



Common Ground

Realisatieprincipes

Leeswijzer

Dit document beschrijft de realisatieprincipes van Common Ground.

Common Ground heeft als doel om een nieuwe, flexibele en meer gezamenlijke informatiehuishouding voor gemeenten te realiseren. In deze notitie worden de door Common Ground gehanteerde *realisatie*principes beschreven.

Deze *realisatie*principes zijn nauw verbonden met de *informatiearchitectuur*principes voor het Gemeentelijk Gegevenslandschap, die de neerslag van Common Ground-uitgangspunten in informatiearchitectuur vormen. Die principes worden in voorliggend document vertaald naar praktische uitgangspunten, die beschrijven hoe we werken en welke keuzes we daarbij maken. De *informatiearchitectuur*principes voor het Gemeentelijk Gegevenslandschap vormen dus een twee-eenheid met de *realisatie*principes voor Common Ground. Beide sets principes zijn belangrijk voor de totstandkoming van een nieuwe, flexibele en meer gezamenlijke gemeentelijke informatiehuishouding. De *informatiearchitectuur*principes voor het Gemeentelijk Gegevenslandschap zijn in een separaat document beschreven.

Dit document is bestemd voor informatiemanagers, adviseurs en architecten bij gemeenten, leveranciers en andere partijen die betrokken zijn bij realisatie van de Common Ground-visie en de ontwikkeling van het Gemeentelijk Gegevenslandschap.

Het document is als volgt opgebouwd:

- Hoofdstuk 2 beschrijft acht architectuurprincipes met daarbij behorende implicaties.

Dit document is in beheer bij VNG-Realisatie

Tabel 1. Documenthistorie

Versie	Toelichting	Datum	Opsteller(s)
0.1	Conceptversies voor reviews door de squad Architectuur Samen Organiseren en Kerngroep Common Ground.	nov/dec 2017	Toine Schijvernars, Ad Gerrits
0.2	Versie voor publicatie op GEMMA Online en externe reviews.	januari 2018	Ad Gerrits

Inhoudsopgave

Leeswijzer	2
inhoudsopgave	3
1. Inleiding	4
2. Realisatieprincipes	5
2.1. Community – we werken als community	5
2.2. Agile - we ontwikkelen incrementeel en iteratief	7
2.3. Nieuw naast oud - we ontwikkelen nieuw naast bestaand	8
2.4. Moderne it – we gebruiken moderne technologie	9
2.5. Open source - we geven de voorkeur aan open source software	10
2.6. Lagen-architectuur - we gebruiken een gelaagd architectuurmodel	11
2.7. Nlx als gateway – we ontwikkelen en gebruiken nlx als gateway	12

1. Inleiding

Onder de noemer 'Common Ground' werken gemeenten en leveranciers aan de realisatie van een nieuw informatielandschap. Deze notitie beschrijft de daarbij gehanteerde realisatieprincipes. Deze beschrijven hoe de Common Ground-beweging verder invulling geeft aan de informatiearchitectuurprincipes voor het Gemeentelijk Gegevenslandschap die in een separate notitie zijn beschreven.

De principes zijn bedoeld om richting te geven aan het stapsgewijs realiseren van een nieuw type informatiehuishouding voor gemeenten. Bij het ontwerpen van nieuwe of aangepaste voorzieningen moet worden aangegeven hoe invulling wordt gegeven aan de principes, en welke overwegingen daarbij worden gemaakt. Daarbij geldt het pas-toe-of-leg-uit principe. Afwijkingen zijn alleen toegestaan als ze met goede argumenten worden onderbouwd en vastgelegd om daar in een later stadium op terug te kunnen komen. Zo wordt voorkomen dat belangrijke zaken over het hoofd worden gezien, en wordt geborgd dat activiteiten in lijn zijn met de Common Ground-visie. VNG Realisatie ziet namens gemeenten toe op naleving van de principes en stelt deze waar nodig bij aan de hand van opgedane leerervaringen en voortschrijdend inzicht.

Bij ieder principe wordt toegelicht waarom het belangrijk is en wat de implicaties van het gebruik ervan zijn. Bij de formulering van principes is de we-vorm gebruikt. Naast gemeenten worden hiermee ook de andere partijen bedoeld die bijdragen aan de realisatie van de Common Ground-visie en de ontwikkeling van het Gemeentelijk Gegevenslandschap, zoals leveranciers van software. In veel gevallen zijn de beschreven implicaties voorzien van een voorbeeld ter verduidelijking.

2. Realisatieprincipes

2.1. COMMUNITY – We werken als community

Net als bij andere ontwikkelingen binnen Samen Organiseren werken gemeenten, leveranciers en VNG Realisatie samen bij het realiseren van de Common Ground-doelstellingen. Alle gemeenten worden regelmatig geïnformeerd over doelen en voortgang en kunnen zowel off- als online inbreng leveren. Via community-building zorgen we er voor dat de nieuw ontwikkelde informatiehuishouding daadwerkelijk van en voor gemeenten wordt.

Implicaties:

1. Aansluitend bij de visie van Samen Organiseren werken we vanuit de gemeentelijke praktijk. We verzamelen bottom-up behoeften bij gemeenten die worden vertaald naar heldere vragen op basis waarvan conform Common Ground ontwikkeld wordt (bijvoorbeeld door vraagstukken bij diverse gemeenten te vertalen naar 'wie-wat-waarom' user-stories en die op een product-backlog te plaatsen).
2. Alle deelnemers aan de Common Ground-community geven aan waar hun prioriteiten liggen. De regio-organisatie stelt de roadmap vast voor wat gezamenlijk ontwikkeld wordt (bijvoorbeeld door ingediende ideeën te prioriteren en beschikbare middelen toe te kennen).
3. Gemeenten en partners kunnen in diverse samenstellingen en op verschillende manieren op eigen initiatief functionaliteit (laten) realiseren (bijvoorbeeld als individuele gemeente of als groep gemeenten samen met leveranciers).
4. Samenwerking tussen gemeenten wordt gestimuleerd (bijvoorbeeld door ontwikkelingen met elkaar te verbinden en gemeenten met elkaar in contact te brengen tijdens fieldlabs).
5. De community biedt ruimte voor marktpartijen om nieuwe diensten en applicaties te ontwikkelen (bijvoorbeeld door op basis van gestandaardiseerde API's onderscheidende frontend-toepassingen te maken).
6. In lijn met de Samen Organiseren-visie stimuleert de community standaardisatie, ook qua techniek, maar legt deze niet op (bijvoorbeeld door leveranciers vrij te laten om gebruik te maken van bij hen bekende tooling).
7. Ontwikkelde producten worden aan de community ter beschikking gesteld (bijvoorbeeld door ontwikkelde software als open source beschikbaar te stellen).
8. Elke gemeente bepaalt zelf wanneer en hoe zij deelneemt en nieuwe voorzieningen gebruikt (bijvoorbeeld door op een voor haar geschikt moment NLX toe te passen op een manier die past bij de gemeentelijke integratiestrategie).
9. Er is vrijheid voor gemeenten op proces- en interactieniveau (bijvoorbeeld door per gemeente te kiezen voor gebruik van webapplicaties en/of apps).
10. Er is vrijheid voor gemeenten om leveranciers en software te kiezen (bijvoorbeeld door al dan niet te kiezen voor leveranciers waarmee al zaken worden gedaan).
11. Bij gebruik van decentrale voorzieningen is de gemeente verantwoordelijk voor installatie, ingebruikname en beheer (bijvoorbeeld bij het in gebruik nemen van NLX als component binnen de gemeentelijke infrastructuur).

12. VNG Realisatie ondersteunt het gedachtegoed van Common Ground en heeft een faciliterende rol binnen de community (bijvoorbeeld door visiedocumenten op te stellen en via instrumentele ondersteuning bij realisatietrajecten).
13. De GEMMA is de referentiearchitectuur voor de informatievoorziening bij één gemeente. VNG Realisatie ontwikkelt en beheert de enterprise-architectuur voor het collectief van gemeenten. Beide architecturen zijn op elkaar afgestemd.

2.2. AGILE - We ontwikkelen incrementeel en iteratief

We ontwikkelen en realiseren vanuit de uitvoeringspraktijk. Rekening houdend met continu optredende veranderingen in de buitenwereld werken we agile. We leveren met korte tussenpozen bruikbare (deel)producten op, verwerken feedback en leren van wat we doen. We geloven in het gebruik van proof-of-concepts en pilots om te kijken of iets werkt. We beschouwen producten niet meer als 'af' maar als 'permanent beta'.

Implicaties:

1. Bedrijfsprocessen en cultuur moeten in lijn zijn met de principes van agile werken om succesvol te zijn. Anders is de invoer niet zonder risico. (bijvoorbeeld door te voorkomen dat hoger management te veel invloed uitoefent op zelfsturende Scrum-teams).
2. Agile gaan werken betekent een paradigmashift voor organisaties. Aanwezige kennis en kunde moet optimaal worden benut. Waar kennis en kunde ontbreken moet daarin worden geïnvesteerd (bijvoorbeeld door voorafgaand aan trajecten medewerkers bij te scholen).
3. Alle initiatieven onder de vlag 'Common Ground' kennen een agile werkwijze. Aspecten hiervan zijn:
 - a. Er zijn vooraf gedefinieerde sprints.
 - b. Sprints kennen een vaste scope.
 - c. Er is een met stakeholders afgestemde backlog.
 - d. De klant is in de rol van productowner betrokken.
 - e. Er wordt regelmatig gecommuniceerd tussen het team en de stakeholders, o.a via periodieke demo's.
 - f. Teams zijn relatief klein zijn (<10) en bestaan uit een mix van diverse disciplines.
 - g. Het zelforganiserende agile team maakt tijdens uitvoering van sprints zelf keuzes.
 - h. Er is aandacht voor samenwerking en vertrouwen tussen teamleden.
4. In tegenstelling tot bij een klassieke projectaanpak kunnen eindproducten bij de start nog niet in detail worden beschreven. Team en opdrachtgever maken daarom 'agile afspraken en contracten' (bijvoorbeeld door minimaal te realiseren functionaliteiten en eigenschappen te benoemen ('Minimum Viable Product')).
5. In de aansturing wordt rekening gehouden worden met agile werken in de uitvoering. Roadmaps, planning en verantwoording op een hoog niveau is uiteraard mogelijk, maar planning en verantwoording binnen deze afgesproken kaders geschiedt volgens de principes en mechanismen van agile werken (bijvoorbeeld door productowners voldoende mandaat te geven om snel te kunnen ontwikkelen).
6. Zowel vanwege de kortere doorlooptijd als vanuit kostenperspectief is het bij snel en frequent opleveren een uitdaging om formele momenten te vinden voor bijvoorbeeld kwetsbaarheid- en/of pentesten. Informatiebeveiliging moet daarom extra aandacht krijgen (bijvoorbeeld door te zorgen dat ontwikkelaars kennis hebben van veilige programmeermethodes en informatiebeveiliging en informatiebeveiligingsaspecten als vast onderdeel van sprints te benoemen).

2.3. NIEUW NAAST OUD - We ontwikkelen nieuw naast bestaand

We ontwikkelen nieuwe voorzieningen conform de Common Ground-visie en de uitgangspunten van het Gemeentelijk Gegevenslandschap, maar houden er rekening mee dat bestaande voorzieningen nog 5 tot 10 jaar in gebruik zullen blijven. Bij nieuwe ontwikkelingen, gemeentelijk en landelijk, zijn de Common Ground-principes leidend. We steken energie in het bouwen van iets nieuws naast het oude. Als 'nieuw' en 'oud' elkaar ontmoeten, moeten oude voorzieningen, eventueel na aanpassing daarvan, gaan aansluiten op het nieuwe. Wat nu als 'nieuw' wordt bestempeld blijft uiteraard niet altijd 'nieuw'. Er zullen continu ontwikkelingen zijn die het nodig maken om te blijven vernieuwen. We formuleren daarom geen eindplaatje en doelarchitectuur, maar zien ontwikkeling als een continu proces ('permanent beta').

Implicaties:

1. We zorgen dat na het opleveren van nieuwe voorzieningen bestaande voorzieningen nog een tijdlang bruikbaar blijven (bijvoorbeeld door bestaande applicaties nog een periode via datadistributie van kopiegegevens te voorzien in plaats van ze zelf via API's gegevens te laten ophalen bij de bron).
2. In gebruik zijnde applicaties die niet conform de principes van Common Ground of het Gemeentelijk Gegevenslandschap werken worden stapsgewijs uitgefaseerd (bijvoorbeeld door bij vervangingstrajecten te eisen dat conform de nieuwe principes wordt gewerkt).
3. We faciliteren in het transitievraagstuk bij gemeenten van de huidige situatie naar Common Ground maar sluiten geen compromissen m.b.t. de gewenste situatie.
4. De transitie van de bestaande informatiehuishouding naar de nieuwe informatiehuishouding zal in overleg met alle betrokken partijen plaatsvinden (bijvoorbeeld met gemeenten, leveranciers, uitvoeringsinstanties).

2.4. MODERNE IT – We gebruiken moderne technologie

Gemeenten zijn onvoldoende meegegroeid met de technologische ontwikkelingen op het vlak van IT. We gaan de kansen die technologie biedt beter benutten door waar mogelijk gebruik te maken van moderne technologie. Daarvoor volgen we technologische ontwikkelingen op de voet, en vertalen die naar praktische toepassingsmogelijkheden. We stellen ons niet afhankelijk op door af te wachten wat leveranciers in de gemeentemarkt doen, maar gaan samen met hen pro-actief onderzoeken waar inzet van nieuwe technologie meerwaarde biedt.

Implicaties:

1. We organiseren als gemeenten voldoende kennis om nieuwe ontwikkelingen te kunnen volgen, te duiden en toe te passen (bijvoorbeeld container- en serverless technologie).
2. Gebruik van moderne techniek stelt vaak andere eisen dan nu gebruikelijk is bij gemeenten. Bijvoorbeeld bij:
 - a. Deployment (bijvoorbeeld via een ci/cd pipeline uitvoeren van softwareupdates).
 - b. Licenties (bijvoorbeeld afsluiten van kort durender contracten bijvoorbeeld in abonnementsvorm)
 - c. Hosting (bijvoorbeeld meer en intensiever gebruiken van PAAS-platformen)
 - d. Beheer (bijvoorbeeld gebruiken van containers in plaats van executables).
3. Passend bij gebruik van nieuwe technieken worden handelingen zoveel mogelijk geautomatiseerd, zoals het:
 - a. Testen en deployen van software (bijvoorbeeld bij het uitvoeren van software-updates).
 - b. Monitoren (bijvoorbeeld van systeembelasting).
 - c. Loggen (bijvoorbeeld van vooraf benoemde verwerkingen zoals opvragingen).
 - d. Notificeren en/of ondernemen van correctieve actie (bijvoorbeeld door bij constatering van onvoldoende performance automatisch capaciteit uit te breiden).
4. De keuze om mee te blijven groeien met technologische ontwikkelingen betekent dat gebruikte platforms en voorzieningen soms sneller vervangen moeten worden dan nu gebruikelijk is. Dit stelt nieuwe eisen aan ontwikkel- en beheerorganisaties. (bijvoorbeeld door gebruik te maken van platform-onafhankelijke technologie).
5. Door mee te blijven groeien met nieuwe ontwikkelingen zijn we als gemeenten meer 'developer-friendly' en worden we aantrekkelijker voor innovatieve kleine bedrijven en voor, eventueel in dienst te nemen, ontwikkelaars.
6. Om innovatie mogelijk te maken kiezen we voor geschikte stijlen en technieken. Reden om bij de start van Common Ground te kiezen voor gebruik van:
 - a. REST als architectuurstijl bij het ontwikkelen van API's .
 - b. JSON als gegevensformaat bij de uitwisseling van gegevens.
 - c. Containertechnologie voor hosting van componenten.
7. Waar we zelf in the lead zijn bij het ontwikkelen van software kiezen we voor moderne methoden en technieken (bijvoorbeeld om zaken als versiebeheer, testen en deployment met de juiste tooling geautomatiseerd te laten uitvoeren).
8. Bij gedeeld gebruik van voorzieningen worden geautomatiseerd gegevens verzameld en vastgelegd om doorbelasting van kosten mogelijk te maken (bijvoorbeeld door betrokken organisaties en het type systeemactie te loggen).

2.5. OPEN SOURCE - We geven de voorkeur aan open source software

In navolging van de Agenda Digitale Overheid stimuleren we het gebruik van open source software “om als overheden minder afhankelijk te zijn van softwareleveranciers én om inzicht te geven in de wijze waarop bepaalde software is opgebouwd”. Open ontwikkeling, zoals die bij open source software gebruikelijk is, past bij de Common Ground-visie waarin we als gemeenten samen met andere partijen innovatieve dienstverlening mogelijk maken. We hebben daarom een voorkeur voor gebruik van open source software bij de realisatie van de beoogde nieuwe informatehuishouding voor gemeenten.

Implicaties:

1. We stimuleren kennisopbouw binnen de Common Ground-community om verantwoord open source-software te (laten) ontwikkelen, testen, beheren en doorontwikkelen.
2. We stellen de code van in opdracht ontwikkelde software als open source beschikbaar waar dit is toegestaan (bijvoorbeeld door de broncode met EUPL-licentie openbaar toegankelijk te publiceren).
3. We gaan als gemeenten niet grootschalig open source software ontwikkelen maar geven hiervoor opdrachten aan marktpartijen (bijvoorbeeld door samen met leveranciers in een agile ontwikkeltraject producten te ontwikkelen).
4. We nemen belemmeringen voor het gebruik of het vrijgeven van open source software zoveel mogelijk weg (bijvoorbeeld door bij gunning van opdrachten voorwaarden op te nemen zodat ontwikkelde broncode kan worden gepubliceerd).
5. De aanschafkosten voor software gaan vaak omlaag, maar we realiseren ons dat er nog steeds onderhoud- en beheerkosten zijn (bijvoorbeeld omdat software periodiek moet worden geüpdate).
6. Open source-software kan kwetsbaar zijn met betrekking tot informatiebeveiliging en privacy omdat de code voor iedereen beschikbaar is. Dit vereist specifieke aandacht (bijvoorbeeld door goed te volgen welke kwetsbaarheden zijn gesignaleerd en zo snel mogelijk passende maatregelen te nemen).
7. Bij keuzes voor open source-software laten we meewegen hoe groot en levendig de community van betrokken ontwikkelaars is omdat die de kwaliteit en continuïteit mede bepaalt (bijvoorbeeld door alleen voor software te kiezen die door meerdere partijen wordt doorontwikkeld en beheerd).
8. Gebruik van open source-software kan de afhankelijkheid van leveranciers verminderen, maar alleen als we hier gericht op sturen (bijvoorbeeld door te zorgen voor voldoende interne kennis en/of het gebruik van verschillende leveranciers).
9. Om goed invulling te geven aan functionele behoeftes is een georganiseerde community met eindgebruikers nodig om de software ‘levend’ te houden (bijvoorbeeld door ze actief te betrekken bij het doorgeven van geconstateerde fouten en nieuwe wensen).

2.6. LAGEN-ARCHITECTUUR - We gebruiken een gelaagd architectuurmodel

We gebruiken een lagenmodel om onze gewenste informatiearchitectuur te beschrijven en te visualiseren.¹ Functionaliteit binnen een laag wordt geleverd door gespecialiseerde componenten. Componenten communiceren met elkaar via standaarden en leveren samen de benodigde functionaliteit. Door het scheiden van verantwoordelijkheden in lagen en componenten zijn we onder meer in staat om verantwoordelijkheden rondom beheer van gegevens en inrichting en uitvoering van processen van elkaar te scheiden.

Implicaties:

1. Nieuwe functionaliteit en componenten worden met behulp van de benoemde architectuurlagen gepositioneerd en beschreven (bijvoorbeeld door aan te geven in welke laag een component past).
2. Iedere laag kent afgebakende functionaliteit die door zelfstandige componenten wordt gerealiseerd (bijvoorbeeld door te beschrijven wat een component geacht wordt wel en niet te leveren).
3. Laaggebonden functionaliteit wordt niet gerealiseerd binnen andere lagen (bijvoorbeeld in de presentatielaag worden voor opvraag van gegevens geen database-queries uitgevoerd maar worden hiervoor beschikbare API's aangeroepen).
4. Bovenliggende lagen kunnen voor hun werking afhankelijk zijn van onderliggende lagen (bijvoorbeeld bij ontwikkeling van een service wordt rekening gehouden met hoe gegevens zijn vastgelegd).
5. Onderliggende lagen zijn onafhankelijk van bovenliggende lagen (bijvoorbeeld bij ontwikkeling van een bronregistratie hoeft geen rekening te worden gehouden met te ontwikkelen services die van de gegevens gebruik maken).
6. Een bovenliggende laag roept alleen functionaliteit in de direct onderliggende laag aan ('closed layer architecture') (bijvoorbeeld een integratievoorziening roept alleen services aan en maakt niet direct verbinding met onderliggende gegevensbronnen). Een uitzondering hierop vormt de situatie dat frontend-componenten uit laag 5 ook direct API's uit laag 3 mogen aanroepen.
7. Bij nieuwe ontwikkelingen of benodigde aanpassingen bezien we holistisch in welke lagen aanpassingen nodig zijn (bijvoorbeeld door bij wensen voor nieuwe functionaliteit te bepalen in welke lagen aanpassingen nodig zijn in plaats van het alleen aanpassingen te doen in de presentatielaag).

¹ In het gebruikte lagenmodel worden vijf lagen onderscheiden: 'gegevensbronnen', 'diensten', 'integratie', 'procesinrichting' en 'interactie'. De beschrijving van de functionaliteit per laag en de uitwisseling tussen lagen is beschreven in het document '[Gemeentelijk Gegevenslandschap - Beschrijving informatiearchitectuur](#)'.

2.7. NLX ALS GATEWAY – We ontwikkelen en gebruiken NLX als gateway

De Common Ground-visie gaat uit van een decentrale architectuur waarin gegevens niet langer worden gekopieerd maar via service-API's worden opgevraagd bij de bron. Ook als die zich buiten de eigen organisatie bevinden. Uitwisselen van gegevens moet daarbij veilig, betrouwbaar en snel plaatsvinden. Aanvullend moet er ook specifieke functionaliteit zijn om gemeenten in staat te stellen te voldoen aan hun wettelijke verplichtingen door bijvoorbeeld doelbindinggegevens te loggen. Hiervoor is specifieke integratiefunctie nodig die de verdere realisatie van de Common Ground visie blijvend ondersteunt. We ontwikkelen en gebruiken daarvoor stapsgewijs NLX. Hiermee wordt de benodigde basisfunctionaliteit geleverd die gegevensuitwisseling conform Common Ground mogelijk maakt.

Implicaties:

1. NLX bevat basisfunctionaliteit om gegevensuitwisseling via API's conform de Common Ground-visie te ondersteunen (bijvoorbeeld door uitwisseling via het HTTP-protocol te ondersteunen en door gefedereerde autorisatie en logging van doelbindingsclaims te ondersteunen).
2. De verantwoordelijkheid voor eventuele aanvullende functionaliteit bij gebruik van services en API's ligt bij de gebruiker (bijvoorbeeld load-balancing, throttling, protocolvertaling, orkestratie).
3. Om gegevens conform Common Ground-principes uit te wisselen wordt NLX door alle betrokken partijen als gateway gebruikt (bijvoorbeeld door gemeenten, andere overheidspartijen en hosting-partijen).
4. NLX is zowel bruikbaar voor gegevensuitwisseling tussen organisaties als voor uitwisseling binnen een organisatie (bijvoorbeeld om lokale applicaties toegang tot lokaal opgeslagen brongegevens te geven).
5. Er is beheer georganiseerd om collectieve NLX-componenten stabiel en betrouwbaar te laten functioneren (bijvoorbeeld door te zorgen voor een betrouwbare servicecatalogus en om storingen te kunnen oplossen).
6. Er is governance georganiseerd voor gebruik en doorontwikkeling van NLX (bijvoorbeeld om gebruiksafspraken en een roadmap vast te stellen en bewaken).
7. De verantwoordelijkheid voor lokale installatie, beheer en gebruik van NLX ligt bij de gebruiker ervan (bijvoorbeeld een gemeente die, al dan niet uitbesteed, NLX toepast, beheert en gebruikt).
8. De broncode van NLX wordt voor zover toegestaan als open source gepubliceerd (bijvoorbeeld door de code in een voor iedereen toegankelijke repository te publiceren).