstandaarden.nl/zorgstandaarden/opiaatverslaving/achtergronddocumenten/verdieping/rom>

2.1.6 Ondersteunende interacties: scenario's en usecases

Koppeltaal biedt via de standaard flexibiliteit, keuzevrijheid en gebruiksgemak in blended care processen. Om te illustreren hoe de standaard dat doet beschrijven we hieronder twee voorbeeld scenario's van verschillende blended care behandelingen. Een in de eerste lijns zorg en een in de gespecialiseerde GGZ zorg. De stappen in de scenario's staan in de linker kolom beschreven, en in de rechterkolom staan de koppeltaal usecases die gebruikt worden in de ondersteuning van de betreffende stap in het proces.

Scenario per stap	Use case / functie Koppeltaal
- Een cliënt meldt zich bij de huisarts met chronische spanning en hoofdpijn. De huisarts constateert dat er veel angst speelt bij deze cliënt en vraagt de praktijkondersteuner GGZ (POH) de cliënt te begeleiden. De POH opent de pagina van deze cliënt in het behandelaarsplatform (KIS) van VitalHealth, en kijkt in de lijst met beschikbare modules welke angstmodule de cliënt het best kan volgen. Ze ziet de module angst en spanning van New Health Collective.	- (Sub)activiteiten publiceren voor gebruik in andere applicaties in het domein
- Ze wijst deze module toe aan de cliënt. Het handige is dat wanneer dit gebeurt, er ook een identiteit voor de cliënt wordt aangemaakt in Pazio, het cliëntportaal van de praktijk. Daar wordt de module ook 'klaargezet'.	- (Sub)activiteiten uit applicaties toekennen aan een gebruiker in een andere applicatie
- De cliënt ontvangt in de mail een uitnodiging voor toegang tot het cliëntportaal van de huisarts. In dit geval Pazio. De cliënt klikt op de link en komt op Pazio terecht. Daar logt de cliënt in. In het cliëntportaal van Pazio staat de module angst en spanning als een 'blok' klaar zoals besproken. De cliënt klikt op het blok en de module wordt geopend. De cliënt gaat aan de slag. "Handig dat ik zo kan starten zonder nog een keer in te loggen", denkt de cliënt. De cliënt werkt een half uur aan de module en sluit deze na afronding af en gaat slapen.	- Single-sign-on realiseren tussen koppeltaal applicaties en het lanceren van de bijbehorende (sub)activiteit

Scenario per stap	Use case / functie Koppeltaal
- De POH komt in de ochtend op kantoor en opent het KIS. Ze ziet een notificatie dat de cliënt de module heeft geopend, afgerond, en afgesloten.	- Voortgang- en statusberichten van (sub)activiteiten delen en ontvangen
- Ze klikt door naar de pagina van de cliënt en kan daar direct de scores van de app en de reflectie van de cliënt daarop lezen.	- Resultaatberichten van (sub)activiteiten delen en ontvangen
- Ze twijfelt of ze de reflectie van de cliënt op de afsluitende opdracht van de module goed begrijpt. Ze klikt door op het resultaat, daarmee wordt de module geopend en kan ze zien wat de cliënt precies gedaan heeft in de opdracht. Nu is ze goed voorbereid voor het gesprek van vanmiddag.	- Single-sign-on realiseren tussen koppeltaal applicaties en het lanceren van de bijbehorende (sub)activiteit

Tabel 1. Eerste lijns zorg scenario - Use Cases

In de specialistische GGZ zou Koppeltaal bijvoorbeeld in het volgende scenario ingezet kunnen worden:

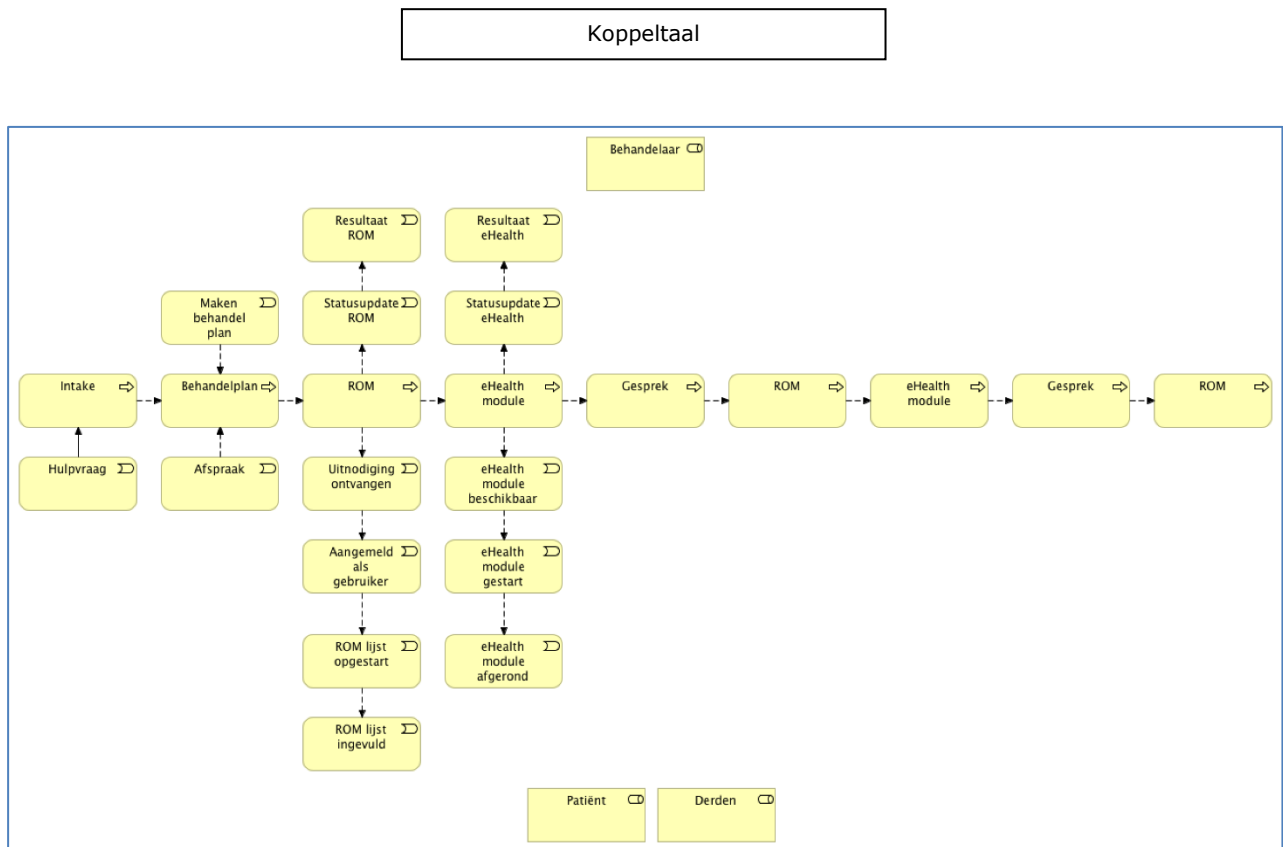
Scenario per stap	Use case / functie Koppeltaal
- Een cliënt is in gesprek met de psychiater. Er is complexe problematiek aan de orde, maar de problematiek is niet ernstig. Er kan behandeld worden met blended care. De psychiater kiest in de cliëntpagina van het EPD (bijvoorbeeld MijnQuarant) een passende ROM vragenlijst (bijvoorbeeld uit VitalHealth Questmanager) om beter beeld te krijgen van de startsituatie van de behandeling.	- (Sub)activiteiten publiceren voor gebruik in andere applicaties in het domein.
- De behandelaar kiest de betreffende vragenlijst en kent deze toe aan de cliënt.	- (Sub)activiteiten uit applicaties toekennen aan een gebruiker in een andere applicatie
- De cliënt wordt, voor de behandelaar automatisch aangemaakt in het eHealthplatform (bijvoorbeeld Karify) en ontvangt via de e-mail een uitnodiging om in te loggen	- Gebruikers (cliënten, behandelaars, derden) aanmaken of updaten tussen applicaties in een domein.

Scenario per stap	Use case / functie Koppeltaal
<ul style="list-style-type: none"> - De cliënt logt in op het eHealthplatform en ziet een knop om de vragenlijst te starten. De cliënt klikt en de vragenlijst wordt geopend. De cliënt vult de lijst in, maar stopt halverwege. Te moeilijk. 	<ul style="list-style-type: none"> - Single-sign-on realiseren tussen koppeltaal applicaties en het lanceren van de bijbehorende (sub)activiteit
<ul style="list-style-type: none"> - De behandelaar krijgt via het EPD een reminder om de status van deze cliënt te checken. Er is al een poosje niets gebeurd. In het EPD ziet de behandelaar dat de cliënt ingelogd is op het eHealthplatform en dat de ROM lijst gestart is, maar halverwege onderbroken, zonder resultaat. Er is nu een week niets gebeurd. De behandelaar besluit de cliënt te bellen om te vragen wat er mis gaat. 	<ul style="list-style-type: none"> - Voortgang- en statusberichten van (sub)activiteiten delen en ontvangen
<ul style="list-style-type: none"> - De cliënt geeft aan dat hij de vragenlijst moeilijk vindt en dat hij daarop afhaakt. Dat is onderdeel van het probleem waarom de cliënt in de eerste plaats kwam. In overleg besluiten behandelaar en cliënt een familielid bij te schakelen om te helpen. Het familielid kan dat op zijn eigen systeem zien wat de cliënt heeft ingevuld en daarop toelichting geven voor de behandelaar. De cliënt nodigt via het eHealthplatform zijn vrouw uit om mee te helpen. De behandelaar krijgt in het EPD een signaal dat er een relatie is toegevoegd aan de pagina van de cliënt. 	<ul style="list-style-type: none"> - Gebruikers (cliënten, behandelaars, derden) aanmaken of updaten in een domein.
<ul style="list-style-type: none"> - Met behulp van zijn vrouw lukt het om de vragenlijst af te ronden. De behandelaar krijgt signaal van de afronding en bekijkt de resultaten ter voorbereiding op het volgende gesprek. 	<ul style="list-style-type: none"> - Resultaatberichten van (sub)activiteiten delen en ontvangen
<ul style="list-style-type: none"> - In gesprek besluiten cliënt en behandelaar voor maatwerk op het gebied van eHealth en monitoring. De cliënt gaat de Journey app gebruiken om zelf doelen te stellen, te monitoren en met zijn familie en vrienden passende content voor 	<ul style="list-style-type: none"> - (Sub)activiteiten uit applicaties toekennen aan een gebruiker in een andere applicatie

Scenario per stap	Use case / functie Koppeltaal
zelfzorg te verzamelen. De behandelaar kent de Journey app toe aan de cliënt.	
- De cliënt logt thuis in op het eHealthplatform en ziet een code om de app te activeren en instructies om de app via de appstore te installeren op zijn iPhone. Hij gebruikt de code uit het eHealthportaal om de app te activeren. (de de app voor die gebruiker wordt aangemeld voor berichten van Koppeltaal)	- Het uitgeven van een unieke code voor een mobiele applicatie om deze te kunnen koppelen aan een behandelplan in een andere applicatie
- De behandelaar ziet met regelmaat status, voortgang en resultaten uit de Journey app van de cliënt langskomen in het EPD.	- Voortgang- en statusberichten van (sub)activiteiten delen en ontvangen - Resultaatberichten van (sub)activiteiten delen en ontvangen.
- Op een gegeven moment denkt de behandelaar aan een filmpje op internet waarvan hij denkt dat het behulpzaam kan zijn voor in de 'rugzak' van deze cliënt. Hij kopieert een link naar het filmpje in een bericht dat hij de cliënt stuurt vanuit het EPD. De cliënt opent de app in de ochtend en ziet daar een tip van zijn behandelaar. De cliënt bekijkt de tip en voegt deze toe aan zijn 'rugzak' in de app.	- Algemene gebruikersberichten delen en ontvangen.

Tabel 2. Specialistische zorg scenario - Use Cases

Schematisch is het Blended Care proces weergegeven in Figuur 4. hieronder. Een Blended Care proces bestaat uit verschillende stappen. Na een intake, om te bepalen of, en zo ja, wat voor behandeling aan de orde is, wordt een behandelplan opgesteld. Aan de start van dat plan wordt via ROM gemeten wat de uitgangssituatie is. Daarna volgt een periode van het werken aan eHealth modules en het voeren van gesprekken, met tussentijds, en aan het eind van de behandeling nog ROM metingen. Het type functies dat Koppeltaal aan behandelaar, cliënt en derden biedt is per type activiteit weergegeven.



Figuur 4. Blended Care proces

Koppeltaal biedt de volgende functionele dekking om scenario's zoals die hierboven mogelijk te maken:

Non functioneel:

- Registratie van applicaties bij een domein
- Berichten delen met, en ontvangen van, andere applicaties in dat domein

Deze functies worden gebruikt bij de configuratie van een domein in Koppeltaal. De verschillende applicaties worden geregistreerd voor dat domein. Daarmee komt er een instantie van de applicatie beschikbaar binnen dat domein. Alle applicaties die via Koppeltaal uit willen wisselen registreren een applicatie instantie voor dat domein.

Daarna kan elke applicatie aangeven welke Koppeltaal functies voor dat domein beschikbaar zijn. En tevens aangeven welke soorten berichten het van de andere applicaties wil ontvangen.

Koppeltaal kan ook functioneren als proxy voor opslag van gegevens. Zo kan een webapplicatie zonder backend geconfigureerd worden om Koppeltaal gegevens in een cliëntportaal op te slaan, zodat de status van die applicatie hersteld kan worden op het moment dat de cliënt er mee verder gaat. Deze functie wordt in de praktijk nog nergens gebruikt.

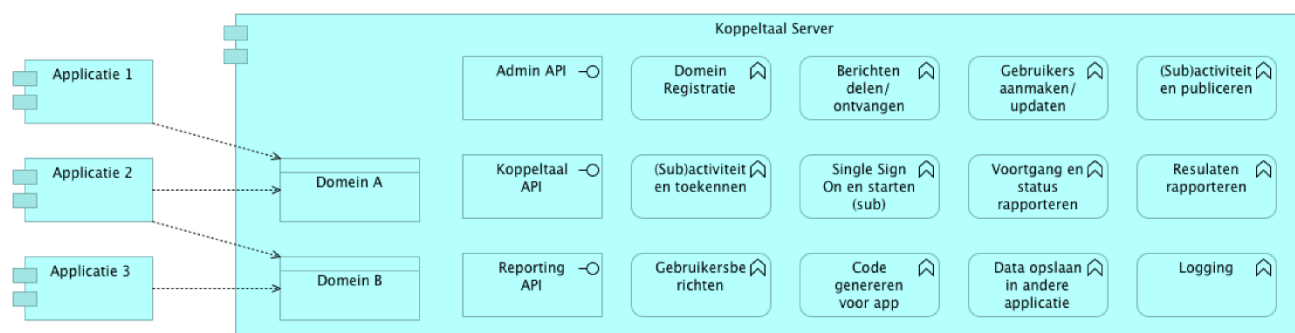
Koppeltaal Server Functies:

1. Gebruikers (cliënten, behandelaars, derden) aanmaken of updaten tussen applicaties in een domein.

2. (Sub)activiteiten publiceren voor gebruik in andere applicaties in het domein
3. (Sub)activiteiten uit applicaties toekennen aan een gebruiker in een andere applicatie dan waar de (Sub)activiteiten zijn opgeslagen
4. Single-sign-on realiseren tussen koppeltaal applicaties en het lanceren van de bijbehorende (sub)activiteit
5. Voortgang- en statusberichten van (sub)activiteiten delen en ontvangen
6. Resultaatberichten van (sub)activiteiten delen en ontvangen
7. Algemene gebruikersberichten delen en ontvangen
8. Het uitgeven van een unieke code voor een mobiele applicatie om deze te kunnen koppelen aan een behandelplan in een andere applicatie.

Deze functies worden in de twee scenario's hierboven nader geïllustreerd.

Het figuur hieronder geeft de functionele dekking weer in de context van de verschillende applicatie rollen in Koppeltaal.



Figuur 5. Applicatie rollen in het Koppeltaal Systeem

2.2 Juridisch kader

In de context van Koppeltaal spelen de volgende juridische concepten, relaties, en regels een rol.

Behandelrelatie. Een behandelrelatie in het kader van de WGBO wordt aangegaan door de GGZ deelnemers van Koppeltaal. De verantwoordelijkheid voor de gegevensverwerking in de context van deze overeenkomst ligt bij de GGZ deelnemers.

De GGZ deelnemers van Koppeltaal hebben contractuele relaties met IT leveranciers die voor hen gegevens verwerken. Deze relatie wordt tevens via een verwerkersovereenkomst geregeld.

GGZ Gebruikers vragen hun IT leveranciers gegevens uit te wisselen via Koppeltaal in de context van de behandelrelatie. IT leveranciers worden hiervoor deelnemer in Koppeltaal en accepteren daartoe de IT deelnemersvoorwaarden. Tevens sluiten ze met Koppeltaal een verwerkersovereenkomst.

Via het privacy beleid van de GGZ deelnemer, en de keten van verwerkersovereenkomsten zoals hierboven beschreven (en de maatregelen die ten

gevolge van die overeenkomsten in de deelnemende organisaties en de Koppeltaal keten worden ingevoerd) voldoet Koppeltaal aan de AVG.

Dit uit zich in de volgende ontwerpbesluiten:

1. Berichten in Koppeltaal worden alleen uitgewisseld tussen geregistreerde applicaties in een domein. Dit domein is analoog aan de zorgaanbieder die de verantwoordelijkheid voor de gegevensuitwisseling neemt.
2. Berichten die via Koppeltaal worden uitgewisseld ten behoeve van gegevensdeling worden direct verwijderd nadat ze succesvol zijn verwerkt door een applicatie. Koppeltaal slaat dus geen gegevens op, maar geeft alleen gegevens door. In 99% van de gevallen, gebeurt de uitwisseling in real-time en worden berichten dus direct verwijderd.
3. Alleen als een applicatie in een domein (tijdelijk) niet online is, blijft een bericht op de server wachten totdat deze applicatie weer online komt. Dat kan bijvoorbeeld gebeuren al een cliënt een poosje niet inlogt op een webapplicatie zonder 'backend' (Alleen "Kickass" maakt van deze structuur gebruik) of een poos 'offline' is op zijn mobiele telefoon (bijvoorbeeld bij de Journey app). Berichten die op deze manier 'wachten' om opgehaald te worden blijven maximaal `MaximumMessageAgeInDays`¹³ dagen staan en worden daarna ook verwijderd. Deze `MaximumMessageAgeInDays` dagen noemen we de *maximum wachttijd ophalen bericht* en deze is tot stand gekomen in overleg met de community van Koppeltaal deelnemers. Het houdt er rekening mee dat een cliënt bijvoorbeeld een aantal weken op vakantie kan zijn en daarna zijn eHealth behandeling wil kunnen hervatten.
4. Koppeltaal berichten vereisen geen verplichte velden. Dat veroorlooft het om de berichten zo in te richten dat cliënten niet geïdentificeerd kunnen worden. Zo kan iedere deelnemer de gegevensdeling minimaliseren in de context van het doel van de uitwisseling. Als het bij het doel past om wel identificeerbare cliëntinformatie te delen, zoals een BSN, dan kan dat overigens wel met Koppeltaal.
5. Cliënten, behandelaars, en derden hebben zelf geen toegang tot Koppeltaal, alleen de applicaties die zij gebruiken maken verbinding met Koppeltaal voor gegevensuitwisseling.
6. Deze verbindingen zijn alleen tot stand te brengen via de Koppeltaal beheerorganisatie. Er is geen directe toegang tot de server voor applicaties.
7. De verbindingen met de server zijn versleuteld middels een servercertificaat.

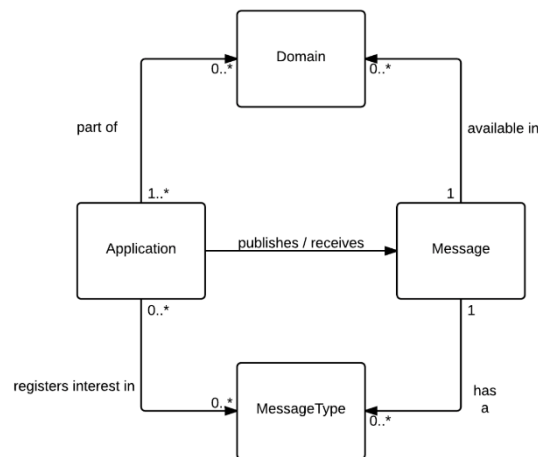
In de documentatie hieronder komen de specificaties van deze uitgangspunten in detail langs. Ze staan hier op een rijtje om overzicht te bieden in de context van het juridische kader.

¹³ `MaximumMessageAgeInDays` is per default 56 dagen en configureerbaar per domein en resource type

3 Interoperabiliteit: Domein en Applicaties

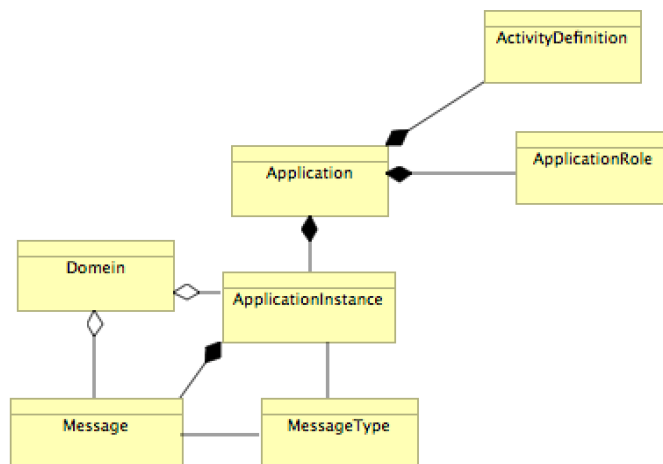
3.1 Context

Het kern context van Koppeltaal is Domein. Een domein representeert een instelling (GGZ) en beperkt de integratiemogelijkheden van de betrokkenen applicaties. Het design van Koppeltaal Applicaties en Koppeltaal Domein wordt gerepresenteerd door het volgende model:



Figuur 6. Registratie op berichttype

Een domein representeert een instelling (GGZ) en beperkt de integratiemogelijkheden van de betrokkenen applicaties. Het design van Koppeltaal Applicaties en Koppeltaal Domein wordt gerepresenteerd door de volgende model:



Figuur 7. Domein en Applicatie model

Er zijn 4 kernconcepten in onze ontologie:

- *Applicatie*
Een partij die via Koppeltaal Server wil communiceren. Applicaties kunnen een *Ro/* en een *ActivityDefinitions* hebben.

- *Applicatie Instance*
Een *Applicatie* die via Koppeltaal Server binnen een specifieke *Domein* wil communiceren.
 - *Domein*
Elke *Applicatie* dat onderdeel van een domein is, wordt *ApplicatieInstance* genoemd. Het domein acteert als een virtuele scheiding voor de het interactie model.
 - *Message*
Een bericht (Message) kan in een domein gepubliceerd worden. Alleen de *Applicaties* binnen dat *Domein* kunnen het bericht zien. Het is niet mogelijk berichten tussen verschillende domeinen uit te wisselen.
 - *MessageType*
Elke bericht heeft een type. Een *Applicatie Instance* zal alleen de berichten kunnen ontvangen van het type waarop de Instance is geabonneerd. De berichten realiseren de functionaliteit van Koppeltaal. In de tabel hieronder staat welke Koppeltaal usecases door welk(e) bericht(en) worden gerealiseerd.
-

Koppeltaal usecase	Koppeltaal bericht type
1. Gebruikers (cliënten, behandelaars, derden) aanmaken of updaten tussen applicaties in een domein.	- CreateOrUpdatePatient - CreateOrUpdatePractitioner - CreateOrUpdateRelatedPerson
2. (Sub)activiteiten publiceren voor gebruik in andere applicaties in het domein	- CreateOrUpdateActivityDefinition
3. (Sub)activiteiten uit applicaties toekennen aan een gebruiker in een andere applicatie dan waar de (Sub)activiteiten zijn opgeslagen	- CreateOrUpdateCarePlan
4. Voortgang- en statusberichten van (sub)activiteiten delen en ontvangen	- UpdateCarePlanActivityStatus
5. Resultaatberichten van (sub)activiteiten delen en ontvangen 6. Het uitgeven van een unieke code voor een mobiele applicatie om deze te kunnen koppelen aan een behandelplan in een andere applicatie.	- CreateOrUpdateCarePlanActivityResult
7. Algemene gebruikersberichten delen en ontvangen	- CreateOrUpdateUserMessage

Tabel 3. Usecase - Koppeltaalbericht

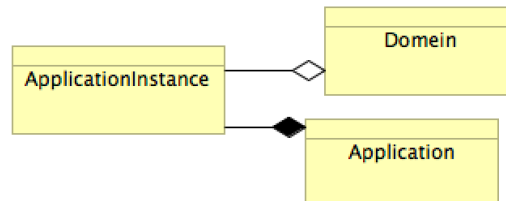
Met deze set berichten kan je de functionaliteit van Koppeltaal realiseren tussen verschillende applicaties. In de praktijk zien we eHealth platforms, cliëntportalen, 'losse' eHealthmodules, behandelaar platforms, EPD's, en ROM-systemen in Koppeltaal gecombineerd worden. Als een applicatie het bericht ondersteunt kan het verwerkt worden.

NB: de single-sign-on functionaliteit wordt niet als bericht afgehandeld. De specificatie daarvan staat hieronder nader weergegeven.

3.2 Koppeltaal componenten

3.2.1 Koppeltaal Domeinen

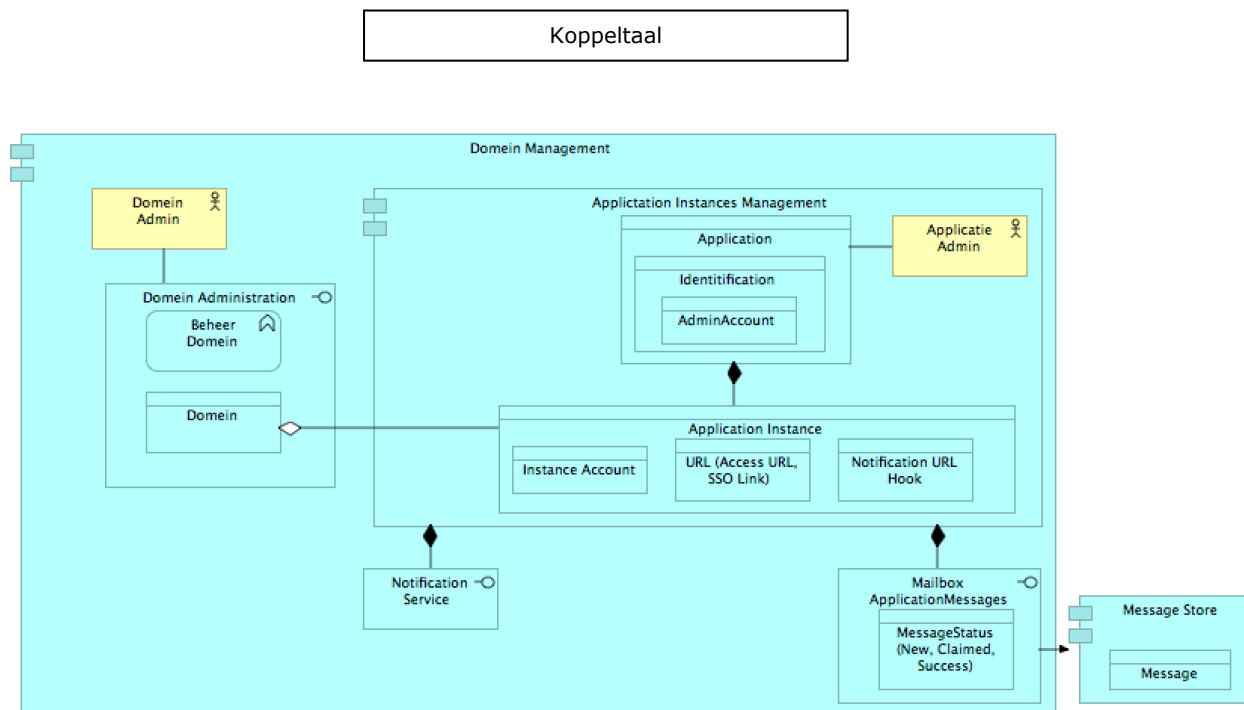
Een domein is gerepresenteerd als een virtuele scheiding van informatiesets. Het is nodig om het functionele model te kunnen ondersteunen en berichten te kunnen isoleren binnen een virtueel domein. Een GGZ-instelling heeft toegang tot één of meerdere eigen domeinen. Deze domeinen bevatten een set van Applicatie Instances die met elkaar kunnen communiceren.



Figuur 8. Applicatie-instantie datamodel

De beschikbare processen ter ondersteuning van de domein functies zijn:

- Een Domein is onderdeel van het Koppeltaal applicatieregister dat onder beheer staat van de Koppeltaal Administrator
- De zorgaanbieder als domein-eigenaar vraagt bij de Koppeltaal administrator een domein aan.
- De Koppeltaal administrator maakt een nieuw domein aan in het Koppeltaal Applicatieregister, en voegt namens de zorgaanbieder een Domein Administrator toe.
- Een *Domein* heeft een of meer **Domein Administrators**
- Een **Domein Administrator** kan Applicaties bij de *Domein* betrekken als *ApplicatieInstances*
- Een applicatie wordt beheerd door een **Applicatie Administrator**.
- Een *Applicatie* kan onderdeel zijn van een *Domein* als een *ApplicatieInstance*
- Een *ApplicatieInstance* kan verschillende configuraties hebben in verschillende domeinen (zie SSO en Notificatie WebHook als voorbeeld)



Figuur 9. Koppeltaal Domein Management

De aangegeven in Figuur 9: NotificatieService, KoppeltaalMailbox en MessageStore componenten zijn Koppeltaal breed en gefilterd per Domein. Elke van deze drie componenten hebben functies die verder per Domein te specificeren zijn.

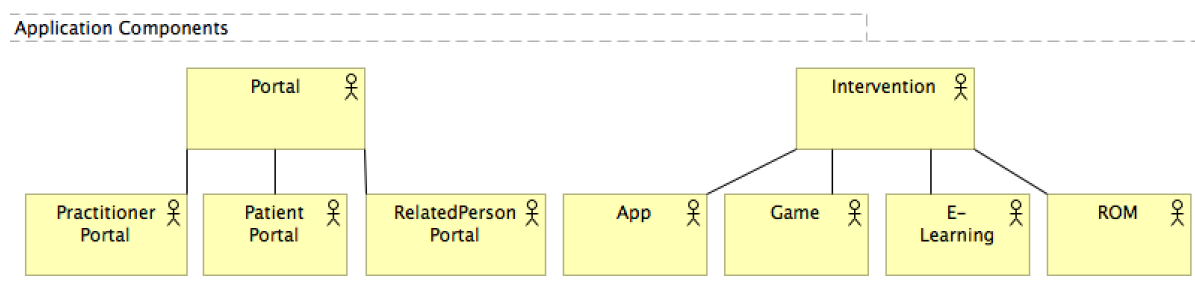
Applicatie Instance: wordt gekoppeld aan een domein via een Account (unieke gebruikersnaam en wachtwoord per instance/domein combinatie).

Applicatie Mailbox: de Berichten (Messages) bestemd voor deze applicatie instantie.

3.2.2 Applicatie model

Er zijn verschillende typen applicaties betrokken ter ondersteuning van een interactief zorgproces.

Functioneel zijn alle applicatie onderdeel van de gehele Koppeltaal omgeving en ondersteunen het interactieproces vanuit verschillende rollen voor de gebruikers. Per domein sluit een applicatie aan met een unieke applicatie-instantie, waarin die rollen binnen dat domein zijn gedefinieerd (zie ook paragraaf x domeinen) Voor elke applicatiecomponent (Portal of Interventie) zijn specifieke rollen te koppelen:



Figuur 10. Applicatie typen

Portalen aan de kant van de Zorgverlener:

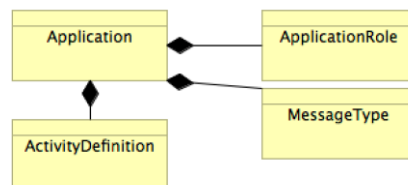
- Behandelaar/Careteam portaal

Portalen aan de kant van de klant in het Zorgproces:

- Patiënten portaal
- Derden portaal

Interventies: zelfstandige applicaties die van een behandelplan deel kunnen worden gemaakt. Deze activiteiten aangeboden door applicaties zijn te integreren in andere applicaties (Portalen) als onderdeel van een blended behandelplan (bijvoorbeeld als SSO remote links).

Een applicatie is gerepresenteerd door een partij die via Koppeltaal Server wil communiceren. Applicaties kunnen één of meerdere rollen en een ActivityDefinitie hebben.



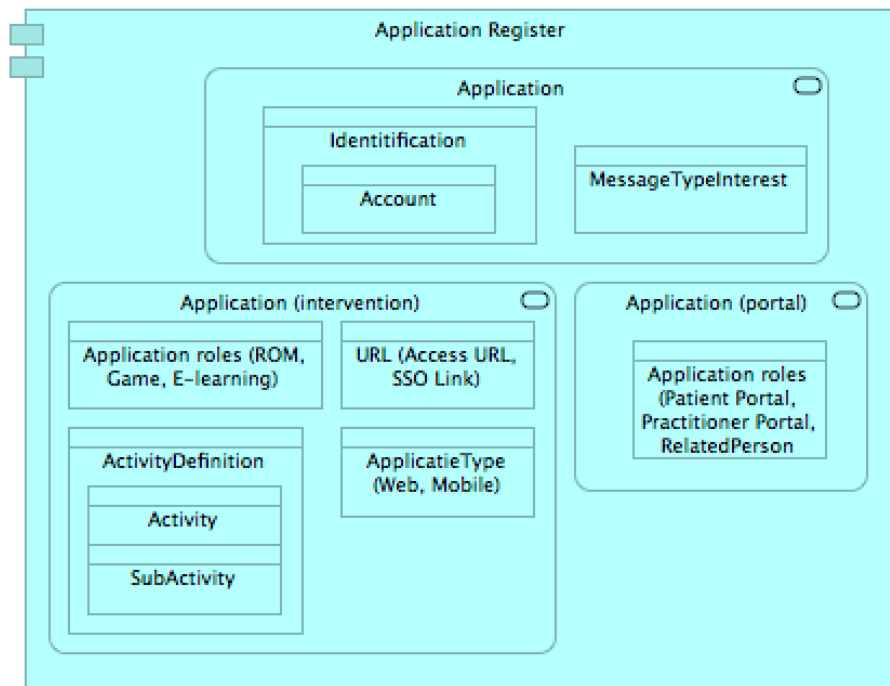
Figuur 11. Applicatie Domein model

Applicaties worden geregistreerd binnen Koppeltaal na afloop van de VZVZ Certificatie Procedure. Deze procedure is onderdeel van het testproces.

Afhankelijk van de Applicatie Rol zijn er verschillende configuraties nodig om een applicatie binnen Koppeltaal te kunnen registreren (zie figuur 10):

1. Portalen: Het Applicatie Rol Portal representeert de omgeving van een gebruiker. Portalen kunnen, naast een normale login, ook opgestart worden via een SSO-link en als ze nog niet in de activiteiten register staan, daar in opgenomen worden.
Soorten portalen: Behandelaar portaal, Patiënten portaal, Derden portaal
2. Interventies: Specifieke applicaties als onderdeel van het [Applicatie Register](#). Applicaties hebben: URL Access, SSO link en Activity Definities. Het applicatie register wordt verder uitgewerkt in paragraaf 6.2.

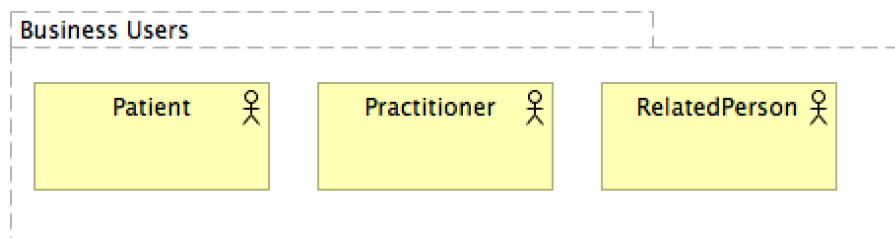
Koppeltaal Subscriptie kan worden gedaan via het MessageType: Interest. Elke applicatie definieert een set van Type Interests waarop een andere applicatie zich kan abonneren.



Figuur 12. Applicatieregister

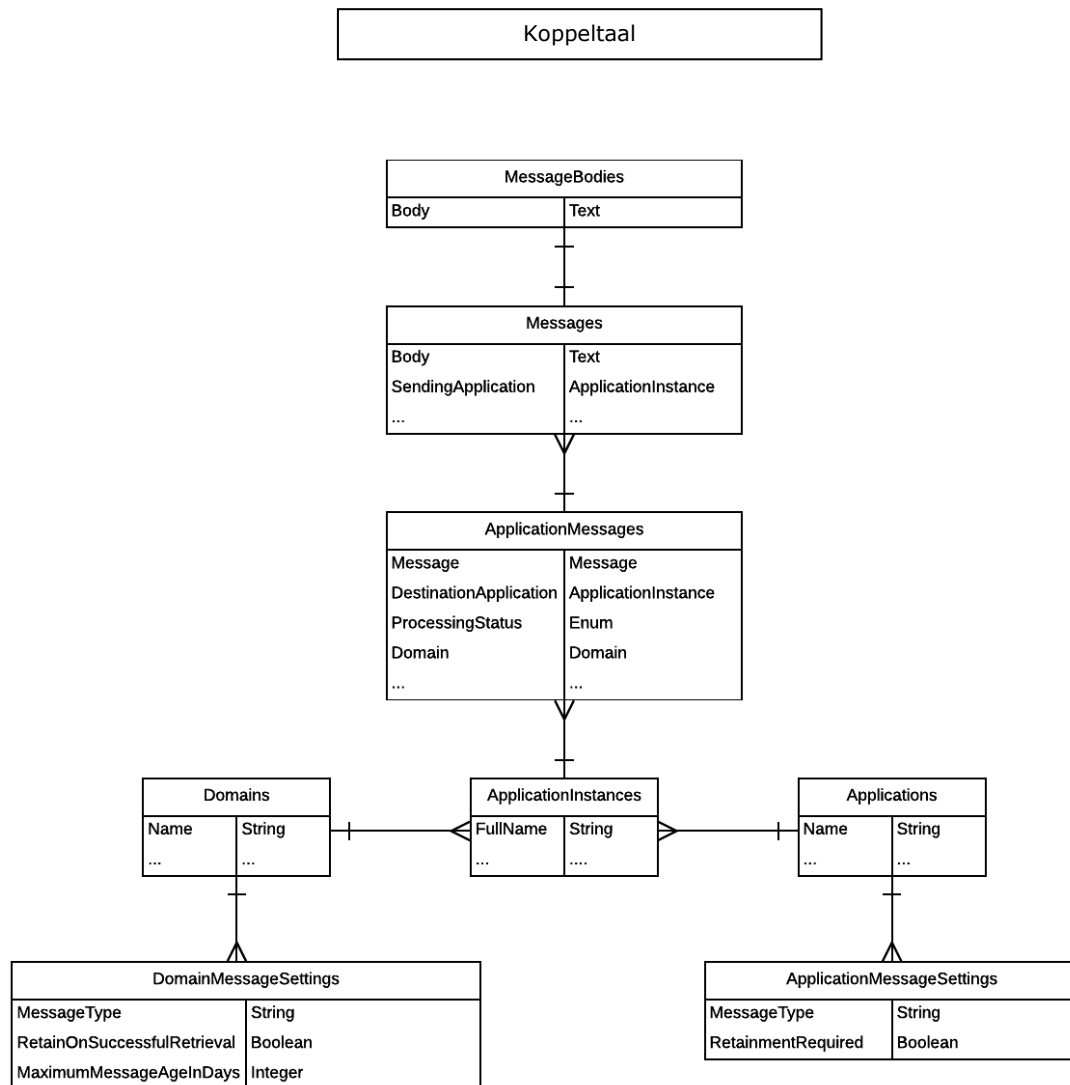
Rollen voor het applicatie model:

- Patiënt
- Behandelaar (Practitioner)
- Careteam (Medebehandelaars)
- Naasten (RelatedPerson)



Figuur 13. Koppeltaal_gebruikers

Elke bericht dat door de Koppeltaal Server wordt gerouteerd, zal intern opgeslagen worden. Binnen de interne datastore, de Koppeltaal Message Body zal apart gepersisteerd worden t.a.v. de Message Header (metadata). De Message Body is opgeslagen in de MessageBodies tabel en de metadata in de Messages Tabel.



Figuur 14. Koppeltaal database tabelstructuur

Voor elke bericht wordt een apart record aangemaakt in de ApplicatieMessage Tabel. Deze tabel bevat ook de processing status (gelezen of niet) als een directe indicatie voor de relatie tussen de applicatie en een bericht: de applicatie outbox (status van de messages binnen de Out-queue).

De interne routing is gebaseerd op referentie-id. Een Applicatie 'queue' wordt gerepresenteerd door een set referenties naar de bron (het bericht) met een gekoppelde status. Er is geen sprake van duplicatie van de berichten (geen additionele load).

Als de processing status verandert, dan gelden de volgende acties:

- als er minstens één geabonneerde applicatie is dat dit bericht nog niet met succes heeft gelezen, dan niets doen.
- als de record DomainMessageSettings voor dit Domein aangezet is en het bericht type met 'RetainOnSuccessfulRetrieval' is 'true', dan niets doen; anders: verwijder MessageBody en mark alle gerelateerde ApplicationMessagesMessages als processingstatus='expired'

Dit proces is actief als gevolg van het beleid een niet opgehaald bericht na uiterlijk `MaximumMessageAgeInDays` dagen te verwijderen.

- als de record `DomainMessageSettings` voor dit Domein en bericht type is aangezet en het bericht zelf is ouder dan de gespecificeerde max tijd, dan verwijder `MessageBody` en markeer alle gerelateerde `ApplicationMessages` als 'expired'

Er wordt in dit geval een extra attribuut toegevoegd aan de `MessageHeader` als een indicatie dat het bericht is verlopen.

4 Informatiemodel

4.1 Architectuur in het kort

Om de interacties tussen zorgverleners onderling en tussen patiënt en zorgverlener te ondersteunen, is een reeks van interfaces gedefinieerd op de Koppeltaal Server. Deze interfaces zijn beschikbaar voor EPD, ROM, EHealth interventies en Portalen. Voor berichtuitwisseling tussen voor eHealth interventies en Portalen wordt gebruik gemaakt van berichten die gebaseerd zijn op de internationale medische berichtenstandaard HL7 FHIR. FHIR-berichten (FHIR Messaging concept) worden verstuurd over HTTPS. eHealth interventies en Portalen worden gekoppeld met Koppeltaal via openbaar Internet op basis van TCP/IP.

4.2 Uitgangspunten

Elke bericht zal self-contained zijn; er worden geen referenties gemaakt naar externe resources: een applicatie zal geen resources nodig moeten hebben om zijn eigen informatie up-to-date te kunnen houden. Tegelijkertijd, helpt dit om:

- niet alle informatie te persisteren
- gebruik te maken van externe links in een complexe architectuur.

Volgens de self-containing principe, zijn binnen een Bundle van een bericht alle benodigde Resources toegevoegd: de versturende applicatie is eigenaar van alle Resources. De andere resources (bijv. `ActivityDefinition` voor elke activiteit binnen een `CreateOrUpdateCarePlan`) kunnen via een URL verwezen worden (Zie [FHIR - basic rules](#)).

4.3 Bericht – Conceptueel Model

FHIR - basisregels

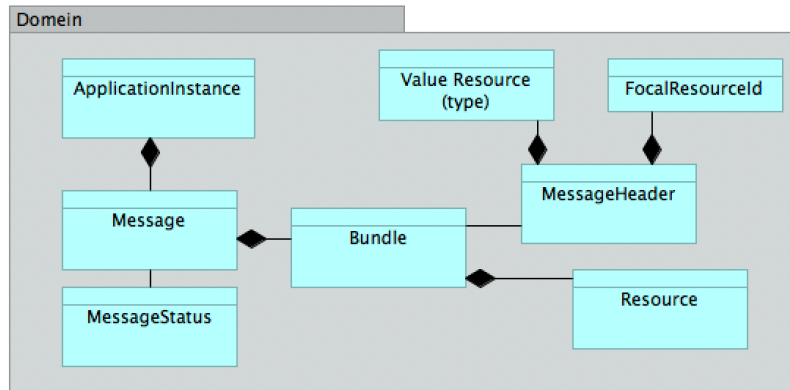
Aan elke applicatie wordt een unieke FHIR basis URL toegewezen, welke de basis vormt voor de aanmaak van resource URLs, bijvoorbeeld `"https://xxx.eengeregistreerdnaam.nl/fhir"`.

Elke Resource dat aan een bericht wordt toegevoegd moet een unieke URL hebben die er als volgt uitziet:

FHIR Basis Url + "/" + Resource Type + "/" + id + "/_history/" + Versie Id

Koppeltaal gebruikt de FHIR Messaging protocol om de routing van de berichten te ondersteunen. Het Message Format heeft twee kern onderdelen:

- [FHIR \(Message\) Bundle](#): een groep van Resources met een [FHIR MessageHeader](#).
- Value Resource/Type: het type van het bericht.
- Focal Resource – de kernresource van het bericht



Figuur 15. Bericht samenstelling

4.4 Resource versionering

Focal resource

In het MessageHeader.data element wordt gerefereerd naar de focal resource van het bericht, oftewel naar de root van het bericht. Deze referentie is geversioneerd zoals in het plaatje hieronder te zien is:

```

<data>
  <reference value="https://koppeltaal.ggz.nl/CarePlan/1234/_history/2018-04-03T11:41:26:178.1210" />
</data>

```

Deze is uiteraard in de Create niet geversioneerd, maar in de Update van het bericht wel.

Overige resources

Naast het Focal Resource is elke resource geversioneerd. De referentie naar die resource is niet geversioneerd. Dit is voor de referenties naar focal resource altijd al het uitgangspunt geweest, omdat het voor de Koppeltaal Server mogelijk moet zijn om de bundle 'as is' op te slaan. Deze referentie zou dan niet meer geldig zijn zodra deze resource een nieuwe versie krijgt.

Ter illustratie: dit geldt in het CarePlan voor de volgende elementen:

- Patient -> Patient resource
- Participant.member -> RelatedPerson|Practitioner
- Activity.participant.member -> RelatedPerson|Practitioner
- Activity.simple.performer RelatedPerson|Practitioner|Patient
- Relation.reference -> Any

Een voorbeeld van resource versioning is uitgewerkt in Appendix: Voorbeeld resource versioning

Een algemeen voorbeeld van versionering:

"https://xxx.jouwomgeving.nl/fhir/Patient/"32324"/_history/"812909

Client stuurt message1 naar Server -> Alleen met focalresource

Server stuurt in reply een versie terug, voorbeeld:

www.emhp.nl/FHIR/CarePlan/1/_history/1

Client stuurt update van message1 naar server -> met laatst ontvangen version:

www.emhp.nl/FHIR/CarePlan/1/_history/1

Server stuurt reply -> www.emhp.nl/FHIR/CarePlan/1/_history/2

Client mag dus nooit een version wijzigen, Koppeltaal Server deelt ze uit.

Versionering verloopt via een datetimestamp.

4.5 Response

Zoals genoemd in punt 4, worden als response op een Mailbox POST in versie 1.3.5 alle uitgekeerde versies teruggegeven.

Voor elke resource wordt een data-element teruggegeven in de MessageHeader. In 1.3.3 was dit alleen het geval voor de focal resource.

Als er aan een 1.3.5 compatibele applicatie een 409 wordt teruggegeven omdat een of meer verkeerde versies zijn meegestuurd, worden hierin de resources teruggegeven waarvan de verkeerde versie was meegestuurd.

5 Zie voor voorbeelden van deze responses Appendix: Voorbeeld resource versioning

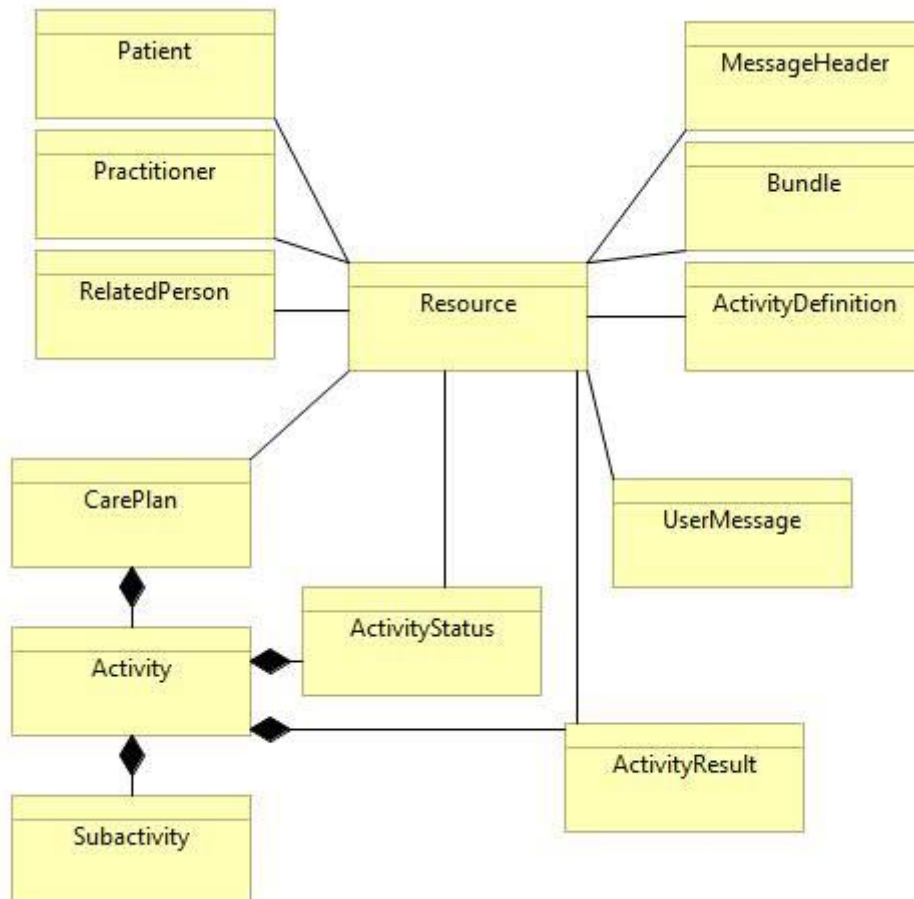
```
<entry>
  <link rel="self" href="http://co.healthonline.nl/Patient/215325/_history/2018-07-11T08:01:22:181.2659" />
  <content type="text/xml">
    <Patient id="215325" xmlns="http://hl7.org/fhir">
      <extension url="http://ggz.koppeltaal.nl/fhir/Koppeltaal/Patient#Age">
        <valueInteger value="44" />
      </extension>
      <name>
        <use value="official" />
        <family value="Todea" />
        <given value="Reli" />
      </name>
      <gender>
        <coding>
          <system value="http://hl7.org/fhir/vs/administrative-gender" />
          <code value="M" />
          <display value="Male" />
        </coding>
      </gender>
      <birthDate value="1972-02-28T00:00:00+01:00" />
    </Patient>
```

</content>
</entry>

Appendix: Response voorbeelden

5.1 FHIR Resources

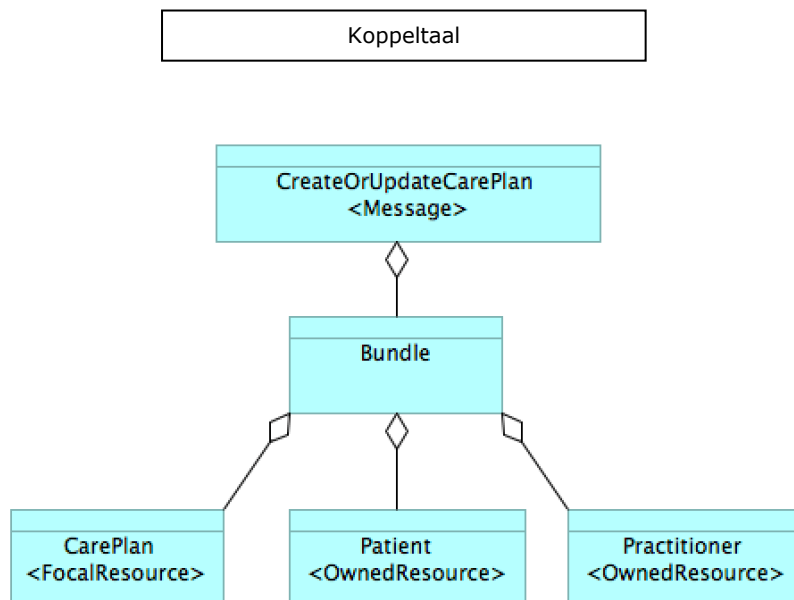
De functionele FHIR Resources binnen Koppeltaal zijn (zie Figuur 16):



Figuur 16. Koppeltaal FHIR Resources

Als een voorbeeld van een Koppeltaal bericht met Owned resources (zelf contained), CreateOrUpdateCarePlan heeft een Bundle met drie resources:

- CarePlan Resources als de Focal Resource
- Patient Resource – de patiënt betrokken bij het CarePlan en gedefinieerd binnen de versturende applicatie (owned resource)
- Practitioner Resource – behandelaar betrokken bij dit CarePlan (owned resource)



Figuur 17. Koppeltaal Bundle

De bekende resources met hun eigen identifiers als gedefinieerd door Koppeltaal internal FHIR Model (zie Tabel 4.):

Resource	Profile Identifiers
ActivityDefinition	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition
Device	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/Application
MessageHeader	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/MessageHeader
Patient	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/Patient
Practitioner	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/Practitioner
RelatedPerson	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/RelatedPerson
Organization	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/Organization
Person	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/Person
CarePlan	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CarePlan
CarePlanActivityStatus	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CarePlanActivityStatus
CarePlanActivityResult	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CarePlanActivityResult
UserMessage	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/UserMessage
CareTeam	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CareTeam

Tabel 4. Resource Type en Profile Identifiers

De informatie-uitwisseling wordt gerealiseerd via FHIR berichten met daarin informatie die is opgenomen in voor gedefinieerde, herbruikbare bouwstenen (FHIR Resources). Bij het mappen van Koppeltaal informatieconcepten (binnenkort volgens de zorginformatiebouwstenen, ZIBs) naar deze FHIR Resources worden FHIR extensies (maatwerk) zo veel mogelijk vermeden. Voor een complete set van Resources verwijzen wij naar de volgende tabel (zie Tabel 5):

Koppeltaal entiteit	FHIR resource
Activity	Other(CarePlanActivity)
ActivityProvider	Is niet mapped. Elke applicatie kan activiteiten aanbieden. Deze zijn gespecificeerd via ActivityDefinitions.
Applicatie	Device
Caregiver/Behandelaar	Practitioner
Carepath/Behandel Plan	CarePlan
Client	Patient
Intervention	Is niet mapped. Elke interventie is een type van activiteit.

Koppeltaal entiteit	FHIR resource
Person	Is niet mapped. Informatie over persoon is gegroepeerd binnen een specifieke rol: Patient, Practitioner or RelatedPerson.
Portal	Is niet mapped. Een portal is gezien als een applicatie.
Related person	RelatedPerson
Result	DiagnosticReport (met extensies)
Role	Is niet mapped. Persoon rol is gelinkt met de gebruikte resource type. I.e., een persoon in de rol van behandelaar zal gezien worden als type Practitioner.
ROM	Is niet mapped. Een ROM is een type activiteit.
Screeener	Is niet mapped. Een screener is een type van activiteit.
Treatment	Is niet mapped. Het behandel traject is gelinkt met de CarePlans gerelateerd aan een specifieke patiënt.

Tabel 5. Koppeltaal ZIB model

6 Interactie tussen Informatiesystemen

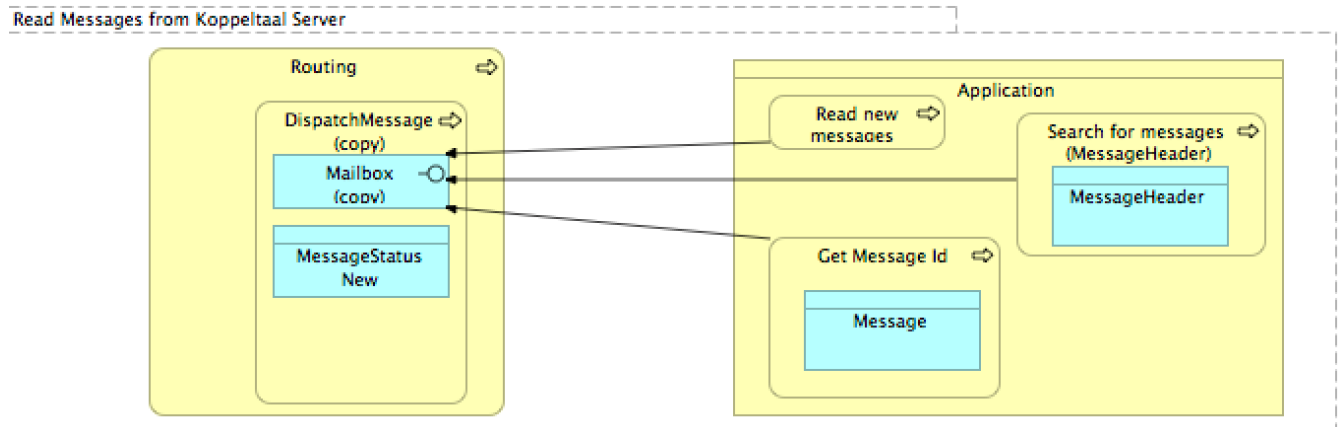
Binnen deze sectie zullen wij kijken naar de mogelijke interacties tussen informatiesystemen. De primaire functies worden hieronder in detail uitgewerkt. De functies die aan de orde komen zijn:

- [Routing](#) (zie 6.2): Post een bericht binnen een Domein. De geabonneerde applicaties zullen dit bericht kunnen zoeken en lezen in hun Applicatie Mailbox.
- [Zoeken](#) (zie 6.1) naar een bericht binnen de Applicatie Mailbox
- [Lezen/Ophalen](#) (zie 6.3) van een specifiek bericht
- [Notificaties](#) (zie 6.4) als optie, het is mogelijk om een notificatie te krijgen van een nieuw bericht in de Applicatie Mailbox
- [SSO](#) (zie 6.5): OAuth flow ter ondersteuning van de LaunchSequence
- [Proxy Storage Service](#) (zie 6.7) (netwerkcomponent): Werkt als een netwerk switch en maakt het mogelijk om BLOB's te persisteren. Koppeltaal heeft een proxy functie (dit is een synchrone activiteit en is geen lange termijn storage op de Server) en niet een Storage functie.

Koppeltaal conceptuele interactie model (zie [Storage model](#) voor details):

- 1) Alle interacties worden onder een Applicatie account gedaan (en daardoor binnen een domein)
- 2) Binnenkomende berichten worden in de inkomende queue geplaatst
- 3) Het bericht wordt opgeslagen op de server in de database
- 4) Gestuurde berichten wordt toegewezen aan de juiste verzendqueue.
- 5) Wanneer gelezen: de status van het bericht op de server wordt bijgewerkt.
- 6) Het bericht wordt verwijderd uit de binnenkomende queue.

6.1 Informatie zoeken en lezen



Figuur 18. Informatie zoeken en lezen

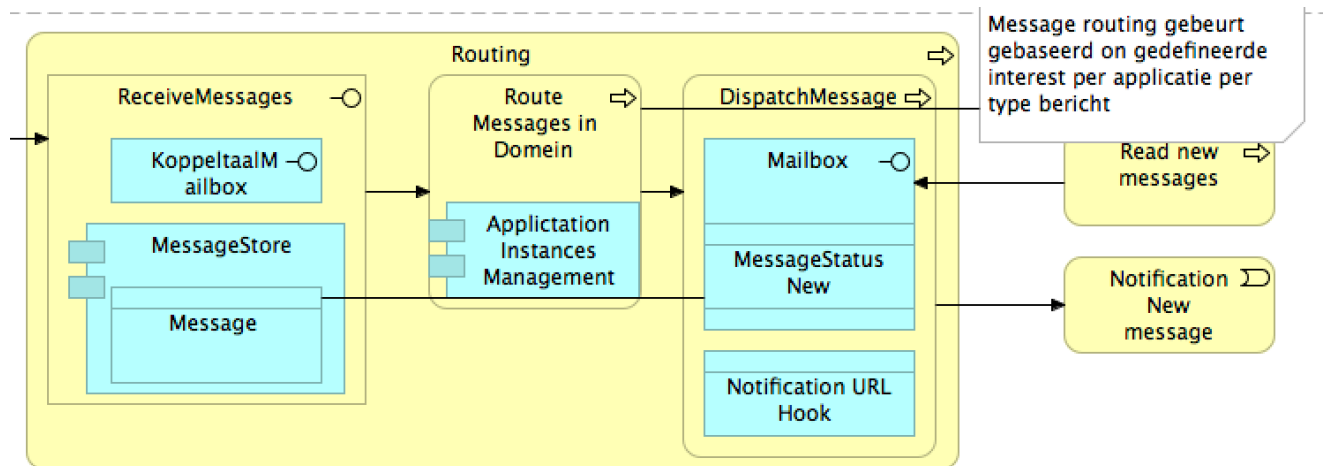
Er zijn 3 mogelijke interacties:

1. https://koppeltaal.nl/FHIR/Koppeltaal/MessageHeader/_search?_query=MessageHeader.GetNextNewAndClaim - zoeken naar berichten met ProcessingStatus= "New", maak ProcessingStatus "Claimed", en stuur een Bundle voor dat specifieke bericht terug. Deze call zal altijd het gevolg moeten zijn van een update van de Message status. Deze interactie geeft tot gevolg dat de berichtstatus wordt aangepast.
2. [https://koppeltaal.nl/FHIR/Koppeltaal/MessageHeader/_search?_count=\[X\]](https://koppeltaal.nl/FHIR/Koppeltaal/MessageHeader/_search?_count=[X]) - deze stuurt een Bundle van MessageHeaders terug om de applicaties te laten zoeken naar een of meerdere specifieke berichten. Een pagesize kan doorgegeven worden met de _count parameter, met een max van 1000.
3. [https://koppeltaal.nl/FHIR/Koppeltaal/MessageHeader/_search?_id=\[id\]](https://koppeltaal.nl/FHIR/Koppeltaal/MessageHeader/_search?_id=[id]) - deze kan gebruikt worden om een complete Bundle voor een specifiek bericht te krijgen (bijv. Als de MessageHeader bekend was door de voorgaande zoek actie)

Het volgende additionele query parameters kunnen gespecificeerd worden:

- Patient: Filtert op de patiënt dossier waar het bericht aan gerelateerd is.
- Event: Filtert op de bericht type
- ProcessingStatus: Filtert op de ProcessingStatus. (New|Claimed|Success|Failed). Deze query parameter kan geen onderdeel zijn van de named query van interactie 1.

6.2 Routing



Figuur 19. Routing

Er zijn vier kern sub-processen:

- bericht sturen
- bericht routeren
- bericht ophalen
- notificaties

6.2.1 Berichten versturen

1. De applicaties gebruiken HTTP calls om berichten te sturen: het POST van de bericht bundle zal naar de kern Koppeltaal inbox gaan:

<https://koppeltaal.nl/FHIR/Koppeltaal/Mailbox>.

2. De structuur van het bericht is gedefinieerd zoals in [Bericht Model](#)

Elk bericht dat gestuurd is via Koppeltaal heeft een versie. Elke andere applicatie dat hetzelfde type bericht stuurt zal een subscriptie voor dit type resource moeten hebben om de versie van de interne resource 'up to date' te houden. Deze actie is ondersteund door de interne structuur zoals beschrijven in het [Storage model](#).

Elk bericht dat gestuurd is, moet de laatste bekende versie hebben. Koppeltaal zal deze versie checken met de laatste versie en het bericht accepteren als dat gelijk is aan dit laatste versienummer.

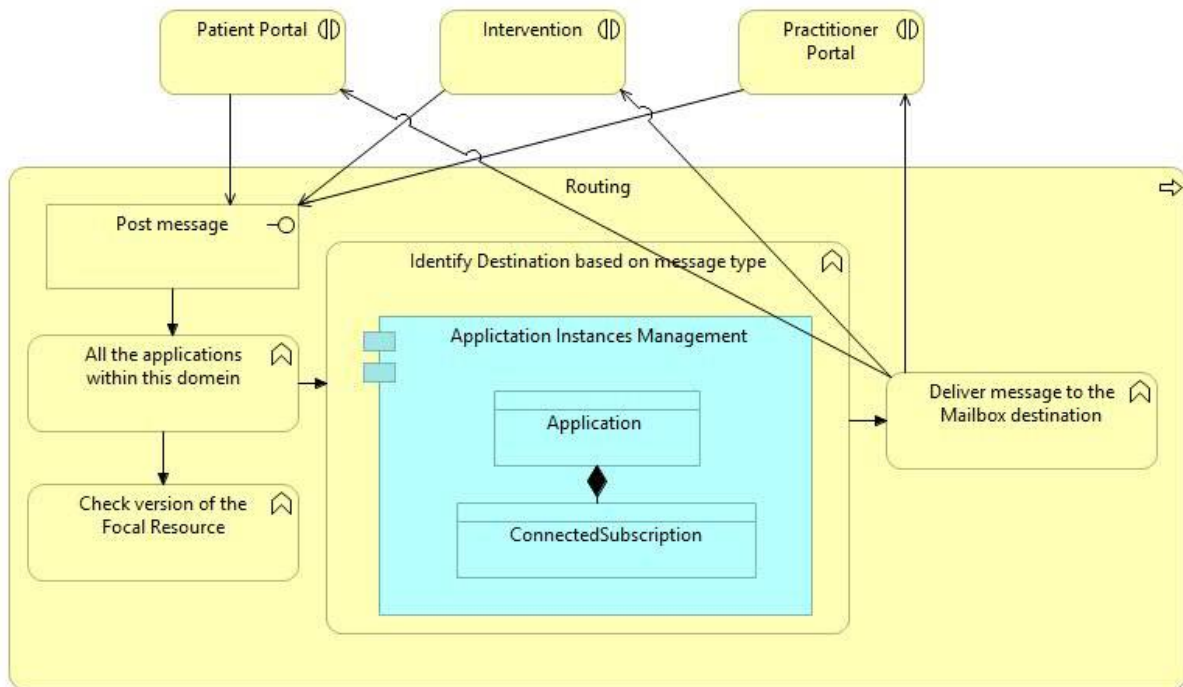
Lezen van de huidige versie:

Er zijn twee mogelijkheden om de laatste versie van de bericht te krijgen:

1. Wanneer een bericht is gestuurd naar de Koppeltaal Server zal de Koppeltaal Server de gegenereerde versie terugsturen als onderdeel van het antwoord.
2. Als er meerdere applicaties een bericht kan sturen met betrekking dezelfde focal resource, zal elk van hen moeten registreren om de updates te krijgen (samen met de versie nummer). Het is nodig te herinneren dat in Koppeltaal Subscriptions zijn gemaakt per Type Bericht. Een voorbeeld van bericht type is *CreateOrUodateCareplan*. Alle geregistreerde applicaties binnen het domein die zijn geregistreerd op de *CreateOrUodateCareplan* zullen dit bericht in hun Mailbox zien met de laatste versie nummer.

1.1.1 Berichten Routeren

Het routeren Berichten gebeurt altijd binnen een Domein. Een valide binnengekomen bericht wordt dan aan alle applicaties opgeleverd waar een subscriptie voor is. De subscripties zijn deel van de applicatie instance configuratie en zijn gespecificeerd per type bericht (e.g. CreateOrUpdateCarePlan, UpdateActivityStatus, etc).



Figuur 20. Bericht Routing

Het complete proces heeft de volgende:

- Een bericht is binnen gekomen als een valide bericht onder een applicatie account
- Dat bericht is nu onderdeel van een domein
- De Focal Resource van dit bericht is onderdeel van de MessageHeader.

In Koppeltaal, alle berichten zijn gerelateerd aan het maken of updaten van een resource. Het zou kunnen dat, op een bepaald moment, meerdere applicaties een kopie van hetzelfde resource hebben. Dit is de reden dat de versies belangrijk zijn: om het concurrency probleem op te lossen (updaten van een oude bericht), het is nodig om elke applicatie de versie van bericht te kennen.

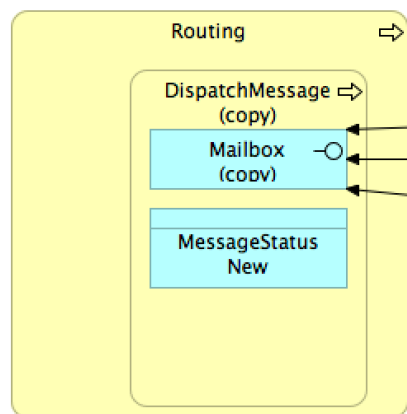
Koppeltaal zal het versienummer van een bericht updaten, wanneer een nieuwe versie van het bericht beschikbaar is. Dat resulteert in het overschrijven van het oude bericht. De applicatie ontvangt altijd de laatste versie van een bericht.

Bijvoorbeeld, het is niet een garantie dat een Client Portaal(applicatie) alle status updates heeft kunnen lezen voor een activiteit, als er een nieuwe statusupdate wordt gepost. Als van een activiteit de status update van "available" naar "in progress", of van "in progress" naar "finished" gaat, dan kan het gebeuren dat één van de status Updates niet gelezen is door de cliënt portaal als een (vervolg)activiteit wordt gestart vóór het moment dat het portaal de mailbox heeft gelezen.

Stel bijvoorbeeld dat een cliënt op een mobiele app werkt aan een behandelplan. Hij is twee weken op vakantie. Zijn behandelaar besluit een activiteit aan het zorgplan toe te voegen. Het EPD stuurt een bericht naar de app van de cliënt met de update van het behandelplan. Een week later besluit de behandelaar nog een activiteit aan het behandelplan toe te voegen. De cliënt is nog steeds op vakantie. Het EPD stuurt een nieuw bericht met de meest recente versie van het behandelplan, met daarin de twee toegevoegde activiteiten. Op het moment dat de cliënt terugkomt van vakantie leest zijn app alleen de meest recente versie van het bericht. De tussentijdse versie waarin één nieuwe activiteit was toegevoegd, hoeft niet gelezen te worden voor Koppeltaal om te werken (wegens het 'self-contained' principe van berichten) en kan ook niet gelezen worden, omdat alleen de meest recente versie van het bericht wordt bewaard in de queue van de app.

6.3 Berichten Ophalen

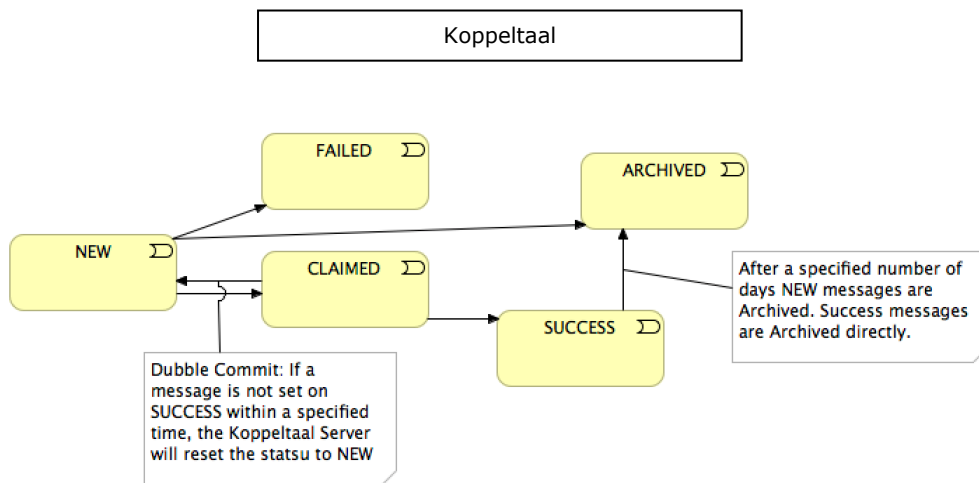
Berichten worden opgehaald van een specifieke mailbox.



Figuur 21. Ophalen berichten

De berichten processing status workflow heeft 6 kern statussen:

- NEW: een ongelezen bericht: een nieuw of een claimed bericht dat niet als SUCCESS is gemeld.
- CLAIMED: een geclaimd bericht: een tijdelijke status want een CLAIMED bericht kan als SUCCESS geboekt worden als het compleet gelezen is of als NEW, wanneer een time-out is opgetreden. (Double-commit strategie)
- SUCCESS: een gelezen bericht
- ARCHIVED: oude NEW berichten (na afloop van de bewaartermijn) of SUCCESS berichten.
- FAILED: een bericht dat een structurele error heeft.
- MaximumRetriesExceeded: na Maximum retries (5)



Figuur 22. Bericht status

Als een bericht met succes opgehaald is, moet de betrokken applicatie de flow afsluiten met een update van de status naar SUCCESS. (de applicatie geeft ter afsluiting van het ophalen van een bericht een bericht:SUCCESS terug). Als de applicatie dat niet doet (na een bepaalde tijd) zet de KTS de status van CLAIMED weer naar NEW.

API URL: [https://koppeltaal.nl/FHIR/Koppeltaal/MessageHeader/\[id\]](https://koppeltaal.nl/FHIR/Koppeltaal/MessageHeader/[id]) (dit is de URL geretourneerd als id link in het MessageHeader bundle)

Als er een error voor komt, wordt de berichtstatus terug naar NEW gezet om verder gelezen te kunnen worden. Als er structurele error optreedt (e.g. verkeerd berichtcontent), zal de ProcessingStatus aangepast worden naar "Failed" samen met een duidelijke beschrijving van de exceptie.

6.4 Notificaties

De Koppeltaal Server biedt een functie om notificaties te versturen als er een nieuw bericht beschikbaar is voor een applicatie binnen een domein. In specifieke gevallen kan je met deze notificatie de performance verbeteren, omdat er dan alleen 'gepulld' hoeft te worden als er daadwerkelijk een bericht is voor de applicatie. Het is aan applicatiebeheerders om te kiezen of ze hiervan gebruik willen maken. Deze notificatie is geïmplementeerd middels REST WebHooks. Om een notificatie te kunnen ontvangen zijn de volgende configuratie acties nodig:

- Een WebHook (http/https) definiëren die Koppeltaal kan aanroepen.
- De lokale implementatie achter de WebHook URL is nodig om notificaties te kunnen interpreteren. Koppeltaal zal een event genereren ter informatie dat er nieuwe berichten beschikbaar zijn. De betreffende app wordt 5 keer 'gecalld' door de notificatie. De applicatie kan daarna het bericht lezen zoals al beschreven in [Berichten Ophalen](#).

6.5 Autorisatie flow – SSO

Koppeltaal ondersteunt twee typen van Single Sign-on flows, met Koppeltaal Server als OAuth Server en de Applicatie als OAuth Client:

- Web Launch, voor het starten van webapplicaties (zoals een eHealthmodule, of een ROM lijst).

- Mobile Launch, voor het starten van een app op een smartphone.

Beide flows in **Fout! Verwijzingsbron niet gevonden.** en **Fout! Verwijzingsbron niet gevonden.** in detail uitgewerkt.

Logical SSO flow

Na het aanmaken van een CarePlan door de behandelaar, klikt de patiënt op een knop in de Portal om een applicatie te starten. De patiënt heeft zowel een Portal als een Applicatie(Game) tot zijn beschikking.

Patiënt gebruikt een knop om de Applicatie (Game) te starten. Zijn patiënt portal zal een URL openen via de KoppeltaalServer. De volgende informatie wordt beschikbaar gemaakt binnen de URL:

1. Application Identifier (gekoppeld aan een specifieke applicatie) – noodzakelijk om de Koppeltaalserver de publicerende applicatie en zijn URL op te kunnen zoeken.
2. CarePlanActivity identifier – gebruikt om te achterhalen welke activiteiten een bepaalde applicatie (Game) herkent.
3. Patient identifier – wordt gebruikt om de patiënt te identificeren door de applicatie (Game).
4. User identifier – wordt gebruikt door de applicatie om correcte views te laten zien voor die gebruiker
5. Optioneel aanvullende applicatie informatie – bijv. dat er een specifieke pagina door de applicatie moet worden geopend.
6. Een tussen de Koppeltaal server en de applicatie afgesproken karaktercombinatie (security token) – de applicatie moet een security token naar de KTS sturen om zo te laten weten dat de applicatie betrouwbaar is en bevoegd is te communiceren met de Koppeltaalserver in de context van domein en applicatie instance. Deze karaktercombinatie bevat ten minste een Hash van de URL van de applicatie en voorkomt dat iemand de URL van de applicatie (het adres) aanpast en opnieuw indient bij de KTS. Tevens bevat de karaktercombinatie een geheime code die zowel bij de applicatie als de KTS bekend is en, mogelijk, een Nonce die voorkomt dat de URL meerder malen (sessies, resource) gebruikt kan worden.

6.6 OAuth2 Specs, Web en Mobile launch

6.6.1 OAuth2 Specs gedistribueerde authenticatie

Koppeltaal Single Sign-On is gebaseerd op de OAuth2 standaard en geïmplementeerd volgens de SMART-on-FHIR voorschriften (zie <http://www.hl7.org/fhir/smart-app-launch/>). Volgens OAuth2 kunnen twee typen Clients worden onderscheiden:

- "Public Client"

Een Public Client draait volledig op een eindgebruiker apparaat. Gevolg is dat de applicatie geen "cliënt secret" kan beschermen in het geval dat er ook geen

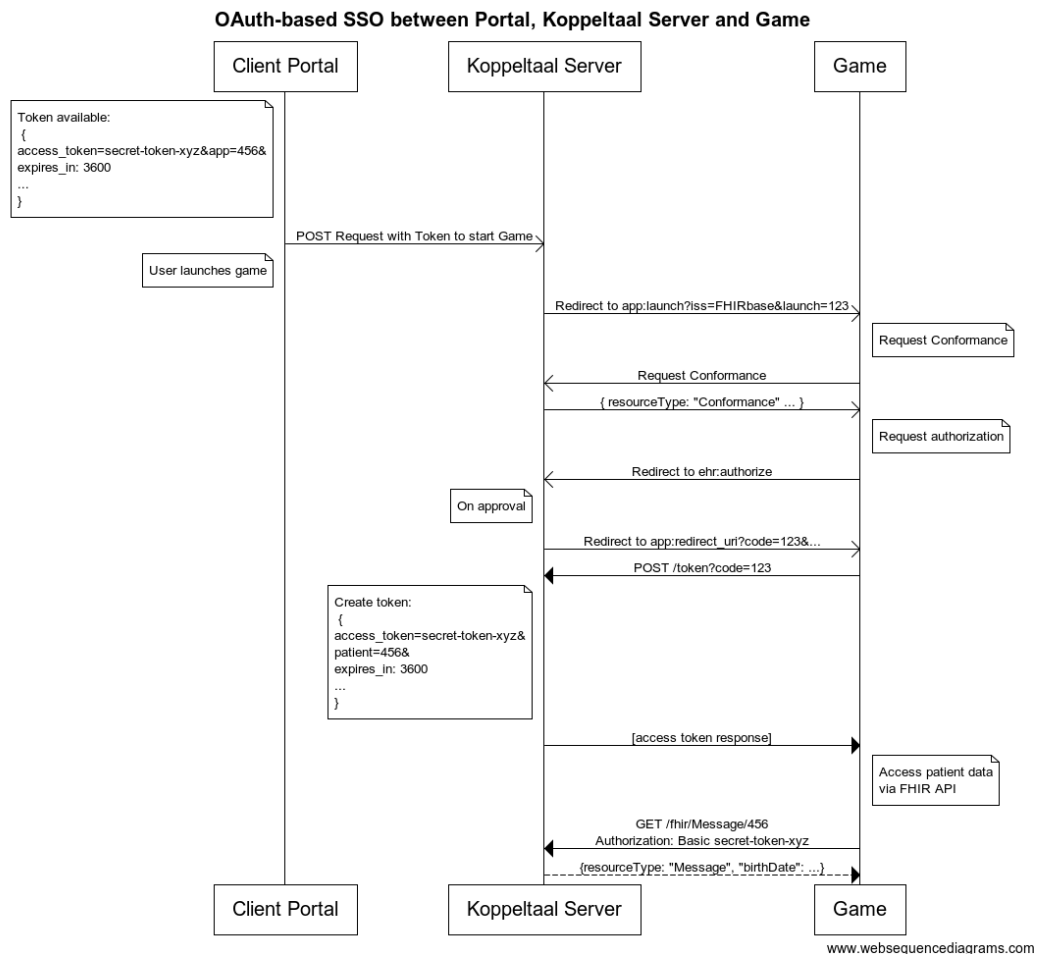
applicatie logica op een server zou draaien.

Voorbeelden JavaScript app in een browser. The Ranj Kick-ASS game is een voorbeeld van een Public Client applicatie.

- "Confidential Client"

Is een applicatie die een "cliënt secret" kan beschermen door gebruik te maken van "server-side business logic". Het grote verschil tussen Publieke en Confidentiële Clients is als de Client de toegang tot het Token endpoint gebruikt, de confidentiële Client het client_id en client_secret als basis authenticatie header kan aanleveren.

6.6.2 Web Launch



Figuur 23. Voorbeeld Game Launch via SSO

1. Portal begint met de 'Launch sequence'
2. Koppeltaal Server redirects naar de Launch URL dat geregistreerd is voor de applicatie
3. Application/Game krijg de Authorization endpoint vanuit de Koppeltaal server
4. Application/Game vraagt Koppeltaal Server om toegang
5. Koppeltaal server geeft toestemming (Authorization) en redirects terug naar de Application/Game
6. Application/Game vraagt de Access token op

7. Application/Game kan berichten vanuit de Koppeltaal server opvragen met de beschikbare Bearer token

6.6.3 Mobile Launch

Mobile Launch lijkt vrij veel op de Web Launch zoals hierboven aangegeven, met kleine verschillen:

- Het eerste verzoek wordt gedaan aan een speciale MobileLaunch Endpoint. Hier zal een mobile activatie code aangevraagd worden. Dit verzoek wordt gedaan (zoals elke Koppeltaal call) onder de credentials van de applicatie instance.

Een Mobile Launch Activatie code opvragen met alle benodigde parameters (zoals bij de Web Launch beschreven):

https://ggz.koppeltaal.nl/OAuth2/Koppeltaal/MobileLaunch?client_id=RANJKA&Patient=https%3A%2F%2Fggzeindhoven.minddistrict.com%2FPatient%2F72308&user=https%3A%2F%2Fggzeindhoven.minddistrict.com%2FRelatedPerson%2F452&resource=RANJKA

- het antwoord is een Activatie Code en een Expiration tijd in dagen:
{"activation_code":"593740","expires_in":7}

Deze activatie code zal vervolgens naar de toekomstige gebruiker doorgegeven moeten worden. (Het is niet binnen de scope van Koppeltaal afsprakenstelsel om deze interactie te beschrijven.) De benodigde beveiliging voor de overdracht van deze code is zeer hoog.

- Bij de eerste opstart van de Mobile App: is deze code aangevraagd?
- De Mobile App (OAuth Client) geeft een gecodeerde configuratie met de FHIR Base URL voor de Koppeltaal Server (verschil met de Web Launch)
- De app zal de Authorize- en Token endpoints moeten ophalen van Koppeltaal Server (OAuth Server) via de Conformance statement (zoals bij de Web Launch).
- De Mobile app zal de Authorize endpoint roepen met de Mobile Activation Code als Launch Code

Voorbeeld om een Authorize op te vragen met alle benodigde parameters:

[https://ggz.koppeltaal.nl/FHIR/Authorize?response_type=code&client_id=RANJKA&redirect_uri=http%3A%2F%2Ftest.kickass.ranjgames.com%2Fbuilds%2Frev-1005%2Findex-game.html%2FAfter-Auth&scope=Patient%2F.read%20launch%3A593740&state=98wrghuwuogerg97](https://ggz.koppeltaal.nl/FHIR/Authorize?response_type=code&client_id=RANJKA&redirect_uri=http%3A%2F%2Ftest.kickass.ranjgames.com%2Fbuilds%2Frev-1005%2Findex-game.html%2FAfter-Auth&scope=Patient%2F*.read%20launch%3A593740&state=98wrghuwuogerg97)*

- Voor het autorisatie verzoek zal de Koppeltaal server een authorization_code teruggeven: {"authorization_code":"0db34c09-201b-41da-af41-deee89302f4b"}
- Als laatste; de Mobile App zal een token vanuit de Koppeltaal Server krijgen net zoals bij de Web Launch Sequence. Deze Access Token kan verderop gebruikt worden om de interactie met de Koppeltaal Server te maken.

Note: Een mobile launch code kan alleen één keer gebruikt kunnen worden.

6.6.4 Gebruik van Refresh Token

Voor een specifieke ClientId kan de Koppeltaal Server geconfigureerd worden om een refresh_token bij te voegen binnen de Token Request.

Voorbeeld van de Token:

```
{  "access_token":  "f3d421f4-d036-468a-b9aa-de9c777ede95","token_type":"Bearer",
  "expires_in":900,"refresh_token":"e54a2533-df44-4e32-bc4d-820c05b2aed0",
  "scope": "Patient/*.*",      "Patient":
  "https://ggzeindhoven.minddistrict.com/Patient/72308","resource":
  "https://ggzeindhoven.minddistrict.com/RelatedPerson/452"}
```

De expiratie tijd gespecificeerd door de "expires_in" is 15 minuten of shorter, met de indicatie dat de access_token gauw niet meer geldig zal zijn. Koppeltaal gebruikt de code 'expired' als OperationOutcome als een verzoek niet gelukt is (FAIL) vanwege een timeout.

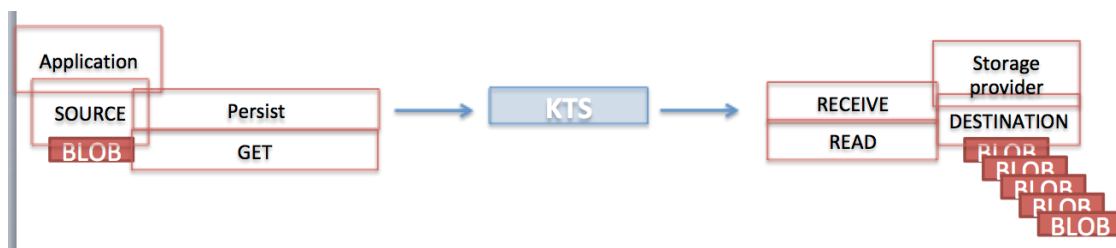
6.7 Storage Service Proxy

Als een applicatie in een domein geen eigen opslag heeft, kan deze indien gewenst gebruik maken van de Storage Service Proxy via de Koppeltaal Server. Zo kan een webapplicatie zonder backend een status opslaan in een cliëntportaal, om met deze informatie de app weer te kunnen starten als de cliënt weer inlogt. Koppeltaal biedt dus zelf geen opslag maar kan als proxy dienen voor een applicatie in het Koppeltaal domein.

De configuratie van storage als dienst in een Koppeltaal domein moet voldoen aan de security en privacy eisen van Koppeltaal en van de betreffende GGZ deelnemer. De termijn en het beleid van opslag van gegevens is onderhevig aan het beleid van de betreffende GGZ deelnemer en de betreffende IT leveranciers.

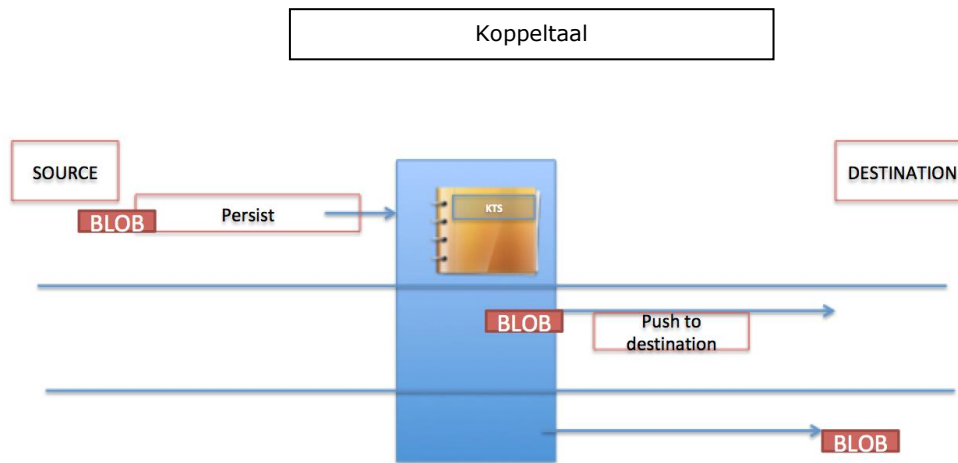
Storage provider: het aangesloten organisatie met deze service (bestemming voor de data)

Applicatie: de op Koppeltaal aangesloten applicatie instance (binnen een domein, de eigenaar van data).



Figuur 24. Storage Service Proxy gebruik

De Storage Service Proxy is geconfigureerd specifiek voor het Domein waarvoor de applicatie is geregistreerd.



Figuur 25. Proxying door het KTS Storage Service

Om gegevens op te slaan, of op te halen kan het eindpunt <https://koppeltaal.nl/FHIR/Koppeltaal/Other?code=StorageItem> worden gebruikt. De header bevat username password en koppelt het bericht aan een application instance. De volgende operaties worden ondersteund:

- POST [Koppeltaal Server]/FHIR/Koppeltaal/Other?code=StorageItem Slaat een nieuw gegeven op. De inhoud van het bericht moet een StorageItem resource bevatten.
- PUT [Koppeltaal Server]/FHIR/Koppeltaal/Other/{id}?code=StorageItem werkt een bestaand opgeslagen gegeven bij. Het argument "id" moet het door de server gegenereerde id van het opgeslagen gegeven zijn. De inhoud van het bericht moet een StorageItem resource bevatten.
- GET[KoppeltaalServer]/FHIR/Koppeltaal/Other/{[id]}?code=StorageItem&Patient=[Patient-id]{&search-arguments} Haalt een opgeslagen geven op dat overeen komt met de id, of zoekargumenten.

Argumenten die zijn toegestaan zijn:

_id (system assigned ID) - is gelijk aan

Object Type - is gelijk aan

Object Key - is gelijk aan / start met / bevat

LastUpdated (filter) - is gelijk aan / is groter dan / is groter dan of gelijk aan / is kleiner dan / is kleiner dan of gelijk aan

LastUpdated (sorteert op- of aflopend)

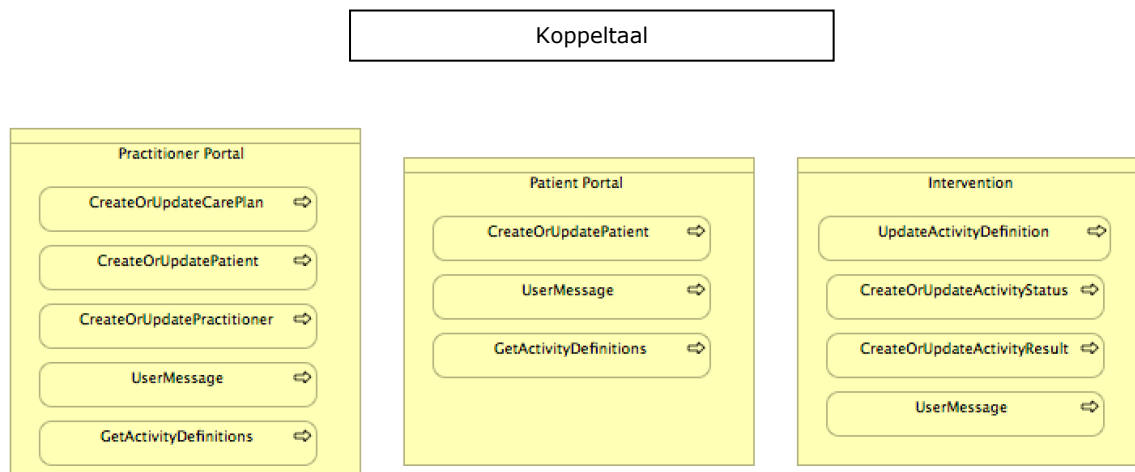
- DELETE [Koppeltaal Server]/FHIR/Koppeltaal/Other/StorageItem:{[id]}

Wist het opgeslagen gegeven met deze id.

7 Primaire interacties: gegevensuitwisseling

7.1 Exchange Informatie model

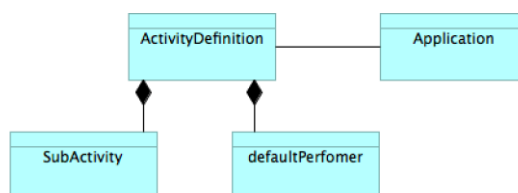
Per type applicatie er zijn specifieke type berichten beschikbaar om de gehele functionele flow van Koppeltaal te ondersteunen:



Figuur 26. Koppeltaal berichten

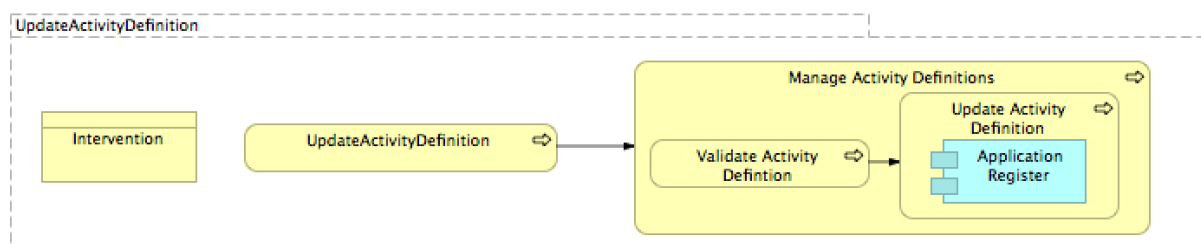
7.2 Activity Definities lezen en updaten

Het Applicatie Register bevat de definities van de interventies ([ActivityDefinition](#)). Deze worden aangemaakt middels het bericht CreateOrUpdateActivityDefinition welke via het certificeringsproces getest zijn op goede en passende werking binnen Koppeltaal. De structuur van de definities is gerepresenteerd door een lijst van de sub-activiteiten die horen bij deze interventie (bijv. level in een game, sectie van een ROM vragenlijst, etc).



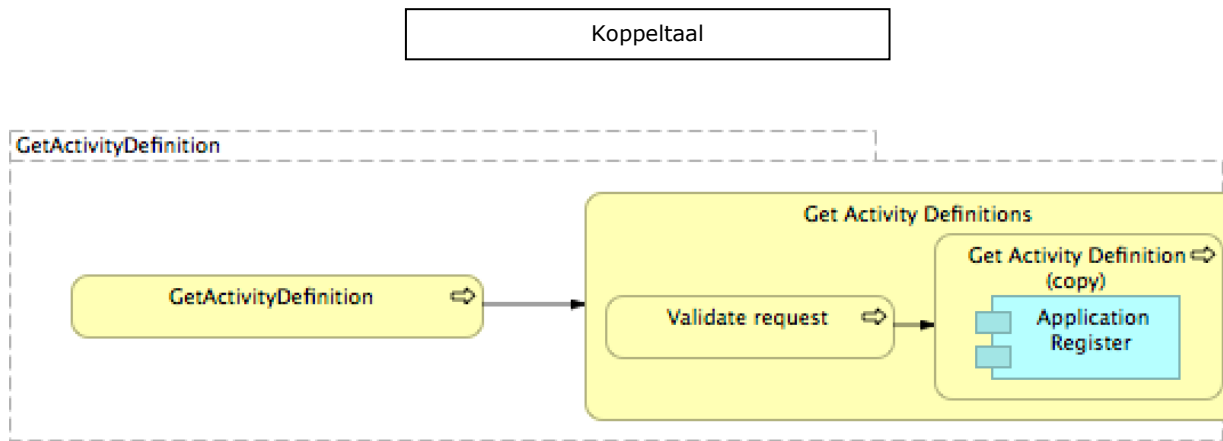
Figuur 27. Activity Definitie

Het proces van het maken en updaten van de definities is dynamisch en direct via REST POST/PUT API Calls voor elke [ActivityDefinition](#). Updates van de definities zal niet in nieuwe activiteiten resulteren.



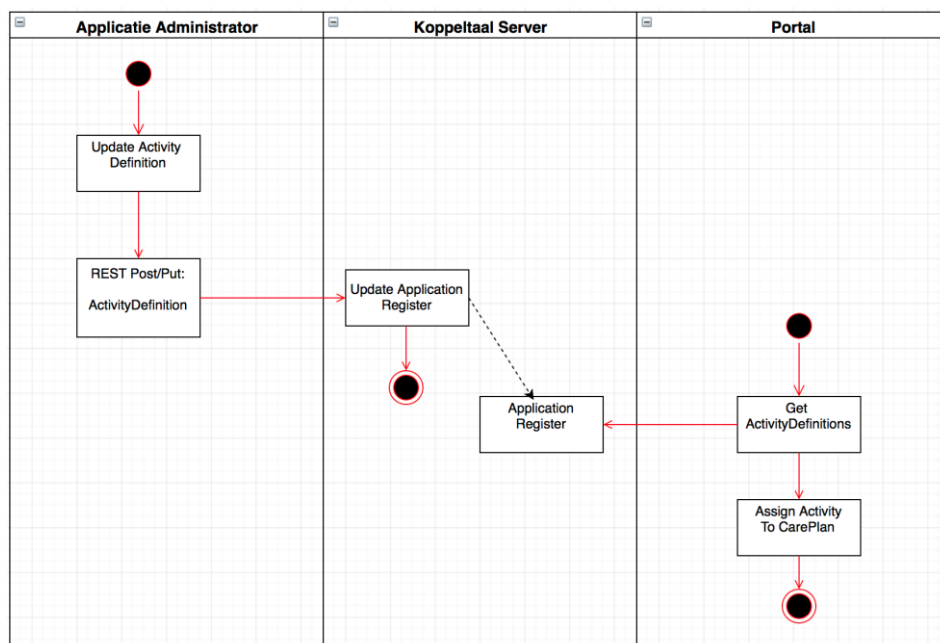
Figuur 28. Update activity definitie

Activity Definities kunnen lezen worden met een specifieke GET REST API:
[https://koppeltaal.nl/FHIR/Koppeltaal/Other/_search?code=ActivityDefinition&\[includearchived=yes\]](https://koppeltaal.nl/FHIR/Koppeltaal/Other/_search?code=ActivityDefinition&[includearchived=yes]).



Figuur 29. Activity Definitie opvragen

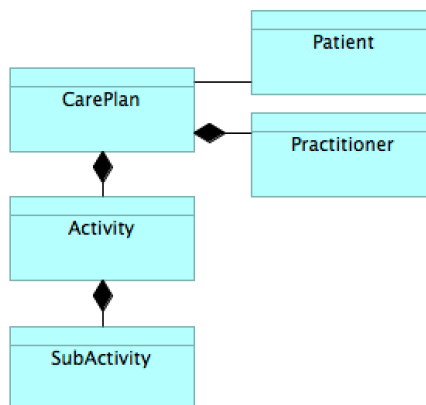
Deze call biedt de Activity Definities aan die beschikbaar zijn de applicatie instantie. Deze Activity Definities kunnen gebruikt worden door een CarePlan (via CarePlan.activity). Als 'includearchived' is 'yes' dan zal de call ook elke Activity Definitie dat gearchiveerd is teruggeven.



Figuur 30. Sync acitvity definities

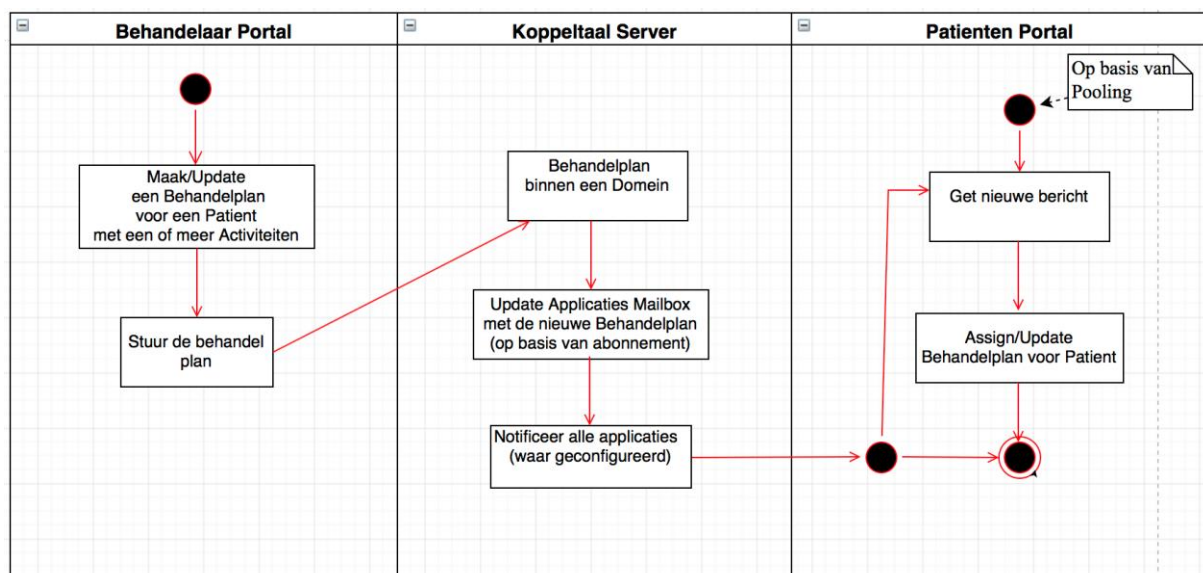
7.3 Care Plan uitwisseling

Een [CarePlan](#) is gemaakt door een behandelaar voor een patiënt. Een behandelplan bevat één of meer activiteiten die door de patiënt kunnen worden uitgevoerd.



Figuur 31. CarePlanstructuur

Als een behandelplan aan een patiënt is toegewezen zijn alle geselecteerde activiteiten direct toegankelijk voor de patiënt. De relatie tussen de patiënt en behandelaar is gemaakt via het behandelplan (en impliciet via de activiteiten daarin).



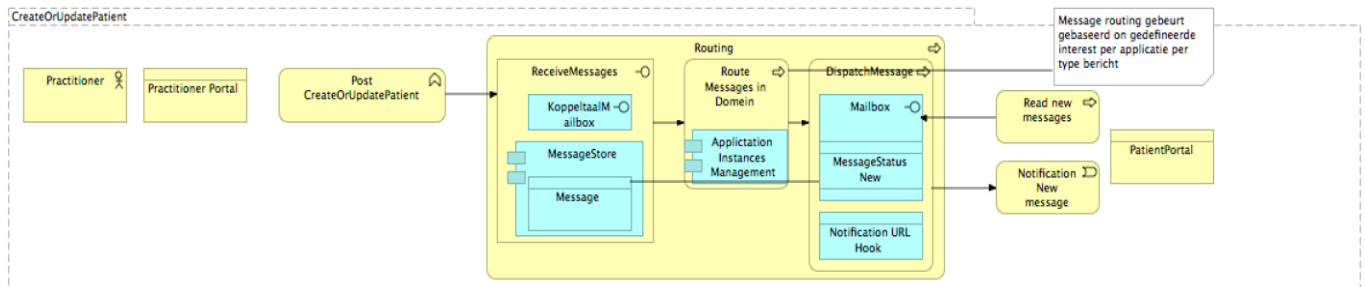
Figuur 32. CarePlan bericht

7.4 Patiënt en behandelaar uitwisseling

Om de synchronisatie tussen de applicaties binnen een domein mogelijk te maken, zijn er twee aparte berichten gedefinieerd: CreateOrUpdatePatient met Patient als FocalResource, en CreateOrUpdatePractitioner met Behandelaar als FocalResource.

Koppeltaal

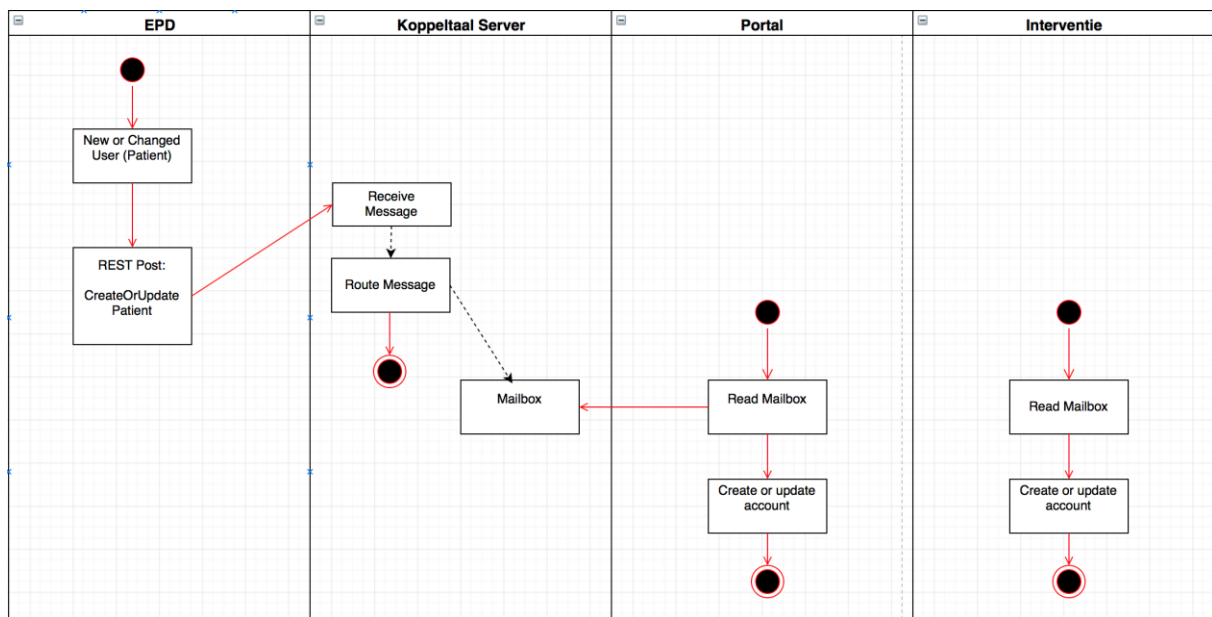
Het doel van deze synchronisatie is om de identificatie van de Gebruiker (Patiënt of behandelaar) gelijk te trekken binnen een Domein. De identificatie van de gebruiker is in Koppeltaal een van de bouwblokken voor de starten van de Interventies.



Figuur 33. Routeren CreateOrUpdatePatient

Als een voorbeeld, een nieuwe gebruiker/inschrijving in de EPD zal een set van acties triggeren:

- het maken van een CreateOrUpdateBericht voor de nieuwe gebruiker
- het posten en routeren via Koppeltaal
- de lokale handling binnen de applicaties om een account voor deze gebruiker te kunnen regelen en de gebruiker ID te koppelen

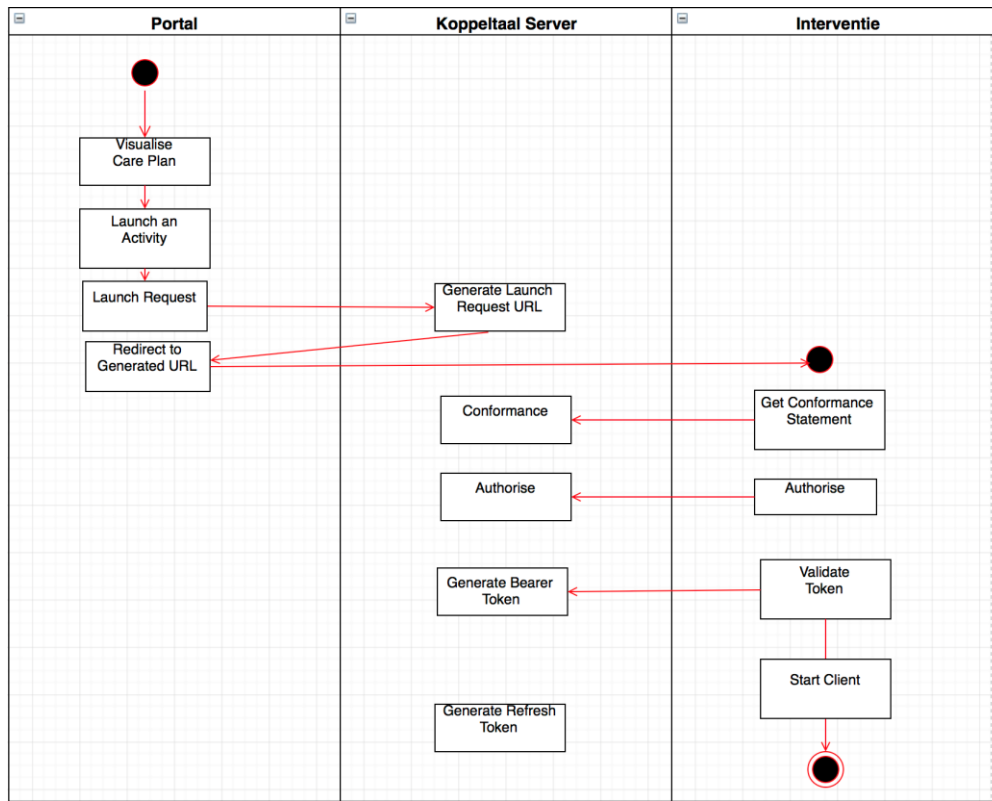


Figuur 34. Patient synchronisatie

7.5 Het starten van een Applicatie

[WebLaunch sequence](#) – het opstarten van een Applicatie gaat volgens de OAuth 2.0 flows zoals beschreven in paragraaf 6.5.

Het voorbeeld laat zien hoe een Interventie kan worden opgestart via deze flow. De gehele interactie is gedaan via de Koppeltaal Server dat acteert als OAuth Server.



Figuur 35. Launch sequence

1. Portal start met de visualisatie van de CarePlan – daarin staat de Activiteiten
2. Gebruiker (Patiënt, behandelaar of naasten) activeert de activiteit: 'Launch sequence' start
3. Koppeltaal Server (Oauth Server) redirects naar de Launch URL die is geregistreerd voor deze Interventie
4. Interventie krijg de Authorization endpoint vanuit de Koppeltaal server via Conformance Statement
5. Interventie (Oauth Client) opvraag Koppeltaal Server voor Authorization
6. Koppeltaal server geeft toestemming (Authorization) en redirects terug naar de Interventie
7. Interventie vraagt de Access token op
8. Interventie kan Berichten vanuit de Koppeltaal server opvragen met de beschikbare Bearer token
9. Refresh Token kan aangevraagd worden als de Bearer token expires

7.6 Activity Status en Result updates

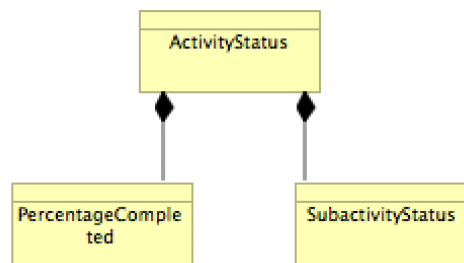
Er wordt een set van berichten gebruikt om de status van de behandel Plan activiteit actueel te houden.

- CreateOrUpdateCarePlan bericht: Dit bericht bevat de informatie over het patiënt behandelplan. Als een behandelaar een of meerdere activiteiten toewijst aan een patiënt, wordt een CarePlan resource gemaakt met de aangegeven activiteiten.

- UpdateCarePlanActivityStatus bericht: Dit bericht wordt gebruikt om de status van activiteiten binnen het behandelplan up to date te houden. Merk op dat een statuswijziging van één CarePlanActivity een statusverandering in een andere CarePlanActivity kan veroorzaken. Als een CarePlanActivity bijvoorbeeld afhankelijk is van een andere, kan het beëindigen van een CarePlanActivity ertoe leiden dat afhankelijke CarePlanActivities beschikbaar komen.
- Zodra een CarePlanActivity beschikbaar is, kunnen er resultaten worden gegenereerd. Resultaten worden gecommuniceerd met behulp van het bericht CreateOrUpdateCarePlanActivityResult. Resultaten kunnen tussentijdse resultaten zijn en als zodanig onderhevig aan verandering.

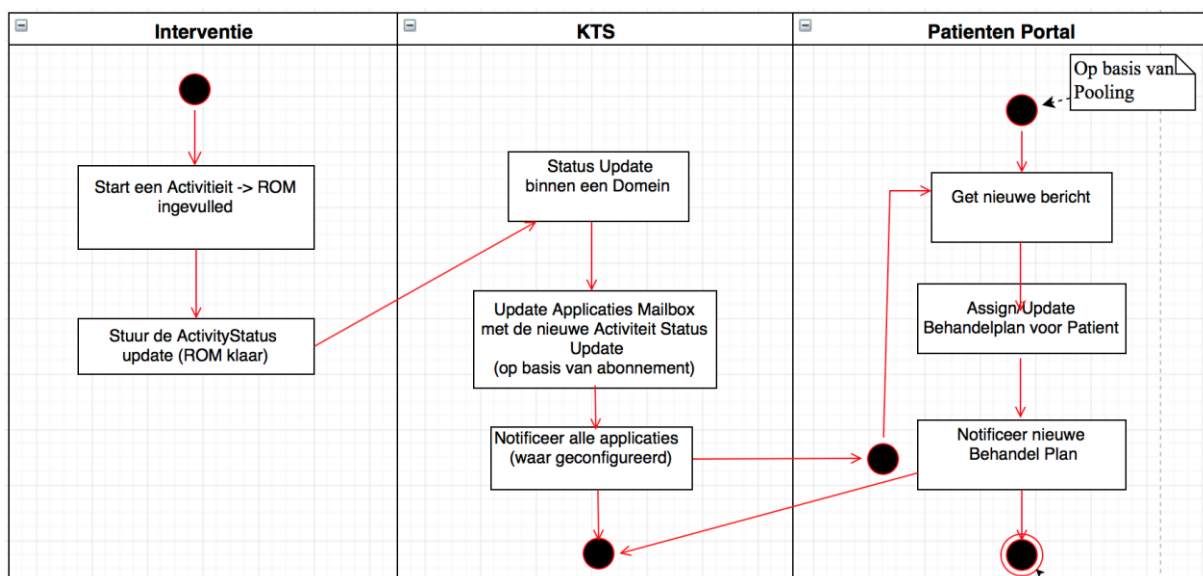
Er zijn twee typen van real time updates:

- Status Updates: met de FHIR specifieke Status : Started



Figuur 36. Activity Status

- Result Updates: om meer informatie bij de Update te kunnen toevoegen is de FHIR "Result" gebruikt



Figuur 37. Interventie en update behandelplan

Wanneer de patiënt bijvoorbeeld begint met het uitvoeren van een CarePlanActivity, verandert zijn status van 'beschikbaar' in 'in uitvoering'.

Vervolgens wordt een bericht van het type `UpdateCarePlanActivityStatus` verzonden met de `CarePlanActivity` waarvan de status is gewijzigd. Dit bericht is bedoeld om de applicatie die eigenaar is van het `CarePlan` op de hoogte te stellen van de verandering in status. De verantwoordelijke toepassing kan vervolgens beurtelings een bericht van het type `CreateOrUpdateCarePlan` verzenden om alle toepassingen die geïnteresseerd zijn in het `CarePlan` als geheel (in plaats van specifieke `CarePlanActivity` in de `CarePlan`) op de hoogte te stellen van de wijziging.

7.7 User berichten

User berichten zijn nodig om eenvoudige ongestructureerde informatie te kunnen uitwisselen tussen patiënt, naasten en behandelaar. Zoals hierboven beschreven in de scenario's en usecases in paragraaf 2.1.6) Dit gebeurt via een `CreateOrUpdateUserMessage`. Deze berichten bevatten alleen tekst.

8 Woordenlijst

Woord (NL/ EN)	Definitie	Synoniemen
Applicatie	alle vormen van ICT-systemen en eHealthplatforms die voor een zorgaanbieder relevant zijn om gegevens tussen uit te wisselen in de context van blended care behandelingen	App/ Application
Applicatie realisatie	Een versie van een applicatie in 1 specifiek domein	Application instance
Blended care	Bij blended care worden reguliere face-to-face gesprekken gecombineerd met online interventies zoals bijvoorbeeld chat, beeldbellen, online behandelmodules en online inzage in het eigen gezondheidsdossier	Zorgproces
CarePlan		ZorgPlan
Cliënt		Patiënt
CreateOrUpdate		MaakOfWijzig
Derden	Anderen dan behandelaar of patient die bij de behandeling worden betrokken, zoals naaste familie.	RelatedPerson
Domein/Domain	Een domein, in Koppeltaal, is synoniem aan een zorgaanbieder, en beperkt de gegevensuitwisseling tot <i>interne gegevensuitwisseling</i> onder verantwoordelijkheid van de betreffende zorgaanbieder. Domeinen worden op de Koppeltaal server afgedwongen.	
eHealth	Het gebruik van informatie- en communicatietechnologie om mensen met psychische klachten en/of aandoeningen te informeren en/of te ondersteunen met betrekking tot hun psychische gezondheid om zo het herstelproces en daarmee hun kwaliteit van leven te verbeteren	eHealth
Interventie	Software en systeem die de cliënt gebruikt in het kader van een behandeling. Een App is een type interventie.	
KTS (tbp)	Centrale server waar Koppeltaal FHIR berichten worden ontvangen en tijdelijk opgeslagen	Koppeltaal server
Leverancier	Leveranciers van applicaties	IT-deelnemer
MaximumMessageAgeInDays	maximum wachttijd ophalen bericht (in aanvang 56 dagen)	
Verzendt Bericht		Dispatch Message

Practitioner		behandelaar
Patiënt	Een person die een behandeling krijgt.	Clïënt
Systeem/System	Platform waar software componenten op worden uitgevoerd.	
Resource	FHIR hanteert de definitie Resource (lijst met mogelijke Resources: https://www.hl7.org/fhir/resourcelist.html)	
Zorgaanbieder	Een instelling dan wel een solistisch werkende zorgverlener	Instelling, Domein

9 Appendix: Voorbeeld resource versioning

```

<entry>
  <link rel="self" href="http://co.healthonline.nl/Patient/215325/_history/2018-07-11T08:01:22:181.2659" />
  <content type="text/xml">
    <Patient id="215325" xmlns="http://hl7.org/fhir">
      <extension url="http://ggz.koppeltaal.nl/fhir/Koppeltaal/Patient#Age">
        <valueInteger value="44" />
      </extension>
      <name>
        <use value="official" />
        <family value="Todea" />
        <given value="Reli" />
      </name>
      <gender>
        <coding>
          <system value="http://hl7.org/fhir/vs/administrative-gender" />
          <code value="M" />
          <display value="Male" />
        </coding>
      </gender>
      <birthDate value="1972-02-28T00:00:00+01:00" />
    </Patient>
  </content>
</entry>

```

10 Appendix: Response voorbeelden

Zoals genoemd in punt 4, worden als response op een Mailbox POST in versie 1.3.5 alle uitgekeerde versies teruggegeven.

Voor elke resource wordt een data-element teruggegeven in de MessageHeader. In 1.3.3 was dit alleen het geval voor de focal resource.

Hieronder een voorbeeldresponse in 1.3.5.

```

<feed xmlns="http://www.w3.org/2005/Atom">
  <id>urn:uuid:5931e3dc-243b-4f29-9200-78c238df9771</id>

```

```

<category term="http://ggz.koppeltaal.nl/fhir/Koppeltaal/Domain#Dev" label="Dev"
scheme="http://hl7.org/fhir/tag/security"/>
<category term="http://hl7.org/fhir/tag/message" scheme="http://hl7.org/fhir/tag"/>
<entry>
  <id>urn:uuid:cab1c156-125c-49d2-9765-cab3e9fddff2</id>
  <content type="text/xml">
    <MessageHeader xmlns="http://hl7.org/fhir">
      <identifier value="urn:uuid:cab1c156-125c-49d2-9765-cab3e9fddff2"/>
      <timestamp value="2018-05-23T09:33:55+02:00"/>
      <event>
        <system value="http://ggz.koppeltaal.nl/fhir/Koppeltaal/MessageEvents"/>
        <code value="CreateOrUpdateCarePlan"/>
        <display value="CreateOrUpdateCarePlan"/>
      </event>
      <response>
        <identifier value="03e2edd0-ef69-49ed-97e9-b075a45a118a"/>
        <code value="ok"/>
      </response>
      <source>
        <name value=""/>
        <software value=""/>
        <version value=""/>
        <endpoint value="https://demo.koppeltaal.nl/FHIR/Koppeltaal/Mailbox"/>
      </source>
      <data>
        <reference value="http://demo.koppeltaal.nl/CarePlan/5044/_history/2018-05-23T07:33:55:708.2583"/>
      </data>
      <data>
        <reference value="http://demo.koppeltaal.nl/Patient/1055/_history/2018-05-23T07:33:55:708.2583"/>
      </data>
    </MessageHeader>
  </content>
</entry>
</feed>

```

Als er aan een 1.3.5 compatibele applicatie een 409 wordt teruggegeven omdat een of meer verkeerde versies zijn meegestuurd, worden hierin de resources teruggegeven waarvan de verkeerde versie was meegestuurd. Dit gebeurt in de volgende response:

```

<OperationOutcome xmlns="http://hl7.org/fhir">
  <text>
    <status value="generated"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p xmlns=""/>
    </div>
  </text>
  <issue>
    <severity value="error"/>
    <type>
      <system value="http://hl7.org/fhir/issue-type"/>
    </type>
  </issue>
</OperationOutcome>

```

```

    <code value="conflict"/>
  </type>
  <extension
url="http://ggz.koppeltaal.nl/fhir/Koppeltaal/OperationOutcome#IssueResource">
    <valueResource>
      <reference value="http://demo.koppeltaal.nl/fhir/Patient/1" />
    </valueResource>
  </extension>
  <details value="The specified resource version is not correct."/>
</issue>
<issue>
  <severity value="error"/>
  <type>
    <system value="http://hl7.org/fhir/issue-type"/>
    <code value="conflict"/>
  </type>
  <extension
url="http://ggz.koppeltaal.nl/fhir/Koppeltaal/OperationOutcome#IssueResource">
    <valueResource>
      <reference value="http://demo.koppeltaal.nl/fhir/Practitioner/12" />
    </valueResource>
  </extension>
  <details value="The specified resource version is not correct."/>
</issue>
</OperationOutcome>

```

Als er een bericht verstuurd wordt naar de Koppeltaal Server met daarin een resource die Koppeltaal niet ondersteunt (in dit voorbeeld 'Condition'), wordt de volgende respons geretourneerd:

```

<OperationOutcome xmlns="http://hl7.org/fhir">
  <text>
    <status value="generated"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p xmlns="">at IExtOnFHIR.FHIRExceptionAction.Run(ActionInput Input, ActionOutputRequest
RequestedOutputs)
      ...</p>
    </div>
  </text>
  <issue>
    <severity value="error"/>
    <details value="The resource type 'Condition' is not supported."/>
  </issue>
</OperationOutcome>

```

Figuur 38. Voorbeeld van een "Condition" respons bericht.

11 Appendix: FHIR Resources

11.1 (Message) Bundle

Definition	A Bundle is a container resource containing one or more FHIR resources that represent the information relevant for the processing of the event.		
Control	1..1		
Bundle.id			
Definition	A unique URI for this bundle.		
Control	1..1		
Type	Uri		
Bundle.title			
Definition	Text statement of purpose.		
Control	1..1		
Type	string		
Bundle.updated			
Definition	When the bundle was built.		
Control	1..1		
Type	instant		
Bundle.tag			
Required	true		
Definition	Tags can be used to add extra context information to a message. Most importantly, it is used to indicate which domain the message is part of.		
Control	2..*		
Type	Tag		
Comments	It is required to always add a domain tag and a message tag:		
	Message tag		
	Term	http://hl7.org/fhir/tag	
	Scheme	http://hl7.org/fhir/tag/security	
	Label	[Domain]	
	<p>Adding the Domain security tag may seem redundant, since the Domain can also be determined from the credentials passed by the source application. However, for applications that operate in multiple domains, this is an extra check that the message is sent with credentials that match the intended domain. http://hl7.org/fhir/tag Note: The ValidateOnly tag indicates that the server should not actually process the message, but only check whether or not the message is valid. It is created using new Tag("http://ggz.koppeltaal.nl/fhir/Koppeltaal/ValidateOnly", Tag.FHIRTAGSCHEME_GENERAL).</p>		
Bundle.entries			
Definition	A bundle contains one entry for each resource that is part of the message. There is one focal resource (the resource referenced by MessageHeader.data) and zero or more resources that are referenced by the focal resource.		
Control	1..*		

Comments It is not necessary for all resources referenced by the focal resource to be included in the bundle, so long as they are not integral to understanding the content of the message.

Bundle.entries.title

Definition Text summary of resource content

Control 1..1

Type string

Bundle.entries.updated

Definition The moment that the resource was last updated.

Control 1..1

Type Instant

Bundle.entries.content

Definition The resource object.

Control 1..1

Type Resource(Any)

11.2 MessageHeader

Definition The header for a message exchange that is either requesting or responding to an action. The resource(s) that are the subject of the action as well as other Information related to the action are typically transmitted in a bundle in which the MessageHeader resource instance is the first resource in the bundle.

Control 1..1

Comments The MessageHeader must be the first resource in every Message Bundle.

MessageHeader.identifier

Definition The identifier of this message.

Control 1..1

Type id

Comments An id is a whole number in the range 0 to $2^{64}-1$ (optionally represented in hex), a uuid, an oid, or any other combination of lowercase letters, numerals, "-" and ".", with a length limit of 36 characters.Regex: `[a-z0-9\-\.\.]{1,36}`

MessageHeader.timestamp

Definition The time that the message was sent.

Control 1..1

Type instant

MessageHeader.event

Definition Code that identifies the event this message represents and connects it with it's definition.

Control 1..1

Binding [MessageEvents](#)

Comments [This field contains the message type](#)

MessageHeader.source

Definition The source application from which this message originated.

Control 1..1

Requirements Allows replies, supports audit.

MessageHeader.source.name

Definition Human-readable name for the source application.

Control 0..1
Type string

MessageHeader.source.software

Definition Name of the software of the source application. May include configuration or other information useful in debugging.

Control 1..1
Type string

Comments Not specifically used in Koppeltaal, but must be present because FHIR mandates it.

MessageHeader.source.version

Definition Version of the software of the source application.

Control 0..1
Type string

Requirements Supports audit and possibly interface engine behavior.

MessageHeader.source.endpoint

Definition Identifies the endpoint of the source application

Control 1..1
Type uri

Comments Not used in Koppeltaal, but must be present because FHIR mandates it. Recommended value is the FHIR Base URL of the source system also used in resource identifiers.

MessageHeader.data

Definition The actual data of the message - a reference to the root/focus resource of the event.

Control 0..*
Type Resource(Any)

Comments Koppeltaal defines per message type what type the focus resource should have. This is also defined in the Conformance.

MessageHeader.Patient

Definition Reference to the Patient resource reflecting the dossier this message belongs to.

Control 0..1
Type Resource(Patient)

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/MessageHeader#Patient>

Comments This field is required for all message types, except those that have no Patient context. The message without Patient context are CreateOrUpdatePractitioner and CreateOrUpdateActivityDefinition.

MessageHeader.processingStatus

Definition The status of the message with regards to the processing cycle.

Control 0..1
Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/MessageHeader#ProcessingStatus>

MessageHeader.processingStatus.status

Definition The status that the message is currently in.

Control 0..1
Type Coding
Binding [ProcessingStatus](#)

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/MessageHeader#ProcessingStatusStatus>

MessageHeader.processingStatus.statusLastChanged

Definition	The time that the message's status was last changed.
Control	1..1
Type	instant
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/MessageHeader#ProcessingStatusStatusLastChanged

MessageHeader.processingStatus.exception

Definition	Details of the exception that occurred.
Control	0..1
Type	string
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/MessageHeader#ProcessingStatusException

MessageHeader.isExpired

Definition	Indicates whether or not the message has expired.
Control	0..1
Type	Boolean
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/MessageHeader#IsExpired
Notes	When empty, 'false' is assumed.

11.3 ActivityDefinition

Definition	An activity definition describes an activity that is made available by a Device. ActivityDefinition is mapped to a FHIR resource of type Other.
------------	---

Control 1..1

ActivityDefinition.code

Definition	Allows Koppeltaal to recognize the Other resource as an ActivityDefinition.
------------	---

Control 1..1

Type CodeableConcept

Binding [OtherResourceUsage](#)**ActivityDefinition.application**

Definition	The application that this ApplicationDefinition is available in.
Control	1..1
Type	Resource(Application)
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#Application

Notes The DisplayName of this relation is the Application's identifier. This value should be used in the SSO sequence as ClientID.

ActivityDefinition.name

Definition	Name of the game, questionnaire, etc. A single application may provide multiple activities, e.g. a ROM provider will provide several different questionnaires.
Type	string
Control	1..1
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#ActivityName

ActivityDefinition.activityDefinitionIdentifier

Definition	A unique identifier for this activity definition.
------------	---

Control	0..1
Type	Identifier
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#ActivityDefinitionIdentifier

ActivityDefinition.identifier

Definition	One or more unique identifier for this activity definition.
Control	0..*
Type	Identifier
Notes	Deprecated

ActivityDefinition.description

Definition	A description of the activity. May be used to judge the intended use of an activity.
Type	string
Control	0..1
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#ActivityDescription

ActivityDefinition.type

Definition	The type of activity.
Control	1..1
Type	CodeableConcept
Binding	ActivityKind
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#ActivityKind

ActivityDefinition.subActivity

Definition	A list of available modules within the activity.
Control	0..*
Comments	For example, within the KickAss game, subactivities may be defined as the missions that are available in the game.
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CarePlan#SubActivity

ActivityDefinition.subActivity.name

Definition	The name of the subactivity.
Control	1..1
Type	string
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#SubActivityName

ActivityDefinition.subActivity.identifier

Definition	An identifier for this specific subactivity.
Control	1..1
Type	string
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#SubActivityIdentifier

ActivityDefinition.subActivity.description

Definition	A description of the subactivity that can be used to judge the intended use of the subactivity.
Control	0..1
Type	string
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#SubActivityDescription

ActivityDefinition.subActivity.isActive

Definition	Indicates if the sub activity is active
Control	0..1

Type	boolean
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#SubActivityIsActive
Comments	If no value is specified, value 'true' is assumed. This is to preserve backwards compatibility.

ActivityDefinition.defaultPerformer

Definition	The person that is normally responsible for performing this activity.
Control	0..1
Type	Code
Binding	ActivityPerformer
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#DefaultPerformer

ActivityDefinition.isActive

Definition	The person that is normally responsible for performing this activity.
Control	0..1
Type	boolean
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#IsActive
Comments	If no value is specified, value 'true' is assumed. This is to preserve backwards compatibility.

ActivityDefinition.isDomainSpecific

Definition	Indicates whether this domain is only available in the current domain or available in all domains that the providing application is part of.
Control	0..1
Type	boolean
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#IsDomainSpecific

ActivityDefinition.launchType

Definition	Indicates how activities of this type should be launched.
Control	0..1
Type	code
Binding	ActivityDefinitionLaunchType
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#LaunchType
Notes	When this field is empty, value 'Web' is assumed.

ActivityDefinition.isArchived

Definition	Indicates if the activity is archived.
Control	0..1
Type	boolean
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/ActivityDefinition#IsArchived
Comments	Archived ActivityDefinitions are by default not returned when GET-ting ActivityDefinitions. If no value is specified, value 'false' is assumed. This is to preserve backwards compatibility.

11.4 CarePlan

Definition The Careplan is a group of activities assigned in the context of care to a single Patient. CarePlanActivities can be assigned to the client, the client's Practitioner, or a person related to the client.

Control 1..1

CarePlan.Patient

Definition Identifies the Patient whose intended care is described by the plan.

Control 0..1

Type Resource([Patient](#))

CarePlan.status

Definition Indicates whether the plan is currently being acted upon, represents future intentions or is now just historical record.

Control 1..1

Binding CarePlanStatus: Indicates whether the plan is currently being acted upon, represents future intentions or is now just historical record. (See <http://hl7.org/fhir/care-plan-status> for values.)

Type Code

Is true

modifier

CarePlan.participant

Definition Identifies all people and organizations who are expected to be involved in the care envisioned by this plan.

Control 0..*

Comments Within the context of Koppeltaal, it is expected that at least the requester of the careplan is given as a participant with role 'Requester'.

CarePlan.participant.role

Control 0..1

Binding [CarePlanParticipantRole](#)

Type CodeableConcept

Comments For the Practitioner that is has requested ('assigned') the careplan the role should be 'Requester'.

CarePlan.participant.member

Definition The specific person or organization who is participating/expected to participate in the CarePlan.

Control 1..1

Type Resource([Organization](#))

CarePlan.goal

Definition Describes the intended objective(s) of carrying out the Care Plan.

Control 0..*

Comments Goal can be achieving a particular change or merely maintaining a current state or even slowing a decline.

CarePlan.goal.description

Definition Human-readable description of a specific desired objective of the care plan.

Control 1..1

Type string

CarePlan.goal.status

Definition Indicates whether the goal has been reached and is still considered relevant.

Control 0..1

Binding CarePlanGoalStatus: Indicates whether the goal has been met and is still being targeted (see <http://hl7.org/fhir/care-plan-goal-status> for values).

Type Code

CarePlan.goal.notes

Definition Any comments related to the goal.

Control 0..1

Type string

Comments May be used for progress notes, concerns or other related information that doesn't actually describe the goal itself.

CarePlan.activity

Definition A list of actions to occur as part of the plan. In effect, a CarePlanActivity is an instance of an ActivityDefinition, meaning that it has been assigned to a Practitioner, RelatedPerson or Patient to be performed.

Control 0..*

CarePlan.activity.id

Koppeltaal true

required

Definition An id used to identify this activity in subsequent status updates.

Control 1..1

Type string

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CarePlan#ActivityID>

CarePlan.activity.identifier

Definition An identifier for this activity. Used when sending an ActivityStatusUpdate.

Control 0..1

Type string

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CarePlan#ActivityIdentifier>

CarePlan.activity.definition

Definition The identifier of the ActivityDefinition that describes the activity to be performed.

Control 0..1

Type string

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CarePlan#ActivityDefinition>

Comments The ActivityDefinition identified by this field may be located either directly in the bundle or in the set of ActivityDefinitions available at the Koppeltaal Server.

CarePlan.activity.type

Definition The type of activity.

Control 1..1

Type Coding

Binding [ActivityKind](#)

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CarePlan#ActivityKind>

Comments **Obsolete! Use the description of Activity Definition referred to by CarePlan.Activity.Definition instead.** Needed for activities that are not defined by an ActivityDefinition; copied from ActivityDefinition otherwise.

CarePlan.activity.description

Definition Description of the activity.

Control 0..1

Type string

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#ActivityDescription>

Comments **Obsolete! Use the description of Activity Definition referred to by CarePlan.Activity.Definition instead.** Needed for activities that are not defined by an ActivityDefinition; copied from ActivityDefinition.description otherwise.

CarePlan.activity.subactivity

Definition A list of subactivities that should be performed.

Control	0..*
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#SubActivity
CarePlan.activity.subactivity	
Definition	A list of subactivities that should be performed.
Control	0..*
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#SubActivity
CarePlan.activity.subactivity.identifier	
Definition	The identifier of the subactivity.
Type	string
Control	1..1
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#SubActivityIdentifier
CarePlan.activity.subactivity.status	
Definition	The status of the subactivity.
Control	0..1
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#SubActivityStatus
Comments	Note that the list of assigned subactivities may differ from the list of subactivities available in the ActivityDefinition. This means that the assigner of the careplan has chosen to not let the performer perform all subactivities. The list of assigned subactivities should be a subset of the subactivities available in the ActivityDefinition.
Binding	CarePlanActivityStatus
Type	Code
CarePlan.activity.goal	
Definition	Describes the intended objective(s) of carrying out this activity.
Control	0..*
Comments	The goal of an activity should be a reference to the ID of a goal in the CarePlan this activity is a part of.
CarePlan.activity.simple.performer	
Definition	Identifies who's expected to be involved in the activity.
Control	0..*
Type	Resource (Patient)
CarePlan.activity.participant	
Definition	Identifies all people and organizations who are involved in the activity.
Control	0..*
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#Participant
CarePlan.activity.participant.role	
Control	0..1
Binding	CarePlanParticipantRole
Type	CodeableConcept
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#ParticipantRole
Comments	Note that 'performer' is not a separate role, but instead is specified in the field CarePlan.activity.simple.performer.
CarePlan.activity.participant.member	
Definition	The specific person or organization who is participating/expected to participate in the activity.
Control	1..1
Type	Resource(Organization)
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#ParticipantMember
CarePlan.activity.startDate	
Definition	The date that this activity should be started.
Control	1..1
Type	dateTime
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#StartDate

CarePlan.activity.CarePlanActivityStatus

Definition	Identifies what progress is being made for the specific activity.
Control	1..1
Binding	CarePlanActivityStatus
Type	Coding
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#ActivityStatus

CarePlan.activity.notes

Definition	Any notes that are entered for this activity.
Control	0..1
Type	string

CarePlan.activity.started

Definition	The date and time when the activity was started.
Control	0..1
Type	instant
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#Started

CarePlan.activity.finished

Definition	The date and time when the activity was completed.
Control	0..1
Type	instant
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#Finished

CarePlan.activity.cancelled

Definition	The date and time when the activity was cancelled or skipped.
Control	0..1
Type	instant
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#Cancelled

CarePlan.activity.endDate

Definition	The date and time after which the activity will no longer be available.
Control	0..1
Type	dateTime
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#EndDate

CarePlan.relation

Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#Relation
Definition	Identifies any relations this careplan may have.
Control	0..*

CarePlan.relation.type

Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#RelationType
Definition	The type of the relation.
Control	1..1
Binding	CarePlanRelationTypes

CarePlan.relation.reference

Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#RelationReference
Definition	The related object.
Control	1..1
Type	Resource(Any)

11.5 CareTeam

Het CareTeam wordt opgenomen in het CreateOrUpdateCarePlan bericht als resource entry als er in het bericht verwezen wordt naar het CareTeam.

11.5.1 Definitie

In de definitiebepaling van het CareTeam is zo veel mogelijk gebruik gemaakt van de [FHIR STU3 CareTeam definitie](#) zodat bij een overgang naar FHIR STU3 en de compatibiliteit van DSTU1 naar STU3 hierin zo min mogelijk verschil is.

Naam	CareTeam
Definitie	Een care team beschrijft welke personen er toegang hebben tot een careplan of careplan activity van waaruit verwezen wordt naar dit object.
Comment	Een CareTeam is geen FHIR DSTU1 resource en is daarom gebaseerd op het FHIR resource type 'Other'

11.5.2 CareTeam elementen

Code

Naam	Careteam.code
Definitie	Geeft Koppeltaal de mogelijkheid om het Other resource te herkennen als een CareTeam
Control	1..1
Type	CodeableConcept
Binding	OtherResourceUsage

Identifier

Naam	Careteam.careTeamIdentifier
Definitie	Identifier voor het care team die het care team binnen het domein uniek identificeert
Control	0..*
Type	Identifier
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CareTeam#CareTeamIdentifier

Status

Naam	Careteam.status
Definitie	Geeft de status van het care team aan.
Control	0..1

Type	Coding
Binding	CareTeamStatus
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CareTeam#Status

Name

Naam	Careteam.name
Definitie	Een label voor menselijk gebruik bedoeld om care teams mee te onderscheiden
Control	0..1
Type	string
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CareTeam#Name

Subject

Naam	Careteam.subject
Definitie	De patiënt aan wie het care team zorg levert
Control	0..1
Type	Resource (Patient)
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CareTeam#Subject

Period

Naam	Careteam.period
Definitie	Geeft aan wanneer het het care team in werking treedt (of is bedoeld) en eindigt.
Control	0..1
Type	Period
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CareTeam#Period

ManagingOrganization

Naam	Careteam.managingOrganization
Definitie	De organisatie die verantwoordelijk is voor het care team
Control	0..*
Type	Resource (Organization)

Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CareTeam#ManagingOrganization
------------------	---

Careteam Valuesets

11.5.3 CareTeamStatus (nieuw)

11.5.3.1 Definitie

Naam	CareTeamStatus
Code system URL	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CareTeamStatus
Value set URL	http://ggz.koppeltaal.nl/fhir/Koppeltaal/ValueSet/CareTeamStatus
Definition	Geeft de status van het care team aan.

11.5.4 Concepts

Code		Display	Definitie
proposed		Proposed	Het care team is samengesteld, maar neemt nog geen deel aan de coördinatie en levering van zorg.
active		Active	Het care team werkt momenteel mee aan de coördinatie en levering van zorg.
suspended		Suspended	Het care team staat op on hold en neemt nog geen deel aan de coördinatie en levering van zorg.
inactive		Inactive	Het care team werkte mee aan de coördinatie en levering van zorg, maar doet dat niet meer.
entered-in-error		Entered in error	Het care team zou niet hebben moeten bestaan.

11.5.5 OtherResourceUsage (update)

Code	Display	Definitie
<i>ActivityDefinition</i>
<i>UserMessage</i>
<i>CarePlanActivityStatus</i>
<i>StorageItem</i>

CareTeam	CareTeam	Wordt gebruikt om de Koppeltaal resource CareTeam te identificeren
----------	----------	--

11.5.6 Toevoeging CarePlan definitie

Het CarePlan heeft op CarePlan en CarePlan.activity niveau 0..* participants. Het participant object krijgt een extra veld genaamd careTeam. Deze toevoeging ziet er als volgt uit:

11.5.7 CarePlan.participant.careTeam

Naam	CarePlan.participant.careTeam
Definitie	De CareTeams waar deze participant onderdeel van is.
Control	0..*
Type	Resource(CareTeam)
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#ParticipantCareTeam

11.5.8 CarePlan.activity.participant.careTeam

Naam	CarePlan.activity.participant.careTeam
Definitie	De CareTeams waar deze participant onderdeel van is.
Control	0..*
Type	Resource(CareTeam)
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlan#ParticipantCareTeam

11.6 CarePlanActivityStatus

Definition Describes the status of a CarePlanActivity in detail.
 Control 1..1
 Comments CarePlanActivityStatus maps to a resource of type Other.

CarePlanActivityStatus.activity

Definition The ID of the activity that is the subject of this message.
 Control 1..1
 Type string
 Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlanActivityStatus#ActivityStatus>
 Notes This must have the same value as CarePlan.activity.identifier for that activity in the CreateOrUpdateCarePlan message.

CarePlanActivityStatus.activityStatus

Definition Identifies what progress is being made for the specific CarePlanActivity.
 Control 1..1
 Binding [CarePlanActivityStatus](#)
 Type Coding
 Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CarePlanActivityStatus#ActivityStatus>

CarePlanActivityStatus.subactivity

Definition The subactivities assigned as part of this activity.
 Control 0..*
 Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CarePlanActivityStatus#SubActivity>
 Comments Note that the list of assigned subactivities may differ from the list of subactivities available in the ActivityDefinition. This means that the assigner of the careplan has chosen to not let the performer perform all subactivities. The list of assigned subactivities should be a subset of the subactivities available in the ActivityDefinition.

CarePlanActivityStatus.subactivity.identifier

Definition The identifier of this subactivity.
 Control 1..1
 Type string
 Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlanActivityStatus#SubActivityIdentifier>
 Comments Must match the identifier of a subactivity as defined in the ActivityDefinition.

CarePlanActivityStatus.subactivity.status

Definition Identifies what progress is being made for this subactivity.
 Control 1..1
 Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlanActivityStatus#SubActivityStatus>
 Binding [CarePlanActivityStatuses](#)
 Type Coding

CarePlanActivityStatus.percentageCompleted

Definition An indication of the progress made on the CarePlanActivity.
 Control 0..1
 Type integer
 Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/CarePlanActivityStatus#PercentageCompleted>

CarePlanActivityStatus.blackBoxState

Definition BlackBoxState allows applications using Koppeltaal to extend messages with information that is not necessarily understood by other applications. The application including BlackBoxState must subscribe to the message to which the BlackBoxState is attached, allowing the application to reload the BlackBoxState next time the application starts for a certain use. BlackBoxState is implemented using the FHIR extension mechanism. Extensions can be nested. For an example, look at how the ProcessingStatus extension is defined for the

MessageHeader resource. Applications using BlackBoxState must create a FHIR profile that describes their extension(s).

Control	0..1
Type	Base64Binary
Extension	Profile#Field as defined by application that owns this
Comments	This extension must be further defined in a profile by the owner.

11.7 CarePlanActivityResult

Definition	The outcome of a CarePlanActivity, including any answers given and calculated scores. The CarePlanActivityResult groups a set of Observation resources and may have a Resource reference to a Questionnaire that holds the answers to questions as entered by the Patient, RelatedPerson or Practitioner. The CarePlanActivityResult resource is an extension of the FHIR resource DiagnosticReport.
------------	---

Control	1..1
Invariants	diagnosticDateTime or a diagnosticPeriod, but not both.
Comments	The CarePlanActivity does not have to be finished in order to have a CarePlanActivityResult. In such cases the CarePlanActivityResult describes the results obtained so far, for example, the scores calculated for the subsections of the questionnaire that have been finished so far.

CarePlanActivityResult.activity

Definition	The id of the activity that this resource is the outcome of.
Control	1..1
Type	string
Extension	http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlanActivityResult#Activity

CarePlanActivityResult.status

Definition	The status of the result.
Control	1..1
Binding	CarePlanActivityResultStatus
Type	code
Is modifier	true
Comments	This is labeled as "Is Modifier" because applications need to take appropriate action if a report is withdrawn.

CarePlanActivityResult.name

Definition	A code or name that describes this diagnostic report.
Control	1..1
Type	string

CarePlanActivityResult.issued

Definition	The date and/or time that this version of the report was released from the source diagnostic service.
Control	1..1
Type	dateTime

CarePlanActivityResult.subject

Definition	The subject of the report.
Control	1..1
Type	Resource(Patient)

CarePlanActivityResult.performer

Definition	The diagnostic service that is responsible for issuing the report.
Control	1..1
Resource	Resource(Organization)

Comments This is not necessarily the source of the atomic data items - it is the entity that takes responsibility for the clinical report.

CarePlanActivityResult.diagnosticDateTime

Definition The date and time at which the observations were made, e.g. date a questionnaire was filled.

Control 0..1

Type dateTime

Invariants diagnosticDateTime or a diagnosticPeriod, but not both.

CarePlanActivityResult.diagnosticPeriod

Definition The period during which the observations were made, e.g. the time period over which a mission in a game was completed.

Control 0..1

Type Period

Invariants diagnosticDateTime or a diagnosticPeriod, but not both.

CarePlanActivityResult.result

Definition Observations that are part of this diagnostic report. Observations can be simple name/value pairs (e.g. "atomic" results), or they can be grouping observations that include references to other members of the group (e.g. "panels").

Control 0..*

Type Resource([Observation](#))

CarePlanActivityResult.presentedForm

Definition Rich text representation of the entire result as issued by the diagnostic service. Multiple formats are allowed but they SHALL be semantically equivalent.

Control 0..*

Type Attachment

CarePlanActivityResult.questionnaire

Definition The answers given by the performer.

Control 0..*

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/CarePlanActivityResult#Questionnaire>

Type Resource([Questionnaire](#))

11.8 Questionnaire

Definition The Questionnaire is used to convey the choices made while playing a game or answering a ROM questionnaire.

Questionnaire.name

Koppeltaal true

required

Definition Structured name for a predefined list of questions this questionnaire is responding to. May not be defined for choices made in a game.

Control 0..1

Type CodeableConcept

Binding Defined by Application

Questionnaire.status

Koppeltaal true

required

Definition The lifecycle status of the questionnaire as a whole.

Control 1..1

Type CodeableConcept

Binding [QuestionnaireStatus](#)

Comments For phase 1, it is recommended to only sent results when the activity has been completed. This is because not all applications support recalling/cancelling of intermediate results.

Questionnaire.authored

Koppeltaal true
 required
 Definition The date and/or time that this the questionnaire was authored.
 Control 1..1
 Type dateTime
 Comments This field is required by FHIR. It can be used to differentiate different versions of the questionnaire.

Questionnaire.group

Koppeltaal true
 required
 Definition A group of questions
 Control 0..1
 Comments The Questionnaire itself has one "root" group with the actual contents of the Questionnaire.

Questionnaire.group.question

Koppeltaal true
 required
 Definition Questions within the group. The order of questions within the group is relevant.
 Control 0..*
 Requirements Must register answers to questions.
 Comments The Questionnaire itself has one "root" group with the actual contents of the Questionnaire.

Questionnaire.group.question.name

Koppeltaal true
 required
 Definition Structured name for the question that identifies this question within the Questionnaire or Group.
 Control 0..1
 Type CodeableConcept
 Binding Is defined by Application
 Requirements Structured name identifies the question so the answers can be extracted and used.

Questionnaire.answer[x] (most likely answerString)

Definition Single-valued answer to the question.
 Control 0..1
 Type string
 Comments Only one of answer[x] or choice can be provided

Questionnaire.group.question.choice

Definition Selections made by the user from the list of options.
 Control 0..*
 Type Coding
 Comments Only one of answer[x] or choice can be provided

Questionnaire.group.question.options

Definition Reference to a valueset containing the possible options.
 Control 0..1
 Type Resource(ValueSet)

11.9 Observation

Definition The Observation provides a single score for a certain aspect of an intervention.

Observation.name

Koppeltaal true

required

Definition Describes what was observed. Sometimes this is called the observation "code".

Control 1..1

Type CodeableConcept

Binding ActivityDefinition.observation

Observation.status

Koppeltaal true

required

Definition The status of the result value.

Control 1..1

Type code

Binding <http://www.hl7.org/implement/standards/fhir/observation-status.html>

Observation.reliability

Koppeltaal true

required

Definition An estimate of the degree to which quality issues have impacted on the value reported.

Control 1..1

Type code

Binding <http://hl7.org/fhir/observation-reliability>

Comments Will usually have the value "ok".

Observation.value[x]

Koppeltaal true

required

Definition The score for the specific observation.

Control 0..1

Type CodeableConcept|Attachment|Ratio|Period|SampledData|string

11.10 UserMessage

Definition A message sent from a user or device to a user.

Control 1..1

UserMessage.context

Definition An uri that describes the context of the message.

Control 0..1

Type Uri

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/UserMessage#Context>

UserMessage.code

Definition Allows Koppeltaal to recognize the Other resource as a UserMessage.

Control 1..1

Type CodeableConcept

Binding [OtherResourceUsage](#)

UserMessage.from

Definition The sender of the message.

Control 1..1

Type Resource([Device](#))

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/UserMessage#From>

UserMessage.to

Definition The intended receiver of the message.

Control 1..1

Type Resource([Patient](#))

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/UserMessage#To>

UserMessage.messageKind

Definition Which kind of message this represents.

Control 1..1

Type CodeableConcept

Binding [MessageKind](#)

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/UserMessage#MessageKind>

UserMessage.subjectString

Definition A short description of the content of the message.

Control 1..1

Type string

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/UserMessage#SubjectString>

Comments This attribute is not called 'subject' because FHIR already defines an attribute with that name for a different purpose.

UserMessage.content

Definition The text content of the message. Can be rich text.

Control 1..1

Type string

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/UserMessage#Content>

11.11 StorageItem

Definition A piece of arbitrary data that can be stored through the [Koppeltaal Server Storage API](#). Based on the Other FHIR resource.

Control 1..1

StorageItem.code

Definition Allows Koppeltaal to recognize the Other resource as a StorageItem.

Control 1..1

Type CodeableConcept

Binding [OtherResourceUsage](#)

StorageItem.contentType

Definition MimeType of the binary content represented as a standard MimeType (BCP 13).

Binding Required: BCP 13 (RFCs 2045, 2046, 2047, 4288, 4289 and 2049)] (The mime type of an attachment)

Control 1..1

Type Code

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/StorageItem#ContentType>

StorageItem.content

Definition The actual content, base64 encoded.

Control 1..1

Type base64Binary

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/StorageItem#Content>

StorageItem.PatientIdentifier

Definition The Patient in whose context this data belongs.

Control 1..1

Type Resource([Patient](#))

Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/StorageItem#PatientIdentifier>

StorageItem.objectType

Definition The type of the object being stored.
 Control 1..1
 Type String
 Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/StorageItem#ObjectType>

StorageItem.objectKey

Definition The key of the object. Should be unique for (at least) the combination of PatientIdentifier and ObjectType.
 Control 1..1
 Type String
 Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/StorageItem#ObjectKey>

StorageItem.lastModified

Definition The instant this object was last modified.
 Control 1..1
 Type Instant
 Extension <http://ggz.koppeltaal.nl/fhir/Koppeltaal/Profile/StorageItem#LastModified>

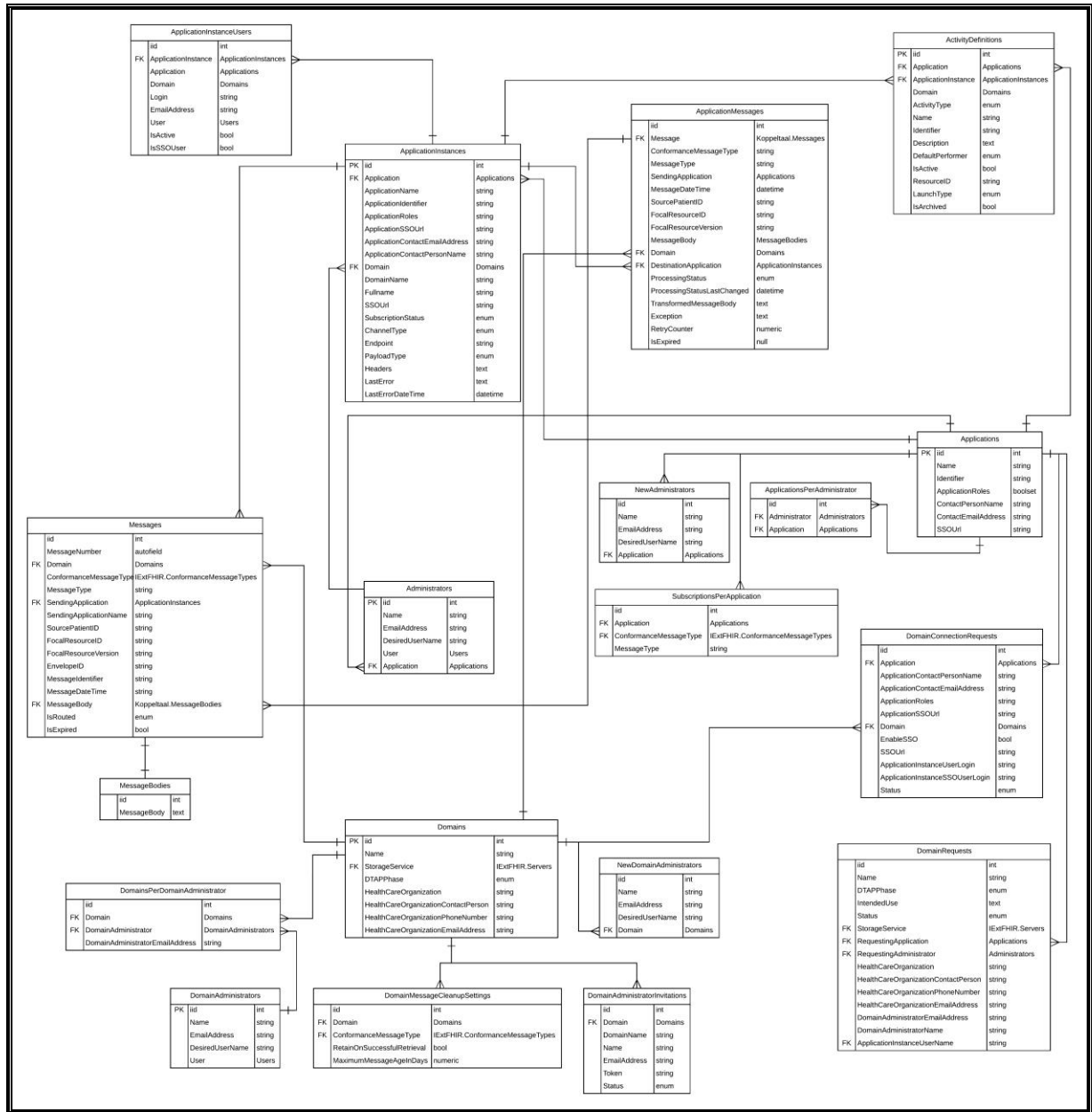
StorageItem.ApplicationIdentifier

Definition Indicates which application stored this item.
 Control 1..1
 Type String
 Comments The Koppeltaal Server ensures this field is filled. It is not possible to set your own value or update it.

StorageItem.DomainName

Definition Indicates which domain this item is part of.
 Control 1..1
 Type String
 Comments The Koppeltaal Server ensures this field is filled. It is not possible to set your own value or update it.

12 Appendix: KoppelTaal server Datadiagram



Figuur 39. KoppelTaal server Datadiagram