

Zaakgericht Werken in het Gemeentelijk Gegevenslandschap



VNG Realisatie

Nassaulaan 12
2514 JS Den Haag

[versie 1.0, Januari 2019]

Inhoud

1.	Inleiding	3
2.	Veranderingen als gevolg van het Gegevenslandschap	4
2.1.	Opknippen Procesapplicaties.....	4
2.2.	Eenmalige opslag vanuit de bron.....	5
2.3.	Vermindering van informatiestromen	5
3.	Schets van de oplossing	6
3.1.	Verandering.....	6
3.2.	Huidige Situatie	6
3.3.	Situatie in een Gegevenslandschap	8
4.	Vraagstukken	10
4.1.	Hoe verhoudt Verzoek zich tot een zaak?	10
4.2.	Waar slaan we procesinformatie op?	12
4.3.	Relatie proces – zaak.....	13
4.4.	Hoe objecten relateren aan een zaak?	13
4.5.	Welke bakjes onderkennen we?	14
4.6.	Wat willen we doen met zaken?.....	15
4.7.	Notificeren over zaken	16

1. Inleiding

Zaakgericht werken is een concept dat veel wordt gebruikt in gemeenten. Het kent vele vormen en implementatievarianten. In de GEMMA is hieraan de laatste jaren meer richting gegeven met het [katern Zaakgericht Werken](#), [Zaakgericht Werken in de praktijk](#) en recentelijk de publicatie van [Zaakgericht Werken in de GEMMA \(informatiearchitectuur\)](#).

Met de beweging “Common Ground” ([commonground.pleio.nl](#)) beogen gemeenten de realisatie van een radicaal nieuw informatie- en applicatielandschap, een “Gemeentelijk Gegevenslandschap”. Hierin wordt de opslag en ontsluiting van gegevens gescheiden van processen en eindgebruikersfunctionaliteit. Gegevens worden dus uit de traditionele procesapplicaties gehaald en zijn in plaats daarvan via gestandaardiseerde API's beschikbaar bij een gegevensbron.

In GEMMA 2, en met name de uitwerking van het zaakgericht werken-deel, is al voor een deel rekening gehouden met de beweging naar een “[Gegevenslandschap](#)”. Zaakregistratiecomponent (ZRC) en Zaakafhandelcomponent (ZAC), Documentregistratiecomponent (DRC) en Documentbeheercomponent (DBC) zijn afzonderlijke referentiecomponenten voor registratie respectievelijk proces/interactie. Zie https://www.gemmaonline.nl/index.php/ZG1_Functies_en_Referentiecomponenten voor uitgebreide beschrijvingen van de componenten en een toelichting hierop.

Toch vraagt de architectuur van het Gegevenslandschap om een aanpassing van de manier waarop we naar de informatiearchitectuur ten dienste van het zaakgericht werken aan kijken. Bovendien behoeft het concept zaakgericht werken zelf enige bijstelling als gevolg van de transitie naar een Gemeentelijk Gegevenslandschap.

Dit stuk onderzoekt deze aanpassingen en schetst enkele oplossingsrichtingen. Het is van belang om voor een aantal van deze punten op korte termijn een oplossing te kiezen. Op dit moment worden immers nieuwe zaakgerichte standaarden (ZGW-API's) ontwikkeld. Omdat de architectuur van het Gegevenslandschap/Common Ground door de Taskforce Samen Organiseren als uitgangspunt voor de toekomstige (collectieve) informatievoorziening van gemeenten is benoemd, is het belangrijk dat deze API's in een zaakgericht werken binnen een Gegevenslandschap kunnen ondersteunen.

Uitgangspunt voor het projectteam dat de nieuwe zaakgerichte standaarden ontwikkelt is GEMMA 2. Waar echter GEMMA 2 nog niet voldoet aan de uitgangspunten van het Gegevenslandschap, gaat het project uit van de architectuur van het Gegevenslandschap. Vraag is hoe backwards compatibility kan worden gerealiseerd. Lukt het om de API's voor zaakgericht werken zo te ontwikkelen dat deze zowel binnen een GEMMA 2-landschap (huidige systemen/leveranciers) als een Gegevenslandschap gebruikt kan worden? Om

deze vraag te kunnen beantwoorden is eerst meer inzicht nodig in hoe (de informatiearchitectuur voor) zaakgericht werken in een Gegevenslandschap vorm krijgt.

2. Veranderingen als gevolg van het Gegevenslandschap

Het Gemeentelijke Gegevenslandschap brengt drie grote veranderingen met zich mee: (1) scheiding van gegevens en proceslogica; (2) (eenmalige) opslag en (meervoudige) bevraging van gegevens bij (of door) de bron en (3) (mede daardoor) zeer beperkte informatie-uitwisseling (geen gegevensdistributie of 'rondpompen', alleen nog notificaties) tussen procesapplicaties.

In GEMMA 2 worden processen zaakgericht uitgevoerd, ondersteund door informatiesystemen die invulling geven aan generieke of specifieke Zaakafhandelcomponenten. De belangrijkste veranderingen als gevolg van het Gegevenslandschap zit in deze componenten. Hieronder gaan we hierop verder in.

2.1. Opknippen Procesapplicaties

In GEMMA 2, en meer bepaald zaakgericht werken hierbinnen, zijn de taakspecifieke processystemen als "[Specifieke Zaakafhandelcomponenten](#)" neergezet. Processen die zaakgericht worden uitgevoerd, worden afgehandeld met een Zaakafhandelcomponent (ZAC, generiek of specifiek).

De generieke ZAC slaat haar gegevens op in de Zaakregistratiecomponent (ZRC). De specifieke ZAC slaat daarentegen alle gegevens zelf op: over de zaak en het proces, het verloop daarvan, en veelal ook de registratie van de objecten waarop dat proces betrekking heeft. Vanuit de specifieke ZAC wordt een zaak geregistreerd (gekopieerd) in de ZRC. Bij wijzigingen in de zaakstatus wordt de informatie in de ZRC vanuit de ZAC bijgewerkt.

Volgens het Gegevenslandschap moeten gegevens gescheiden van de procesapplicatie worden opgeslagen. In plaats daarvan worden ze in één of meerdere registraties ondergebracht, en via gestandaardiseerde API's ontsloten.

Voor de Generieke Zaakafhandelcomponent is hierin al voorzien. Aangezien hierin alleen generieke zaakgegevens worden gebruikt, kan de generieke ZAC gebruikmaken van de gegevens die worden bijgehouden in de Zaakregistratiecomponent (ZRC). Voor een specifieke ZAC waarin veel domein- of processpecifieke gegevens worden bijgehouden heeft de scheiding tussen applicatie en gegevens echter (vergaande) consequenties.

2.2. Eenmalige opslag vanuit de bron

Common Ground gaat uit van het beschikbaar stellen en bevragen van gegevens bij de bron. Kopiëren en distribueren van gegevens (gegevensmagazijnen, gegevensdistributie, ..) wordt in een Gegevenslandschap voorkomen. ‘Beschikbaar stellen bij de bron’ moet evenwel niet altijd letterlijk genomen worden. Anders zouden er precies evenveel registraties als procesapplicaties ontstaan: iedere procesapplicatie zou immers moeten beschikken over een zelfstandige bron voor de gegevens die in die applicatie zijn ontstaan. Dit zou een nogal gefragmenteerde gemeentelijke informatiehuishouding opleveren. Om een overzicht te krijgen van de gegevens die horen bij alle lopende zaken zouden bovendien telkens al deze bronregistraties bevestigd moeten worden.

“Soortgelijke gegevens opslaan bij soortgelijke gegevens” is een meer realistisch uitgangspunt. Zo ontstaan diverse “(kern)registraties”: coherente en integere registraties met meervoudig gebruikte soortgelijke gegevens die vanuit verschillende procesapplicaties worden bijgehouden. Bijvoorbeeld één gemeentelijke kernregistratie medewerkers, en niet een (deel-)registratie per afdeling/proces.

Voor het zaakgericht werken heeft dit als gevolg dat veel objecten die nu in een Zakenmagazijn of procesapplicatie worden bijgehouden, (behandelend medewerker, betreffend object, aanvrager (klant, ..)) vervangen moeten worden door een verwijzing naar de kernregistratie waar gegevens over die objecten zijn opgeslagen.

2.3. Vermindering van informatiestromen

In de huidige gemeentelijke informatievoorziening vindt de interactie tussen processen plaats door middel van gegevensuitwisselingsberichten tussen bij dat proces betrokken applicaties (c.q. referentiecomponenten). De zo verkregen informatie wordt veelal weer opgeslagen door en in de ontvangende procesapplicatie. Denk bijvoorbeeld aan een mutatie van het verblijfsadres van een burger (als gevolg van een verhuizing) die door de GBA- (c.q. BRP-)beheerapplicatie verstrekt wordt aan bijvoorbeeld een vergunningenapplicatie, belastingenapplicatie en sociaal domeinapplicatie.

In een gegevenslandschap is er van dergelijke gegevensdistributie geen sprake meer. Hooguit wordt een proces genotificeerd over een gebeurtenis (zoals een verhuizing). De daarbij betrokken gegevens kunnen vervolgens, indien gewenst, door de genotificeerde applicatie opgevraagd worden bij de betrokken registratie(s). Daarmee kan de voor het proces relevante afhandeling worden uitgevoerd, maar de ontvangende applicaties mag niet voor eigen gebruik een kopie van de gegevens vastleggen. Het versturen van berichten met inhoud tussen verschillende referentiecomponenten wordt dus zoveel mogelijk beperkt en

vervangen door het sturen van notificaties en het opvragen van de bijbehorende gegevens uit een registratie. We noemen dit “informatie-arm notificeren”.

Specifieke aandacht verdienen aanvragen, meldingen en dergelijke die ‘op een website’ (of vanuit een App) ingediend worden. Het indienen wordt ondersteund door de referentiecomponent voor het aanvragen van producten en diensten (“e-Formulierencomponent” in GEMMA 2).

In GEMMA 2 wordt een dergelijke aanvraag gerouteerd (in een bericht) naar de van betrokken generieke of specifieke ZAC. In een Gegevenslandschap kan van dit routeren van gegevens geen sprake meer zijn: gegevens dienen bij de bron opgevraagd te worden. Voor de hand liggend is dat de e-Formulierencomponent die een product- of dienstaanvraag opslaat, bijvoorbeeld in een (kern)registratie verzoeken”, een notificatie stuurt naar de desbetreffende ZAC, die naar aanleiding daarvan de behandeling start (en daartoe onder meer de zojuist genoemde ‘Verzoekenregistratie’ bevrucht).

3. Schets van de oplossing

3.1. Verandering

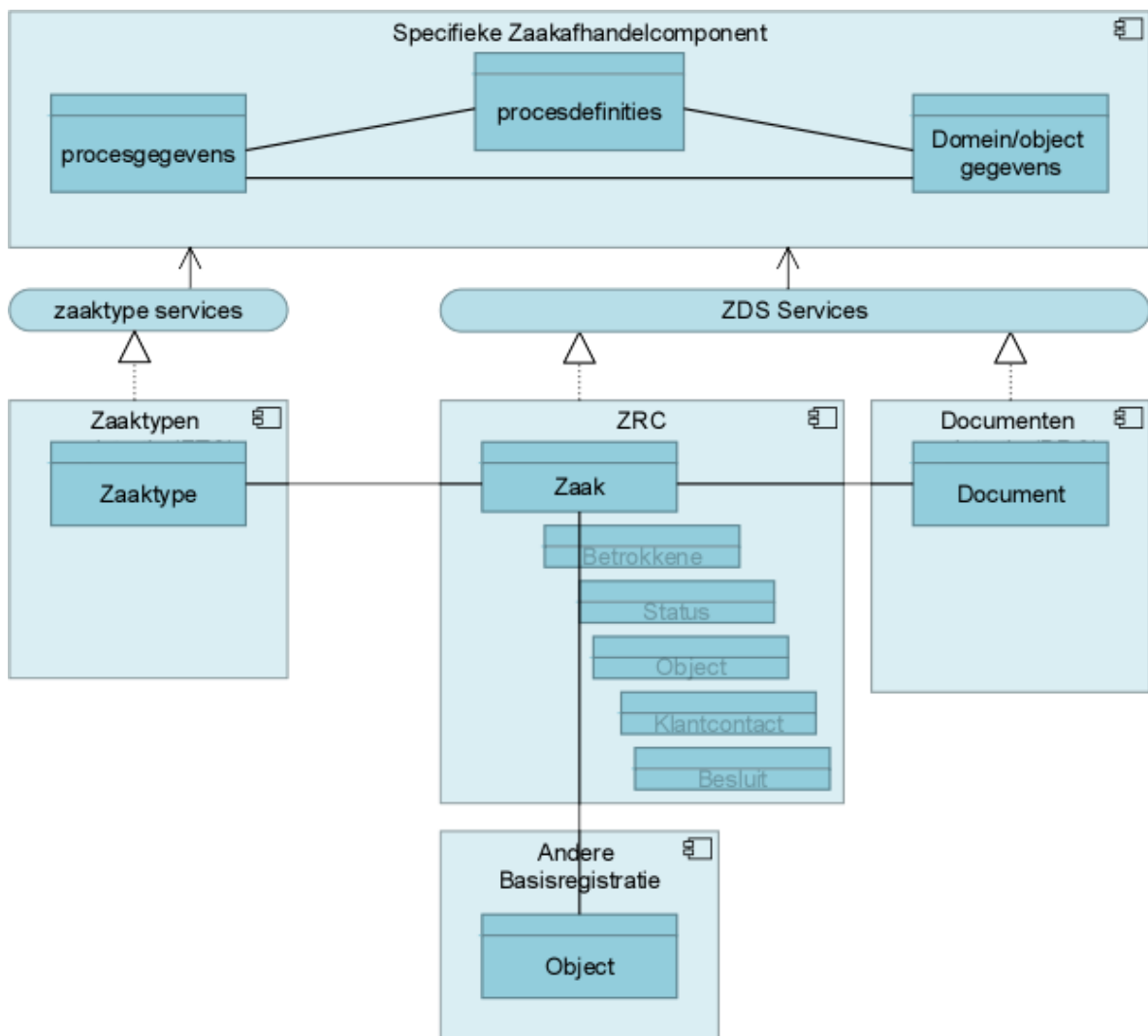
De hierboven besproken verandering past goed bij de ontwikkelrichting die zaakgericht werken al doorlopen heeft: van een zakenmagazijn naast backoffice-silo's die ongeschikt waren voor klantgerichte (online-)dienstverlening, via alles-in-een zaaksystemen, naar zaakgericht registreren en generieke en specifieke zaakafhandelcomponenten.

Veel uitdagingen die aanleiding hebben gegeven tot het zaakgericht werken, zoals het verbeteren van de dienstverlening, een beter inzicht in proces- en voortgang uit backoffice applicaties, en digitaal uniform archiveren zijn in het Gegevenslandschap integraal in applicaties en registraties verwerkt. Zaakinformatie wordt als gevolg hiervan niet meer op één centrale plaats vastgelegd, maar wordt gedistribueerd over meerdere registraties.

Grof geschetst bewegen we van een situatie zoals in figuur 1 naar een situatie zoals in figuur 2.

3.2. Huidige Situatie

Figuur 1 toont een schets van de huidige situatie zoals min of meer beschreven door GEMMA 2.



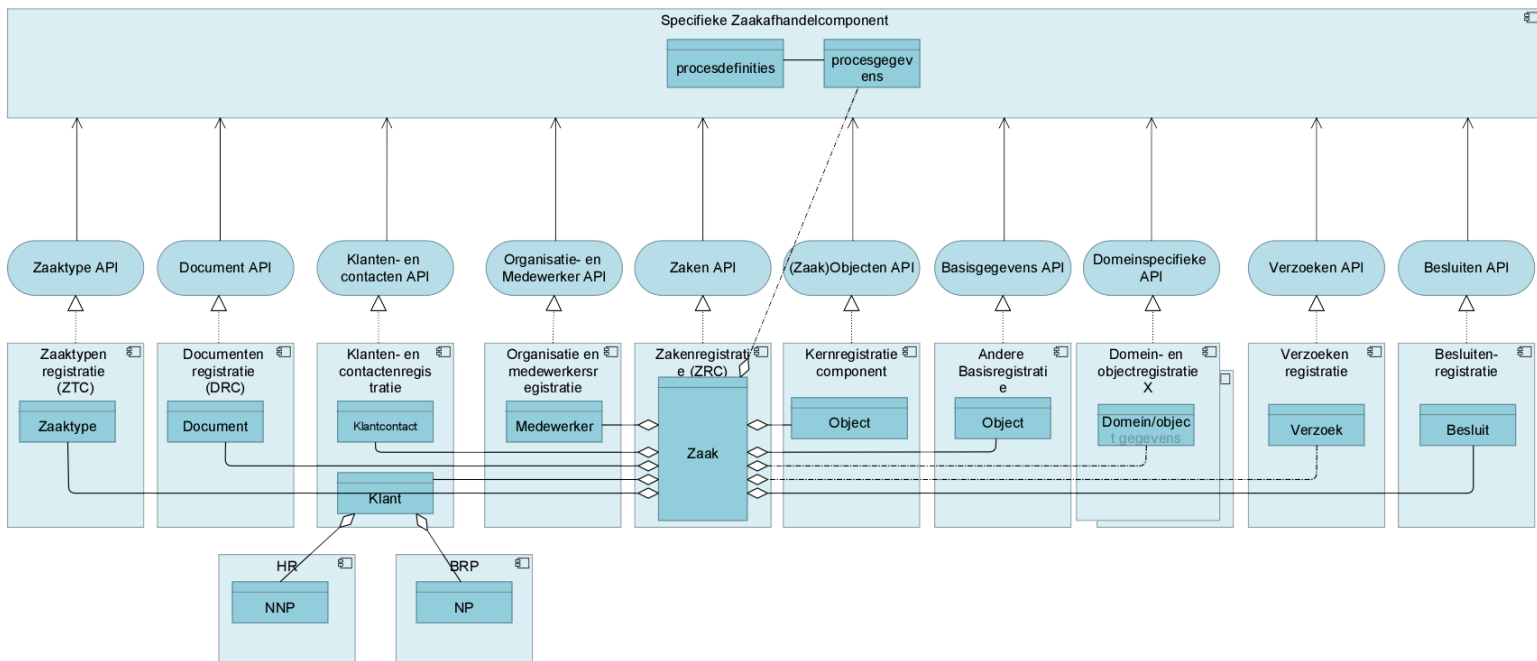
Figuur 1 - ZGW in de huidige situatie (~GEMMA 2)

- Een Zaak is geregistreerd in een ZRC. Aan een Zaak gerelateerde objecten (RGBZ) worden ook geregistreerd in de ZRC: Betrokkenen, Besluiten, Klantcontacten, Zaakobjecten en eventuele Zaaktypespecifieke Objecten met zaaktypespecifieke eigenschappen.
- Vanuit een Zaak liggen er verwijzingen naar Zaaktype (ZTC), document(en)/informatieobject(en) (DRC) en eventuele objecten in Basisregistraties (in theorie), of een gegevensmagazijn (in de praktijk vaak het geval).
- Voor Betrokkene is in theorie ook sprake van een verwijzing (naar BRP en HR, kernregistratie medewerkers, kernregistratie bedrijven, kernregistratie klanten,...), maar in de praktijk worden deze gegevens meestal in de ZRC zelf opgeslagen.
- De Zaakafhandelcomponent bevat nog veel gegevens:

- procesdefinities (hoe moet een proces lopen, welke keuzes zijn er mogelijk);
- procesgegevens (waar in het proces zitten we, welke keuzes zijn in het proces gemaakt, wie is de behandelaar, en op basis van welke gegevens, ..);
- domein/objectgegevens: gegevens over objecten in het domein waartoe het proces behoort. Voor een Subsidiesysteem bijvoorbeeld de gegevens over een aangevraagde subsidie, doelen die men met subsidies wil stimuleren en de voorwaarden en uitnutting van diverse subsidiepotjes.
- “Aanvraag” ontbreekt in dit plaatje. Dat klopt ook. De bijbehorende gegevens maken onderdeel uit van de domein/objectgegevens. In de huidige praktijk wordt een aanvraag nog vaak (onterecht) met een zaak vereenzelvigd. De aanvraag wordt dan in vorm van een Informatieobject als PDF-je opgeslagen bij een zaak. Naast dit “verlies” van gestructureerde gegevens uit het formulier, mis je hierdoor ook de flexibiliteit om meerdere meldingen in één zaak af te handelen, of de behandeling van één melding te splitsen in twee zaken.
- De ZRC kent alleen zaakobjecten en zaaktypespecifieke gegevens. Er bestaat in deze component dus geen rechtstreekse relatie tussen het merendeel van de domein/objectgegevens en een Zaak. Hetzelfde geldt voor gedetailleerde procesgegevens bij een zaak. Met andere woorden: in het geval van complexe processen kent de zaak in de ZRC van het proces alleen de hoofdlijnen (zaakstatussen).
- De hierboven bedoelde relaties tussen proces en de bijbehorende domein/objectregistratie(s), worden bijgehouden in de specifieke ZAC. Daarin is (impliciet of expliciet) een proces geconfigureerd (=procesdefinitie of “procestype”), bijvoorbeeld “Afhandelen evenementensubsidieaanvraag”). Het uitvoeren van dit proces van dat proces volgt dit procestype.
- De specifieke ZAC heeft wel uiteraard weet van welk Zaaktype en welke Zaak er bij een bepaald procestype en bij een bepaalde procesinstantie horen, maar vanuit een zaak is dit dus niet direct opvraagbaar.

3.3. Situatie in een Gegevenslandschap

Door het scheiden van proces en data, eenmalige opslag en bevraging ‘bij de bron’ en het verminderen van informatiestromen ziet de situatie zoals weergegeven in figuur 1 er in een Gegevenslandschap er heel anders uit. De nieuwe situatie is geïllustreerd in figuur 2.



Figuur 2 - ZGW in een Gegevenslandschap

- De domein/objectgegevens gegevens zijn uit de ZAC verdwenen. Procesdefinities en procesgegevens blijven daar echter achter. Dit is ook precies het punt waar procesapplicaties zich op kunnen onderscheiden: goede procesondersteuning.
- Uitgangspunt in het gegevenslandschap is: alle gegevens worden eenmalig opgeslagen. En alle soortgelijke gegevens worden bij elkaar opgeslagen. Er komt dus een registratie met besluiten, een registratie met verzoeken (aanvragen en meldingen), een registratie met klanten, een registratie medewerkers. Etc.
- Objecttypen die ook betekenis hebben of voorkomen buiten het zaakgericht werken worden zo als eigenstandige registratie onderkend.
- De Zaak is een belangrijk verbindend element tussen de objecten in de registraties. In figuur 2 wordt dit getoond door aggregatierelaties met deze objecten. Zaak fungeert zo als 'sleutel' in een grote 'verwijsindex' met links naar objecten in andere registraties. Die kunnen op hun beurt weer relateren aan objecten in andere registraties, bv. de BRP of het HR.
- De Zakenregistratiecomponent is dus niet meer één grote database met daarin alle gegevens over een zaak. In plaats daarvan is sprake van een beperktere registratie van alleen gegevens over zaken, met veel links (URL's) naar objecten in andere registraties (en vice versa). Dit principe toont sterke overeenkomsten met het concept achter Linked data.

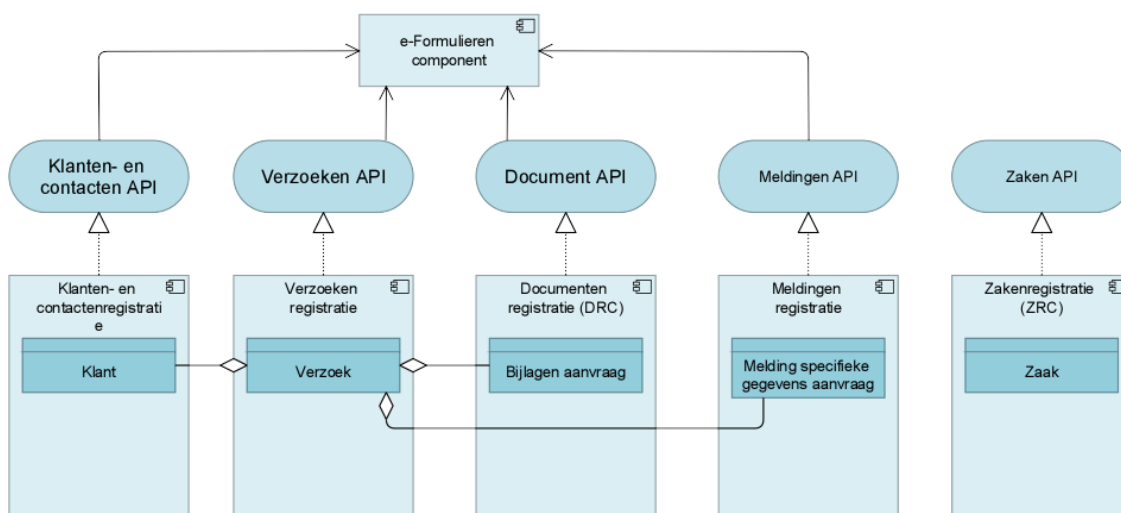
4. Vraagstukken

Figuur 2 vormt een goede illustratie van het concept zaakgericht werken in een Gegevenslandschap. Toch zijn er nog enkele openstaande vragen te beantwoorden. Voor een deel is dit al af te leiden uit de stippellijnen in bovenstaande plaat:

4.1. Hoe verhoudt Verzoek zich tot een zaak?

In figuur 2 is de relatie tussen Melding/Aanvraag (als generieke naam hiervoor hanteren we Verzoek) en Zaak als een stippellijn getekend¹. Hiervoor is gekozen omdat Verzoek niet in het RGBZ voorkomt (zat wel in GFO Zaken), en we dus niet weten hoe Verzoek en Zaak zich tot elkaar verhouden.

Essentie van de relatie tussen de twee lijkt te zijn dat een verzoek een op zichzelf staand 'iets' is, en wordt behandeld als zaak. Er is echter geen sprake van een strikte een-op-een-relatie: één verzoek kan leiden tot meerdere zaken, maar meerdere Verzoeken kunnen ook leiden tot één zaak. Denk aan meerdere meldingen openbare ruimte die als één zaak worden afgehandeld. Verzoek heeft bovendien juridisch een eigen betekenis, bijvoorbeeld in het kader van archivering en het bepalen van de afhandeltermijn ('ontvangstdatum Verzoek').



Figuur 3: Verzoeken

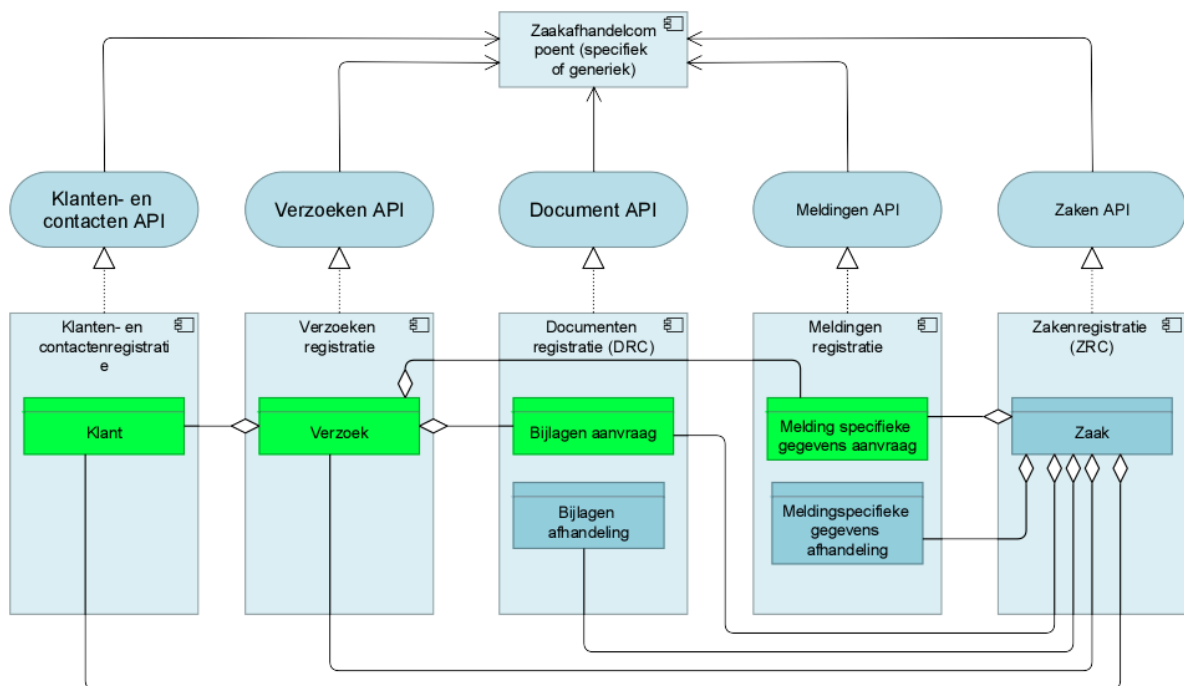
We kiezen ervoor om Verzoeken als aparte registratie te onderkennen en op te nemen in het RGBZ (als apart Informatiemodel Verzoeken met relatie met Zaak). Zo voorkomen we de eerder genoemde omzetting van gestructureerde gegevens

¹ Merk op dat er ook zaakgericht gewerkt kan worden in processen die niet naar aanleiding van een verzoek van de klant worden opgestart, maar bv. n.a.v. een binnengekomen gebeurtenis. In dat geval is er geen verzoek.

die in een formulier zijn ingevuld naar bijvoorbeeld PDF en het bijbehorende verlies van informatie.

Een e-formulierencomponent (of ander systeem waarin een burger of bedrijf een aanvraag kan doen) registreert het ingediend verzoek in de Verzoekenregistratie. De gegevens van de aanvrager (in figuur 3 een burger) worden geregistreerd in de klantenregistratie, voor zover die niet al bestonden. De documenten bij diezelfde aanvraag worden in de documentenregistratie geregistreerd, terwijl verzoeks specifieke gegevens worden opgeslagen in de domeinspecifieke registratie voor het bij de melding horende domein. Denk bij een melding openbare ruimte aan de aard van de melding, omschrijving van de klacht etc.

De relatie tussen verzoek en alle informatie die bij de aanvraag aanwezig was, of eventueel later is aangevuld, wordt vastgelegd en mag niet meer gewijzigd worden. Zo is te allen tijde duidelijk welke gegevens er bij de aanvraag verstrekt zijn.



Figuur 4: Van Verzoek naar Zaak

Als het verzoek is geregistreerd, moet het in behandeling worden genomen. Dit gebeurt door een medewerker die in de regel met een taakapplicatie zal werken. Deze applicatie wordt ofwel genotificeerd over een nieuw binnengekomen verzoek, ofwel controleert de applicatie periodiek, bijvoorbeeld ieder uur, of er nog nieuwe verzoeken van een bepaald type binnen zijn gekomen.

Nadat het verzoek in behandeling is genomen, wordt vanuit de toepasselijke taakapplicatie via de Zaakregistratiecomponent een zaak aangemaakt. Het

Verzoek wordt gerelateerd aan deze zaak (de relatie hiertussen moet in het informatiemodel nog geformaliseerd worden). Objecten die aan Verzoek zijn gerelateerd (groen gekleurd in figuur 4: Klant, Bijlagen, Meldingspecifieke gegevens), worden – indien gewenst – ook rechtstreeks aan Zaak gerelateerd.

Tijdens de behandeling in de Zaakafhandelcomponent ontstaat bijkomende informatie in de Documentenregistratie en de Meldingenregistratie.

De relatie tussen Verzoek en Zaak wordt nog verder uitgewerkt in het informatiemodel. Zie het issue: <https://github.com/VNG-Realisatie/Squad-Architectuur-Samen-Organiseren/issues/154>

4.2. Waar slaan we procesinformatie op?

Een andere vraag is waar we in een gegevenslandschap ‘procesinformatie’ willen opslaan.

De relatie tussen procesdefinitiegegevens en procesgegevens is vergelijkbaar met die tussen zaaktypes en zaken. Procesdefinitiegegevens beschrijven het te volgen proces in de afhandelapplicatie. Uit welke stappen bestaat dat proces? Welke keuzes kunnen binnen het proces gemaakt worden? Welke beslisregels worden gehanteerd? Procesgegevens beschrijven hoe dit proces voor één bepaald geval is doorlopen. Welke behandelaar heeft wanneer welke stap uitgevoerd? Welke keuzes zijn gemaakt op basis van de bekende gegevens?

In figuur 2 is het opslaan van procesinformatie – samen met procesdefinitiegegevens – gepositioneerd in de Zaakafhandelcomponent (zoals die nu in de diverse Taaksystemen zijn opgeslagen). Dit lijkt (mogelijk) in strijd met het uitgangspunt van het Gegevenslandschap. Dat veronderstelt immers dat functionaliteit en gegevens van elkaar gescheiden zijn. In de praktijk is het echter niet eenvoudig om proces- en procesdefinitiegegevens uit de afhandelende systemen te halen. In veel gevallen is de procesdefinitie zelfs impliciet (denk aan hard gecodeerde flow tussen schermen): dat valt dan niet eens los te knippen.

Toewerken naar een situatie waarbij procesgegevens onafhankelijk van de afhandelende applicatie in een leveranciersafhankelijke “procesregistratie” komen te staan lijkt ons niet haalbaar. Het meest generieke formaat wat we over alle processen konden afspreken hebben we al, namelijk zaken en zaaktypen. Dat is voor het sturen en registreren van de afhandeling niet toereikend. Een alternatief is van alle taakapplicaties te eisen dat ze op basis van in BPMN gedefinieerde processen gaan werken, en één centrale processenrepository gaan gebruiken. Deze oplossing lijkt ons, gezien de grote diversiteit aan leveranciers en applicaties op de markt, te ingrijpend. Bovendien is het de vraag of de markt voor afhandelapplicaties voldoende interessant blijft als vrijwel alle specificaties vooraf zijn vastgelegd.

Een laatste aandachtspunt betreft het archiveren. Het duurzaam beschikbaar kunnen maken van procesinformatie is nodig om het handelen als overheid te verantwoorden: het procesverloop moet dus gearhiveerd worden. Wie heeft op welke datum welke stap gezet? Welke procesflow is er doorlopen en welke keuzes zijn daarbij gemaakt? Welke informatie was wanneer beschikbaar? Hoe luidde de (deel)berekening die heeft geleid tot een bepaald resultaat? Dit is informatie die niet in één van de andere registraties past, en voor het grootste deel niet in de ZRC wordt vastgelegd: die ZRC kent immers meestal slechts één behandelaar, en heeft alleen kennis van het proces op hoofdlijnen via de statussen.

De conclusie is dat proces- en procesdefinitiegegevens in de afhandelapplicaties blijven. Idealiter maken deze een scheiding tussen schermen (interactie) en afhandeling (proces), en stellen ze hun procesgegevens in een open formaat (via een API) beschikbaar. Alternatief is een “log” van hun procesverloop als document (PDF) als document aan het dossier van de zaak toe te voegen.

4.3. Relatie proces – zaak

In de figuur staat een stippellijn getekend tussen Zaak en Procesgegevens. “Proces” is niet gedefinieerd in het informatiemodel RGBZ. Daardoor ligt de relatie tussen zaak en proces niet eenduidig vast (in GFO zaken lag dat met behulp van “stap” anders).

Uit het voorgaande, en vooral uit de prominente rol van ‘status’ in het RGBZ, blijkt al dat het moeilijk is om alle soorten processen op een eenduidige manier te registreren. Uiteindelijk kom je dan uit bij een informatiemodel met de rijkheid van BPMN (en voor de meer adaptieve processen iets dat lijkt op CMMN), wat uiteraard niet de bedoeling is.

Echter is het, indachtig de vraag: “waar registreren we procesinformatie?”, noodzakelijk dat de relatie zaak-proces goed is gedefinieerd, zodat als we nadere procesinformatie bij een zaak - indien beschikbaar - bij een zaaknummer kunnen opvragen.

Issue om deze vraag te beantwoorden: <https://github.com/VNG-Realisatie/Squad-Architectuur-Samen-Organiseren/issues/156>.

4.4. Hoe objecten relateren aan een zaak?

Objecten worden in het RGBZ aan een zaak gerelateerd door middel van “Zaakobject”: een Object waarop de zaak betrekking heeft’. In theorie kun je hier veel kanten mee op. In de praktijk is het gebruik hiervan beperkt tot voornamelijk basisregistratieobjecten. In een gegevenslandschap moeten veel meer soorten objecten aan een zaak gerelateerd kunnen worden. De zaak vormt immers de

sleutel om deze bij elkaar te kunnen houden. Denk bijvoorbeeld aan verleende subsidies, uitgegeven marktplaatsen of grafrechten.

In het verleden werden objectenregistraties bijgehouden in het processysteem, waar impliciet de relatie tussen de zaak en deze objecten werd gelegd. Het Gegevenslandschap vraagt om een andere, meer integrale benadering. Het is daarom de vraag of de relatie zaak - object zoals gedefinieerd in het RGBZ (zaak heeft betrekking op object) hier voldoet. Dit laatste geldt zeker voor de complexere processen waarbij domeingegevens uit meerdere objecten zullen bestaan. Relateer je dan al die objecten aan een zaak?

Discussie en verdere verdieping op dit issue: <https://github.com/VNG-Realisatie/Squad-Architectuur-Samen-Organiseren/issues/155>

4.5. Welke bakjes onderkennen we?

In de uitwerking hierboven hebben gezien dat er aparte registraties - “bakjes” - ontstaan voor verschillende aan zaak gerelateerde objecten die nu nog in de ZRC worden bijgehouden. Voor deze registraties moeten er aparte, aan zaak (RGBZ) gekoppelde informatiemodellen komen. De objecten worden via aparte API's ontsloten.

Rondom het zaakgericht werken onderkennen we de volgende registraties:

- Zaaktypenregistratie (nu ZTC)(Zaaktype, Statustypen, Resultaattypen,..);
- Documentenregistratie (nu DRC) (documenten c.q. informatieobjecten);
- Klanten- en contactenregistratie (voor initiators, belanghebbenden, gemachtigden e.d.). Deze registratie verwijst voor persoonsgegevens naar de BRP, en voor bedrijven en instellingen naar het HR. Aanvullend daarop wordt alleen bijkomende informatie, bijvoorbeeld contactvoorkeuren en de momenten waarop contact is geweest met klanten opgeslagen.
- Organisatie en medewerkersregistratie (o.a. voor behandelaars, adviseurs, afdelingen, rollen en bevoegdheden);
- Verzoekenregistratie (verzoeken met verwijzingen naar alle bij een verzoek horende gegevens);
- Besluitenregistratie (besluiten);
- Domein/objectregistraties (domein/objectgegevens; meerdere registraties: één per domein/proces, bijvoorbeeld voor meldingen of subsidies);
- Basisregistraties (personen, bedrijven, panden, terreinen, percelen e.d.);
- Eventuele andere kernregistraties (Zaakobjecten die nog niet genoemd zijn, maar wel behoren tot kerngegevens (meervoudig gebruikt));
- En natuurlijk, last but not least: de Zakenregistratie (nu ZRC);

Let wel dat we hier nadrukkelijk spreken over “registraties” en het in GEMMA 2 gebruikelijke achtervoegsel “-component” hebben weggelaten. Waar deze registraties worden geïmplementeerd is niet zo van belang. Niet iedere registratie is per sé een nieuw referentiecomponent.

4.6. Wat willen we doen met zaken?

Zaakgericht werken kent een lange geschiedenis. Het concept ontstond in de tijd dat dienstverleningsambities hoger werden en backoffice-applicaties deze ambities niet of nauwelijks konden realiseren. Daarna was er de gedachte dat een “alles-in-één-zaaksysteem” voor veel processen voldoende zou zijn. Inmiddels draait de slinger een beetje terug en zien we taakspecifieke systemen moderner en opener worden. In een Gegevenslandschap is dit helemaal het geval.

In de loop van de jaren zijn er diverse doelen voor zaakgericht werken ontstaan, zoals het verbeteren van dienstverlening, het vormen van een basis voor archiveren, en het bieden van managementinformatie. Bij een aantal van deze doelen is het de vraag is of je ze in een modern applicatielandschap nog met zaakgericht werken wil oplossen.

In het geval van een gemeente die met een dominant generiek zaaksysteem werkt ligt een oplossingsrichting op basis van zaken voor de hand. In een Gegevenslandschap veel minder. Bovendien zien we op vele vlakken een terugtrekkende beweging rondom het “gemeentebrede” dienstverleningsconcept, ten faveure van domeinspecifieke apps en portalen. Een meer domeingerichte dienstverlening en informatievoorziening (in plaats van generiek op basis van zaken) is in de toekomst dus waarschijnlijk.

Enkele voorbeelden:

1. Een gemeente wil alle Meldingen Openbare Ruimte op een kaart tonen. Vraag je hiertoe alle melding-zaken op met een generiek ZDS-koppelvlak “toon zaken op de kaart”? Wetende dat de AVG dit voor het overgrote deel aan zaken/zaaktypen niet toestaat? Of kies je voor een meldingen-specifiek koppelvlak, dat mogelijk functioneel rijker is: type melding, prioriteit van de melding, tekstuele toelichting...allemaal wellicht nuttig om op een dergelijke kaart te kunnen weergeven. Maar niet of slechts met moeite (zaaktypespecifieke eigenschappen) met een generiek zaken-koppelvlak te realiseren.
2. Hoe wil je een proceseigenaar/afdelingshoofd van managementinformatie voorzien? Op basis van zaken kun je generieke metrieken weergeven. Aantal zaken per afdeling, aantal zaken per status of zaken die voor of achter lopen op plandatum. Toch is het vergelijken van een subsidiezaak met een vergunningenzaak appels met peren vergelijken. Bovendien zit een

manager vergunningen vermoedelijk op heel andere metrieken te wachten: aantal vergunningen per wijk, of aantal afhandeluren per bouwvolume-eenheid.

3. Hoe wil je status-informatie weergeven aan de klant? Heel generiek op basis van alleen voorgedefinieerde 'zaakstatussen', of wil je meer gedetailleerde informatie bieden binnen een domeinspecifiek portal, bijvoorbeeld specifiek voor subsidies?

Het is belangrijk om als gemeente aandachtig stil te staan bij vragen als die hierboven. Zaakgericht werken is geen panacee voor alle problemen en behoeften. Maar zaakgericht werken is wel een manier van procesgericht werken die zodanig is uitgewerkt dat ook relatieve buitenstaanders een proces kunnen begrijpen, en wel op het niveau dat voor zo'n buitenstaander waarschijnlijk volstaat: de hoofdlijnen. Dit 'Esperanto' blijft een onderscheidend kenmerk van zaakgericht werken en zaakgerichte API's boven domeinspecifieke API's.

De voorgaande paragrafen laten zien dat twee niveaus ontstaan: domeinspecifiek en domeinoverstijgend. Daar waar uniformiteit over de domeinen belangrijk is, gebruiken we zaken (en zaakgerichte standaarden). Is er echter voldoende zwaarwegende behoefte aan domeinspecifieke functionaliteit, dan kiezen we voor een domeinspecifieke oplossing en dito standaarden. De afwegingen rondom de keuze voor de een of de ander moeten van geval tot geval gemaakt worden.

4.7. Notificeren over zaken

In een gegevenslandschap willen we geen zaken (of andere informatie) meer "rondsturen". Bijbehorende vraag is hoe een behandelaar met een taakapplicatie te weten komt dat er een verzoek is binnengekomen dat behandeld moet worden. En dat is maar één van de scenario's waarin het nuttig of nodig is dat een partij actief wordt genotificeerd over een zaak(status). Andere momenten waarop notificaties waardevol kunnen zijn, zijn bijvoorbeeld bij meerdere behandelaars die met meerdere systemen werken (of moeten we dat niet willen?), of wanneer er een nieuw document aan een zaak is toegevoegd door een KCC-medewerker.

Het is nuttig om enkele van deze scenario's uit te werken binnen de context van een Gegevenslandschap en hierbij een generieke oplossing voor te stellen - een verdere uitwerking van

https://www.gemmaonline.nl/index.php/ZGW_in_GEMMA_2_compleet#Notificeren_over_Zaken binnen een gegevenslandschap.

Nog verder uit te werken in issue: <https://github.com/VNG-Realisatie/Squad-Architectuur-Samen-Organiseren/issues/153>