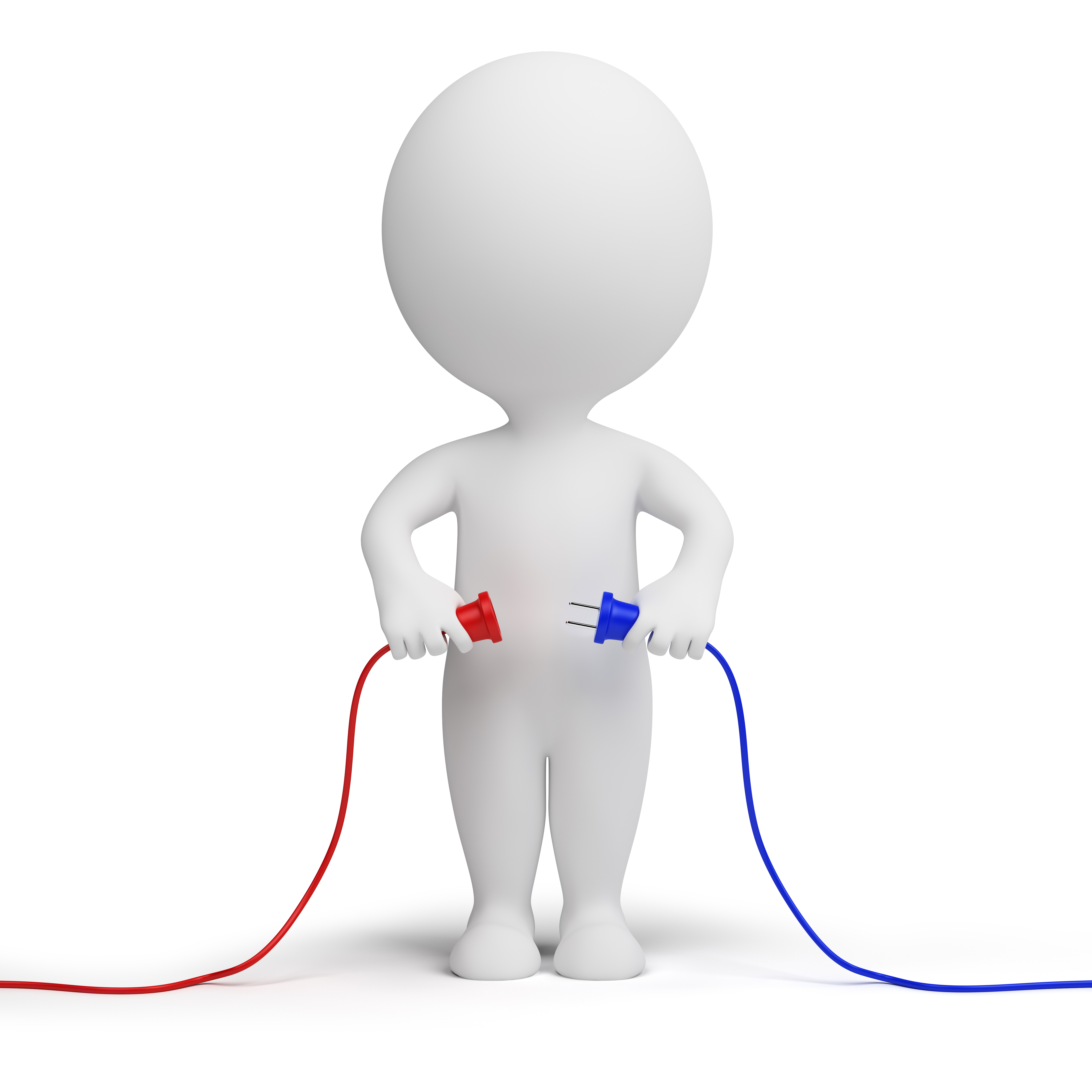
**Functionele inrichtingsprincipes API’s**

**Basis- en Kerngegevens**



Datum: 09 januari 2019

Auteurs: Dennis de Wit

Versie: 0.92

Status: Concept

Inhoud

[1. Inleiding 3](#_Toc534814846)

[2. Principes API's 5](#_Toc534814847)

[3. Patronen API’s 14](#_Toc534814848)

[Bijlage: Kaders en Ontwikkelingen 16](#_Toc534814849)

# Inleiding

Dit document geeft richting aan zowel het proces, het gedrag als de inhoud van het ontwikkelen, beschikbaar stellen en aanroepen van Application Programming Interfaces (API’s) door gemeenten voor het uitwisselen van Basis- en Kerngegevens binnen de Common Ground beweging. Dit document is tot stand gekomen met de medewerking van de volgende gemeenten: Gemeenschappelijke Regeling Drechtsteden, Gemeente Amsterdam, Gemeente 's-Hertogenbosch, Gemeente Haarlem en Gemeente Utrecht[[1]](#footnote-1).

Het document wordt ter vaststelling aangeboden aan het 'Kennisplatform API's' dat is aangewezen binnen Samen Organiseren/Common Ground om kaders te stellen bij de uitrol van de API’s.

**Achtergrond**

In opdracht van VNG Realisatie heeft Solventa B.V. in januari 2017 een advies opgesteld aangaande de behoefte aan vervangende eindproductstandaarden, dan wel toevoegingen aan bestaande eindproductstandaarden[[2]](#footnote-2). Dit advies staat aan de basis van het inhoudelijke deel van dit document.

Vanuit de Common Ground beweging wordt afscheid genomen van het gebruik van halffabricaat- of horizontale sectormodelstandaarden, zoals StUF-BG. De generieke berichten uit de horizontale sectormodellen worden vervangen door API’s. Dit document gaat specifiek in op API’s voor Basis- en Kerngegevens. De Common Ground beweging omvat veel meer en beperkt zich dus tot Basis- en Kerngegevens. In het verlengde van dit document zouden ook voor andere thema’s de functionele inrichtingsprincipes van API’s opgesteld moeten worden.

**API**

Een API definieert de toegang tot de functionaliteit die er achter schuil gaat. Het is de technische specificatie van een dienst. In de context van dit document beschrijft een API de interactie tussen aanbieder (bron) en afnemer (afnemer) om gegevens uit te wisselen.

De API-strategie van de DSO[[3]](#footnote-3) stelt technische kaders voor API’s. Die kaders worden nu ook overheidsbreed gesteld en bieden daarmee ook houvast voor de in dit document beschreven API's.

**Leeswijzer**

In hoofdstuk 2 van dit document worden principes beschreven waarlangs API's ingericht worden voor Basis- en Kerngegevens. Hoofdstuk 3 geeft een toelichting op de patronen die gevolgd worden.

**Terminologie**

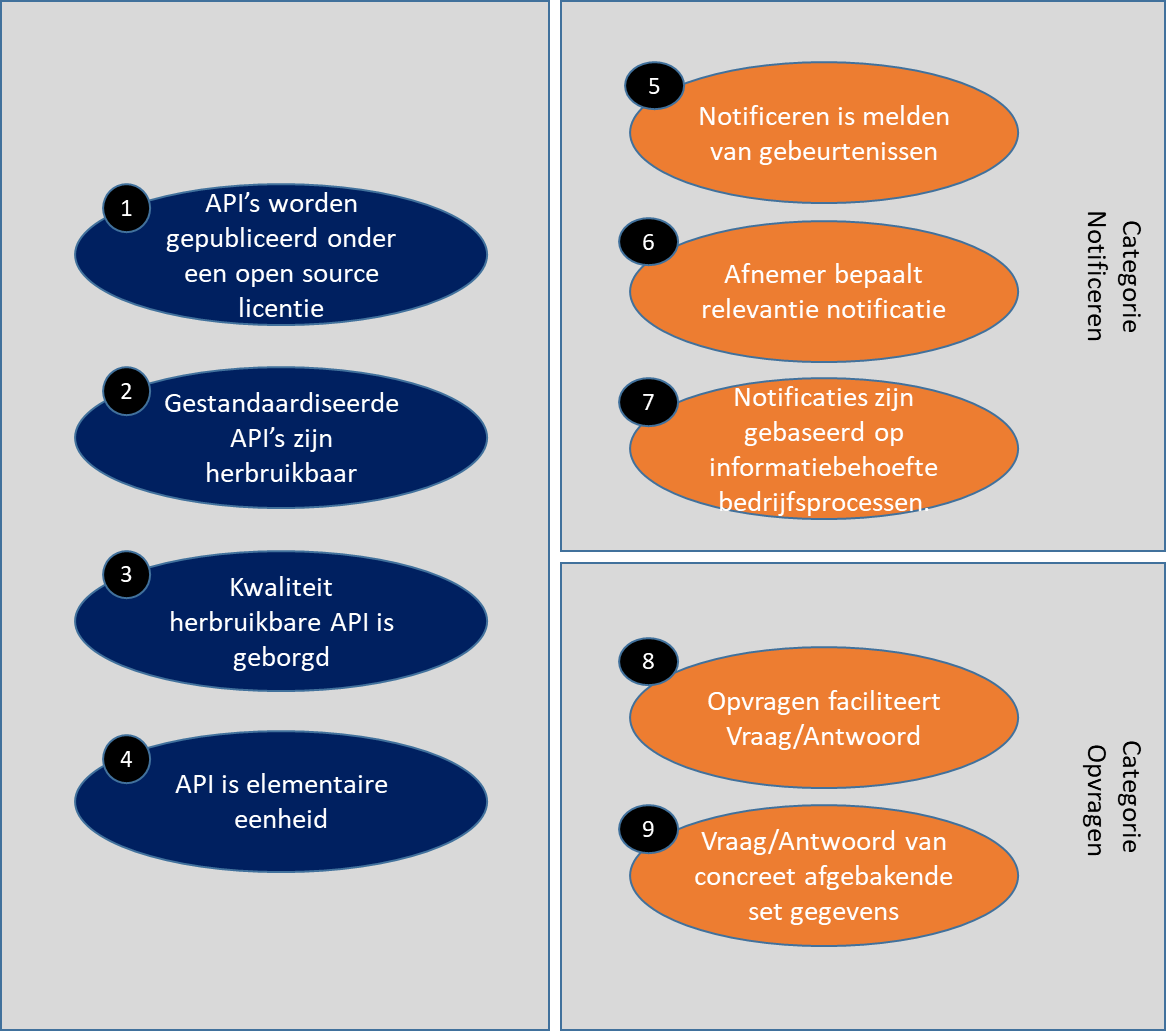
Om dit document op de juiste manier te interpreteren is het van belang dat de termen die gehanteerd worden duidelijk beschreven zijn. In de tabel hieronder volgt de uitleg.

|  |  |
| --- | --- |
| **Term** | **Uitleg** |
| Bronhouder | Een [bronhouder](http://tax.kadaster.nl/resource?subject=http%3A%2F%2Ftax.kadaster.nl%2Fid%2Fbegrip%2FBronhouder) is een organisatorische eenheid die verantwoordelijk is voor het inwinnen en bijhouden van de authentieke en niet-authentieke gegevens in een [basisregistratie](http://tax.kadaster.nl/resource?subject=http%3A%2F%2Ftax.kadaster.nl%2Fid%2Fbegrip%2FBasisregistratie) of kerngegevens in een kernregistratie en voor het borgen van de kwaliteit van die gegevens.  Voor binnengemeentelijke bronhouders is de organisatorische eenheid de afdeling die de bijhouding van de basisregistratie uitvoert (bijvoorbeeld de afdeling Burgerzaken voor de Basisregistratie Personen). Voor buitengemeentelijke bronhouders zijn het Houders Landelijke Voorzieningen (HLV’s) die de bijhouding van de basisregistratie uitvoeren (bijvoorbeeld het Kadaster voor de Basisregistratie Kadaster).  Voorbeelden van kerngegevens: een Belastingapplicatie die (niet authentieke) correspondentieadressen binnengemeentelijk wil distribueren, een Belastingapplicatie die (niet authentieke) persoonsgegevens (bijvoorbeeld buitenlanders die eigenaar zijn van onroerend goed) levert aan een centraal gegevensmagazijn, een BAG-applicatie die gegevens over Buurt en Wijk levert aan een kernregistratie. |
| Afnemer | Een afnemer is organisatorische eenheid die verantwoordelijk is voor het afnemen van gegevens voor gebruik in de eigen processen. Afnemers kunnen gegevens afnemen uit basisregistraties of kernregistraties.  Voorbeelden: een gemeentelijke applicatie voor het Sociaal domein neemt persoonsgegevens af uit de BRP, een Zaaksysteem neemt gegevens uit het Handelsregister af, en een generieke raadpleegmodule (Geo-viewer) neemt gegevens over wijken en buurten af uit de kernregistratie. |
| API | Een Application Programming Interface (API) is een combinatie van technische bestanden, documentatie en andere ondersteuning die helpen bij het aanroepen van externe applicaties. Het is een soort koppelvlak plus documentatie en hulpmiddelen. De Data is geen onderdeel van de API.  De kerneigenschappen van API’s zijn eenvoud, bruikbaarheid en schaalbaarheid. Deze eigenschappen maken het voor een platform goed mogelijk om data en diensten beschikbaar te stellen en zoveel mogelijk gebruikers te bereiken. |
| Gebeurtenis | Een gebeurtenis is een voorval dat in veel gevallen leidt tot het creëren, wijzigen of verwijderen van een data-object/registratie.  Er worden twee soorten gebeurtenissen onderscheiden: een real life gebeurtenis en een systeemgebeurtenis. Als gevolg van een gebeurtenis (lees: trigger, impuls) worden gegevens uitgewisseld.  De melding van een gebeurtenis is niet hetzelfde als de uitwisseling van alle gegevens behorend bij de gebeurtenis.  Voorbeeld gebeurtenis: Persoon met BSN 123456789 is overleden.  Voorbeeld gegevensuitwisseling: Categorie 7 van Persoonslijst BRP. |
| Resource | In dit document wordt het begrip ‘resource’ gehanteerd. Daarbij sluit het aan op de terminologie in het document API-strategie. Een resource is niet per definitie gelijk aan een StUF Objecttype (of wel 'StUF entiteit'). |

# Principes API's

Met de principes uit dit document kunnen de API’s nauwkeurig worden gedefinieerd. Door de principes te gebruiken wordt proces, gedrag en inhoud van de API’s ook in de toekomst geborgd. Bij wijziging van de principes geldt dat dit impact heeft op alle API’s. Bij het opstellen van de principes is rekening gehouden met kaders en ontwikkelingen (zie bijlage 1).

In de visualisatie hieronder is de structuur van de principes zichtbaar. De blauw gekleurde principes zijn overkoepelend. Verder vertalen de principes zich naar een categorisering van API’s die ook in onderstaand figuur wordt weergegeven (zie oranje gekleurde principes). Deze principes (5 t/m 9) hebben specifiek betrekking op Basis- en Kerngegevens.



Figuur 1: Visualisatie van de principes

Hieronder worden de principes uitgewerkt.

|  |  |  |
| --- | --- | --- |
| Principe | API’s worden gepubliceerd onder een open source licentie | #1 |
| Toelichting | API’s worden gerealiseerd door of in opdracht van gemeenten, in samenspraak met de bronhouder en afnemers. In veel gevallen speelt VNG Realisatie (als vertegenwoordiger van alle gemeenten) een rol in het traject om te komen tot een set aan API’s, echter is dit niet noodzakelijk. API’s kunnen ook door lokale initiatieven ontstaan en uiteindelijk resulteren in landelijke gebruik. Eenmaal gerealiseerde API’s vallen onder een open source licentie. API’s zijn voor gemeenschappelijk gebruik en daarom voor alle overheidspartijen vrij beschikbaar. | |
| Ratio | * De overheid investeert in de realisatie van API’s om hergebruik mogelijk te maken, als het eenmaal werkt dan werkt het voor iedereen (op dezelfde manier). Hierdoor betaalt de overheid als geheel maar eenmalig voor de realisatie van een API. * Door aan te sluiten op bestaande documenten (ook opgesteld door of in opdracht van de overheid) wordt de deskundigheid van alle domeinexperts benut, naast de deskundigheid van stakeholders ten aanzien van het binnengemeentelijk gebruik van basisgegevens. | |
| Implicaties | * Softwareleveranciers ontwikkelen API’s in opdracht van een gemeente. De ontwikkeling van een API kan een softwareleverancier maar eenmalig in rekening brengen bij de overheid (dit moet terugkomen in de Gibit inkoopvoorwaarden). Gemeenten wordt een gebruikerslicentie (EUPL) toegekend, zodat meerdere gemeenten er gebruik van kunnen maken. * De definities van API’s zijn voor basisregistraties per definitie direct gebaseerd op, door of namens de overheid opgestelde documentatie (gebeurteniscatalogi, definitie documenten van basisregistratie of informatiemodellen). * Een API kan worden beschreven direct na het beschikbaar komen van een gebeurtenis (of na beschikbaar komen van een gehele gebeurteniscatalogus). * Het gebruik van documentatie door gemeenten is hiermee transparant en ondersteunt de benadering dat de informatiebehoefte van gemeenten bekend is. * De wijzigingen in API’s worden voorspelbaar, deze volgt direct op wijzigingen in gebruikte documentatie. * De verbetering van API’s worden door gemeenten beïnvloed mits zij deelnemen aan de totstandkoming van de gebruikte documentatie. * Een softwareleverancier/gemeente kan specifieke API’s beschikbaar stellen die gebaseerd zijn op door VNG Realisatie geaccordeerde standaard API’s. Dit kunnen (zogenaamde outer) API’s zijn die een combinatie van elementaire (inner) API’s aanroepen. * Het staat iedereen vrij om wijzigingsvoorstellen te doen op bestaande API’s om zo de doorontwikkeling te bevorderen. | |

|  |  |  |
| --- | --- | --- |
| Principe | Gestandaardiseerde API's zijn herbruikbaar. | #2 |
| Toelichting | Gemeenten en andere overheidsorganisaties worstelen met het zelfde vraagstuk van gegevensuitwisseling en toegang tot Basis- en Kerngegevens. De beweging Common Ground en de API-benadering daarbinnen maakt het mogelijk gestandaardiseerde en herbruikbare oplossingen voor gegevensuitwisseling in te richten. Omgekeerd geldt dat herbruikbaarheid vraagt om bepaalde mate van standaardisatie. Dit principe gaat in op de eisen aan standaardisatie.  API's die worden ontwikkeld moeten in lijn zijn met de opgestelde principes. Voorafgaand aan beschikbaarstelling voor hergebruik toetst VNG Realisatie dit (zie principe 3). | |
| Ratio | * Gestandaardiseerde API's zijn voor iedere gemeentelijk toepassingen inzetbaar (het opvragen van BRP-gegevens is identiek voor sociale dienst processen en belastingheffing processen). Door hergebruik van eenmalig ontwikkelde API's voor gegevensuitwisseling kan een softwareleverancier zich juist focussen op het bieden van toegevoegde waarde in de functionele ondersteuning van processen in taakgebieden. Het doel is om een situatie te bereiken waarin API's hergebruikt en gedeeld worden (generiek inzetbaar). * Enkel API's worden gemaakt waar een behoefte aan is (gemeente is initiatiefnemer dus de opdrachtgever/afnemer bepaalt). * Enkel API’s worden gemaakt waarvoor nog geen API aanwezig is, waardoor er niet méér API's gemaakt worden dan nodig zijn. Op enig moment is hierdoor (mogelijk) de totale landelijke behoefte afgedekt. * Nadat een API gerealiseerd is wordt het bestaan ervan gedeeld met andere gemeenten (wordt opgenomen in een centrale respository). Om andere gemeente in staat te stellen deze API ook te gebruiken (plug&play) moet een API voldoen aan bepaalde voorwaarden (zie principe 3). * Een API wisselt gegevens uit tussen twee beschreven endpoints (applicaties). * Gemeente die een API aanbiedt draagt geen verantwoordelijkheid voor gebruik van de API tussen andere endpoints. | |
| Implicaties | * Als een gemeente een informatiebehoefte heeft die niet door een beschikbare API is afgedekt neemt zij het initiatief tot ontwikkeling van die API. * Gemeente maakt afspraken met bronhouder (voor landelijke voorzieningen bij voorkeur samen met VNG Realisatie). * Gemeente ontwikkelt (of laat ontwikkelen) een API. * Documentatie is onderdeel van de realisatie van de API. * Documentatie voldoet aan vormvereisten (nader te bepalen) * Gemeente is verantwoordelijk voor het testen van de API. * Gemeente betaalt de totstandkoming en het beheer[[4]](#footnote-4) van de API. * Voor gemeenten die integraal API's gebruiken moeten de kosten voor het ontsluiten van basis- en kerngegevens drastisch dalen t.o.v. de situatie waarin StUF-berichten gebruikt worden. * Gemeenten stellen het gebruik van API’s als eis in alle programma's van eisen van alle toepassingen waar gebruik van data uit (externe) bronnen relevant is. * Koplopergemeenten zullen meer geld uitgeven aan de realisatie van API's dan gemeenten die volgen.   *Aanname: desondanks zullen de kosten lager zijn dan in de huidige StUF-constructie.*   * Softwareleveranciers hoeven niet zelf te investeren in API’s voor gegevensuitwisseling die al door anderen ontwikkeld zijn en gepubliceerd zijn. * Softwareleveranciers incorporeren het gebruik van de beschikbare API's in hun software. | |

|  |  |  |
| --- | --- | --- |
| Principe | Kwaliteit van herbruikbare API is geborgd. | #3 |
| Toelichting | Het is niet efficiënt als gemeenten bilateraal API's moeten uitwisselen om hergebruik mogelijk te maken. Gemeente wil bij hergebruik weten wat de kwaliteit is en de zekerheid hebben dat het werkt en beheerd wordt. De aanbieder van een API toont de kwaliteit van de API aan bij publicatie. VNG Realisatie toetst steekproefsgewijs de gepubliceerde API's op kwaliteit en heeft daarin een controlerende rol. | |
| Ratio | * Voor feitelijk hergebruik kan plaatsvinden moet een onpartijdig oordeel over de kwaliteit van de API uitgesproken worden. VNG Realisatie kan als vertegenwoordiger van alle gemeenten die onpartijdige toets uitvoeren op de gerealiseerde API's. * De werking van de API moet gegarandeerd zijn voor de hergebruiker (bij toepassing op dezelfde eindpunten) om de kostenreductie van gegevensuitwisseling te kunnen realiseren. | |
| Implicaties | * Bij publicatie van een API is er minimaal één referentie-implementatie, namelijk de gemeente die de API aanbiedt ter toetsing. * Door, of in opdracht van gemeenten ontwikkelde API’s worden landelijk centraal beschikbaar gesteld (als open source) op Github (<https://github.com/VNG-Realisatie/>) t.b.v. hergebruik. * VNG Realisatie toetst een gepubliceerde API bij controle op volgende voorwaarden: * API voldoet aan de kaders van RSGB of het informatiemodel van een Kernregistratie. * API voldoet aan de principes uit 'Functionele inrichtingsprincipes API's voor Basis- en Kernregistratie' (onderhavig document). * API voldoet technisch aan de principes uit 'API strategie DSO'. * VNG Realisatie toetst voorts een ingediende API op volgende voorwaarden: <https://github.com/VNG-Realisatie/Architectuur-en-Standaarden/blob/master/criteria%20voor%20een%20koppelvlakstandaard.md> * VNG Realisatie toetst de API’s niet op inhoudelijke compleetheid, aangezien de API tot stand gekomen is op basis van de functionele wens van de opdrachtgever voor realisatie van de API. * VNG Realisatie publiceert conformiteit van toepassingen van softwareleveranciers aan API's op [www.softwarecatalogus.nl](http://www.softwarecatalogus.nl). | |

|  |  |  |
| --- | --- | --- |
| Principe | Een inner API is een elementaire eenheid. | #4 |
| Toelichting | Een API is de technische specificatie van een dienst. Een API (in de context van dit document) beschrijft de interactie tussen aanbieder (bron) en afnemer (afnemer) om gegevens uit te wisselen.  De functionaliteit die schuil gaat achter een inner API is een zo klein mogelijke dienst die op zichzelf toegevoegde waarde biedt en daarmee bestaansrecht heeft. Naast inner API’s bestaan er ook outer API’s. Een outer API is een API die meerdere inner API’s onder water kan aanroepen). Zowel innner als outer API’s kunnen aan consumers worden aangeboden.  API’s kunnen worden ingedeeld in categorieën om logische groeperingen en structuur aan te kunnen brengen is de hoeveelheid API’s. API’s worden opgedeeld in twee categorieën (verschillende uitwisselingspatronen):   * Categorie Notificeren: API’s voor het melden van een gebeurtenis (proces) * Categorie Opvragen: API’s voor het verstrekken van gegevens (data)   Zie een visuele toelichting op het onderscheid in verschillende soorten API’s in figuur 2. | |
| Ratio | Door API’s als elementaire brokken te beschouwen is iedere API overzichtelijk en eenvoudig.  Er zijn veel API’s te bedenken, voor verschillende functies, voor verschillende sets van gegevens, voor verschillende eigenschappen (bv. historie) et cetera. Het vinden van de juiste granulariteit in het aanbieden van API’s is een taak van de opsteller van de API’s in samenspraak met de afnemers. Essentieel is hierbij dat de RESTful principes hiervoor worden gehanteerd (zie API-strategie DSO).  Alleen de afnemer weet welke combinaties van elementaire eenheden voor hem relevant zijn.  Het onderscheid tussen Opvragen en Notificeren API’s wordt gemaakt om structuur aan te brengen en duidelijk te kunnen duiden welke API’s bedoeld zijn voor welke functionele doeleinden en welke referentiecomponenten de API’s ondersteunen.  Conformiteitstoets kan plaatsvinden op enkelvoudige API’s of op een logische groep van API’s (cluster). | |
| Implicaties | * Iedere combinatie van categorie Opvragen en Notificeren met een basis- of kernregistratie wordt aangeduid met de term ‘cluster’. Voorbeelden van clusters zijn dus: Opvragen BRP, Notificeren BAG. * Clusters bestaan uit één of meerdere elementaire API’s. * Eén API behoort tot maximaal één cluster. * Elementaire eenheden kenmerken zich door eenvoud: makkelijk te realiseren, makkelijk te begrijpen, makkelijk te gebruiken. * De toepassing van een cluster is eenduidig. Het is eenvoudig te bepalen welke clusters en daarmee API’s vereist zijn om een koppelvlak voor een referentiecomponent vast te stellen. | |

|  |  |  |
| --- | --- | --- |
| Principe | Een API in de categorie **Notificeren** faciliteert het melden van gebeurtenissen aan een afnemer. | #5 |
| Toelichting | Een notificatie is een melding van een gebeurtenis die een procestrigger vormt. Een gebeurtenis kan een wijziging in de gegevens van de bronhouder tot gevolg hebben en daarom kan het zinvol zijn voor een afnemer hierover genotificeerd te worden. Gebeurtenissen worden te allen tijde door bronhouder gecommuniceerd richting een broker (bijvoorbeeld ESB/API-Gateway) zo kort mogelijk na het optreden van de gebeurtenis, ongeacht of er afnemers zijn. Indien er een afnemer is brengt de notificatie de afnemer op de hoogte van de gebeurtenis. Dit triggert bij de afnemer een proces: De afnemer is aan zet om te oordelen wat de passende vervolgacties zijn bij het ontvangen van de notificatie.  Omdat een notificatie zelf geen content bevat (wel identificerende functionele gegevens) is er geen historie aanwezig in notificaties.  Notificaties worden toegezonden aan afnemers die middels afnemerindicaties hebben aangegeven notificaties te willen ontvangen. Het kenbaar maken en het beëindigen van deze wens vindt plaats middels een API binnen dit deze categorie. | |
| Ratio | * De notificatie van een gebeurtenis is niet hetzelfde als de uitwisseling van gegevens die de gebeurtenis beschrijven. De gegevens die de gebeurtenis beschrijven (de content) zijn beschikbaar via API’s in de categorie Opvragen.   *Voorbeeld gebeurtenis: Persoon met BSN 123456789 is overleden.*   * Een afnemer bepaalt de interesse in notificaties (via het plaatsen van afnemerindicaties). * De afnemer bepaalt de relevantie van een individuele notificatie. Dit is intelligentie binnen de functionaliteit van de afnemer. Het ontvangen van een gebeurtenissen kan voor de afnemer een trigger zijn voor het initiëren van een bedrijfsproces (en daarmee het opvragen van gegevens). | |
| Implicaties | * De afnemer is verantwoordelijk voor het afnemen van gegevens. * Door de afnemer middels een notificatie slechts te informeren over een gebeurtenis bepaalt de afnemer zelf of en hoe de gebeurtenis wordt opgevolgd. * Indien de afnemer na ontvangst/ van een notificatie gegevens wenst te ontvangen kan de afnemer – na bepaald te hebben welke gegevens – deze middels API’s uit categorie Opvragen verkrijgen. * Afnemers die enkel op basis van interne triggers acteren hebben geen behoefte aan notificaties m.b.t. wijzigingen op bestaande objecten. De actuele gegevens van deze objecten worden namelijk opgehaald indien een intern proces start. * De verantwoordelijkheid van de afnemer omvat de doelbinding. De afnemer is verantwoordelijk de opvolging van een gebeurtenis uit te voeren binnen zijn kader van doelbinding. * De afnemer volgt een vast patroon bij het notificeren (zie figuur 2 in hoofdstuk 3 voor de visuele weergave van dat patroon). | |

|  |  |  |
| --- | --- | --- |
| Principe | Afnemer bepaalt relevantie van een notificatie. | #6 |
| Toelichting | Afnemers die via API Notificeren gegevens willen ontvangen nemen op het niveau van een resource (dus niet instelbaar op een specifiek data-object) in de breedte van het geografische verzorgingsgebied gebeurtenissen af.  Uitzondering: Voor de BRP (en eventuele registraties of data waar privacy op rust) is hier gelet op de privacywetgeving een uitzondering op. Voor het afnemen van personen uit de BRP dient per persoon een specifieke afnemerindicatie geplaatst te zijn door de afnemer, in lijn met zijn doelbinding. Geadviseerd wordt afnemerindicaties te plaatsen door het uitwisselen van het 'BSN' en niet door het uitwisselen van een technische sleutel. | |
| Ratio | * De complexiteit van het binnengemeentelijk distribueren van gegevens uit basisregistraties wordt enorm vereenvoudigd door afnemers enkel te notificeren over gebeurtenissen. * De afnemer bepaalt de relevantie van de notificaties in het licht van zijn eigen bedrijfsprocessen, al deze wijzigen kan de afnemer zonder tussenkomst cq. zonder afhankelijkheid van een bronhouder zijn verwerkingsroutine hierop aanpassen. | |
| Implicaties | * Bronhouder communiceert gebeurtenissen (vergelijk een radio-uitzending). * Het is niet aan de bronhouder van gegevens om te bepalen wie geïnteresseerd is in gegevens/gebeurtenissen. * Afnemers zijn bevoegd (als doelbinding aansluit) te luisteren naar de gebeurtenissen, maar zijn het niet verplicht. * De afnemer abonneert zich bij de bronhouder op alle of juist specifieke gebeurtenissen behorend bij een resource (zie figuur 3 in hoofdstuk 3). * Een uitzondering vormt het volgen van gebeurtenissen op persoonsgegevens (of objecten die persoonsgegevens bevatten). Afnemerindicaties worden hierbij wel op persoonsniveau geplaatst. | |

|  |  |  |
| --- | --- | --- |
| Principe | Notificaties zijn gebaseerd op de informatiebehoefte van bedrijfsprocessen. | #7 |
| Toelichting | Afhankelijk van de informatiebehoefte bij afnemers voor het triggeren van bedrijfsprocessen worden notificaties onderkend.  In grote lijnen komt dit overeen met de gebeurtenissen die in de ontwerpdocumenten of berichtencatalogi van de basisregistraties zijn onderkend. | |
| Ratio | De overheid stuurt op gebeurtenis-gericht beschikbaar stellen van basisgegevens. De bronhouder is weliswaar de meest deskundige partij voor het bepalen van haar gebeurtenissen, echter weet de afnemer het beste in welke gebeurtenissen hij geïnteresseerd is. Aangezien bepaalde gebeurtenissen bij de bronhouder voor geen enkele afnemer interessant zullen zijn zal het aantal gebeurtenissen waarover genotificeerd wordt naar afnemers altijd kleiner zijn dan het daadwerkelijke aantal gebeurtenissen. Met deze regel wordt die deskundigheid maximaal benut. Bij wijzigingen in de gebeurtenissen is tevens geborgd dat het gegevensverkeer dit volgt.  Niet alle gebeurtenissen uit de catalogi van de basisregistraties zijn echter voor binnengemeentelijk gebruik interessant. Inventariseer welke gebeurtenissen niet relevant zijn en neem die niet op in de te realiseren API’s. | |
| Implicaties | * De definitie van API’s in de categorie Notificeren is bepaald door de behoefte van de afnemers in combinatie met de gebeurteniscatalogi van bronhouders. | |

|  |  |  |
| --- | --- | --- |
| Principe | Een API in de categorie **Opvragen** faciliteert het uitwisselen van gegevens op basis van een vraag-antwoord constructie. | #8 |
| Toelichting | De categorie Opvragen wordt gebruikt voor het op afroep verkrijgen van gegevens uit een bron via synchrone vraag-/antwoordcalls.  De vragen die in RSGB-bevragingen zijn onderkend vormen de basis, deze worden één-op-één vertaald naar API’s, Voor de API's die daarin niet beschreven worden vormt het RSGB[[5]](#footnote-5) het kader voor het opstellen van API's. Voor kernregistraties (niet beschreven in het RSGB) zal een lokaal informatiemodel de basis vormen.  Naargelang de informatiebehoefte van de afnemer kunnen API’s in de categorie Opvragen worden ingericht voor het verstrekken van actuele gegevens en/of materiële historie en/of formele historie. | |
| Ratio | * De clusters Opvragen voor de verschillende basisregistraties zijn waar mogelijk gebaseerd op RSGB-bevragingen: deze bevragingen zijn opgesteld vanuit het perspectief van de afnemer, door deze afbakening over te nemen sluiten de API’s nauw aan op het perspectief van de afnemer. * In informatiemodellen zit de mogelijkheid om historie vast te leggen op specifieke gegevens. Dit wordt zeker niet door iedere afnemer veelvuldig gebruikt en de materie kan complex zijn, maar er zouden wel API’s in de clusters Opvragen moeten zijn die de historie kunnen opvragen indien die behoefte er is (dit geldt voor zowel formele als materiële historie). | |
| Implicaties | * De definitie van API’s in voor de clusters Opvragen is bepaald in de standaard RSGB-bevragen. * Index API's worden ondersteund om *fuzzy search* mogelijk te maken. * De afnemer (en ook uitsluitend de afnemer) kent de doelbinding van af te nemen gegevens. * Afnemers zijn verantwoordelijk voor, en deskundig in het feitelijk afnemen van gegevens uit basis- of kernregistraties, zij kennen de condities waaronder zij gegevens wel of niet willen afnemen. * Gegevensuitwisseling staat primair ten dienste van de afnemer maar volgt de mogelijkheden en beperkingen van de bron. * Een afnemer neemt gegevens af bij de bron. De basisregistratie worden in deze als bron gezien en niet bijvoorbeeld de afzonderlijke burgerzakenapplicaties. * Opvragen vindt per definitie synchroon plaats. * Door direct afnemen bij de bron te faciliteren is het afschaffen van lokale gegevensmagazijn mogelijk. Dit reduceert de beheerskosten aanzienlijk. * Door direct afnemen bij de bron te faciliteren is het afschaffen vertaalsoftware van landelijke naar gemeentelijke berichtenstandaarden mogelijk (in technische termen: transformaties op ESB-niveau, als voorbeeld hoe momenteel wordt getransformeerd tussen StUF-HR en StUF-BG). | |

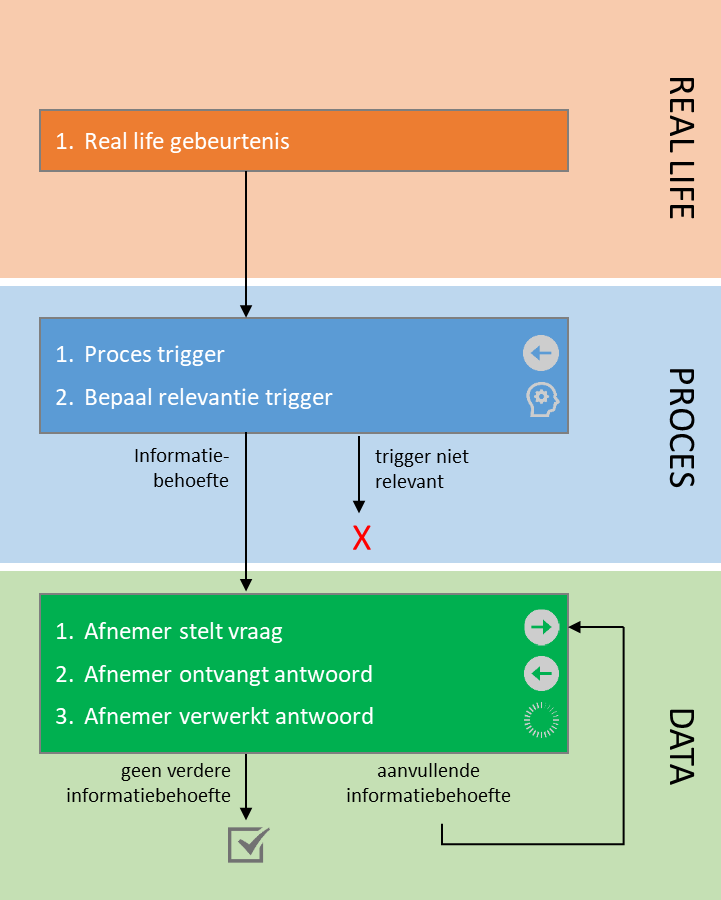
|  |  |  |
| --- | --- | --- |
| Principe | Een vraag/antwoord API ondersteunt het opvragen van een concreet afgebakende set gegevens. | #9 |
| Toelichting | Elke response van de categorie Opvragen omvat maximaal (afhankelijk van autorisatie, zie [design decision BRP API](https://github.com/VNG-Realisatie/Bevragingen-ingeschreven-personen/blob/master/docs/design_decisions.md#de-api-filtert-terug-te-geven-gegevens-op-autorisatie-van-de-organisatie)) de attributen, relaties en de identificerende gegevens van de gerelateerde, maar niet de gerelateerde resources in zijn geheel. Het verkrijgen van de gerelateerde resources vereist het stellen van een nieuwe vraag (nieuwe API-call uit categorie Opvragen). Het gebruik van *expand = true* is niet toegestaan. | |
| Ratio | * Opvragen van gerelateerde gegevens wordt geacht te worden gedaan door het initiëren van een aanvullende vraag-antwoordcall, of indien dit voordeel oplevert gerelateerden direct op te halen via een ‘expand’ in de call op te nemen. * Relaties uit dezelfde bron kunnen embed worden. Zie [design decision BRP API](https://github.com/VNG-Realisatie/Bevragingen-ingeschreven-personen/blob/master/docs/design_decisions.md#relaties-uit-dezelfde-bron-kunnen-embed-worden). * Voor het gebruik van de API is het gebruiken van expand=true om alle relaties embed te krijgen niet toegestaan. Het embedden van gerelateerde resources moet bewust worden gebruikt. Zie [design decision BRP API.](https://github.com/VNG-Realisatie/Bevragingen-ingeschreven-personen/blob/master/docs/design_decisions.md#gebruik-van-expandtrue-wordt-uitgesloten) * Relaties kunnen maximaal één niveau diep worden embed. Zie [design decision BRP API](https://github.com/VNG-Realisatie/Bevragingen-ingeschreven-personen/blob/master/docs/design_decisions.md#relaties-kunnen-maximaal-%C3%A9%C3%A9n-niveau-diep-worden-embed). * Autorisatie wordt gerealiseerd in een specifiek daarvoor ingerichten autorisatie-mechanisme. Vaak wordt dit in een tussenlaag (broker) gerealiseerd. De bron beschikt niet over middelen om applicatie of zelfs gebruikers-specifiek autorisatie in te richten. Wel kan een bron dit op organisatie-niveau. | |
| Implicaties | * De afnemer is verantwoordelijk voor het afnemen van gegevens. * De afnemer volgt een vast patroon bij het opvragen van gegevens (zie figuur 2 in hoofdstuk 3 voor de visuele weergave van dat patroon). * De provider bepaalt het maximum aantal responses en niet de consumer. De API geeft een foutmelding wanneer er teveel resultaten zijn op de vraag. * De afnemer kent de doelbinding van het verzoek en stuurt de aanduiding van de doelbinding mee in de call. * Met parameter ‘fields’ in de call is het mogelijk om op attribuutniveau aan te geven welke gegevens gewenst zijn in de response. Het is niet verplicht een fields-parameter op te geven. * Voor sommige API’s is het verplicht een combinatie van zoek parameters te gebruiken (zo mag de BRP bijvoorbeeld niet enkel op huisnummer bevraagd worden). * Er vindt een validatie plaats op zoek parameters. Voor het gebruik van invalide parameters wordt een foutmelding teruggegeven aan de consumer. * Een gemeente draagt zelf zorg voor een fijnmazig autorisatie-mechanisme om binnen de organisatie onderscheid te kunnen maken in autorisatie per applicatie en of gebruiker. | |

# Patronen API’s

Voor het notificeren en opvragen van gegevens worden patronen gevolgd. Deze patronen worden hieronder visueel weergegeven (zie figuur 2). Merk op dat het patroon van opvragen ook terugkomt bij notificeren, dit is een resultaat van de ontwikkeling: "Van kopiëren van data naar bevragen bij de bron" (zie bijlage 1).

Toelichting bij de visualisatie:

Real-life gebeurtenissen liggen ten grondslag aan alles wat in systemen processen triggert. Indien een gebeurtenis wordt gevolgd door een afnemer (via een geplaatste afnemerindicatie) ontvangt de afnemer hierover notificaties. Op basis van ontvangen notificaties bepaalt een afnemer of de gebeurtenis voor hem interessant is. Indien de notificatie voldoende interessant is kan het leiden tot het opstarten van een vraag/antwoord verwerking. De afnemer stopt met het stellen van vragen als er geen verdere informatiebehoefte is. Merk op dat het verkrijgen van data enkel via vraag/antwoord plaatsvindt.

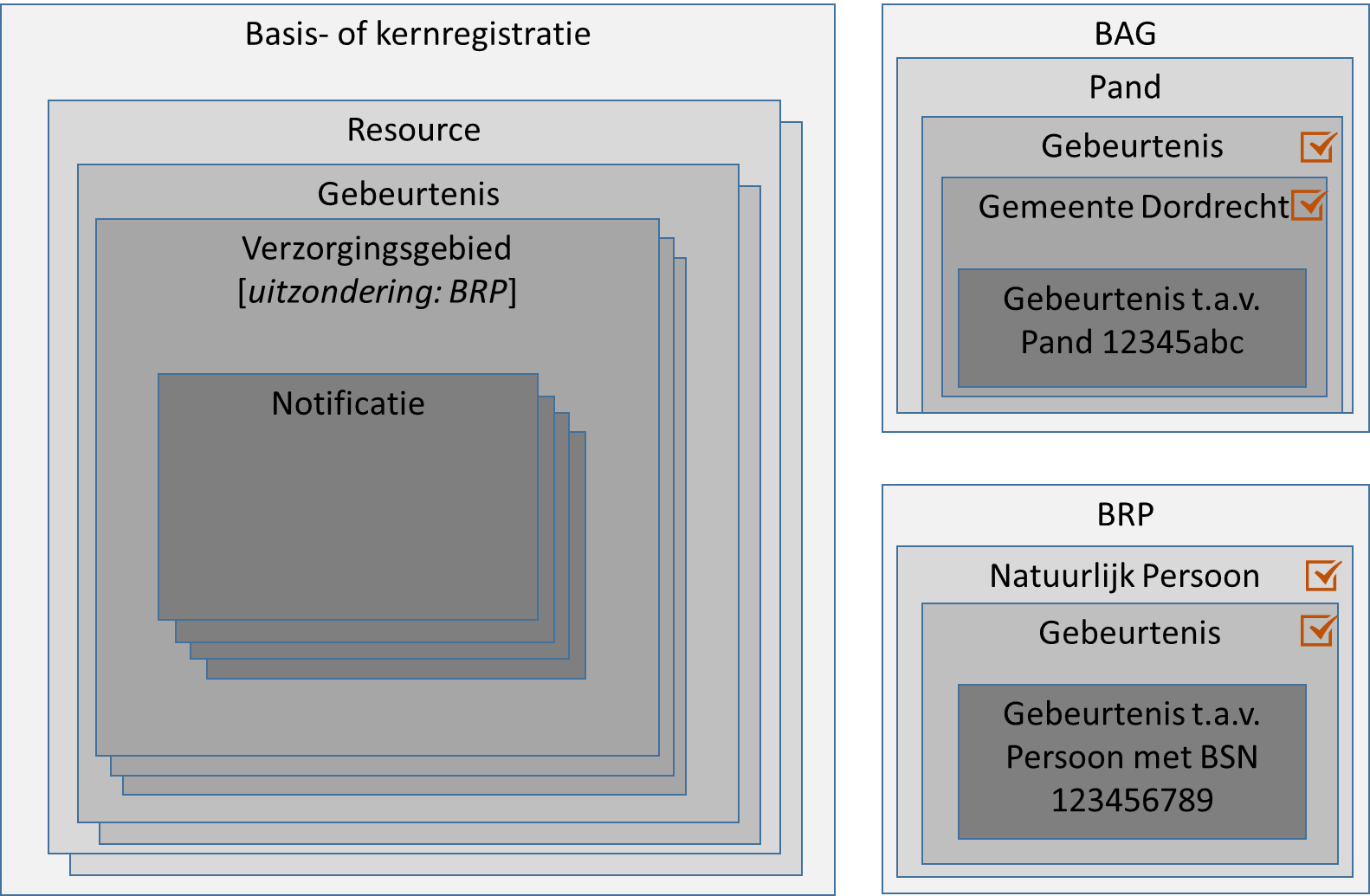


Figuur 2: Visualisatie van gelaagdheid tussen real life, proces en data.

**Notificeren**

Afnemerindicaties worden geplaatst op niveau van gebeurtenis en verzorgingsgebied. Hierdoor ontvangt de afnemer de gebeurtenissen van alle resources in dat geografische gebied. Het ontvangen van deze gebeurtenissen kan voor de afnemer een trigger zijn voor het opvragen van gegevens.

BRP (en eventueel ook HR) vormt een uitzondering, hiervoor geldt dat afnemerindicaties op persoonsniveau i.c.m. gebeurtenis worden geplaatst (het verzorgingsgebied is niet van belang en is dus gelijk aan heel Nederland). Hieronder wordt dit visueel weergegeven.



Figuur 3: Visualisatie van de wijze van notificeren en plaatsen van afnemerindicaties

Toelichting op de voorbeelden:

* BAG: een afnemer volgt één of meerdere BAG-gebeurtenissen uit een bepaald verzorgingsgebied (zoals hierboven gemeente Dordrecht). Een voorbeeld van een BAG-gebeurtenis is een wijziging gebruiksdoel (zie 4.2 voor de lijst met gebeurtenissen voor de BAG). Een notificatie wordt verzonden naar de afnemer indien de combinatie gebeurtenis/verzorgingsgebied optreedt en gevolgd wordt door de afnemer.
* BRP: een afnemer volgt één of meerdere BRP-gebeurtenissen van een bepaald ingeschreven natuurlijk persoon. Een voorbeeld van een BRP-gebeurtenis is een wijziging adresgegevens (zie 4.1 voor de lijst met gebeurtenissen voor de BRP). Een notificatie wordt verzonden naar de afnemer indien de combinatie persoon/gebeurtenis optreedt en gevolgd wordt door de afnemer.

# Bijlage: Kaders en Ontwikkelingen

**Kaders**

De beschrijvingen in GEMMA2.0 gelden als kader bij het afbakenen van API’s. Daarmee zijn ook NORA en RSGB relevante kaders.

*API-strategie van de overheid*Gezien de ontwikkeling naar een digitale samenleving waar veel digitale diensten op een eenvoudige wijze met elkaar kunnen, willen en moeten samenwerken dient er conform een API-strategie gewerkt te worden.

API’s worden technisch gezien uitgerold conform de bepalingen binnen de API-strategie. Een aantal van de belangrijkste kaders die gesteld worden vanuit deze API-strategie:

* API’s zijn op een uniforme manier opgezet en werken op een uniforme manier (zie paragraaf 2.2 API-strategie).
* API’s zijn in principe, conform het open stelsel gedachte, voor derden beschikbaar. Hierdoor kunnen derden aanvullende toepassingen ontwikkelen op basis van de data en API’s van het stelsel (zie paragraaf 2.2 API-strategie).
* API’s moeten toegankelijk genoeg zijn om een brede ontwikkel-community aan te spreken (ontwikkelvriendelijk zijn). (zie paragraaf 2.2 API-strategie).
* Formele koppelvlakken zijn op basis van Digikoppeling (zie paragraaf 2.2 API-strategie).
* Alle API’s zijn goed gedocumenteerd (zie paragraaf 2.6.3 API-strategie).
* API’s bevatten altijd een versienummer (zie paragraaf 2.6.4 API-strategie).
* JSON is het primaire representatieformaat voor API’s (zie paragraaf 2.6.5. API-strategie).

**Ontwikkelingen**

* Afnemen staat centraal (de informatiebehoefte van afnemers is leidend): van kopiëren van data naar bevragen bij de bron.

Het RSGB kadert het informatiemodel voor de basisgegevens binnen het gemeentelijk speelveld af. Het RSGB bevat een deelverzameling van de basisregistraties die door binnengemeentelijke afnemers worden gebruikt. RSGB conformeert zich daarbij maximaal aan het domeinmodel van de basisregistratie en wijkt alleen af als dat nodig is voor de integratie van de verschillende basisregistraties.

* Gebeurtenis gebaseerd  
  Er is een trend in het uitwisselen van gegevens op basis van gebeurtenissen. De landelijke basisregistraties worden bij vernieuwingen hier meer en meer op ingericht. De omschakeling van ‘gegevens uitwisselen’ naar ‘gegevens op basis van gebeurtenissen uitwisselen’ heeft behoorlijke impact op de inrichting van de informatievoorziening. Dientengevolge zal deze trend ook impact hebben op de standaarden voor gegevensuitwisseling.
* Van silo’s naar lagen

Alle applicaties dienen dezelfde software infrastructure te hanteren. Onderscheid in de software tussen de volgende lagen 'interaction', 'control', 'integration', 'services' en 'data'. Dit is conform het 5-lagen model dat voorgeschreven is vanuit de Common Ground beweging. Domein specifieke silo’s zijn daarmee eindig.

* Grip op data

Om grip te krijgen op het beschikbaar stellen van data binnen de overheid en de verwerking ervan verschuift autorisatie per oplossing naar autorisatie op centraal niveau (binnen de API-gateway). Een oplossing hiervoor wordt verder uitgewerkt binnen een landelijke (technische) voorziening voor ontsluiting van services, namelijk NLX.

*NLX; is een open source peer-to-peer system dat federatieve authenticatie, beveiligde verbindingen en protocollering faciliteert in een dynamisch en grootschalig API-landschap tussen veel verschillende organisaties.*

1. Allen nog niet bekend met laatste versie (0.92). [↑](#footnote-ref-1)
2. <https://www.gemmaonline.nl/images/gemmaonline/0/08/Eindproductstandaarden_-_vervanging_StUF-BG_en_StUF-ZKN.pdf> [↑](#footnote-ref-2)
3. https://github.com/Geonovum/KP-APIs-APIStrategie/blob/master/API%20strategie%20voorbeelden/dso-api-strategie\_-\_12\_maart\_2018.pdf [↑](#footnote-ref-3)
4. Of komt er een landelijke beheerorganisatie hiervoor? Dit is onder de aandacht bij VNG-Realisatie. [↑](#footnote-ref-4)
5. Het RSGB 3.0 omvat niet geheel de verzameling van informatiebehoeftes van gemeentelijke afnemers op het gebied van Basisregistraties. Hieraan wordt gewerkt [↑](#footnote-ref-5)