Status	Concept
Versie	0.9
Auteur	Dennis de Wit
Datum	10-12-2020

Versiebeheer

Versie	Datum	Wijzigingen	Auteur
0.1	05-10-2020	Eerste versie, gebaseerd op specificatie Drechtsteden	Dennis de Wit
0.2	10-11-2020	Review John Zwart, Joost Farla en Ronald Doest verwerkt	Dennis de Wit
0.3	06-12-2020	Na overleg met Joost Farla en Ronald Doest	Dennis de Wit
0.9	10-12-2020	Na overleg met John Zwart, Joost Farla en Ronald Doest	Dennis de Wit

Inhoudsopgave

5 Architectuur

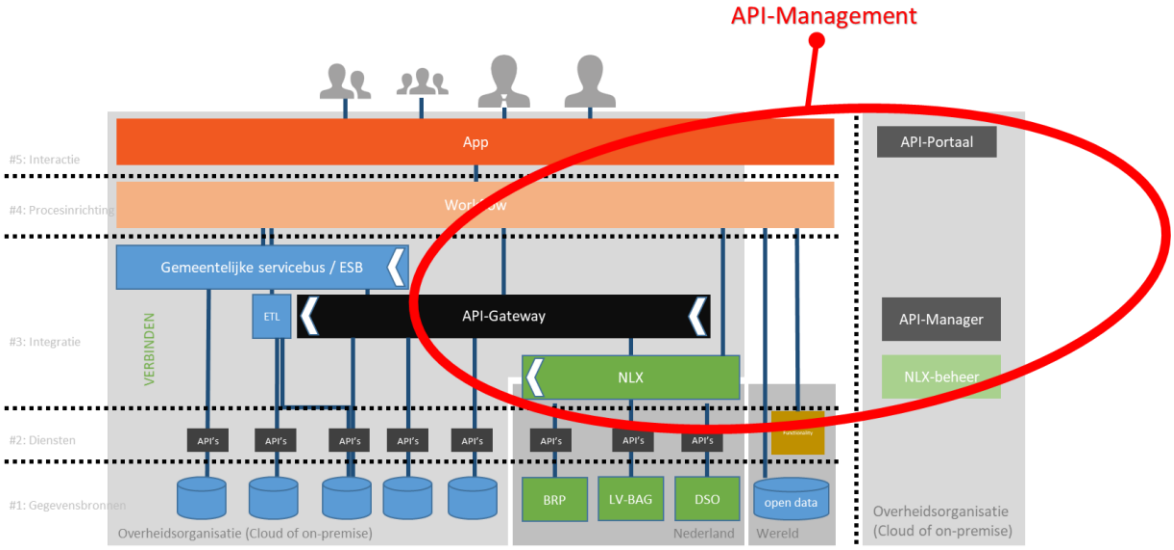
5.1 Inleiding

In dit hoofdstuk wordt aandacht besteed aan de positionering van API-Management binnen de informatie architectuur van een overheidsorganisatie. Daarbij wordt stilgestaan bij de generieke specificaties waaraan dit soort tooling moet voldoen om API management goed op te kunnen zetten. De kaders en richtlijnen die in dit hoofdstuk worden benoemd zijn in lijn met landelijke geldende architectuurrichtlijnen (zoals NORA, GEMMA en Common Ground), hiermee kan dit document worden gezien als een referentiearchitectuur op het gebied van API-Management.

5.2 Informatie architectuur

Alle overheden hebben een uitdaging op het gebied van data integratie. Hoe faciliteer je gegevensuitwisseling tussen bronnen en afnemers op een efficiënte en beheersbare manier die voldoet aan de eisen van wet- en regelgeving?

Een overheidsinstantie geeft hier het beste invulling aan door organisatiebreed een zogenaamde 'data integratielaag' binnen de infrastructuur te positioneren. Dit kan je beschouwen als de gereedschapskist waarbinnen verschillende tools beschikbaar zijn voor data-integratie. Gezien de wereldwijde ontwikkelingen in het gebruik van API's, kan API-Management tooling hierbinnen niet ontbreken. Het is voor overheden essentiële functionaliteit om de in lijn met de NORA te kunnen opereren (of specifieker in lijn met Common Ground).
In figuur 2 is dit gevisualiseerd.



Figuur 1: Visualisatie 'Positionering API-Management binnen de integratielaag'

Overheden zijn er op dit moment bij gebaat om de integratiefunctionaliteit van API-Management lokaal op grote schaal technisch in te richten. Een directe stap naar het enkel gebruiken van een landelijke integratiefunctionaliteit (zoals 'NLX' dit zou kunnen gaan invullen) en geen gebruik meer te maken van de lokale integratiefunctionaliteit is (op uitzonderingen na) in deze tijd niet realistisch.

Enerzijds is deze stap voor het complete applicatielandschap veel te groot. Landelijke integratiefunctionaliteit leunt op termijn bijvoorbeeld op de gedachte dat Identity & Access Management (IAM) binnen de deelnemende overheden op orde is. Ook is de verwachting dat er, gedurende de transitie, behoefte zal zijn aan transformatie-, orkestratie- en autorisatiefunctionaliteit. Nog niet duidelijk is hoe dit in de landelijke integratiefunctionaliteit wordt gepositioneerd. Daarnaast zijn er veel onzekerheden in de

adoptiegraad van de landelijke integratiefunctionaliteit door het volledige ecosysteem (partijen in de informatieketen). Het tijdspad van de transitie (van lokale integratiefunctionaliteit naar landelijke integratiefunctionaliteit) zal naar verwachting jaren in beslag nemen.

Rol van servicebus
Binnen de integratielaag opereert veelal ook een organisatiebrede servicebus, vaak is er veel energie gestoken in het faciliteren van gegevensstromen via de servicebus (met name op basis van StUF[1]). Het is voor overheden geen doel op zich om bestaande verbindingen te elimineren of de servicebus uit te faseren. De API-Management tooling komt naast de servicebus te staan en kan zo aanvullende functionaliteit bieden binnen de integratielaag.

Wel lijkt gezien de wereldwijde ontwikkelingen de aandacht van integratievraagstukken te gaan verschuiven van de inzet van een organisatiebrede servicebus naar een landschap waarin lightweight API-Management tooling een belangrijke rol speelt. Nieuwe verbindingen (met name voor het ophalen van data) zullen vaker gelegd gaan worden via enkel een API Gateway en de inzet van de servicebus wordt teruggedrongen. Enkel op het gebied waar de huidige servicebus specifieke toegevoegde waarde levert, wordt deze voor overheden nog ingezet voor nieuwe verbindingen (eventueel in combinatie met een API Gateway. Dit zal naar verwachting voor overheden de meest voor de hand liggende oplossing zijn, gezien het huidige applicatielandschap.

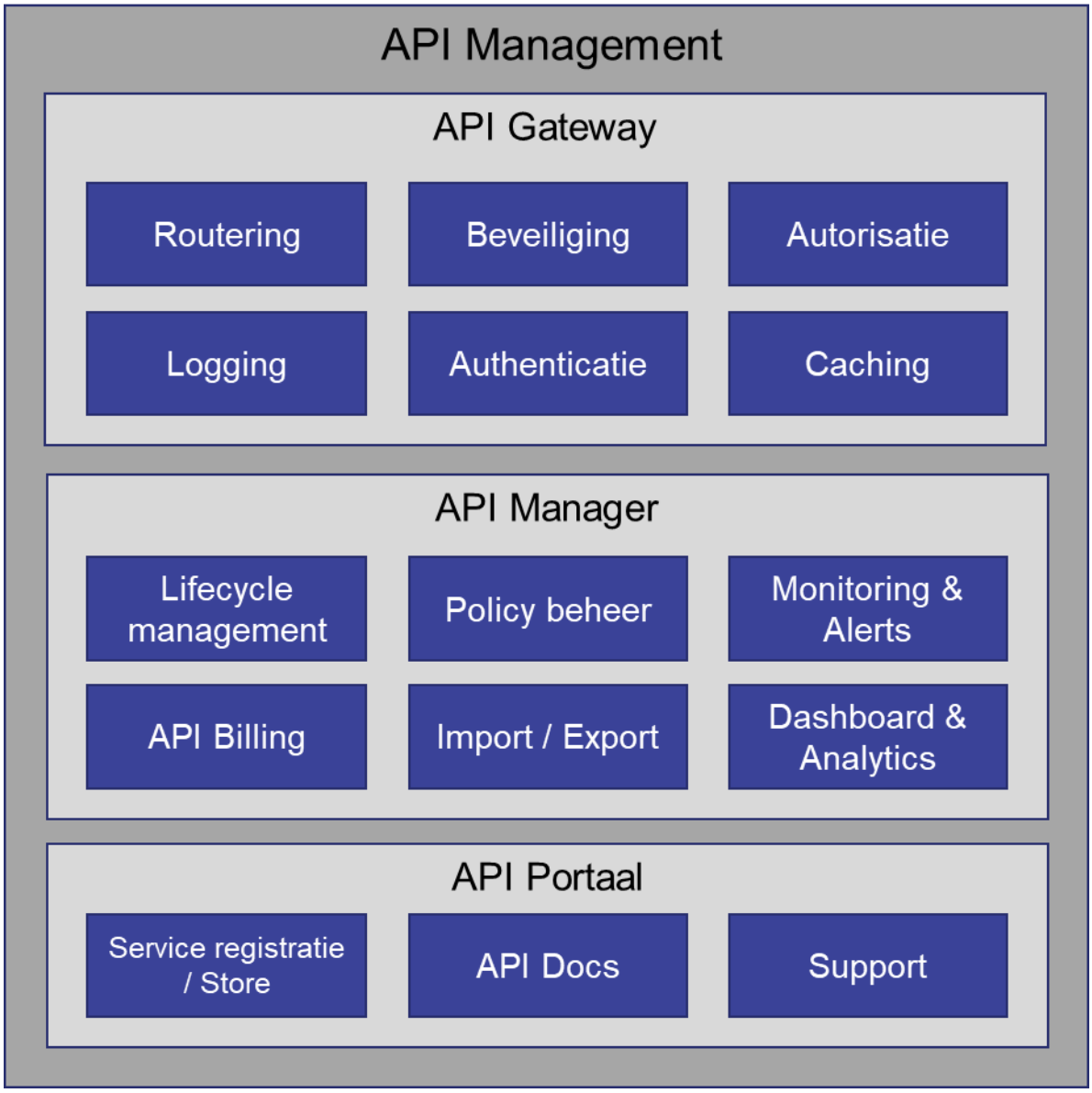
Toegevoegde waarde servicebus i.r.t. API-Gateway:
- StUF-koppelvlakken
- Complexe transformaties
- Orkestratie/logica
- Gegevensautorisatie op doelbinding

[1] Zie StUF Berichtenstandaard - GEMMA Online

Referentiecomponenten
API-Management tooling omvat de referentiecomponenten API-Gateway, API-Manager en een API-Portal. Hieronder is beknopt beschreven wat overheidsorganisaties hieronder kunnen verstaan:

- API-Gateway: hiermee worden gegevensverbindingen technisch gefaciliteerd, beveiligd en gemonitord. Alleen de applicaties (afnemers en aanbieders van API's) gebruiken de gateway.
- API-Manager: zorgt voor de configuratie van de gateway en het beheer van de API's, op basis van patronen en zogenaamde policies.
- API-Portal: een portaal waarin de aangeboden API's aan het brede publiek worden gepresenteerd. Gebruikers zijn medewerkers geïnteresseerd in de ontwikkeling en het gebruik van API's. Dit kunnen zowel medewerkers van de overheid zijn, als ook externen (bijvoorbeeld van software leveranciers of ketenpartners).

In de volgende paragrafen zijn per component de generieke functionaliteit in detail beschreven.



Figuur 3: Functionaliteit binnen API-Management

5.3 Functionaliteit API-Gateway

Een API-Gateway wordt ingezet als poort tot het achterliggende datalandschap.
In cloud-native implementaties zie je steeds vaker een meer gedistribueerd model m.b.v. micro gateways (als ingang voor iedere Cloud-omgeving) i.p.v. een corporate API-Gateway die alles regelt, eventueel in combinatie met een service mesh[2]. Een hybride opstelling is ook mogelijk.

Routering

- **Inrichting van routering**
  Het is mogelijk om (light weight) routering te definiëren binnen de API-Management tooling. Een routering kan ingericht worden op basis van verschillende criteria, zoals bijvoorbeeld afzender of inhoud.

- **Gegevensuitwisseling protocollen en formaten**
  Binnen het gegevensuitwisselingsdomein worden verschillende uitwisselprotocollen- en formaten toegepast, zoals: SOAP, REST, WFS3.0, JSON, GeoJSON, XML. Waarbij ook de transformatie tussen de verschillende protocollen en formaten voor de meest voorkomende gevallen wordt ondersteund door de API-Gateway.

- **Integratie met landelijke integratiefunctionaliteit**
  Het moet mogelijk zijn om makkelijk verbinding te kunnen opzetten met een integratieoplossing voor system-2-system communicatie (bijvoorbeeld NLX).

- **Rate limiting**
  Rate limiting (throttling) op de API Gateway biedt bescherming tegen een bovenmatig aantal verzoeken die worden afgevuurd op de omgeving van de overheidsorganisatie, waardoor er storingen op de backend systemen kunnen optreden. Dit kan bij reguliere bedrijfsuren het 'spitsuur' zijn waardoor deze beveiliging moet worden ingeschakeld. Rate limiting is ook een manier om een SLA af te dwingen o.b.v. een contract.
  Er kunnen verschillende profielen of plannen worden aangemaakt waarin deze quota worden geconfigureerd. Of het toekennen van voorrang op de afhandeling van bepaalde calls is mogelijk, bijvoorbeeld door bulk-processen voor een korte periode voorrang te verlenen op andere calls. Hierdoor is gecontroleerd traffic management mogelijk, waardoor eventuele verstoringen op een later moment door piekbelasting juist kunnen worden voorkomen.. Een profiel of plan is te koppelen aan een specifieke API.

Beveiliging

- Zie voor authenticatie op basis van OAuth2.0 API Designrules Extensions (Nederlandse API Strategie IIb) (geonovum.github.io)
- **Hardening**
  Zorgt ervoor dat de tooling is opgewassen tegen bedreigingen (o.a. hacking) van buitenaf.
- **Validatie**
  Het valideren van inkomende en uitgaande API-calls tegen geldende semantische definities (onderdeel van API Policies).
- **Geoptimaliseerd voor laagdrempelige en meest gebruikte protocollen/standaarden.**
  Voorbeelden:
  - Application Level Security (OSI level 7): Een verbinding over een internet moet veilig zijn, de standaard voor het veilig uitwisselen van gegevens

[2] Service mesh - Wikipedia

tussen twee of meerdere devices is het HTTPS Protocol. Bij gebruik van HTTPS worden de gegevens versleuteld, waardoor het voor een buitenstaander, bijvoorbeeld iemand die afluistert, het onmogelijk zou moeten zijn om te weten welke gegevens er worden verstuurd.

- Transport Level Security (OSI level 4): Voor het beveiligen van de transportlaag binnen het netwerk wordt gebruik gemaakt van TLS (Transport Layer Security-protocol). Het is het meest gebruikte protocol voor het opzetten en gebruiken van een cryptografisch beveiligde verbinding tussen twee computersystemen (client en server), waarbij versie 2.0 de minimale standaard is. Er wordt hierbij gebruik gemaakt van beveiligingscertificaten die zorgen voor de cryptografische versleuteling.

Autorisatie
De gateway zal op runtime de ingestelde autorisatie afhandelen.

Per profiel (overeenkomstig met een rol, raakvlak met Role Based Access Control (RBAC)) is het grofmazig instelbaar welke data opgevraagd mag worden door een gebruiker via de API (in dit geval een applicatie i.c.m. de aanduiding van de gebruiker op persoonsniveau). Dit geeft mede invulling aan de eisen die worden gesteld vanuit wet- en regelgeving, namelijk dat autorisatie is ingericht conform doelbinding.

Fijnmazige (domein specifieke) autorisatie wordt idealiter afgehandeld door de domein applicatie. Echter ondersteunen verschillende gebruikers niet de correcte manier van API's aanroepen en doelbindingsregisters ontbreken veelal nog, ook is IAM in veel overheidsorganisaties niet in die mate op orde dat hierin volledig sturing aan gegeven kan worden. Fijnmazige autorisatie kan hierdoor eventueel nog niet door de backend worden afgehandeld, een alternatief hiervoor is dat er verschillende API's naast elkaar worden aangeboden vanuit de gateway.

Logging
De gateway zal op runtime logging opbouwen.
De logging van de gebruikte API's is noodzakelijk voor audit doeleinden (i.h.k.v. privacy, beveiliging en transparantie). Om deze achteraf te kunnen inzien:

- Wie heeft data geraadpleegd?
- Welke data is geraadpleegd? Enkel de metadata ervan is zichtbaar.
- Waarom de data is geraadpleegd? Doelbinding wordt vastgelegd.
- Wanneer de data is geraadpleegd
- Hoe de data is geraadpleegd?

De technische logging over het afgelopen jaar is op een efficiënte wijze raadpleegbaar en doorzoekbaar. Voor technische en functionele logging is de API-gateway in staat om deze weg te schrijven en te exporteren naar een externe logging of monitoring tool.

Authenticatie
De gateway op runtime de ingestelde authenticatie afhandelen (zie meer informatie onder API-Manager).

Caching
In bepaalde gevallen kan het van meerwaarde zijn dat een bron niet opnieuw bevraagd hoeft te worden, maar dat de laatst opgehaalde gegevens in een zogenaamde 'cache' voor korte tijd bewaard blijven. Caching kan worden ingevuld door de API-Gateway. De caching van de API-Gateway is instelbaar.

5.4 Functionaliteit API-Manager

Life cycle management
Life cycle management is essentieel onderdeel in een applicatielandschap dat continu verandert en een onzekere toekomst kent. Life cycle management omvat bijvoorbeeld de volgende functionaliteit:

- het beschikbaar stellen van API's, door het importeren van externe API configuraties/definities of door het creëren van een eigen API. Een eigen API maakt het mogelijk om voor een organisatie specifieke verbindingen op te zetten, zoals voorbeeld:

    - specifieke convenience API's[3] te creëren op landelijke system API's (voorbeeld 1 convenience API die 3 system API's aanroept).
    - Organisatie specifieke bronnen te ontsluiten (zoals kernregistraties)
    - JSON API's te creëren boven op interne SOAP API's.

    Het volgende kan per API worden vastgelegd:
    - Het definiëren van de URL waarmee de API aangeroepen kan worden door gebruikers.
    - Het definiëren van het endpoint (voor data ontsluiting of datamutaties).
    - Het inrichten van policies (zie beveiliging).

- het aanbieden van nieuwe versies (eventueel via oplossingen zoals GIT)
- het uitfaseren van oude versies.

Policy beheer
Voor iedere verbinding is het instelbaar welke maatregelen er genomen moeten worden om de API zo specifiek mogelijk open te zetten voor afnemers (ook wel aangeduid als afnemers)

Monitoring & Alerts
Het versturen van een (SMS en/of mail) notificatie aan beheerders indien er enkele bepaalde gebeurtenis optreedt. (bijvoorbeeld wanneer een bepaalde gegevensverbinding uit de lucht is of een certificaat op korte termijn zal verlopen).

API Billing (Monetizing)
Voor inzicht in rapportages en dashboards zijn API's goed te gebruiken hierdoor het mogelijk is om het gebruik van API's door te belasten aan de afnemers. Dit is voor overheden interessant om zo eventuele kosten door te belasten op de afzonderlijke organisaties, maar ook richting ketenpartners.

Import/Export tussen overheidsorganisaties
Overheden kunnen grote efficiency voordelen behalen door de API-serviceregistratie met elkaar uit te wisselen via de Open API Specificatie versie 3.0 (OAS3.0[4]). Het is ook mogelijk om (delen van) de API-serviceregistratie van een overheid te exporteren (incl. policies), zodat andere overheden die ook kunnen gebruiken. Daarnaast is het mogelijk om (delen van) de configuratie van een andere overheid / departement te importeren.

Dashboarding & Analytics

- **API use dashboards/Analytics**
  Voor ontwikkelaars, beheerders en management is het interessant om inzicht te hebben in welke API's wanneer en hoe vaak worden gebruikt en wat de performance is. Op basis van deze informatie kan bijvoorbeeld gesleuteld worden

[3] Zie 3.4.2 in de landelijke API-standaard
[4] https://swagger.io/specification/

aan de inrichting om het applicatielandschap efficiënter te laten functioneren. Deze dashboarding/analyse kan vanuit verzamelde data (o.a. logging) worden opgebouwd.

- **Auditing**
  Voor audit-doeleinden is mogelijk om logs weg te schrijven naar een logging voorziening.

5.5 Functionaliteit API-Portaal

Serviceregistratie (ook wel aangeduid als API-Explorer of API-Gallery)
Voor medewerkers die betrokken zijn bij de software ontwikkeling is het interessant te weten welke API's beschikbaar zijn, welke data (of functionaliteit) ermee kan worden opgehaald en hoe de API moet worden aangeroepen (zie ook API docs).

De serviceregistratie is eenvoudig doorzoekbaar. Daarnaast is registratie/publicatie via de Open API Specificatie versie 3.0 (OAS3.0) wenselijk, zodat API-definities van andere (externe) bronnen op basis van de (OAS3.0) makkelijk beschikbaar gesteld kunnen worden.

API Docs
Alle documentatie gerelateerd aan API’s is via het portaal eenvoudig te beheren en terug te vinden. Bijvoorbeeld voorbeeldcode: Ontwikkelaars die software maken die ook een API-call kunnen versturen zijn gebaat bij het verkrijgen van de voorbeeldcode. Hierdoor kan de ontwikkeltijd aan afnemer zijde nog verder teruggebracht.

Support
Het is de voorkeur van overheden dat vanuit de tooling helder op te maken is waar je terecht kan voor ondersteuning.

Note: Landelijke ontwikkelingen op dit vlak i.r.t. developer.overheid.nl kunnen er voor zorgen dat deze functionaliteit op termijn centraal landelijk beschikbaar is.

Bijlage: Openstaande punten en onduidelijkheden

In deze bijlage zijn de belangrijkste voor de werkgroep Architectuur nog openstaande punten en onduidelijkheden opgenomen. De werkgroep verwacht hier in samenwerking met VNG Realisatie Architectuur & Standaards, Kennisplatform API, verschillende teams vanuit Common Ground, overheden en marktpartijen nader invulling aan te kunnen geven.

- NLX als landelijke integratiefunctionaliteit
  Het is nog niet helder welke rol NLX op welke termijn exact gaan pakken. De overlap met functionaliteit uit een API-Gateway is groot. In deze requirements specificatie is wel uitgegaan van het scenario dat NLX de komende jaren een steeds belangrijkere rol gaat pakken in gegevensuitwisseling voor overheidsorganisaties. De ontwikkelingen vanuit Common Ground/NLX-team en Groeipact worden gevolgd om de positie goed te kunnen bepalen.
- Digitale identiteitsplatformen
  Voor overheidsorganisaties is het van meerwaarde om authenticatie-voorzieningen zoals eIDAS, DigiD, eHerkenning en IRMA via de integratielaag aan te roepen. Hierdoor vermindert het aantal connecties met deze landelijke brokers, waardoor ook audits op het gebruik van deze voorzieningen gereduceerd kunnen worden.
- Rol IAM in het gegevenslandschap
  De werkgroep architectuur is in deze requirements specificatie uitgegaan van het toepassen van Role Based Access Control (RBAC) en bewust nog niet van Attribute Based Access Control (ABAC), aangezien het voor veel overheidsorganisaties voor nu een stap te ver lijkt. Voor het toepassen van RBAC moet eerst een flinke professionaliseringsslag gemaakt worden binnen IAM. Momenteel is het onduidelijk wanneer dit mogelijk is.
- Toepassing GraphQL
  De volgende stap na de inzet van JSON REST API’s lijkt de inzet van GraphQL. Hoe de inzet van GraphQL zich binnen de markt verder ontwikkelt is momenteel onduidelijk. Het geeft tevens aan dat de techniek altijd zal blijven doorontwikkelen en dat het voor overheidsorganisaties belangrijk is een model te ondersteunen wat hier op voorsorteert.