# Lessons learned (obv github)

1. **API beschikbaarheid, API bruikbaarheid en API compleetheid**

*beschikbaarheid*

Aanvankelijk was de gedachten bij de start van het project, dat een ruime beschikbaarheid van API’s zou zorgen voor een brede beschikbaarheid van data. Het juist combineren van deze data zou een schat aan informatie opleveren.

Echter, in tegenstelling blijkt dat API’s niet ruim beschikbaar zijn. De BAG beschikt over een goed opgezette API die zowel relationele als geografische data biedt. De BAG API onderscheidt zich daar dan ook mee.

*bruikbaarheid*

Meer dan 80% van de gebruikte data wordt beschikbaar gesteld als kaart laag, die zich niet als API laat gebruiken. Daarbij is de BAG API ook de enige API die zich geografisch laat bevragen.

*compleetheid*

Daarnaast stelt een API niet per definitie alle data beschikbaar. De BAG API stelt bijvoorbeeld het gebruiksdoel van een verblijfsobject niet beschikbaar via de API.

*Overwegingen voor vervolg (Linda en Dick)*

* Op de huidige pas-toe-of-leguit-lijst van Forum Standaardisatie staan nu de Nederlandse profielen van versies van WMS en WFS. Deze ‘verplichting’ staat een andere doorontwikkeling in de weg. Dit zou op een bepaald moment van de lijst af moeten.
* In plaats daarvan zou op de lijst API’s moeten staan en/of Linked Data.
* Een deel van de oplossing ligt in het zodanig vernieuwen van de versies van WMS en WFS dat deze als API functioneren. Voor WFS is met WFS 3.0 dit al mogelijk, voor WMS wordt aan een nieuwe versie gewerkt.

1. **API als datasilo**

* Eén constatering ten aanzien van APIs lijkt te zijn dat een API gezien kan worden als data silo. De API deelt de data uit de silo en bevat veelal geen enkele relatie naar data (of een dataset) buiten de API.
* Gebruikelijk lijkt te zijn, dat als een API meerdere requests aanbiedt, het antwoord van de ene request verwijst naar een andere request binnen diezelfde API. Zo verwijst de data over een verblijfsobject naar de API request om pand data op te vragen.

*Overweging voor vervolg of voor opname in API-strategie*

* Om data uit de datasilo in samenhang beschikbaar te maken, zou er verwezen moeten worden naar identifiers. Hierdoor kan er generiek omgegaan worden met het ophalen van data van verschillende APIs.
* Daarnaast zouden APIs kunnen verwijzen naar identifiers van andere APIs. Zo beschrijft de data van de Rijksdienst voor Cultureel Erfgoed (RCE) adressen op monumenten, maar deze laten zich lastig vergelijken en brengt grove foutmarges met zich mee. Idealiter zou de RCE API (ook) verwijzen naar een adres identifier beschikbaar binnen het BAG.

*Overweging voor vervolg of voor opname in API-strategie*

* Om het stelsel en aanpalende gegevensverzamelingen in samenhang te kunnen bevragen **moeten** APIs verwijzen naar identifiers uit de samenhangende objectenregistratie als die relaties er zijn.
* Het moet ook duidelijk zijn wat de betekenis is van die relatie. Soms wordt bijvoorbeeld aan een object een adres gerelateerd voor de vindbaarheid terwijl in dat object meerdere verblijfsobjecten met adressen kunnen voorkomen. Hieruit mag niet worden geconcludeerd dat alleen dat ene adres relevant is.
* Voor het organiseren (leggen en beheren) van die relaties is governance en budget nodig!

1. **Eén en tweezijdige verwijzingen en afwijkingen**Datamodellen leggen restricties op. Zo legt bijvoorbeeld het BAG datamodel vast dat een verblijfsobject naar een pand verwijst, maar een pand niet naar een verblijfsobject. Echter de API doet dit wel en wijkt daarmee af van het datamodel. Dit maakt het gebruik van een API erg specifiek.

*Overweging voor vervolg*

* een relatie tussen twee objecten moet altijd in twee richtingen uitvraagbaar zijn.
* Het semantisch model moet zo robuust zijn dat een eventuele orchestratielaag over APIs heen zo compact mogelijk wordt gehouden.

1. **Wisselende API resultaten**

Een API kan wisselende resultaten teruggeven. Een veelvoorkomende situatie is het teruggeven van één of meerdere resultaten. Dit houdt bijvoorbeeld als het resultaat één instantie bevat wordt deze direct terug gegeven in de resultaten. Echter als het resultaat meerdere instanties teruggeeft, worden deze gebundeld in bijvoorbeeld een JSON array. Normaliter is dit niet persé een probleem. In dit project vormt dit wel een probleem. De resultaten van een API worden omgezet naar een semantisch formaat. Dit wordt gedaan door middel van een mapping. Hoe de instanties gebundeld worden, moet in de mapping verwerkt worden, anders kan deze niet goed uitgevoerd worden. *Overweging voor vervolg of voor opname in API-strategie*

* Er kan overwogen worden om in de API-strategie op te nemen dat een antwoord dat een of meer resultaten kan bevatten altijd een bundeling is, ook als het slechts één resultaat is. Hiermee wordt de orchestratielaag klein en compact gehouden.

1. **Adresgegevens onvergelijkbaar**

Adresgegevens blijken moeilijke gegevens te zijn. In de verschillende datasets worden adresgegevens gebruikt, echter laten deze zich moeilijk vergelijken. Aannemende dat het Kadaster het meest complexe model op adresgegevens hanteert lijkt dit ook de meest nauwkeurige en daardoor meest bruikbare. Een adres bestaat uit een aantal facetten (woonplaats, straat, huisnummer, huisletter, toevoeging). Echter een groot deel van de datasets/API’s hanteren deze data door elkaar. Hierdoor is het lastig om adresdata te vergelijken.

*Overweging voor vervolg of voor opname in API-strategie*

• Wederom: Om het stelsel en aanpalende gegevensverzamelingen in samenhang te kunnen bevragen moeten APIs verwijzen naar identifiers uit de samenhangende objectenregistratie als die relaties er zijn. Relaties op basis van beschrijvende elementen zoals adres moeten worden uitgesloten.

1. **Stelselcatalogus geen relatie met bron**

Om de semantische relatie tussen de data van de verschillende API’s te ontdekken, moet in de semantische orchestratie laag gedefinieerd worden hoe data zich tot elkaar verhoudt. Dit wordt prima gedaan door de al bestaande stelselcatalogus. Deze catalogus bevat enkel de basisregistraties, maar dezelfde structuur wordt in de orchestratielaag ook toegepast op de registraties die hier niet in zitten.

Een deel van de data is ook al beschikbaar als semantische data bij de bron. Echter is er geen relatie tussen de semantische relaties bij de bron en de equivalente relaties in de stelselcatalogus. Hierdoor is het onmogelijk om de orchestratielaag te gebruiken om de al reeds semantische data op te vragen en aan de demonstrator toe te voegen.

*Overweging voor vervolg*

* De semantische relaties van de samenhangende objectenregistratie (opvolger van de stelselcatalogus, in ieder geval voor de geo-basisregistraties) moeten altijd gerelateerd zijn aan het semantische model van de datasets.

1. **Wat als dat van meerdere bronnen komt?**

Tijdens het ontwikkelen van de Disgeo demonstrator is de aanname gedaan dat een data geleverd wordt door één enkele API. Data omtrent een verblijfsobject zal altijd van het kadaster komen. Op het moment data deze aanname ongeldig wordt, treedt het probleem op dat het haast onmogelijk is om te achterhalen waar een bepaald object opgevraagd moet worden.

*Overweging voor vervolg*

* Basisgegevens moeten enkel en alleen bij de samenhangende objectenregistratie worden opgehaald. Dit blijft een uitgangspunt!

1. **Herkomst van data**

Een API biedt in geen van de gevallen metadata over het object. Het is daardoor onmogelijk om de actualiteit, nauwkeurigheid en betrouwbaarheid van data te valideren.

*Overweging voor vervolg of voor opname in API-strategie*

* Onderzocht moet worden of APIs hiervoor geschikt te maken zijn.
* Linked data biedt hiervoor goed de mogelijkheid

1. **Configuratie last**

Om de API’s aan elkaar te kunnen relateren, de resultaten van een API om te zetten in semantisch formaat en om de API configuratie te beschrijven is een enorme configuratie last nodig. Voor de beperkte API’s op dit moment hebben hier al ruim 4000 regels aan configuratie nodig. Ook het onderhoud van deze configuratie zal een redelijke last met zich mee brengen.

*Overweging voor vervolg*

* Dit pleit er extra voor de orchestratielaag zo compact mogelijk te houden.
* NB Linked data maakt dit helemaal overbodig !!!

1. **Geografische vraag kenmerkt zich door missende relatie (en waar naar te zoeken)**

Bij de geografische vraag gaat de demonstrator op zoek naar objecten die geen relatie hebben tot elkaar, deze objecten moet binnen een bepaalde straal van elkaar liggen. Echter zou dit mogelijk zijn op alle objecten die gedefinieerd zijn in de Stelselcatalogus waarvan de gedefinieerde API geografische query’s ondersteund. Echter zou dit betekenen dat met een groeiend aantal APIs dit een flinke belasting op de performance wordt. Daarnaast betekent dit, dat bijvoorbeeld alle panden worden opgevraagd in een bepaalde straal. Dit levert ontzettend veel data op, die in veel gevallen waarschijnlijk niet de vraag van de gebruiker zal beantwoorden. Op dit moment is gekozen om in de configuratie vast te leggen op welke objecten geografisch gezocht moeten worden vanuit een bepaald start object. Hierdoor kan er sturing plaats vinden.

*Overweging voor vervolg*

* Hier moet nog nader onderzoek naar worden gedaan wat de balans is tussen op alles te kunnen zoeken en het aantal resultaten behapbaar te houden en het inbouwen van gerichte zoekpatronen / manieren om het aantal zoekrichtingen te kanaliseren

1. **Geografische relatie obv GeoSPARQl**

Geografische data laten zich makkelijk in samenhang gebruiken. Geografische data worden in het kader van deze demonstrator in alle gevallen aangeboden in een geo-standaard. Hierdoor kunnen verschillende tools en libraries eenvoudig omgaan met geografische data ongeacht de standaard die toegepast is. Ook tools die semantisch werken kunnen goed met deze data omgaan om dat het eenvoudig te gebruiken is met Semantische Geo standaarden.

*Overweging voor vervolg*

* Dit is waardevol, op deze weg blijven doorgaan.