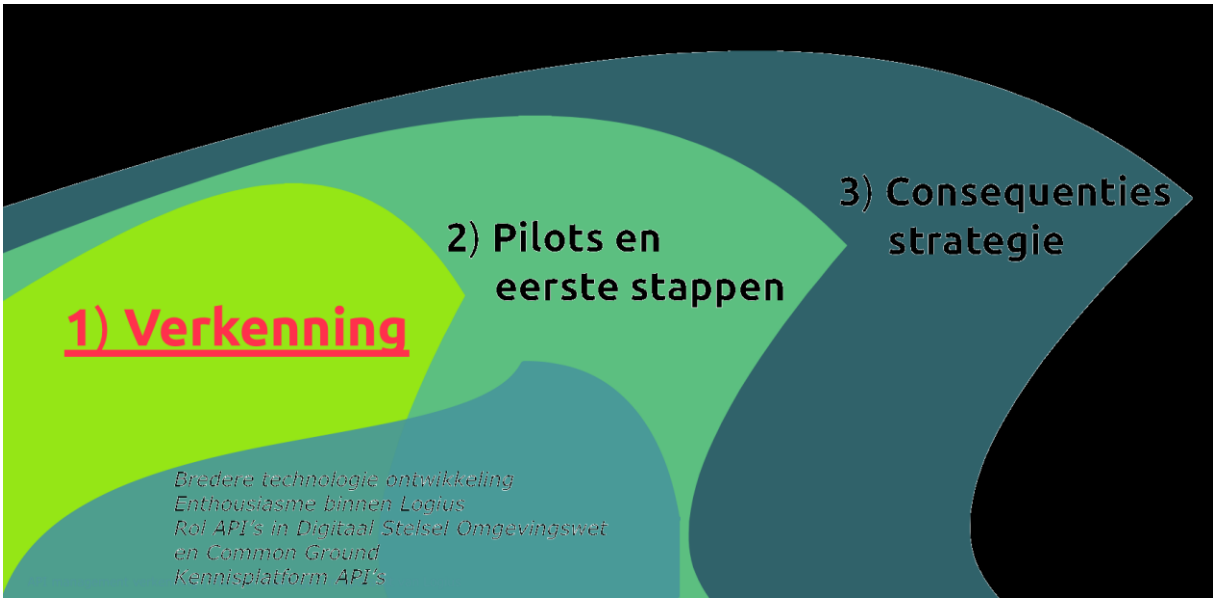
API’s versnellen digitalisering

Het API mes snijdt aan twee kanten: versnelling van de digitaliseringsagenda en doorbreken van verzuiling tussen de GDI voorzieningen

Versie 1.0 – januari 2019



Managementsamenvatting

API's zijn dè integratievorm en dè dienstverleningsvorm van digitalisering

API-gewijze dienstverlening is sinds een aantal jaren sterk gegroeid. API's zijn een de facto standaard voor integratie in de hyper-vernetwerkte wereld. Wij benutten voortdurend API's wanneer wij de diensten op onze smartphone gebruiken. Voor gebruikers is dit onzichtbaar, toch hebben API's niet alleen technologische relevantie. De eenvoud van API-integratie heeft het mogelijk gemaakt dat organisaties losse functies toegankelijk maken voor (verantwoord) gebruik door derden. Dit is daadwerkelijk een "andere manier van werken" en de visie op dienstverlening die "Silicon Valley" groot gemaakt heeft. Daarmee is API-gewijze dienstverlening belangrijk voor de Agenda Digitale Overheid NL DIGIbeter.

Logius en andere uitvoeringsorganisaties hebben de eerste stappen gezet op weg naar API-gewijze dienstverlening door de overheid. API-gewijze dienstverlening speelt bijvoorbeeld een rol in het Digitaal Stelsel Omgevingswet en in Common Ground van de VNG.

Het API mes snijdt bovendien aan twee kanten: naast versnelling van de digitaliseringsagenda kunnen API's een grote bijdrage leveren aan het doorbreken van de verzuiling tussen voorzieningen van de Generieke Digitale Infrastructuur (GDI). Dit maakt API's voor Logius dubbel aantrekkelijk!

Eenvoud heeft een prijs: nieuwe standaarden en doorontwikkelde GDI

API's bieden eenvoudige integratie voor de afnemers ervan. De aanbieder van API's – in deze verkenning is dat Logius – moet hier veel voor inrichten. Iedere API vormt een product met een eigen doelgroep die hoge eisen stelt. Dit vereist een wendbare organisatie die ingericht is op het leveren van een continue stroom kleine verbeteringen (Agile en devops). Technologisch vereist het een API-managementplatform en nieuwe vormen van authenticatie. Hiervoor is in de markt een volwassen scala producten beschikbaar, zowel van grote platformleveranciers als open source, zowel Cloud gebaseerd als on site. Meerdere overheidsorganisaties hebben een dergelijk platform in gebruik of zijn daarmee bezig. Verder vereist API-gewijze dienstverlening een cultuur van developercommunities. API-gewijze dienstverlening is niet denkbaar zonder "developer.logius.nl en developer.overheid.nl portalen waar het aanbod aan API's voor afnemers beschikbaar is. Mede daarom zijn succesvolle API-strategiëen gebouwd op het adagium dat voor interne integraties dezelfde eisen worden toegepast als voor externe dienstverlening ("eat your own dog food").

Gezien de taak die Logius heeft in de standaardisatie binnen de Nederlandse overheid is het belangrijk dat Logius mede vormgeeft aan de nieuwe standaarden die voor API-gewijze dienstverlening belangrijk zijn. Deze standaardisatie kan het beste in samenhang met de bestaande familie van Digikoppelingstandaarden worden gerealiseerd.

Impact van API-strategie op Logius: wendbaarder en vernieuwde GDI

API technologie wordt al toegepast door Logius. Wanneer Logius kiest voor een expliciete API-strategie dan wordt dit fors versneld. Een dergelijke keuze gaat goed samen met de lopende transformatie naar Agile werken (invoering SAFe). Een API-strategie leidt tot een andere relatie met afnemers omdat aan de bestaande situatie communities worden toegevoegd die tot directere feedback leiden maar tegelijk grotere wendbaarheid in de reactie vereisen. Bovendien zou een API-strategie een inhoudelijk (enterprise architectuur) element worden in het portfolio. Alleen op die manier kunnen de benodigdheden voor API-gewijze dienstverlening gerealiseerd worden over de naast elkaar staande GDI-voorzieningen heen. De sinds begin dit jaar ingerichte Agile release train die zowel MijnOverheid als DigiD omvat is de voor de hand liggende plek om deze strategie te beginnen, daarnaast liggen er ook mogelijkheden in de Stelseldiensten release train. Naast het uitvoeren van pilots, hetgeen al gepland is, zijn het uitspreken dat API-gewijze dienstverlening de gewenste richting is en het toewerken naar een plan daarvoor, belangrijke stappen om de al in gang gezette beweging verder te stimuleren.

Met een expliciete API-strategie bevordert Logius vanuit eigen kracht de digitalisering en vernieuwing van de GDI. API-gewijze dienstverlening vormt qua technologie en dienstverleningsvorm het adequate evenwicht tussen volwassenheid en innovatie (early majority). Bovendien draagt Logius daarmee maximaal bij aan versnelling en verdere uitvoering van NLDigibeter en de steeds diepgaandere digitalisering van haar afnemers. Dit maakt een zichtbare en effectieve verbetering van de digitale dienstverlening aan burgers en bedrijven mogelijk en versterkt dat maatschappelijke baten van een doorontwikkelde generieke digitale infrastructuur.

Leeswijzer deze verkenning

Het doel van deze verkenning[1] is Logius en haar afnemers het materiaal te bieden op basis waarvan de keuze voor een explicietere API-strategie kan worden afgewogen. Het eerste hoofdstuk schetst wat API's zijn, wat API-gewijze dienstverlening inhoudt en hoe dit in de bredere context van digitalisering past. Het tweede hoofdstuk biedt op basis van een beknopte marktscan inzicht in de volwassenheid van benodigde ICT-producten. Het derde hoofdstuk gaat uitgebreid in op de benodigde standaardisatie in het licht van de bestaande Digikoppeling standaard, vervolgens geeft het vierde hoofdstuk een belanghebbende analyse. In hoofdstuk 5 wordt de impact op Logius als organisatie beschreven. Dit wordt afgesloten met een advies in hoofdstuk zes en een aantal bijlagen over specifieke punten.

[1] Voor de opdrachtomschrijving zie bijlage A.

1. API-gewijze dienstverlening is belangrijk voor Logius

1.1 Snelle maatschappelijke verandering! U hebt vandaag al een API gebruikt!

API-gewijze dienstverlening is sinds een aantal jaren sterk gegroeid. In diverse sectoren is het een volwassen vorm van dienstverlening, deze sectoren zijn daardoor wezenlijk veranderd. Bijvoorbeeld het onderscheid tussen klassieke televisie en Netflix of het boeken van een vliegtuig via een reisbureau of via vliegtickets.nl. Vrijwel zeker hebt u vandaag, voor u dit las, al meerdere API's gebruikt[2].

Gezien het feit dat Salesforce en eBay de eerste API's in de hier bedoelde vorm in 2000 lanceerden, kan met recht worden gezegd dat API-gewijze dienstverlening volwassen is. API-gewijze dienstverlening is onderdeel "van de nieuwe digitale technologieën die onze maatschappij en economie razendsnel veranderen" en daarmee uitgangspunt voor de digitaliseringsstrategie van het kabinet[3].

API's zelf zijn niet nieuw. Interfaces, waarmee het ene programma met het andere praat, bestaan sinds de eerste computerprogramma's. De eerste vijftig jaar van ICT zijn deze interfaces technische interfaces, het domein van de ICT-afdeling en system integrators. De term API, letterlijk Application Programming Interface", verraadt niet meteen dat er sindsdien echt wat veranderd is. Hoezo spreken we nu van een **API-economie**[4] en van API-gewijze dienstverlening? De interface ofwel de API is zelf een "product" geworden. API's zijn in veel organisaties dè communicatievorm met de buitenwereld. Dit is het resultaat van een zichzelf versterkende cirkelbeweging tussen een groeiend vraag naar integratie en technologie die integratie eenvoudiger maakt. De gegroeide behoefte aan ketenintegratie hoort bij het hyper-vernetwerkt zijn, het internet dat alles met elkaar verbindt. De mogelijkheden voor eenvoudiger integratie zijn enorm gegroeid door verdere standaardisatie. Deze zichzelf versterkende cirkel heeft geleid tot "de API" zoals we die nu kennen.

1.2 Wat is een API? Wat moet *iedereen*[5] weten over API's?

Een Application Programming Interface (API[6]) *is een goed gedocumenteerde interface om (derden) digitaal toegang te geven tot stukjes van uw taakuitvoering en dit op een manier die eenvoudig te gebruiken is en enorm goed schaalbaar tot internetvolumes.*

Belangrijk is dat API's bedoeld zijn voor toepassing in andere applicaties. Een API is een gestandaardiseerde toegang (interface of koppelvlak) die bepaalde functionaliteit voor andere toepassingen ontsluit. Digitale processen bevatten veel van dergelijke aanroepen. Net zoals een webpagina informatie ontsluit voor een algemeen publiek van lezers, ontsluit een API geprogrammeerde functionaliteit voor andere applicaties. Meestal worden met API's dan ook "**web-API's**" bedoeld: API's die toegankelijk zijn via het internet. Net als voor

[2] Check: Netflix en vergelijkbare aanbieders leveren 100% API-gewijze dienstverlening. De andere manier van televisiekijken die hiermee verbonden is, volgt mede uit deze technologiestrategie. Voor vliegtickets, zie het voorbeeld verder op. Tenslotte benut vrijwel iedere app op uw telefoon API's en 90% van de Nederlanders heeft een smartphone [https://www.cbs.nl/nl-nl/nieuws/2018/05/nederland-koploper-in-europa-met-internettoegang] en gebruikt deze heel veel keren per dag https://www2.deloitte.com/nl/nl/pages/technologie-media-telecom/articles/global-mobile-consumer-survey.html kijk de video's!

[3] Nederlandse Digitaliseringsstrategie, 15 juni 2018, 26 543 nr. 541, paragraaf 1

[4] The API Economy: Turning Your Business Into a Platform (or Your Platform Into a Business), Gartner G00280448

[5] NL DIGIbeter stelt (pag. 23) dat iedere beslisser binnen de overheid in staat moet zijn om digitaliseringsaspecten mee te nemen in beslissingen en dat iedere ambtenaar een bepaald minimum aan kennis over functioneel gebruik van ICT dient te hebben. Wat betreft API's formuleert deze paragraaf dit minimum.

[6] Vanaf dit punt in het document worden de technische, meestal Engelstalige, termen die van belang zijn voor het onderwerp de eerste keer dat ze gebruikt worden gemarkeerd. Deze termen zijn opgenomen in de begrippenlijst.

websites geldt overigens dat er openbaar toegankelijke API's zijn en API's waarvoor eerst wordt ingelogd. De openheid van API's kan gepaard gaan met hoge veiligheid.

De afnemers van API's zijn applicaties. Deze worden gemaakt door ontwikkelaars. Daarom zijn documentatie en eenvoud zo essentieel. Omdat op voorhand niet precies te voorspellen is hoe derden een web-API zullen toepassen, is het heel belangrijk dat deze schaalbaar is. Een redenering die we ook kunnen omkeren: dankzij de eenvoud en standaardisatie van internet is het inmiddels mogelijk om primaire bedrijfsfuncties op basis van web-API's binnenstebuiten te keren en verantwoord aan te bieden aan buitenstaanders.

Eenvoud, schaalbaarheid en verantwoord aanbieden zijn dus essentieel voor API's. Dit heeft geleid tot het toepassen van bepaalde standaarden en architecturen voor API-gewijze dienstverlening. "**REST**" is een veel gebruikte aanduiding in dit kader. Het staat voor een vorm van communicatie tussen de afnemende applicatie en de functionaliteit die achter de API-interface zit. Samengevat is dit "in de beperking herkent zich de meester" voor integratie tussen systemen. REST is niet de enige manier om API's te realiseren, maar wel de de facto standaard. (Meer over REST oftewel **RESTful API**'s in bijlage B).

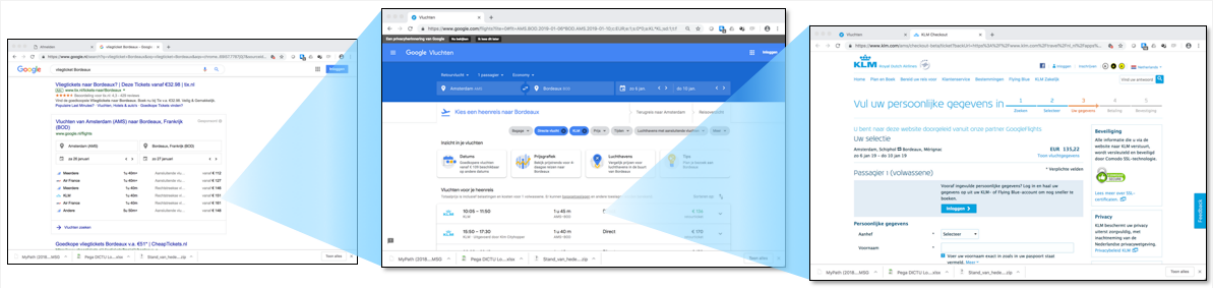
Voorspelbaarheid, schaalbaarheid, het veilig en verantwoord aanbieden van API-gewijze dienstverlening worden bereikt door een **API-managementplatform** toe te passen.

Voorbeeld: Stel ik wil de definitie van API van Wikipedia opnemen in mijn website. Dan kan ik deze opzoeken en kopiëren. Als deze definitie in Wikipedia aangepast wordt[7], dan toont mijn website de verouderde versie. Ik kan ook de API-gewijze versie van Wikipedia gebruiken. Ik neem dan de onderstaande link op in mijn website.

https://nl.wikipedia.org/w/api.php?format=json&action=query&prop=extracts&exintro&explaintext&redirects=1&titles=Application Programming Interface

Iedere keer wanneer mijn website geladen wordt zal deze de actuele versie van de definitie ophalen en tonen. Probeer dit zelf door de link in een browser te kopiëren[8].

Voorbeeld: Stel ik wil naar Bordeaux vliegen. Bij Google zoek ik op "vliegen Bordeaux". Meteen na de advertenties verschijnen de google.com/flights resultaten. Als ik daar de KLM aanklik dan gebruikt Google de API's van KLM om mij meer details te tonen. Vervolgens kan ik het hele boekingsproces doorlopen. Ik hoef de gegevens die ik bij Google heb ingevoerd niet opnieuw in te voeren.



Om dit mogelijk te maken heeft Google de API van KLM gebruikt. KLM weet niet op voorhand wie de API's die zij aanbiedt gaan gebruiken. In die zin zijn API's net als websites: je maakt een website voor een algemeen publiek en zet hem openbaar. Alleen door gebruikersstatistieken te meten krijg je inzicht in de afnemers. Om een indruk te krijgen van de documentatie die KLM aanbiedt, ga naar:

https://developer.airfranceklm.com

Door in dit voorbeeld te vergelijken hoe resultaten van verschillende luchtvaartmaatschappijen getoond worden, zie je dat de ene maatschappij deze API's veel beter voor elkaar heeft dan de ander.

Zo zien we dat '`developer.logius.nl`' de voor de hand liggende naam zou zijn van de plek waar Logius in de toekomst haar API's aanbiedt.

1.3 Context: API-management en platformen die dat faciliteren

De andere kant van de medaille van de eenvoud van API's is dat er behoorlijk wat faciliteiten ingericht moeten worden om API's goed te kunnen benutten. Dit zijn faciliteiten waar de aanbieder van de API's in voorziet. Het is dus vooral de afnemer die profiteert van de eenvoud. Goed aanbieden van API's vereist veel meer dan enkel technologie. De aanbieder richt daartoe "API-management" in. De ICT-voorzieningen die daarbij worden gebruikt noemt men API-managementplatformen. API-management omvat alles wat nodig is om de gehele levenscyclus van API's te managen:

[7] En het is zeker mogelijk de definitie van Wikipedia te verbeteren aangezien die nogal in techneutentaal gesteld is ☺.
[8] En de poweruser mag "nl" in "en" veranderen en "json" in "xml"

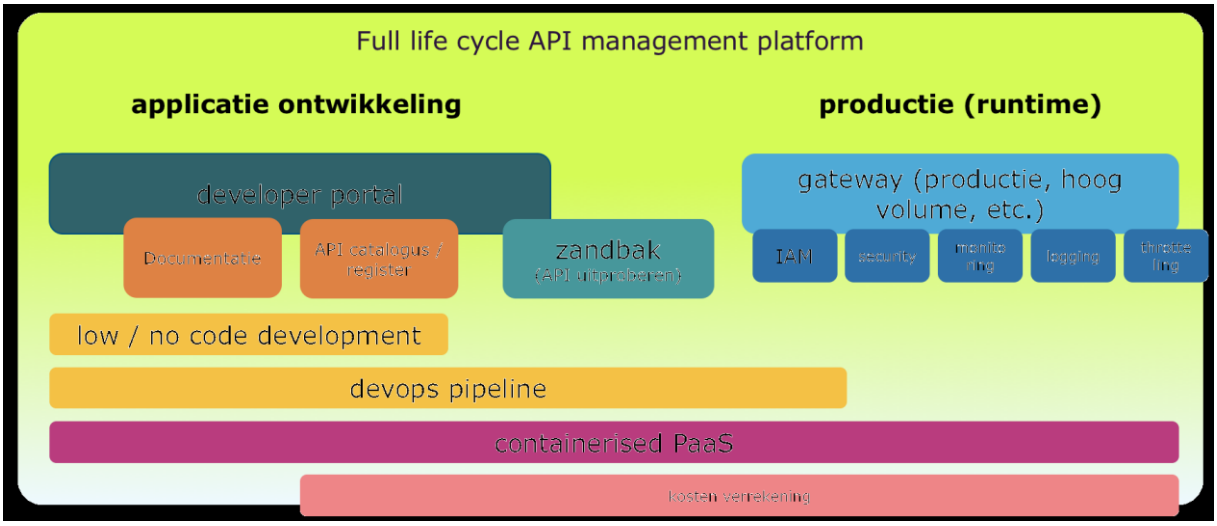
- planning, ontwerp,
- realisatie en testen,
- documentatie gericht op externe afnemers,
- publiceren,
- operationeel houden van de API's voor uitproberen door afnemers,
- operationeel houden van de API's voor benutten door afnemers in productie,
- monitoring,
- onderhouden, versiebeheer en
- uitfasering.

Deze voorzieningen, het API-managementplatform, omvatten een ontwikkelaarsportaal gericht op de community van ontwikkelaars die de API's benutten en het runtime management van API's. De runtime component, dat wil zeggen de productie-omgeving, wordt veelal API gateway genoemd.

Een API ontwikkelaarsportaal bevat:

- een catalogus van API's,
- documentatie,
- hulpmiddelen om API's snel en makkelijk te benutten en
- testomgevingen om API's uit te proberen tijdens het ontwikkelproces van applicaties waarin ze worden opgenomen.

De gateway omvat alles wat nodig is om verantwoord API's aan te bieden aan buitenstaanders: beveiliging, maximering van gebruik, monitoring, meting van gebruik (en dus kosten), etc. In feite biedt een dergelijk platform de gehele ICT-architectuur om met API's te werken.



Het is onmogelijk om een effectief API-programma te realiseren zonder een volledige API management voorziening[9].

Grootschalige toepassing van API's gaat samen met de technologische ontwikkelingen op het gebied van **microservices** en **containerisering** van infrastructuren (meer daarover in kader aan eind van dit hoofdstuk). Om succesvol te zijn met API-gewijze dienstverlening moeten organisaties daarom hun ICT architectuur en ontwikkelaanpak aanpassen.

[9] Zie o.a. Gartner https://www.gartner.com/reviews/market/full-life-cycle-api-management

1.4 API-gewijze dienstverlening heeft forse organisatorische impact

API-gewijze dienstverlening omvat het als product leveren van API's op het internet. Het leveren van de API's zelf is een vorm van productmanagement. Een API heeft een levenscyclus net als andere producten[10]. Dat is een wezenlijke verandering ten opzichte van eerdere interfaces, die bleven puur technologie voor interne integratie. API's zijn dienstverlening. Het organisatorisch goed inrichten van de levenscyclus ("**API-lifecycle**") en het productmanagement, zijn een vereiste voor API-gewijze dienstverlening. Dit omvat standaard productmanagementzaken als ontwerpen, realiseren, in gebruik nemen, monitoren van gebruik en vernieuwen of vervangen. Het belangrijkste daarvan is de communicatie met afnemers en het wendbaar reageren op hun verwachtingen. Dit is een eerste aspect van de organisatorische impact van API-gewijze dienstverlening.

Vervolgens zijn er verdere specifieke API aspecten, de belangrijkste daarvan is "community management".

Succesvolle API gebaseerde organisaties besteden hier veel aandacht aan. Deze **community** is de de facto standaard voor het overleg tussen de aanbieder van een API en de menselijke gebruikers ervan. De traditionele scheidslijnen tussen "business" en "IT" worden daarin doorbroken. De afnemers zijn technisch vaardig, maar benutten API's vanuit het business doel dat zij willen bereiken. De aanbieders kiezen ervoor hun business, hun eigen taakuitvoering, bedrijfsfuncties, business processen (hoe je het noemen wilt) open te stellen. Dat vereist inzicht in de businesswaarde, gekoppeld aan inzicht in technologische vereisten zoals het strikt toepassen van de gebruikelijke standaarden. Om deze interactie te versnellen kiezen organisaties ervoor om in hun digitaliseringsstrategie ook de eigen ontwikkelaars te verplichten intern op basis van API's te integreren. Daarbij verwachten ontwikkelaars goed bruikbare zelfservices waarmee ze zonder helpdesks e.d. direct aan de slag kunnen[11].

API-gewijze dienstverlening wijzigt dus de relatie en interactie met afnemers, met de buitenwereld. Intern is er ook impact. API's passen in de trend naar kleinere brokjes functionaliteit die losjes gekoppeld (**loosely coupled**) samenwerken. Dit heeft duidelijke organisatorische gevolgen: het vereist Agile aansturing en DevOps werkwijze en het vereist een door de hele organisatie gedeeld platform voor API management en een volgens PaaS ingerichte infrastructuur. Dit hangt in het algemeen samen met een centraal sturende architectuurfunctie (die echter tegelijk zeer Agile moet zijn en terughoudend ten aanzien van details, want dat zou API-gewijze dienstverlening juist belemmeren[12]).

---

[10] https://www.linkedin.com/pulse/architecture-perspective-api-enablement-using-part-1-david-rutter/ zie onder "Who".

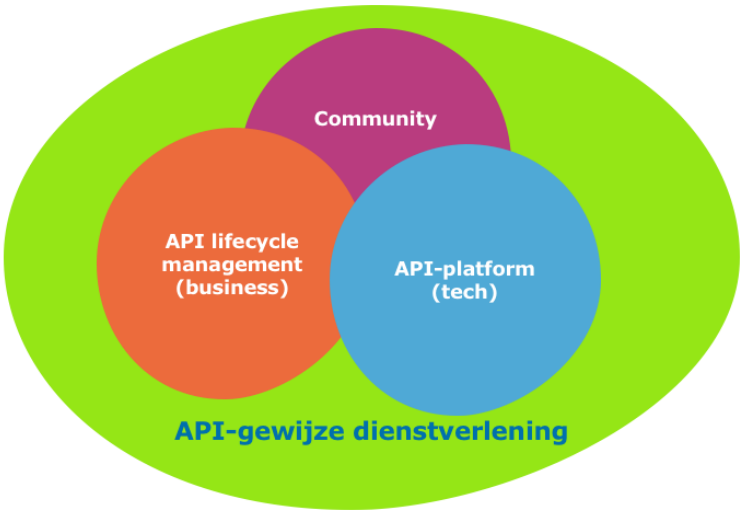
[11] Zelfservices: ontwikkelaars willen 24 x 7 onmiddellijk dingen kunnen uitproberen, geautomatiseerd een test toegangscode krijgen of een test-API uitproberen. Succesvolle developer portals richten zich op deze korte "time to first Hello World" apifriends.com/api-management/api-program-time-first-hello-world/

[12] Zie opmerking over Agile SAFe portfolio management in hoofdstuk 5.

1.5 API succes begint met intern toepassen van API-integratie!

De huidige generatie web-API's die de motor van de API-economie vormen, zijn dus gericht op de buitenwereld. Toch begint het toepassen van API's meestal intern. De kracht van API's voor integratie met de buitenwereld is ook intern geldig. Vrijwel alle organisaties van enige omvang en die meerdere jaren bestaan, kennen interne integratievraagstukken. De kracht van API's is ook van toepassing op deze interne integratievraagstukken. Een succesvolle API-strategie laat het mes dus aan twee kanten snijden: API's worden zowel voor interne integratie als voor dienstverlening aan de buitenwereld ingezet. Dit leidt tot een sterke prikkel voor intern hergebruik en vergroot snel het aantal API's dat – met de nodige extra randvoorwaarden – aan de buitenwereld kan worden aangeboden. Dit is de strategie waarmee digitale ondernemingen snel groeien en wendbaar blijven.

Voor een organisatie als Logius die organisch gegroeid is tot verzuild landschap met functioneel afgebakende silo's, is dat een grote kans. API-gewijze dienstverlening biedt zo een interventie op verzuiling van voorzieningen. Relevant daarbij is dat er tevens goede best practices bestaan om op basis van een API-strategie legacy voorzieningen stap voor stap te moderniseren. Uit de interviews die in het kader van deze verkenning binnen Logius gehouden zijn, blijkt enthousiasme om voor een dergelijke API-gebaseerde technologiestrategie te kiezen, bovendien zijn er al meerdere API's gerealiseerd[13].



1.6 API-gewijze dienstverlening is een enabler voor NL DIGIbeter

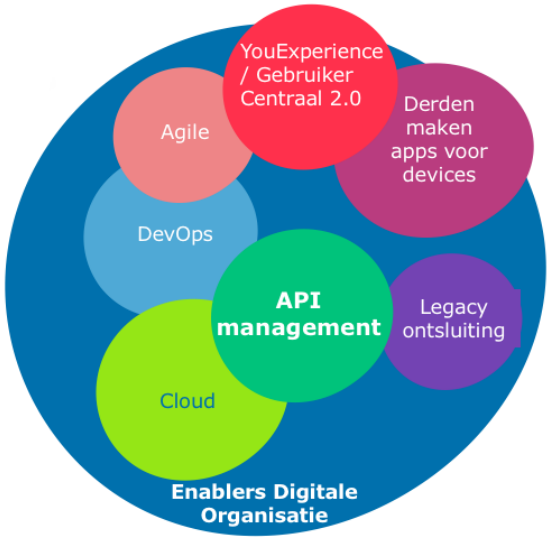
API-gewijze dienstverlening begint aldus met het toepassen voor interne integraties. Vervolgens maakt het een andere uitvoering van de eigen (business)taken mogelijk. Consequent doorvoeren van API-gewijze dienstverlening leidt tot het "binnenstebuiten keren van de functies van een organisatie" (**externalisatie**). Losse functies worden toegankelijk gemaakt voor (verantwoord) gebruik door derden. De eigen taken en processen zijn dan zo ingericht dat ze zelf bestaan uit een opeenvolging van soortgelijke losse functies. Dit is daadwerkelijk een "andere manier van werken"[14] en de visie op dienstverlening die "Silicon Valley" groot gemaakt heeft. Het houdt een visie op digitalisering in, waarbij "digitalisering veel meer is dan bedrijfsvoering"[15]. Evenzeer kan gesproken worden van "van push naar pull". Het "binnenstebuiten keren van de eigen functies" betekent een radicale digitalisering die "de kern van de primaire processen betreft"[16].

[13] Zie Bijlage C
[14] NL DIGIbeter blz. 8.
[15] Idem
[16] Zie rapport Maak Waar!

Daarmee is API-gewijze dienstverlening een duidelijke enabler van de Agenda Digitale Overheid NL DIGIbeter[17]. API-gewijze dienstverlening is namelijk dè strategie om levensgebeurtenis gerichte diensten aan te bieden; het is tevens een voor de hand liggende element voor de modernisering van overheidsportalen[18].

1.7 API-gewijze dienstverlening is een vorm van digitale transformatie

API-gewijze dienstverlening staat niet op zichzelf. API's vormen één van de onderdelen van een "**digital enterprise**", een digitale organisatie. Marktkenners[19] stellen dat API-gewijze dienstverlening de gebruikelijke vorm is om digitale transformatie te bewerkstelligen. Dit is de bredere context waarin we API-gewijze dienstverlening als enabler voor NL DIGIbeter kunnen plaatsen. Het hoort integraal bij de platformisering. Afnemers van Logius doorlopen deze digitale transformatie en baseren hun verwachtingen daarop. Aangezien gemeenten ook afnemers zijn van GDI is de Common Ground ontwikkeling een relevant voorbeeld[20]. En voor de afnemers van de overheid, burgers en bedrijven, is dat niet anders. Sterker nog, zoals NL DIGIbeter aangeeft, de hele samenleving maakt deze transformatie door en dit vereist een overheid die deze kansen benut. Bredere toepassing van API-gewijze dienstverlening levert een duidelijke bijdrage aan het zelf snel wendbaarder worden van de overheid, zoals verwoord in de Nederlandse Digitaliseringsstrategie[21].



API-gewijze dienstverlening zal in de komende jaren hard blijven groeien. Deze groei wordt veroorzaakt door verschillende bewegingen, zoals **IoT**, open banking en **PSD2** wetgeving, conversational chat bots, verdere groei gebruik mobiele devices en sociale media. Relevante voorbeelden als Medmij en Ockto.

Impact van API-gewijze dienstverlening in bankensector

De Europese Payment Services Directive 2 (PSD2) is nieuwe wetgeving die banken dwingt bankrekeningen open te stellen op basis van API-gewijze dienstverlening. Banken moeten rekeninghouders API's aanbieden die toegang geven tot de rekeninggegevens. Bovendien moeten rekeninghouders de mogelijkheid hebben om derden toestemming te geven die API's namens hen te gebruiken. Het betreft in feite de gegevens en handelingen waar

[17] Voor een voorbeeld van uitwerking van deze visie zie https://github.com/Geonovum/KP-APIs-Gebruikerswensen/blob/master/discussie%20document/Hello%20world%20geef%20me%205%20minuten.pdf
[18] Levensgebeurtenissen en modernisering van overheidsportalen zijn concrete doelstellingen van NL DIGIbeter.
[19] Bijvoorbeeld: Forrester, Now Tech: API Strategy And Delivery Service Providers, Q1 2018 https://www.forrester.com/report/Now+Tech+API+Strategy+And+Delivery+Service+Providers+Q1+2018/-/E-RES142594# ; Gartner https://www.gartner.com/doc/3217617/api-economy-turning-business-platform.
[20] https://github.com/VNG-Realisatie/common-ground/blob/master/cg-vision.md
[21] Nederlandse Digitaliseringsstrategie, 15 juni 2018, 26 543 nr. 541, slotparagraaf, pag. 3

rekeninghouders nu via de internetbankier applicatie of app van hun eigen bank toegang toe hebben. Zonder PSD2 is het enkel de bank die bepaalt hoe die applicatie werkt en wat je ermee kunt doen. In de PSD2 situatie kan de rekeninghouder zelf of via een andere dienstverlener betalingsdiensten regelen en ook – met toestemming – derden geld laten overmaken. Het „monopolie" van de bank op de internetbankierapplicatie wordt doorbroken, bovendien worden nieuwe betaaldiensten mogelijk. Ten aanzien van PSD2 spelen vergelijkbare privacy afwegingen als voor overheidsdiensten, met name de vraag in hoeverre de eindgebruiker beschermd moet worden tegen onverstandig gebruik van de nieuwe mogelijkheden van deze API's. De betreffende Europese richtlijn is sinds kort (december 2018 wetsvoorstel aangenomen door eerste kamer) in Nederlandse wetgeving verwerkt.

De **API-strategie** van Logius zal worden bepaald door de taken die Logius binnen de overheid vervult. De bijbehorende impact van API-gewijze dienstverlening voor Logius wordt in hoofdstuk 4 weergegeven op basis van een aantal scenario's.

Voordat de impact in kaart gebracht wordt is het relevant in te gaan op de markt voor API-managementvoorzieningen, worden betrokkenen binnen de overheid in beeld gebracht en wordt ingegaan op standaardisatie. Daarna volgt de organisatorische impact voor Logius aan de hand van de hier geschetste scenario's.

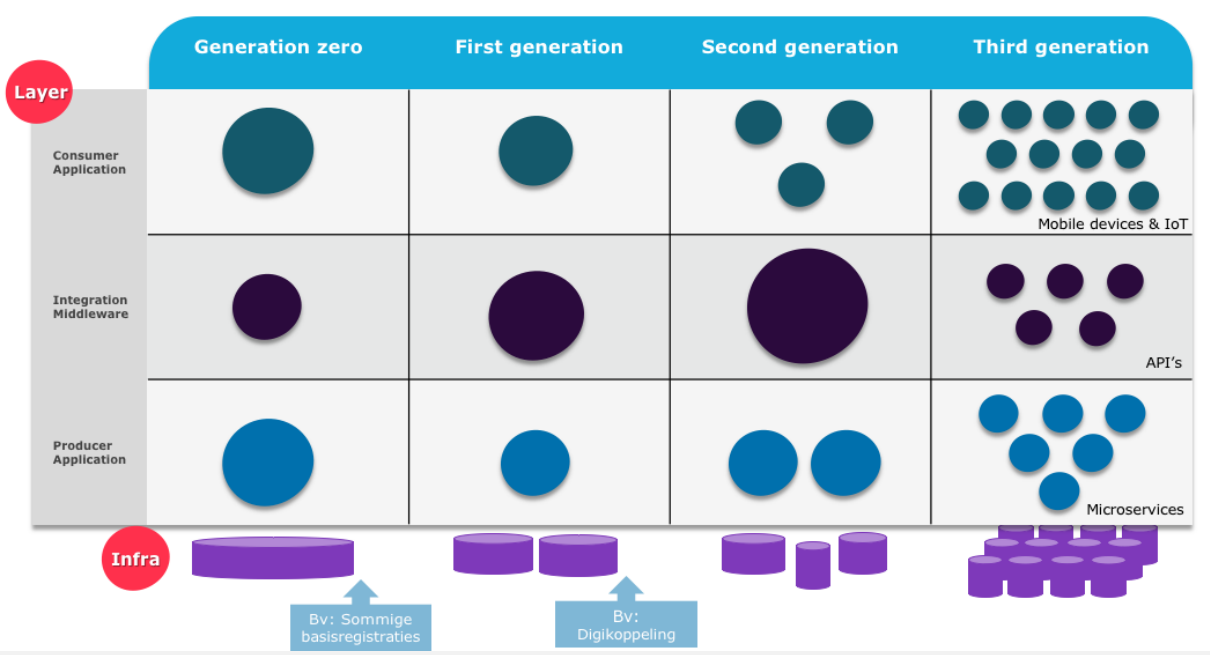
Enkele technologische onderwerpen (uitleg voor beslissers)

RESTful API's API-gewijze dienstverlening wordt vaak in één adem genoemd met **REST** of **RESTful API**'s. REST is een aantal architectuur best practices gericht op voor ontwikkelaars eenvoudig te benutten API's. Het is de de facto standaard voor de hierboven geschetste ontwikkelingen. API-gewijze dienstverlening is op zich ook mogelijk zonder deze best practices en de bijbehorende standaarden toe te passen. Bijvoorbeeld de huidige services van BKWI zijn een voorbeeld van API-gewijze dienstverlening die gebaseerd is op de **WUS** familie van standaarden. Dit neemt niet weg dat RESTful API's en de bijbehorende standaarden, in het bijzonder **HTTP** voor transport, **OAuth** voor authenticatie en **JSON** voor berichtinhoud, wel degelijk iets toevoegen boven bestaande standaarden (en dat betreft onder andere de standaarden waarop de huidige versie van Digikoppeling is gebaseerd). Naast technologische innovatie is het feit dat de huidige generatie ontwikkelaars een sterke voorkeur heeft voor REST in de praktijk relevant. Het volgen van de hoofdstroom van de expertise verkleint de afhankelijkheid van specifieke kennis en vermindert de risico's op dure vervangingen als gevolg van achterhaalde technologie.

Langjarige technologie trend: naar kleinere brokstukken die loosely coupled zijn

API-gewijze dienstverlening en REST API's worden vaak in geassocieerd met **microservices**. Dit heeft te maken met de langjarige trend van de technologie. Hieronder weergegeven van links naar rechts. Generatie nul betreft de grote mainframe voorzieningen van de jaren tachtig gevolgd door (generatie 1) client-service voorzieningen in de jaren negentig van de vorige eeuw. Sommigen van de basisregistraties bevatten nog technologie van die vorm. Over de afgelopen twintig jaar heen gaat de technologie naar kleinere brokjes functionaliteit. Dit gebeurt op meerdere lagen (hier boven elkaar getekend) tegelijk. Van een uitgebreide client applicatie voor een specifiek proces (eerste generatie), zijn we gegaan naar portalen voor eindgebruikers (tweede generatie) en zien we nu een enorme toename van het aantal afnemende applicaties in de vorm van apps (derde generatie). Veelal apps die ieder op zich slechts één taak, één functioneel aspect betreffen. Aan de serverkant betreft het de ontwikkeling van een mainframe, naar b.v. J2EE gebaseerde systemen met steeds meer service georienteerde aspecten (tweede generatie). Microservices vormen daar de derde generatie. Daartussen bevindt zich middleware. In de eerste generatie betreft dit e-mailachtige berichtenverkeer of ftp-achtig file transport. Inmiddels (in de tweede

generatie) betreft dit **ebMS** reliable messaging en dergelijke. In plaats van het periodiek up to date houden van een kopiebestand worden berichten met events uitgewisseld. REST API's met een API-gateway vormen de middleware van de derde generatie. Deze trend zien we tenslotte ook in de infrastructuur. Virtualisatie hoort in de tweede generatie. De huidige trend naar containers is op die laag de trend naar kleinere korrelgrootte.



Op iedere laag geldt: hoe kleiner de korrel, hoe belangrijker standaarden en allerlei faciliterende zaken er omheen die de eenvoud en taakgerichtheid van iedere zelfstandige korrel mogelijk maken. **Loosely coupled**, ofwel losjes gekoppeld betekent daarbij dat de ene component geen kennis heeft van de andere component. Hierdoor kan een component anders geïmplementeerd worden zonder afhankelijkheden met andere componenten, zolang de interface ongewijzigd blijft. In de API-wereld communiceren componenten met elkaar via API's, dezelfde API's als waarmee ze ook met de buitenwereld communiceren (indien dat van toepassing is).

Bovenstaande trend is belangrijk omdat deze ook zichtbaar maakt dat een als silo georganiseerde voorziening geen goed gebruik kan maken van de mogelijkheden van de derde generatie (dat was in de tweede generatie deels ook al zo) en dat het ontbreken van een container gebaseerde **PaaS** infrastructuur een belemmering kan zijn in de derde generatie.

Dat neemt overigens niet weg dat ook een mainframe van de eerste generatie haar functies kan externaliseren via REST API's als middleware. Daarbij geldt het zogenoemde strangler pattern[22] als een van de best practices voor gestage en verantwoorde migratie van oudere generaties legacy naar microservices.

Van boven naar beneden kijkend kun je stellen dat de huidige generatie apps de hoogste performance en schaalbaarheid bereiken met REST API's gebaseerd op microservices en draaiend op een containerised infrastructuur. Maar dit kan best samengaan met delen van de functies die zich nog in eerdere generaties bevinden.

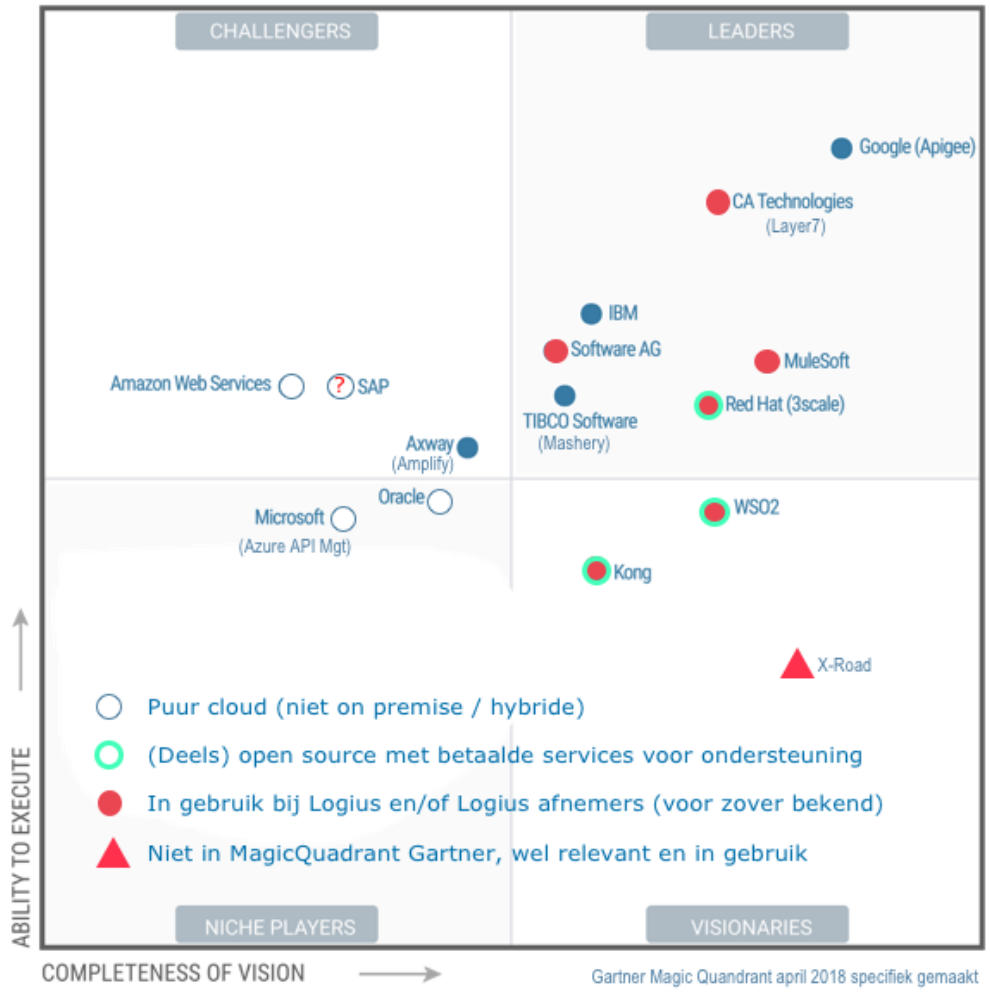
[22] https://books.google.nl/books?id=DJdMDwAAQBAJ&pg=PA278&lpg=PA278&dq=strangler+Cloud+Native+Development+Patterns+and+Best+Practices+by+John+Gilbert&source=bl&ots=GB1HskjAUl&sig=ACfU3U2HCjaoAIjKRljzurSmdpnHEOCmgQ&hl=nl&sa=X&ved=2ahUKEwjAr-Lw4P_fAhUEZVAKHbE3Cx8Q6AEwAnoECAcQAQ#v=onepage&q=strangler%20Cloud%20Native%20Development%20Patterns%20and%20Best%20Practices%20by%20John%20Gilbert&f=false

2. Marktscan API-managementplatformen

2.1 Een volwassen markt van API-managementplatformen

Alle grote spelers hebben in afgelopen jaren eigen **API-managementplatformen** op de markt gebracht en/of kleinere spelers overgenomen om op basis van de innovaties van die spelers hun eigen aanbod op pijl te brengen. Dit geeft een Gartner Magic Kwadrant dat goed gevuld is[23]. Gartner spreekt van "Full Life Cycle API management", dit betreft platformen die het gehele in vorige hoofdstuk beschreven model van API-gewijze dienstverlening mogelijk maken.

Het voor Logius relevante productaanbod omvat ons inziens in ieder geval de volgende producten (die ook in Gartners Magic Quadrant voorkomen).



Een subset hieruit van de voor Logius meest relevante producten moet in ieder geval CA Technologies, RedHat (nu overgenomen door IBM) en WSO2 bevatten gezien het feit dat deze al bij Logius zelf of afnemers zijn ingezet en voldoende goed scoren.

Een nadere analyse van de volwassenheid van de REST ontwikkelingen binnen X-Road mag daarbij niet ontbreken. Dit is de Estse oplossing[24] waar in kader van NLX/Common Ground

[23] Als basis voor de marktscan gaan we uit van "Magic Quadrant for Full Life Cycle API Management", 30 April 2018, ID: G00319327. In deze werkversie zijn teksten die daar (min of meer) letterlijk uitgehaald zijn gekleurd.

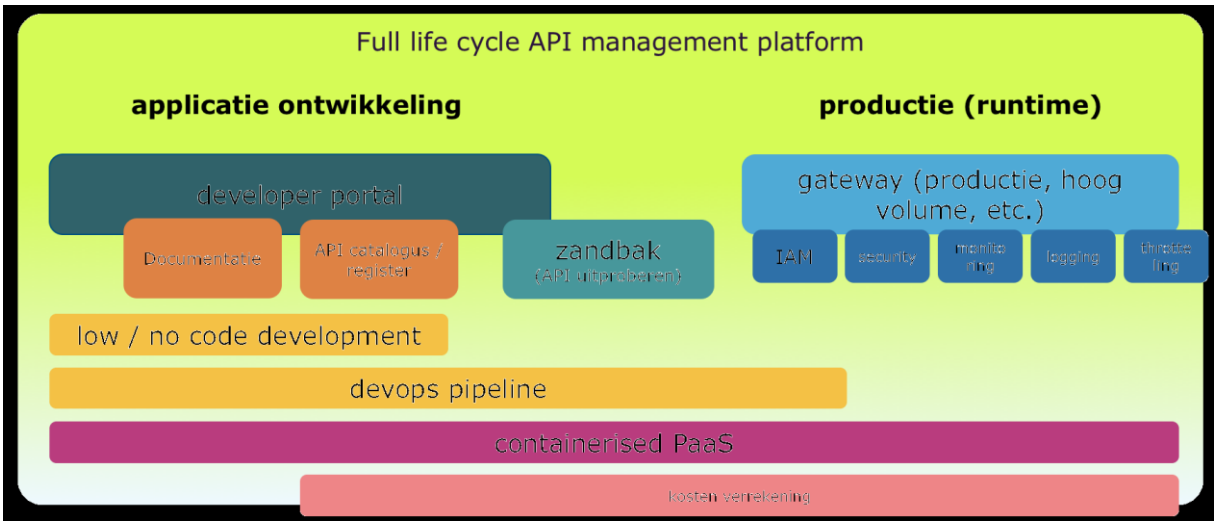
[24] zie https://www.ria.ee/en/state-information-system/x-tee.html en https://www.niis.org/blog/2018/10/3/x-road-rest-support-where-are-we-today

ook naar gekeken is. Indien voor een open source strategie gekozen wordt vanuit het specifieke publieke/overheidsbelang, dan kan samenwerking met X-Road belangrijke voordelen hebben boven het benutten van een open source product uit de reguliere markt.

2.2 Belangrijke aandachtspunten bij selectie API-managementplatform

Belangrijke criteria om de producten te vergelijken:

Benodigde functionaliteit. De producten in het Leaders Quadrant omvatten allemaal een volledige set API management functionaliteiten. Dat is niet in iedere situatie noodzakelijk. Met name het zich onderscheiden in development tools voor zeer wendbare ontwikkeling (low code tooling) is in de context van Logius minder van belang. De nadelen van de lock-in op een specifiek platform zijn waarschijnlijk groter dan de noodzaak om extreem hoge wendbaarheid te bereiken.



Prijsmodel. Meerdere van deze producten zijn (deels) open source. Dat maakt de totale kosten echter niet per se lager. De vereiste technische kennis om deze producten goed te benutten als basis voor hoogwaardige dienstverlening kan schaars zijn en dus kostbaar. De ons bekende voorbeelden van gebruik binnen de Nederlandse overheid hebben allen sterke vendor lock in, hetzij bij een beperkte groep ZZP-ers, hetzij bij een grotere ICT-dienstverlener. Daarnaast wordt de doorontwikkeling medebepaald door de kwaliteit van de community rond het product. Het beheren van een open source API gateway kan gemakkelijk een budget in de orde van 1 mio per jaar vergen[25]. Dit maakt duidelijk dat het gewenst is deze kosten over meerdere overheidsorganisaties te delen.

Ontwikkeling. Alle producten hebben een voor dit moment volledige platformfunctionaliteit. Voor de middellange termijn is echter de doorontwikkeling zeer bepalend. Het is niet gezegd dat alle aspecten van een full life cycle API-managementplatform voor Logius noodzakelijk zijn. Dit betekent ook dat de rijkheid aan functionaliteit van de in de markt meest zichtbare platformen (het leaderskwadrant) niet het enig relevante is. Degelijkheid aan de gebruikskant (runtime) is belangrijker.

Cloudmodel. Alle relevante producten zijn ook in of op Cloudplatformen beschikbaar. Op dit moment is (openbare) Cloud niet het voorkeursmodel van de overheid. Er is een stevige afhankelijkheid tussen de selectie van API-managementplatformen en de selectie van Cloudtechnologie (voor toepassing in een Nederlandse overheidscloud die er linksom of rechtsom in komende jaren zal komen).

[25] Zie bijvoorbeeld http://www.mynewsdesk.com/niis/pressreleases/estonia-and-finland-conduct-a-public-procurement-of-x-road-software-core-development-2443642 waarbij X-Road vermoedelijk minder compleet en dus minder uitgebreid en mogelijk minder kostbaar in beheer dan de door Gartner beoordeelde producten.

Verder is het relevant dat pogingen om een Full Life Cycle API-managementplatform te realiseren op basis van een combinatie van onderdelen van deze producten, al dan niet in combinatie met specifieke andere deelproducten, veelal geen goede resultaten opleveren[26].

Aanbeveling: Bouw zelf kennis op van API-managementplatformen.

2.3 Binnen Nederlandse overheid al gebruikte API-managementplatformen

Binnen de Nederlandse overheid zien we – voor zover we weten – de volgende oplossingen in gebruik:

Oplossing	Partij	
WSO2	Ministerie I&W, DSO	Standaard Platform
CA	Kadaster, UWV	
Centrasite van Software AG	UWV	In combinatie met CA voor catalogus
Mulesoft	Rijkswaterstaat	
Red Hat 3scale	Logius,	Gemeente Amsterdam gebruikt Red Hat Keycloak als IdP voor data.amsterdam.nl
Kong	gemeente Amsterdam	Is beproefd door data.amsterdam.nl. Maar wegens gebrek aan toegevoegde waarde is men momenteel verder gegaan op basis haproxy als gateway.
IBM	Logius (e.a.)	Onvoldoende gegevens ten aanzien van welke onderdelen waar in gebruik zijn. In gebruik bij Logius, maar niet de API-managementfunctionaliteit
Axway	Logius (e.a.)	In gebruik bij Logius MijnOverheid, maar niet de API-managementfunctionaliteit.
Tibco (Mashery)	Logius (e.a.)	Tibco middleware is in gebruik bij Logius, maar niet de API-managementfunctionaliteit.
Microsoft Azure	Mogelijk bij NS	Azure wordt breed gebruikt, maar concrete voorbeelden van inzet van Azure voor API management hebben we niet geconstateerd.
Google (Apigee)		Voor zover bekend niet in gebruik bij de overheid.
Oracle		Wordt breed gebruikt, maar concrete voorbeelden van inzet van Oracle API management voorzieningen hebben we niet geconstateerd.
Nota bene: Ontbrekende voorbeelden in deze lijst graag doorgeven.		

Aanbeveling: Investeer in kennisuitwisseling tussen de verschillende overheidsorganisaties aangaande de ervaringen met deze platformen. Faciliteer uitwisseling op werkvloerniveau waar echte praktijkervaringen uitgewisseld kunnen worden zoals de mate waarin platformen ontwikkelaars echt ondersteunen, standaarden goed implementeren, voorzien zijn van goede mechanismen voor updates en performance gegevens uit de praktijk.

[26] Wat niet wegneemt dat voor PoC's of pilot een veel eenvoudigere oplossing op basis van combinatie van goed gekozen open source producten meer voor de hand ligt. Bijvoorbeeld tyk gateway en keycloak identity manager.

3. Modernisering PTOLU kan adoptie API's versnellen

3.1 API-gewijze dienstverlening en de bestaande PTOLU lijst

De Generieke Digitale Infrastructuur zal gaan vallen onder het regime van de Wet Digitale Overheid. Deze omvat een wettelijk regime voor het Forum Standaardisatie en de pas-toe-of-leg-uit lijst van standaarden. In deze lijst is per standaard een toepassingsgebied en een werkingsgebied gedefinieerd. Dit toepassingsgebied is van belang om te bepalen op welke situatie een standaard van toepassing is. Dit is een belangrijk middel om te voorkomen dat overheidsorganisaties op oneigenlijke gronden te weinig meewerken aan het daadwerkelijk toepassen van standaarden. Het nadeel kan echter zijn dat een eenmaal vastgestelde standaard met een toepassingsgebied innovatie vertraagt. Innovatie bestaat immers vaak uit nieuwe werkwijzen die niet voldoen aan de standaarden van de vorige generatie. Dit geldt ook voor API-gewijze dienstverlening volgens de REST best practice. Gelukkig is op grond van artikel 26 van het wetsvoorstel[27] voorzien in een bepaling om voor dergelijke innovaties andere afspraken te maken.

Voor het vervolg nemen we aan dat dit wetsvoorstel zodanig wordt ingevoerd dat de huidige pas-toe-of-leg-uit lijst en toepassings- en werkingsgebieden als mechanisme blijven bestaan. Uitgaande van de huidige lijst zijn dan de volgende standaarden met hun toepassingsgebied[28] relevant voor API-gewijze dienstverlening.

Verplichte standaarden		
Naam standaard	Toepassingsgebied en werkingsgebied	Relevantie inzake API's
Digikoppeling	Toepassingsgebied: Digikoppeling[29] moet worden toegepast op alle[30] digitale gegevensuitwisseling met behulp van gestructureerde berichten die plaatsvindt met voorzieningen die onderdeel zijn van de GDI, waaronder de basisregistraties, of die sector-overstijgend is. Organisatorisch werkingsgebied: Nederlandse overheden (Rijk, provincies, gemeenten en waterschappen) en instellingen uit de (semi-)publieke sector.	API's leveren veelal een antwoord in de vorm van een gestructureerd bericht, potentieel is er dus overlap met dit toepassingsgebied. De voorzieningen van Logius vallen onder de GDI.
OpenAPI Specification	Toepassingsgebied: OAS moet worden toegepast op het beschrijven/specificeren van een REST API. Organisatorisch werkingsgebied: Identiek aan Digikoppeling.	Geldt voor REST API's.
SAML	SAML moet worden toegepast op de uitwisseling van authenticatie- en autorisatiegegevens om gebruikers na eenmalig inloggen toegang te geven tot meerdere diensten. Organisatorisch werkingsgebied: Identiek aan Digikoppeling.	Een dienst die bestaat uit meerdere API's waarvoor autorisatie vereist is, lijkt hier onder te vallen. SAML is echter een standaard die niet aansluit bij API's.
STUF	Uitwisseling en bevraging van basisgegevens die behoren tot een aantal wettelijk vastgestelde basisregistraties, zoals Personen (GBA), Adressen (BRA), Gebouwen	Bevat voorschriften ten aanzien van onderliggende koppelvlakken gebaseerd op Digikop-

[27] Momenteel in behandeling in de Tweede Kamer: www.tweedekamer.nl/kamerstukken/wetsvoorstellen/detail?cfg=wetsvoorsteldetails&qry=wetsvoorstel%3A34972
[28] Uitgaande van de lijst d.d. 28 september 2018 op https://www.forumstandaardisatie.nl/open-standaarden/lijst/verplicht
[29] https://www.forumstandaardisatie.nl/standaard/digikoppeling
[30] Geautomatiseerde gegevensuitwisseling tussen informatiesystemen op basis van NEN3610, dat wil zeggen geoinformatie, is uitgesloten van het functioneel toepassingsgebied.

	(BGA), Kadaster (BRK), Nieuw Handelsregister (NHR) en Waarde Onroerende Zaken (WOZ); Uitwisseling en bevraging van zaakgegevens die behoren tot de producten- en dienstenportfolio van gemeenten; Uitwisseling van domein- of sectorspecifieke gegevens waarin ook basis- en/of zaakgegevens voorkomen en waarvoor geen andere (inter)nationale (XML-gebaseerde) berichtenstandaard is vastgesteld.	peling hetgeen impliciet inhoudt dat STUF niet direct compatibel is met RESTful API's[31] Dit neemt niet weg dat er wel stappen gezet zijn om REST voor specifieke services te benutten.
Overige	Zie Discussiedocument RESTful APIs Versie 1.1 – juni 2016 blz. 26 voor een overzicht van verdere relaties.	Dit overzicht geeft aan dat in enkele gevallen er (formele) belemmeringen zijn om betreffende standaarden in combinatie met REST API's te gebruiken.

Aanbevolen standaarden		
Naam standaard	Toepassingsgebied	Relevantie inzake API's
JSON	Toepassingsgebied: Objectnotatie voor het uitwisselen van datastructuren. Bijvoorbeeld in webapplicaties die asynchroon gegevens ophalen van de webserver.	Meest gebruikte en eenvoudigste uitwisselingsformaat bij REST API's
OData	Toepassingsgebied: OData kan worden toegepast voor het bouwen en gebruiken van REST APIs met als doel het gestructureerd ontsluiten van (statistische) open datasets.	OData maakt het makkelijker om interoperabele REST-implementaties te bouwen, waardoor data die via API's wordt uitgewisseld op een uniforme wijze weergegeven kan worden.
SOAP	Toepassingsgebied: Uitwisselen van gegevens als XML bericht	Het aanbevelen van REST API's impliceert een alternatief voor SOAP bevragingen.

De huidige stand van zaken geeft de volgende duidelijke aandachtspunten indien API-gewijze dienstverlening gewenst is:

- De huidige versie van Digikoppeling kan gezien worden als een "ontmoediging" op gebruik van REST API's voor gegevensuitwisseling binnen de overheid van- en naar GDI voorzieningen en voor sector-overstijgende gegevensuitwisseling (zie bovenstaande tekst van het toepassingsgebied). REST API voldoen immers niet aan de huidige Digikoppeling standaarden. Toepassing van REST API's voor de communicatie van gebruikerstoepassingen of apps voor burgers en bedrijven naar overheidsvoorzieningen valt buiten het werkingsgebied. Daar geldt de beperking dus niet. Bijvoorbeeld binnen het Digitaal Stelsel Omgevingswet zie je dit in praktijk gebracht: REST API's voor de gebruikerstoepassingen en Digikoppeling voor de communicatie tussen overheden.

- SAML en meer in het algemeen authenticatie en autorisatie, vormen eigenlijk een groter issue. SAML is de standaard waarmee het huidige DigiD en eHerkenning functioneren. Voor authenticatie, autorisatie, vormen van machtiging en het doorgeven van authenticaties wordt in API uitwisseling meestal Oauth gebruikt (en de uitbreiding daarop Open ID Connect). Dit issue is breder dan de pas-toe-of-leg-uit lijst. Artikel 7 van het

[31] Binnen de STUF community wordt hier wel over gesproken, bijvoorbeeld https://discussie.kinggemeenten.nl/discussie/gemma/stuf-301/geschikt-maken-van-stuf-voor-rest-architecturen

wetsvoorstel Digitale Overheid verbiedt het namelijk aan overheden om te werken met niet toegelaten identificatiemiddelen. Doel van dit artikel is het gebruik van DigiD, eHerkenning en andere middelen onder het identificatiestelsel van de GDI vallend te verplichten. Deze middelen zijn nog niet goed compatibel met Oauth en met de vormen van inloggen die buiten de overheid in de API-economie gebruikelijk zijn.

Beide issues en het nut van standaardisatie omtrent REST API's zijn door het Forum Standaardisatie[32] onderkend, gezien het eerdere advies: "Monitor de ontwikkelingen rondom de standaarden met betrekking tot het gebruik van API's (waaronder RESTful API's) om vroegtijdig de behoefte om dergelijke standaarden op te nemen op de lijst voor open standaarden te identificeren." Dit monitoren vindt onder ander plaats in het kennisplatform API's waarin Logius participeert. Het feit dat de OpenAPI Specification al op de PTOLU-lijst staat kan worden gezien als eerste stap in die richting. Als vervolgstappen ligt het voor de hand om de standaardisatie van API-gewijze gegevensuitwisseling op basis van REST voort te zetten. Doelstelling daarbij is voldoende interoperabiliteit, gezien de aard van REST kan dit betekenen dat er minder vastgelegd wordt dan bij Digikoppeling.

Wat betreft het tweede issue geldt dat de procedure voor opname van OAuth 2.0 op de 'pas toe of leg uit'-lijst loopt. In diverse projecten binnen de overheid wordt inmiddels met Oauth (en de uitbreiding daarop in de vorm van Open ID connect) gewerkt, zoals Digitaal Stelsel Omgevingswet. Het is gewenst deze stappen te vervolgen, maar wel zodanig dat authenticatie en autorisatie in het kader van RESTful API's ook onder artikel 7 van het wetsvoorstel Digitale Overheid kan vallen. Het is gewenst daarbij ook JSON, OData en SOAP, die nu als aanbevolen standaard zijn genoemd, consistent te maken met de verplichte lijst.

De andere standaarden betreffen ofwel de semantiek of beschrijving van gegevens, ofwel zaken die grotendeels compatibel zijn met API-gewijze dienstverlening, ofwel zaken die volledig buiten scope zijn. Volledig compatibel zijn: HTTPS, HSTS, IPv4, IPv6, DNSSEC en TLS.

3.2 API's standaardisatie opnemen in Digikoppeling geeft duidelijk signaal

Dat standaardisatie in het kader van API-dienstverlening gewenst is, is duidelijk. Daarmee is de vraag hoe zich dit tot Digikoppeling verhoudt echter nog niet beantwoord. Aangezien het toepassingsgebied gedeeltelijk overlapt met Digikoppeling zijn er twee mogelijkheden: API-gewijze dienstverlening wordt binnen Digikoppeling gestandaardiseerd of daarnaast.

Functioneel richt API-gewijze dienstverlening zich op uitwisselen van kleinere, meer op één enkele deeltaak of processtap gerichte, brokjes informatie. Dit betekent dat de behoefte aan de bestaande Digikoppeling standaarden voor berichtenuitwisseling blijft bestaan. Het betreft verschillende interactievormen die naast elkaar bestaan. Er kan dus voor gekozen worden Digikoppeling met het huidige toepassings- en werkingsgebied te handhaven voor het berichtenverkeer binnen de overheid en daarnaast een RESTful API standaard te plaatsen met als toepassings- en werkingsgebied de interactie met apps van eindgebruikers (burgers en bedrijven).

In de API-economie, zie hoofdstuk 1, zie je echter dat snelle digitalisering veelal gepaard gaat met de bewuste keuze om uitwisseling met externe eindgebruikers volgens dezelfde technologie te realiseren als interne uitwisseling. Dat is "het mes dat aan twee kanten snijdt". Dit biedt een krachtig medicijn tegen silo-vorming. In de taal van ontwikkelaar wordt dat "eat your own dog food" genoemd[33]. Deze verbinding tussen intern en extern

[32] Zie https://www.forumstandaardisatie.nl/thema/application-programming-interfaces-api en https://www.forumstandaardisatie.nl/standaard/oauth.
[33] https://en.wikipedia.org/wiki/Eating_your_own_dog_food

benutten van API's komt veel beter tot haar recht wanneer dit geheel binnen één consistente Digikoppeling standaard valt.

Relevant daarbij is dat de nieuwe technologische mogelijkheden (zie kader aan eind van hoofdstuk 1) tot verschuiving van de architectuur kunnen leiden: van uitgebreide webapplicaties – veelal met nog restanten van vroegere papieren formulieren – met daarachter ketenprocessen naar apps voor een specifieke taak – meer levensgebeurtenis georiënteerd (zie Bijlage D). Deze volgende stap in de digitalisering kan gepaard gaan met procesvereenvoudiging. Als de technologische mogelijkheden goed benut worden, kan dit leiden tot minder complexe ketenprocessen en dus uiteindelijk tot afname van de behoefte aan de huidige berichtuitwisselingsstandaarden. Tegelijk leidt het tot een explosie aan apps en API's "aan de voorkant" en dus tot een noodzaak wildgroei op dat gebied te voorkomen en benodigde standaardisatie af te spreken. De standaardisatievraagstukken rond REST en Digikoppeling zijn dus niet opgelost wanneer REST enkel voor interactie met apps van burgers en bedrijven wordt toegevoegd. Het feit dat Digikoppeling, openAPI en SAML momenteel een identiek organisatorisch werkgebied hebben in de PTOLU-lijst laat ook zien dat het gewenst is hier duidelijkere afspraken over te maken.

De keuze om de standaarden voor REST API's op te nemen in de Digikoppelingfamilie zou dus een duidelijk signaal geven dat deze samenhang tussen het toepassingsgebied bevraging voor burgers en bevragingen binnen de overheid gewenst is.

Aanbeveling: Kies ervoor om REST API's als variant toe te voegen aan de Digikoppeling standaard. Het toevoegen biedt de beste mogelijkheden om aan te geven hoe de verschillende varianten zich tot elkaar verhouden en samenwerken en is de kortste route naar opname van API-gewijze dienstverlening als optie binnen de PTOLU standaarden.

Aanbeveling: Omarm het "eating your own dog food" adagium. Neem dit voor een aantal voorzieningen op als eis voor verdere doorontwikkeling. MijnOverheid en DigiD liggen daarbij het meest voor de hand. Kies vanwege de samenhang tussen interne en externe koppelingen voor opname van een derde variant voor RESTful API's binnen de bestaande Digikoppeling standaard. Daarmee blijft de naam "Digikoppeling" als dè standaard voor gegevensuitwisseling tussen overheden in stand.

3.3 Samenhang tussen meerdere standaarden vastleggen in Digikoppeling

De huidige Digikoppeling standaardisatie omvat een hiërarchie aan onderwerpen: architectuur, identificatie & authenticatie, verdere voorschriften en koppelvlakbeschrijving. Op al deze vlakken zijn aanpassingen nodig om interoperabiliteit rond REST API's te bevorderen[34]. Daarbij zijn de volgende internationale standaarden van belang:

- OpenAPI Specification (al opgenomen op de PTOLU lijst).
- Nederlands profiel Oauth
- Open ID connect (aangemelding bij Forum wordt overwogen, een Nederlands profiel kan volgens dezelfde werkwijze als kennisplatform toepast voor Oauth, beschreven worden).
- JSON
- OData

[34] De Digikoppeling architectuur (versie 1.5.1) hanteert NORA als uitgangspunt. Een echt consistente doorontwikkeling van API-gewijze dienstverlening zou ook tot uiting moeten komen in doorontwikkeling van de NORA. Uit interviews komt naar voren dat de huidige NORA werkwijze is dat dergelijke ontwikkelingen eerst uitgewerkt worden in NORA dochters en pas na een relatief lang proces mogelijk in de NORA zelf verwerkt worden. Dit betekent in toenemende mate dat de verwachting van bestuurders (en andere niet-architecten) dat "voldoen aan de NORA in huidige vorm" relevant is voor interoperabiliteit en effectiviteit van ICT voorzieningen van de overheid, achterhaald is. Het NORA team is zich daar overigens van bewust, zie https://www.noraonline.nl/wiki/API. Vanuit Agile werkwijzen is het belang van een uitgeschrveen NORA overigens sowieso beperkt. Zie https://www.scaledagileframework.com/agile-architecture/

Verder biedt de API-strategie van het Digitaal Stelsel Omgevingswet een goede basis voor de onderwerpen waarover afspraken gemaakt moeten worden. Dit is ook de aanpak die inmiddels in het kennisplatform API's is gevolgd[35]. De aanduiding "strategie" moet niet worden begrepen als strategisch in bestuurlijke zin. Het document biedt standaardisatie op een niveau vergelijkbaar met Digikoppeling.

Een ander aandachtspunt zijn standaardisatie-afspraken van de Uniform Resource Identifiers (URI's)[36] die veelal worden gebruikt om een API aan te roepen. Afhankelijk van de gekozen standaardisatie kunnen er (kleine) conflicten met bestaande standaarden optreden indien daarin strijdige eisen aan URI's zijn opgenomen. De DSO API-strategie biedt ook op dit punt voldoende kader. In de verdere standaardisatie is dit wel een belangrijk aandachtspunt en is het van belang dat er draagvlak is voor een eenduidige toepassing van URI's.

Aanbeveling: De zogenoemde API-strategie van DSO biedt een goed uitgangspunt voor de benodigde afspraken voor een REST API variant binnen Digikoppeling. DSO heeft geen belang bij het beheer hiervan op langere termijn. Deze werkwijze wordt nu al in het kennisplatform API's gevolgd. De meest pragmatische werkwijze is om een en ander eerst losstaand te realiseren en in de bestaande Digikoppeling standaard te verwijzen naar deze ontwikkeling. Daarna is het gewenst o.a. het Digikoppeling architectuurdocument aan de nieuwe ontwikkelingen aan te passen zodat een samenhangend geheel ontstaat.

Aanbeveling: Afronding van het Nederlandse Oauth profiel betekent dat de belangrijkste belemmering voor standaardisatie wordt weggenomen. Dit profiel wordt momenteel beschreven in kennisplatform[37]. Dit biedt een goed momentum om de uitbreiding van Digikoppeling aan te kondigen. Dit werk wordt voor eind 2018 door het kennisplatform API's afgerond en is gebaseerd op het voor Nederland specifiek maken van een internationaal profiel[38].

Aanbeveling: In vervolg op Oauth dient ook een profiel voor Open ID Connect te worden vastgesteld. Net als voor Oauth kan hiervoor aangesloten worden op een internationale profiel[39]. Koppel dit aan een roadmap voor het uitbreiden van DigiD met app-to-app en API gerichte authenticatie en autorisatie.

Aandachtspunt: Dit zou betekenen dat Logius tenminste een deel van de doelstellingen van het kennisplatform API's tot haar taak gaat rekenen. Dit dient uiteraard in goed overleg en vanuit een faciliterende inzet naar de belangen van het hele ecosysteem te worden ingericht.

Het overkoepelende toepassings- en werkingsgebied blijft ongewijzigd. Wel is het nodig nader uit te werken welke van de vier varianten voor welk interactiepatroon bedoeld zijn. Voor bevragingen vanuit apps en gebruikerstoepassingen van burgers en bedrijven is het gewenst toe te werken naar voorschrijven van REST API's. Dit zal op den duur bepaalde van de huidige WUS bevragingen vervangen.

Qua verwachtingen en communicatie is het belangrijk te benadrukken dat dit enerzijds een volwaardige variant is, maar anderzijds de behoefte aan de bestaande varianten WUS en ebMS niet wegneemt. Er is geen reden bestaande koppelingen uit te faseren of om te bouwen. Voor transacties en meldingen die reliable messaging vereisen (ebMS) vormt REST, zeker voorlopig, geen alternatief.

[35] Huidige werkversie van het kennisplatform: https://geonovum.github.io/KP-APIs/
[36] De algemenere vorm van een URL, bijvoorbeeld https://nl.wikipedia.org/wiki/Uniform_resource_identifier
[37] De huidige werkversie staat op https://geonovum.github.io/KP-APIs-OAuthNL/
[38] https://openid.net/specs/openid-igov-oauth2-1_0.html
[39] https://openid.net/specs/openid-igov-openid-connect-1_0.html

Aanbeveling: Ga in overleg met het programma Digitaal Stelsel Omgevingswet over de DSO API-strategie. Stel aan DSO voor deze als basis te nemen voor de bovengenoemde derde Digikoppeling variant. Aangezien standaardisatie buiten haar eigen domein niet de taak is van dit programma, kan dit goed als een win-win worden gerealiseerd.

Aanbeveling: Veranker de uitgangspunten van Common Ground[40] expliciet in de architectuur en werk vanuit draagvlak en gezamenlijkheid aan een standaardisatie die ook op de ontwikkelingen bij gemeenten en andere bestuurslagen aansluit. Maak daarbij tijdig afspraken over de overgang van de huidige op innovatie gerichte ontwikkeling van o.a. NLX en het belang tijdig voor te sorteren op een stabiele beheer, opschaal en doorontwikkelfase. Dat laatste is een mogelijke rol voor Logius.

3.4 API-gewijze dienstverlening en semantische interoperabiliteit

Het feit dat bij API-gewijze dienstverlening sprake is van uitwisseling van kleinere brokjes informatie roept de vraag op hoe zich dit verhoudt tot de meer semantische PTOLU standaarden[41]. Deze beschrijven en standaardiseren omvangrijke berichten die veelal een geheel probleemdomein betreffen. Wanneer in een dergelijk domein API-gewijze dienstverlening wordt ingevoerd dan zal dit omvangrijke bericht vermoedelijk worden opgeknipt in meerdere eenvoudigere berichten. Afhankelijk van hoe deze standaarden precies zijn beschreven, kan dit – op z'n minst in formele zin – betekenen dat niet meer aan betreffende PTOLU standaard wordt voldaan (nalopen van al deze standaard op dit aspect valt buiten de scope van deze verkenning). Mogelijk is er echter een eenvoudige oplossing. Het doel van deze standaarden is interoperabiliteit op semantisch niveau. Dat wil zeggen dat de overeenkomende betekenis van de uitgewisselde begrippen aan beide kanten van de lijn moet worden gewaarborgd. Dat staat meestal los van de vraag of informatie in één keer in een omvangrijk totaalbericht wordt uitgewisseld of in kleinere brokjes. Het zal dus veelal mogelijk zijn de API-gewijze uitwisseling in een dergelijk domein zo in te richten dat op semantisch niveau aan de huidige standaarden voldaan wordt. Op den duur kunnen alle betreffende standaarden worden nagelopen op de vraag of dit altijd opgaat. Voor de korte termijn is het echter interessanter om te onderzoeken of een aantal algemene regels hoe bij API-gewijze dienstverlening aan de semantische standaarden kan worden voldaan, volstaat. De toepassing van API-gewijze dienstverlening hoeft dan niet te wachten tot dergelijke standaarden zijn aangepast[42].

Aanbeveling: Doe nader onderzoek naar effect van API-gewijze interactie op semantische standaarden. Indien dit het vermoeden bevestigt dat de semantische operabiliteit goed kan worden behouden, kan een overgangsregeling worden opgesteld in de vorm van een best practice. Daarmee zouden formele belemmeringen voor API-gewijze dienstverlening in betreffende toepassingsgebieden worden voorkomen. Dit onderzoek is eerder een taak van Forum Standaardisatie dan van andere Logius onderdelen. Deze aanbeveling is in feite gelijk aan de aanbeveling[43] "kwaliteitstoets" die in 2016 al aan Forum Standaardisatie gedaan is. Daarvan is toen besloten dat dit een taak is van de betreffende beheerorganisaties. Het is zinvol te bekijken in hoeverre betreffende beheerorganisaties dit inmiddels hebben opgepakt.

3.5 Juridische interoperabiliteit

Uit enkele van de interviews zijn relevante juridische vragen naar boven gekomen. In het bijzonder de vraag in hoeverre API-gewijze dienstverlening op gespannen voet staat met

[40] https://github.com/VNG-Realisatie/common-ground/blob/master/cg-vision.md
[41] Zie ook https://github.com/VNG-Realisatie/common-ground/blob/master/cg-vision.md#moving-from-complex-all-encompassing-services-to-fit-for-purpose-services
[42] Voor de Geo-standaarden is deze redenering al gevolgd, zie https://docs.geostandaarden.nl/wp/basis-wpgs/#x3-3-1-opkomst-rest-convenience-api-s evenals binnen de STUF community https://discussie.kinggemeenten.nl/discussie/gemma/stuf-301/rfc-compacte-eenvoudige-vrije-berichten-tbv-rest.
[43] Blz 3. Discussiedocument RESTful APIs Versie 1.1 – juni 2016, Forum Standaardisatie

doelbinding. Doelbinding is een uitgangspunt in de huidige privacywetgeving. Voor toepassingen binnen de overheid geldt dat deze gebaseerd zijn op een wettelijke taak en vanuit deze taak kan het doel van een handeling in principe worden afgeleid. Het doel moet vooraf duidelijk zijn. De wendbare inrichting van API-gewijze dienstverlening kan er toe leiden dat het doel van de gebruiker onzichtbaar is voor de betrokkenen. Het doorgeven van een "doelbindingsverklaring (declaration of purpose)" als voorgesteld door Common Ground[44] biedt een goede oplossing voor het behoud van doelbinding in een zeer wendbaar API-ecosysteem. Zonder dergelijke oplossingen kan "het recht op inzicht wie, op welk moment en voor welk doel, gegevens inziet, gebruikt of aan anderen geeft" zoals dit door het kabinet aan burgers beloofd wordt (in o.a. NL DIGIbeter), niet worden gerealiseerd[45].

Op het moment dat API's daadwerkelijk worden opengesteld voor gebruik door de buitenwereld, is het de vraag of deze vorm van doelbinding nog de gewenste bescherming van de privacy vormt. Belangrijk daarbij is dat op basis van API's zeer eenvoudige vragen als "is achttien" en "heeft momenteel inkomen onder x" kunnen worden geïmplementeerd. Essentieel bij dergelijke API's is dat de betrokkene, de burger zelf, expliciet toestemming geeft. Veelal zal het de burger zelf zijn die het antwoord dat de API geeft via zijn mobiel aan een derde toont of aan een andere webtoepassing doorgeeft. Op het moment dat de betrokkene dit zelf doet, is het juist ongewenst om een "doelbindingsverklaring" vast te leggen. Het is niet nodig en zelfs slecht voor de privacy indien bijvoorbeeld bij de BRP een overzicht ontstaat van alle burgers die de isachttien-API gebruiken om alcohol te kopen. Formeel zijn er nu belemmeringen om dergelijke algemene vragen op onder doelbinding vallende overheidsvoorziening te baseren. Dit vereist nader onderzoek en het inzicht dat nieuwe technologische mogelijkheden om nieuwe juridische vormen vragen, die de onderliggende grondrechten en publieke waarden in de nieuwe context verankeren.

Aanbeveling: Vraag aan het ministerie van BZK om in kaart te brengen wat er al over dit vraagstuk uitgewerkt is. Vermoedelijk zijn deze vraagstukken op basis van goede doordenking van de AVG oplosbaar. Dit neemt niet weg dat het goede uitleg vraagt. Toegankelijke best practices op dit gebied kunnen zorgen dat onnodige barrières voor innovatie verdwijnen en grootschalige misstappen tijdig worden gesignaleerd. Een bepaalde mate van acceptatie van onevenwichtigheid in de experimenteerfase hoort daarbij. Juist omdat dit een aandachtspunt in het verwachtingenmanagement is het opstellen van best practices en actief uitdragen daarvan, vanuit beleidsperspectief van belang.

Bij het verder standaardiseren van API's is het van belang om uit te gaan van het doel van deze standaardisaties: interoperabiliteit binnen de publieke sector. Belangrijk is dat REST geen standaard is maar een architectuur. Dat is een patroon met best practices, maar ook met de flexibiliteit om in een bepaalde situatie de beste toepassing voor een eindgebruiker te realiseren.

[44] Zie https://github.com/VNG-Realisatie/common-ground/blob/master/cg-vision.md#moving-from-manual-to-automated-accountability

[45] Voor de duidelijkheid: Dit recht bestaat al, o.a. in BRP-wetgeving en in AVG. Voor de BRP echter is het op dit moment niet mogelijk dit recht volledig uit te voeren, tenzij handmatig. Momenteel kan wel getoond worden welke organisaties gegevens afnemen, maar doordat deze vervolgens kopieën maken kan niet op globaal niveau transparant gemaakt worden wat er daarna met de gegevens gedaan wordt.

4. Wie zijn bij API-gewijze dienstverlening betrokken?

Belanghebbende	Belang t.a.v. API management Logius	Eigen rol	Gewenste rol Logius / perceptie
BZK	Uitvoering en concretisering NL DIGIbeter.	DIO: Opdrachtgever Logius, SG: eigenaar Logius	Beheer en doorontwikkeling GDI.
VNG Realisatie / Common Ground	Nader bekijken in hoeverre NL X een federatief stelsel van API-gateways veroorzaakt. Zie https://docs.nlx.io/	API-gebaseerde visie op digitalisering voor gemeenten realiseren.	
BKWI	Doorontwikkeling reliable messaging standaarden. Aandacht voor stabiele beheerfase (na innovatiefase)	Betrouwbaar berichtenverkeer in Sociaal Domein. Geen eigen API-strategie voorzien.	Doorontwikkelen Digikoppeling standaarden.
Geonovum	Standaardisatie buiten het geo-domein	Medeverantwoordelijk kennisplatform.	Beheer standaarden. Geen overheidscatalogus API's. Doorontwikkeling DigiD t.b.v. apps en API's kan alleen landelijk.
Kadaster	Standaardisatie buiten het geo-domein	Heeft met RvIG en KvK voorstel voor API doorontwikkeling bij BZK akkoord gekregen.	
DSO Programma ADSMO	Standaardisatie buiten het geo-domein	Realiseren DSO (mede op basis van API-strategie)	Beheer API-strategie (het niet DSO specifieke deel van API-standaarden)
UWV	Standaardisatie	Vernieuwing ICT landschap en vergroten wendbaarheid.	
Belastingdienst			
RIVG		Heeft met Kadaster en KvK voorstel voor API doorontwikkeling bij BZK akkoord gekregen.	
KvK		Heeft met RvIG en Kadaster voorstel voor API doorontwikkeling bij BZK akkoord gekregen.	
CBS	Standaardisatie. GDI met goede financiering.		
DUO	Standaardisatie. GDI met goede financiering.		

De voornaamste belanghebbenden bij API-gewijze dienstverlening zijn uiteindelijk burgers en bedrijven. Tenminste als we uitgaan van digitalisering in het maatschappelijk belang.

Daarnaast zijn de medewerkers van overheidsorganisaties, ambtenaren, belanghebbende. API-gewijze dienstverlening kan een impuls geven aan betere ondersteuning voor hun werkzaamheden en taakuitvoering; en dat heeft als het goed is weer een positief effect op burgers en bedrijven.

Dit belang is aanleiding voor individuele overheidsorganisaties (zie boven) om API's aan te bieden. Dat is echter onvoldoende voor een bredere versnelling van de digitalisering op basis van API-gewijze dienstverlening omdat belangrijke faciliteiten die de grenzen van één specifieke taak of doelgroep overschrijden niet tot stand komen. Vanuit haar rol, samen met EZK en J&V, in de gezamenlijke kabinetsbrede aanpak heeft het ministerie van BZK een belang om daarin beleidskeuzes te maken. Doel van deze beleidskeuzes moet zijn om de gezamenlijke faciliteiten voor API-gewijze dienstverlening tot stand te laten komen. Het is belangrijk dat dit blijft aansluiten op initiatieven "van onderaf" zoals Common Ground en het kennisplatform API's, maar dat neemt niet weg dat duidelijke keuzes de gehele beweging kunnen versnellen. Dat is een keuze voor doorontwikkeling van Generieke Digitale Infrastructuur. Voor Logius zijn dergelijke keuzes uiteraard van groot strategisch belang omdat het keuzes betreffende de taken en toekomst van Logius zijn.

Standaardisatie kan ook gezien worden als centrale faciliteit en daarin heeft Logius, met het Forum Standaardisatie, een duidelijke taak.

Uit de interviews komen twee specifieke belangen naar voren die aandacht behoeven: bekostiging en het belang van bestaande beheerpartijen.

Bekostigingsbelang. Bij private partijen zijn er twee sterke prikkels voor API-gewijze dienstverlening: API's worden met een per-tik-betaal (pay per use) mechanisme aangeboden en dit opent een nieuw verdienmodel naast het bestaande of API's dragen bij aan grotere verkoop tegen lagere kosten in een bestaand product. Bij de overheid ontbreekt deze prikkel. Bekostiging van generieke faciliteiten volgens het profijtbeginsel, zoals dat nu voor bestaande GDI wordt ingevoerd, kan niet zonder een centraal beleid dat de bekostiging uit algemene middelen van eigen alternatieven stap voor stap beëindigt. Een afdeling van een overheidsorganisatie die een taak met bestaande financiering en bestaande middelen uitvoert zal altijd weerstand bieden tegen een uitvoering die anders is en verschuiving van bekostigingsgeldstroom veroorzaakt. Dit maakt dat het gewenst is – wetende dat het een complexe ambtelijke opgave is – om middellange termijn afspraken over bekostiging van GDI voor API-gewijze dienstverlening onderdeel te maken van het beleid. Het is mogelijk en aanlokkelijk om eerst op basis van enkel programmafinanciering de relatief makkelijke korte termijnresultaten met API's te realiseren. Voor Logius biedt dat echter niet het gewenste perspectief om met volle kracht op de benodigde vernieuwde GDI te te zetten.

Daarnaast is het gewenst API-gewijze dienstverlening van het eerste moment af zo op te zetten dat de werkelijke kosten voortdurend gemeten worden en dat het ontwerp gericht is op de meeste economische vormgeving met minimale overhead.

Aanbeveling: Zet voor doorontwikkeling van GDI naar API-gewijze dienstverlening in op het van start af opzetten van structurele financiering en bouw benodigde voorzieningen voor kostenverrekening in.

Beheerbelang. Beheerders van bestaande voorzieningen hebben niet altijd belang bij nieuwe standaardisatie en technologische vernieuwing. Dit geldt voor Logius zelf, maar ook bijvoorbeeld voor BKWI. Consistentie en een duidelijke middellange termijn agenda rond standaardisatie zijn daarbij van groot belang. De wereld van beheer is veelal niet ingericht op snelle innovatie en op het uitproberen van nieuwe mogelijkheden. Een fase van pilots is daarom belangrijk (en in feite al gaande). Minstens even belangrijk is het om al tijdens deze fase met de beheerpartijen na te denken over opschaling en beheer in de

toekomst, indien nieuwe standaarden en technologieën op grote schaal benut gaan worden.

Relevant hierbij is dat ook private partijen, als leverancier van deze beheerders, grote belangen kunnen hebben die modernisering kunnen remmen. Een consistente meerjarige agenda die voldoende concreet is en stapsgewijs daadwerkelijk gerealiseerd wordt, leidt ook naar die partijen toe tot de benodigde prikkels.

Aanbeveling: Stel een middellange termijnagenda op voor doorontwikkeling van Digikoppeling en communiceer deze parallel aan initiatieven rond API-gewijze dienstverlening.

Het laatste belang waarbij in een beleid voor API-gewijze dienstverlening rekening gehouden moet worden is een goede afbakening van generiek versus specifiek. De rol van Logius en van de GDI betreft algemene voorzieningen en generieke standaarden. Een organisatie als Geonovum kan daar goed mee samenwerken, zolang deze generieke ontwikkeling haar rol in de specifieke standaarden (in dit voorbeeld voor het geo-domein) versterken en niet hinderen. Hetzelfde geldt voor andere domeinen. Genericiteit van GDI en generieke standaarden dient getoetst te kunnen worden door organisaties die een rol hebben in de belangrijkste specifieke domeinen.

Een belangrijk argument voor een stevige rol van Logius is dat juist de eenvoud van API's voor afnemers goede generieke voorzieningen vereisen. Zonder investering in de GDI is het risico groot dat de digitalisering geremd wordt. Er kunnen dan slechts in beperkte mate apps gerealiseerd worden, het realiseren van apps die een echt verschil maken omdat ze toegang hebben tot dieperliggende functies zal een probleem blijven vormen.

Aanbeveling: Laat andere partijen expliciet toetsen dat Logius zich tot generieke zaken en standaardisatie beperkt.

5. Implicaties voor de dienstverlening van Logius

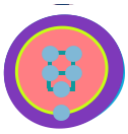
5.1 Implicaties hangen af van scenario bredere digitaliseringsstrategie

Deze verkenning richt zich op de impact die API-gewijze dienstverlening kan hebben op de stelselvoorzieningen en mutatis mutandis evenzeer op de andere GDI. Het gaat daarbij met name om het in beeld brengen van de gevolgen voor de taken en dienstverlening van Logius. De impact voor Logius wordt door de eigen keuzes van Logius bepaald, maar evenzeer door de bredere beleidskeuzes en de snelheid van de ontwikkelingen in de digitalisering van de publieke sector als geheel.

De impact voor Logius wordt besproken aan de hand van vier scenario's. Het eerste scenario betreft de snelste en maximaal diepgaande digitale transformatie van het voor Logius relevante deel van de publieke sector. De scenario's zijn opgesteld in afnemende intensiteit van digitalisering. Datgene wat geldt voor de minder digitaliserende scenario's, geldt altijd ook in de meer radicale scenario's. Alles wat geldt voor scenario 4, zal dus ook gelden voor 3 t/m 1, en zo voorts.

1. Radicaal Digitaal De publieke sector gaat voor radicale digitalisering. Logius speelt hierin een belangrijke rol (voor het deel van de overheid dat zij bedient). Bv: alle basisregistraties en sectorale registraties via web-API's beschikbaar en nieuwe services als www.overheid.nl/isachtien-api realiseren[46].

Logius faciliteert het API-management voor meerdere van haar afnemers en speelt een hoofdrol in de standaardisatie voor API's binnen de Nederlandse overheid. Dit is een soort variant op route die banken in kader van PSD2 afleggen.



2. Digitaal Logius Logius zelf wordt een digital enterprise en heeft daarin hogere snelheid dan de overheden waarvoor het werkt. Logius heeft een eigen API-managementplatform en speelt een actieve rol in overheidsbrede standaarden voor API's. Bv: OIN-register API.

3. Ecosysteem Logius functioneert in een ecosysteem waarin haar partners en afnemers sterk digitaliseren. De strategie is gericht op het benutten van API's van anderen en niet op het zelf leveren. Waar Logius eigen API's aan buitenwereld levert, worden API-managementvoorzieningen en standaarden van anderen benut.

Logius zelf verandert minder sterk en ook een deel van de afnemers blijft op huidige manier functioneren. Andere partijen nemen mogelijk een sterkere positie in voor wat betreft API-gewijze dienstverlening binnen de overheid.

4. Tech Only Logius behoudt de huidige rol. "Elektronische overheid" gaat door digitalisering op in bredere "digitale samenleving"; De rol van Logius wordt mogelijk kleiner omdat andere vormen van digitalisering deze rol op den duur overnemen. De strategie is dus enkel gericht op beheer en kosteneffectiviteit. API management is relevant waar het daar in de technische architectuur aan bijdraagt.

In het Radicaal Digitaal scenario heeft Logius een specifieke rol als beheerder van voor API-management doorontwikkelde GDI. In de andere scenario's kan voor Logius ook een andere uitvoeringsorganisatie worden ingevuld.

[46] De in bijlage opgenomen lijst van mogelijke API-gewijze dienstverleningsinitiatieven is ingedeeld naar deze scenario's. Deze voorbeelden worden daar nader toegelicht.

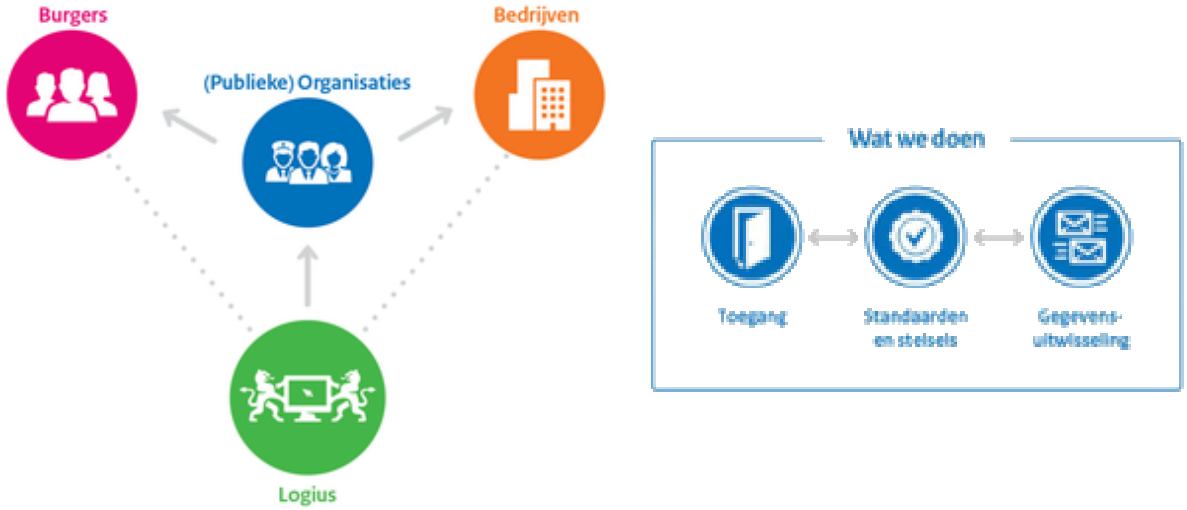
5.2 Implicaties in context van 'Doorontwikkeling Logius'

Logius werkt – met oog voor burgers en bedrijven – aan digitale dienstverlening door publieke organisaties[47]. In het kader van deze dienstverlening heeft Logius ook zelf direct contact met burgers en bedrijven, veelal zonder dat ze daarbij op de voorgrond treedt.

Momenteel is de uitvoering van Logius als volgt verdeeld[48]:

- Toegangsdiensten
- Keteninformatiediensten (inclusief stelseldiensten)
- Portaaldiensten (in onderstaande "wat we doen" onderdeel van Toegang).

Daarnaast is er een afdeling Infrastructuur & Servicecentrum (I&S) en zijn er stafdiensten. De standaardisatietaak is voor wat betreft het vaststellingsproces in de afzonderlijke unit Bureau en College Forum Standaardisatie ondergebracht en voor de uitvoering van door Logius beheerder standaarden bij het Centrum voor Standaarden (onderdeel van I&S)[49].



In het kader van de invoering van SAFe (Scaled Agile Framework) wordt dit in 2019 gewijzigd in een productiehuis waarin bovengenoemde uitvoering komt met daarnaast een afdeling Strategie en Regie (inclusief standaarden en stelsels) en een afdeling Bedrijfsvoering & Organisatieontwikkeling. De implicaties van API-gewijze dienstverlening raken deze veranderende organisatie op drie vlakken:

Standaardisatie. Dit raakt naast Forum Standaardisatie en Centrum voor Standaarden alle voorzieningen aangezien Logius zelf deze standaarden toepast, gezien de relatie met Digikoppeling raakt het ook Stelseldiensten.

Uitvoering in het productiehuis. Afhankelijk van de breedte van de API-gewijze dienstverlening (zie scenario's hierna) worden één of meerdere voorzieningen geraakt. Het feit dat deze in één productiehuis worden ondergebracht en dat daarbinnen op basis van een eenduidig sturingsmodel (SAFe) gewerkt wordt, is daarbij behulpzaam (zie § **Fout! Verwijzingsbron niet gevonden.**).

[47] Het visiedocument "Dit is Logius" vormt de achtergrond van deze beschrijving van de impact
[48] Men zou deze indeling kunnen zien als bedrijfsfuncties, of meer in Agile termen als Capabilities. In het kader van de Agile transformatie is het relevant deze functies meer en meer om te vormen naar Value streams, zie https://www.scaledagileframework.com/value-streams/
[49] Bron https://www.logius.nl/fileadmin/logius/ns/over-logius/Organogram-Logius-2018-maart.pdf

Portfoliomanagement. In de meer verregaande scenario's beperkt API-gewijze dienstverlening zich niet tot één enkele voorziening. Dit vereist een vorm van portfoliomanagement, zeker indien API-gewijze dienstverlening ingezet wordt om de digitalisering te versnellen. Aangenomen wordt dat dit in de nieuwe afdeling Strategie en Regie past.

5.3 Scenario's Tech only gebeurt al en kan eenvoudig gestimuleerd worden

In alle voorzieningen waarvan in het kader van deze verkenning medewerkers geïnterviewd zijn, worden al API's toegepast voor integratie binnen de voorziening. Onder architecten en lead developers bestaat een goed beeld van de technische mogelijkheden van REST API's. Voor nieuwe voorzieningen en grotere doorontwikkelingen wordt expliciet gekozen voor toepassen van API's als integratietechnologie, bijvoorbeeld voor DigiD Machtigen en voor MOvO. Het boven geschetste scenario "Tech Only" is dus al werkelijkheid. Zie bijlage Bijlage C voor API's die momenteel al binnen Logius worden toegepast.

Aanbeveling: Logius doet er verstandig aan deze ontwikkeling actief te (blijven) stimuleren. De meest voor de hand liggende vorm daarvan is het organiseren van kennissessies met architecten van de verschillende voorzieningen en het consequent geven van demo's vanuit een voorziening waar bepaalde aspecten van API's zijn gerealiseerd aan de teams van andere voorzieningen. Dit kan in de vorm van een community of practice[50].

Bij een aantal voorzieningen speelt de ontwikkeling naar verdere DevOps werkwijze. Om de vruchten van deze werkwijze te plukken is verregaande automatisering van deployments, of eigenlijk breder, automatisering tot volledige continuous integration – continuous deployment (CICD) van belang[51]. Deze automatisering maakt veelal gebruik van het feit dat de meeste componenten van deze keten met API's te "besturen" zijn. In plaats van bij iedere deployment van een component handmatig en procedureel de benodigde verbindingen door een firewall heen in te regelen, kan vooraf worden geprogrammeerd hoe deze verbindingen op het moment van deployment worden gerealiseerd doordat geautomatiseerd de betreffende API's van de firewall worden aangeroepen (een web applicatie firewall).

Aanbeveling: De mogelijkheden van automatisering binnen de software ontwikkeling en deployment groeien zeer snel en zijn onderdeel van dezelfde technologievernieuwing op basis van API's. Het wegnemen van belemmeringen om deze te benutten, vereist managementaandacht. Globale kennis van de verregaande mogelijkheden is van belang om te voorkomen dat beslissingen over verantwoordelijkheidsverdeling tussen afdelingen, de implementatie van SAFe of sourcingsvraagstukken leiden tot hindernissen voor deze verdere automatisering. Inzet van leveranciers om geheel in deze verregaande automatisering mee te draaien dient een eis te zijn in nieuwe opdrachten.

Tech only scenario samengevat: Het tech only scenario treedt duidelijk al op. In dit scenario past Logius API's toe als integratie technologie en bij de automatisering van het voortbrengingsproces. Dit gebeurt vooral per voorziening afzonderlijk. Daardoor is er weinig hergebruik. Het management kan dit eenvoudig verder stimuleren door ruimte te geven aan een community of practice rond REST API's als integratietechnologie en door hindernissen voor automatisering van software ontwikkelproces weg te nemen.

5.4 Logius functioneert al in een ecosysteem van API-gewijze dienstverlening

Logius voert de digitale dienstverlening van en voor de overheid samen met diverse andere organisaties uit. API-gewijze dienstverlening kan worden benut voor de integraties tussen deze organisaties en voor ketens die door deze partijen heen lopen. Op beperkte schaal gebeurt dit al. Zo neemt MijnOverheid de REST API van de KvK af en wordt het OIN-

[50] https://www.scaledagileframework.com/communities-of-practice/
[51] Zie https://www.scaledagileframework.com/continuous-delivery-pipeline/

register via een API ontsloten voor DSO. De deelname aan het kennisplatform API's laat zien dat er al sprake is van een ecosysteem waarin API-gewijze dienstverlening ontstaat.

Gezien de rol die Logius heeft in de generieke digitale infrastructuur en gezien de positie van Forum Standaardisatie wordt naar Logius gekeken voor de verdere standaardisatie van de toepassing van API's binnen de Nederlandse publieke sector. Het oppakken van deze rol past in de Kompas-ambities "regie op samenhang binnen de generieke digitale infrastructuur" en "bundelen kennis en expertise". Zie hoofdstuk 3 voor de concrete invulling hiervan en bijbehorende aanbevelingen.

Aanbeveling: Beleg het bijdragen aan het kennisplatform API's en de samenhang met de verdere standaardisatie eenduidig in de nieuwe afdeling Strategie en Regie. Teken het bijbehorende manifest en draag uit dat Logius de nieuwe standaarden zelf in haar voorzieningen toepast. Dit zijn stappen waarmee Logius vanuit eigen kracht invulling geeft aan groei van API-gewijze dienstverlening.

Eco-systeem scenario samengevat: De eerste contouren van een ecosysteem rond API's tekenen zich af. In dit scenario gaat Logius de rol in standaardisatie die momenteel in de generieke digitale infrastructuur gespeeld wordt, in het bijzonder rond Digikoppeling ook spelen ten aanzien van API-gewijze dienstverlening. Logius vervult daarin tenminste een faciliterende en stimulerende rol. Begrip van API-gewijze dienstverlening en sturing op de benodigde stappen vereisen een duidelijk managementcommitment en de benodigde afstemming met BZK en de in vorige hoofdstuk genoemde andere belanghebbenden.

5.5 Verdergaande organisatorische impact van een Digitaal Logius

In dit scenario maakt Logius de strategische keuze om zelf een "digital enterprise" te worden. De consequentie daarvan is maximale inzet op API-gewijze dienstverlening. Dit heeft duidelijk grotere organisatorische impact dan de voorgaande twee scenario's. Deze impact wordt geïllustreerd aan de hand van "Dit is Logius" (zie volgende pagina).

Dit scenario omvat een aantal elementen:

a. Voortvarende aanpak van benodigde standaardisatie, bij voorkeur binnen de Digikoppelingfamilie (als beschreven in hoofdstuk 3).
b. Een actief beheerde developercommunity en ruimte om à tempo op wensen van deze nieuwe doelgroep in te gaan.
c. API's als strategie voor hergebruik tussen voorzieningen (doorbreken van verzuiling), ten minste voor een deel van de voorzieningen.
d. Koppeling tussen deze interne strategie en API-gewijze dienstverlening aan de buitenwereld (eat your own dogfood)
e. Een API-first strategie voor MijnOverheid.nl als meest voor de handliggende startpunt van API-gewijze dienstverlening door Logius.
f. Een duidelijke roadmap voor aanpassing van DigiD voor app diensten en API's die autorisatie vereisen.
g. API-gewijze dienstverlening als voorkeursarchitectuur voor doorontwikkeling van GDI.
h. Toewerken naar een Logius brede API-managementvoorziening met voldoende schaalbaarheid en specifieke voorzieningen voor kostenverrekening, toezicht en privacybescherming in de overheidscontext.
i. Doorontwikkeling van de stelselcatalogus tot een instrument dat zowel beleidsmakers als ontwikkelaars ondersteunt in efficiënt hergebruik van overheidsinformatie en wettelijke definities.

Op hoofdlijn wijzigt dit de missie en taak van Logius niet, wel leidt het op alle belangrijke aspecten van missie, visie en strategie tot belangrijke accenten.

Allereerst komt er een nieuwe doelgroep bij in de vorm van de ontwikkelaars die de door Logius aangeboden API's benutten. In dit scenario betreft dit in eerste instantie ontwikkelaars die in opdracht van andere overheden werken, die apps leveren aan andere overheden. Aangezien ontwikkelaars het beste begrijpen wat andere ontwikkelaars verwachten is de betrokkenheid van eigen ontwikkelaars bij deze community essentieel. Dat betekent dat hiervoor capaciteit moet worden ingepland in het productiehuis. In het portfoliomanagement en de regiefunctie is een proces nodig om prioriteitsconflicten tussen wensen vanuit beleid van het ministerie en vanuit deze developercommunity adequaat af te wegen. Zeker in de beginfase is het gewenst dat er naast bijdragen vanuit alle teams die API's aanbieden één eindverantwoordelijke is en aandacht voor kwaliteit van het geheel. Dat is een nieuwe taak.

Ten tweede is portfoliomanagement en enterprise architectuur over de voorzieningen heen (g) noodzakelijk om API's als strategie voor hergebruik (c), gekoppeld aan het effect in de buitenwereld (d) door te zetten naar de voorzieningen (e, f, h, i). De inrichting van een afdeling Strategie en Regie en diverse SAFe best practices bieden daar een goede basis voor. Deze functie zal richting moeten geven aan toepassen van API-gewijze dienstverlening in de doorontwikkeling van voorzieningen, in de stapsgewijze ombouw van legacy en in het toepassen van microservices in nieuwe te realiseren voorzieningen.

Ten derde zal in dit scenario het aantal en het gebruik van API's snel groeien. Dat vereist opbouw van een API-managementplatform. Dit is een andersoortige voorziening dan de huidige voorzieningen omdat deze faciliterend is aan de andere voorzieningen maar ook dienstverlenend. Dit API-managementplatform stelt ook eisen aan de toekomstige infrastructuur, maar valt daar niet mee samen. Een op cloudtechnologie en containers gebaseerde infrastructuur is noodzakelijk in dit scenario. Het SAFe framework biedt handvaten hoe dergelijke faciliteiten (enablers) ingepland worden en door de teams gerealiseerd. Bijvoorbeeld goed functionerende architectural runways en een system team zijn hiervoor belangrijke concept. Samenwerking met bestaande leveranciers en verankering hiervan in nieuwe contracten is gezien het regiekarakter van Logius daarbij essentieel.

Aanbeveling: Maak onderscheid tussen het API-managementplatform met bijbehorende dienstverlening en community management en de onderliggende infrastructurele platformen. Beide zijn faciliteiten die lateraal staan op de voorzieningen. Beide dragen bij aan het doorbreken van verzuiling en hergebruik. Echter, wanneer de API-gewijze dienstverlening gereduceerd wordt tot "infrastructuur" dan is het moeilijker de bijbehorende cultuurverandering en gerichtheid op dienstverlening te realiseren.

Naast deze nieuwe taken en organisatorische impact heeft dit scenario gevolgen voor de doorontwikkeling van de GDI. Het is niet nodig om deze verandering voor alle GDI tegelijk in te zetten. Het ligt het meest voor de hand in ieder geval met DigiD (f) en MijnOverheid (e) te starten.

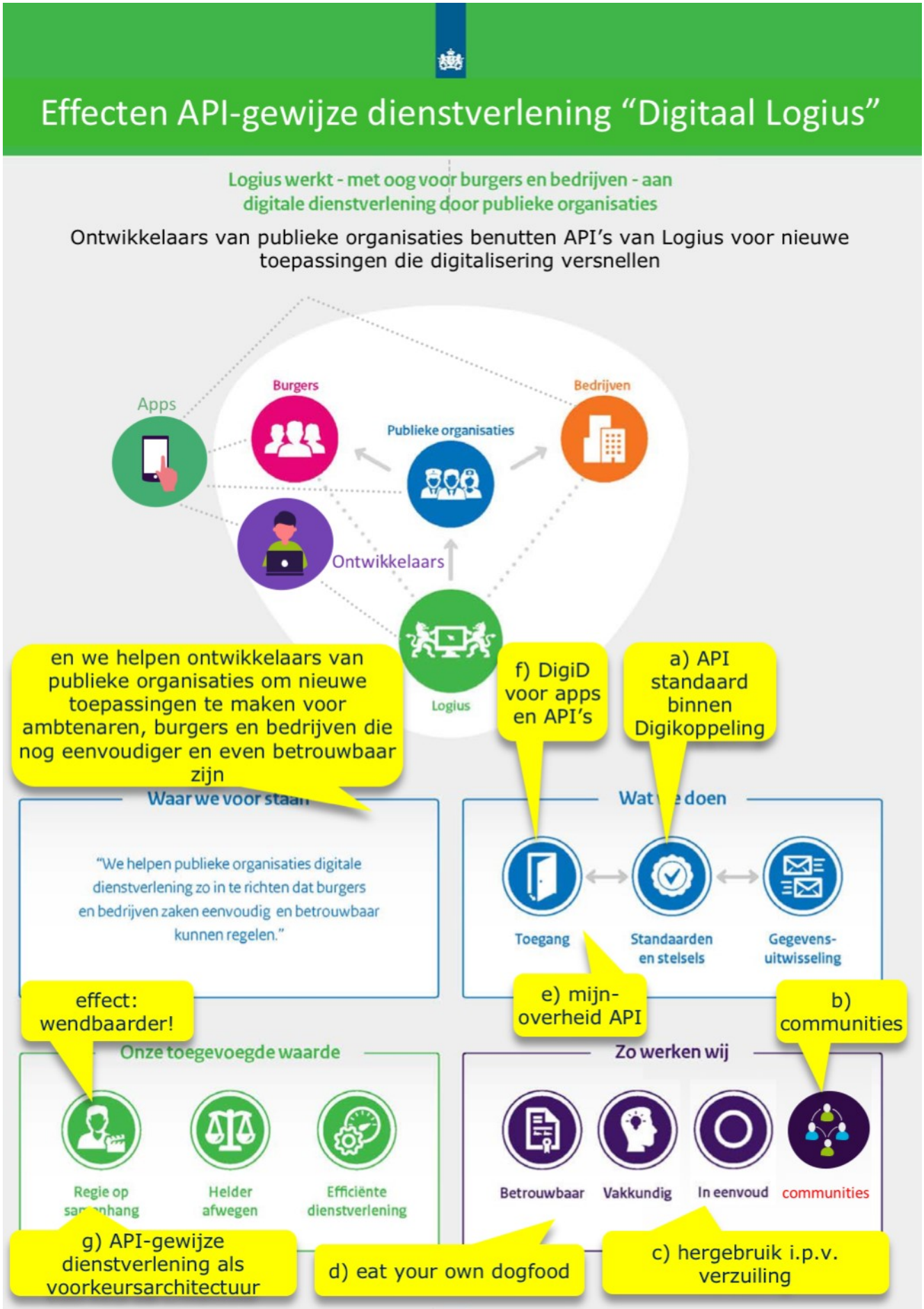
Een API-first strategie voor MijnOverheid.nl betekent dat voor alle gegevens die een burger daar over zichzelf kan inzien ook een API beschikbaar is die los van het portaal in een app benut kan worden. Conform eat your own dogfood zal het portaal zelf ook meer en meer deze API's benutten.

Omdat een groot deel van de voor burgers relevante gegevens niet publiekelijk toegankelijke zijn, vereist dit de inrichting van nieuwe Oauth gebaseerde authenticatie-mechanismen om afnemende apps mee te identificeren zodanig dat alleen door Logius goedgekeurde apps toegang krijgen. In dit scenario betreft het apps van andere overheden. Daarnaast is verdere doorontwikkeling van de app-to-app koppeling op basis van open ID connect hiervoor nodig. Dit vereist doorontwikkeling van DigiD.

Dit scenario kan niet worden uitgevoerd zonder een duidelijke en goed voorbereide strategische keuze en vereist een aanvullende taak en budget vanuit BZK.

Aanbeveling: Werk de API-gewijze dienstverlening voor MijnOverheid en DigiD in samenhang uit en betrek daarin ook stelseldiensten. Werk vanuit het SAFe concept value stream, gericht op de waarde voor burgers.

Aanbeveling: Voer de geplande pilots zo uit dat ervaring opgedaan wordt met API-managementplatformen en het vormgeven van een community van developers.


Effecten API-gewijze dienstverlening “Digitaal Logius”
Logius werkt - met oog voor burgers en bedrijven - aan digitale dienstverlening door publieke organisaties
Ontwikkelaars van publieke organisaties benutten API’s van Logius voor nieuwe toepassingen die digitalisering versnellen
Apps
Burgers
Bedrijven
Publieke organisaties
Ontwikkelaars
Logius
en we helpen ontwikkelaars van publieke organisaties om nieuwe toepassingen te maken voor ambtenaren, burgers en bedrijven die nog eenvoudiger en even betrouwbaar zijn
f) DigiD voor apps en API’s
a) API standaard binnen Digikoppeling
Waar we voor staan
“We helpen publieke organisaties digitale dienstverlening zo in te richten dat burgers en bedrijven zaken eenvoudig en betrouwbaar kunnen regelen.”
Wat we doen
Toegang
Standaarden en stelsels
Gegevens-uitwisseling
effect: wendbaarder!
e) mijn-overheid API
b) communities
Onze toegevoegde waarde
Regie op samenhang
Helder afwegen
Efficiënte dienstverlening
Zo werken wij
Betrouwbaar
Vakkundig
In eenvoud
communities
g) API-gewijze dienstverlening als voorkeursarchitectuur
d) eat your own dogfood
c) hergebruik i.p.v. verzuiling

5.6 Een radicalere digitale strategie met Logius in kernrol

In dit scenario is het niet Logius die een strategische keuze maakt maar het ministerie van BZK. Dat betekent een meer politiek besluit, dat uiteraard wel medegebaseerd kan zijn op de input van Logius vanuit haar regie-rol. Het leitmotiv van dit scenario is dat API-gewijze dienstverlening bij uitstek een invulling is van NLDIGIbeter.

Een dergelijk besluit zal zich vertalen in andere accenten in de doorontwikkeling van de GDI en vereist een intensivering daarvan. De radicaal digitaal strategie vereist invulling van nieuwe taken. Bij ieder van die taken dient afgewogen te worden in hoeverre het een generieke voorziening betreft die georganiseerd kan worden als een GDI.

Het betreft in ieder geval de volgende taken:

- Uitvoering developer.overheid.nl. Daarmee wordt de mogelijkheid om toepassingen te bouwen voor de interactie tussen overheid en burgers bewust uit handen gegeven aan burgers en bedrijven.
- DigiD voor applicaties. Openstellen van API's vereist kan alleen op een verantwoorde manier als de benodigde authenticatie- en autorisatiemechanismen worden ingericht. Er zal een "DigiD voor applicaties" moeten zijn en er zal actief beheer nodig zijn om de toegang tot overheids-API's voor applicaties die de regels overtreden de toegang te ontzeggen. Voor meer gevoelige API's zal een toelatingsproces moeten worden ingericht. De mogelijkheden voor een dergelijke open overheid nemen toe wanneer de in DigiD Hoog ingebouwde polymorfe pseudoniemen als alternatief voor het BSN in omloop komen.
- Dataminimalisatie API's. Basis API's die gegevens leveren uit bijvoorbeeld basisregistraties kunnen het meest effectief bij de uitvoerder van betreffende basisregistratie worden gerealiseerd. Ten behoeve van privacy enhancing is er behoefte aan een laag van API's daarbovenop die gegevens afschermt, die bijvoorbeeld de geboortedatum uit de basis-API omzet naar een ja/nee antwoord op de vraag "is meerderjarig?"

Het moge duidelijk zijn dat dit scenario niet van vandaag of morgen zal ontstaan. Onderdelen uit dit scenario, bijvoorbeeld developer.overheid.nl, kunnen ook in het voorgaande minder radicale scenario gerealiseerd worden.

6. Advies: API-gewijze dienstverlening is chefsache en vraagt concrete actie

6.1 Kernadvies: Maak dubbelslag op basis van API-gewijze dienstverlening

Wij adviseren Logius in te zetten op een dubbelslag gebaseerd op API-gewijze dienstverlening. Dubbel omdat API-gewijze dienstverlening zowel een goede basis biedt voor de concretisering van beleidswensen van de opdrachtgevers op basis van NL DIGI beter als invulling geeft aan de benodigde interventie op verzuiling van GDI voorzieningen. Dit is een invulling van de "heldere keuzes ten aanzien van producten en diensten die bij Logius passen" en een actueel en relevant strategisch vraagstuk voor Logius omdat het taak en positie op middellange termijn beïnvloedt.

Uit de interviews komt naar voren dat hiervoor intern draagvlak en enthousiasme bestaat. Bij Logius afnemers en belanghebbenden vanuit al gestarte API-initiatieven binnen de overheid liggen geen grote belemmeringen voor een rol van Logius op het gebied van API-gewijze dienstverlening op voorwaarde dat dit vanuit overleg en faciliterende rol wordt opgepakt en op voorwaarde van goede afspraken over de (toekomstige) kosten.

De mate en snelheid waarmee deze strategie uitgevoerd kan worden, hangt mede af van het gesprek tussen Logius en ministerie van BZK. Op een aantal punten zijn richtinggevende beleidsuitspraken gewenst. Anderzijds is het voor Logius van groot belang verwachtingen te wekken die ook daadwerkelijk gerealiseerd kunnen worden. Dit staat niet los van de "doorontwikkeling Logius" en de SAFe invoering.

Met deze strategie bevordert Logius vanuit eigen kracht het scenario "Digitaal Logius" en draagt ze bij aan bredere ontwikkeling richting het scenario "Radicaal Digitaal".

6.2 API-gewijze dienstverlening is voor Logius van strategisch belang

Het bovengenoemde kernadvies impliceert dat API-gewijze dienstverlening voor Logius strategisch belang is. Voor het Logius MT, haar BZK-opdrachtgevers en de beslissers van belangrijke Logius afnemers is inzicht in het belang van API-gewijze dienstverlening voor digitalisering daarom relevant.

Dit inzicht betreft met name de niet-technologische aspecten, namelijk:
– API's „externaliseren" stukjes van de eigen taakuitvoering. Dit "binnenste buiten keren" heeft belangrijke effecten op hoe die taakuitvoering plaatsvindt. Het leidt tot een andere relatie met afnemers, het dwingt tot "van buiten naar binnen denken" en biedt een sterke prikkel voor hoogwaardige dienstverlening. Logius heeft de potentie hierin binnen de overheid voorop te lopen.
– De trend gaat richting verder opknippen in kleine voor zich eenvoudiger brokken losjes gekoppelde functionaliteit op alle ICT lagen van de besturingslaag, via apps bij de gebruiker tot containers in de infrastructuur.
– De focus op hergebruik via API's en het meebewegen in de trend naar kleinere en meer zelfstandige stukjes functionaliteit komt met een prijs. Deze prijs is de noodzaak veel aandacht te besteden aan de faciliterende laag. Een kwalitatief goede faciliterende laag is een harde voorwaarde om deze ontwikkeling te maken. Dit betreft ICT voorzieningen in infrastructuur en tooling en het betreft de benodigde managementsteun om deze faciliteiten goed en Logius-breed te organiseren en te bekostigen.

Het advies is door te gaan met het agenderen van deze inhoudelijke thema's in het Logius MT, het enthousiasme hiervoor binnen de organisatie te steunen en te laten groeien en te sturen op leren van belemmeringen en groei van successen, mede gerelateerd aan de "doorontwikkeling Logius" en de invoering van SAFe.

Daarnaast is het aanbevelenswaardig BZK voor te stellen enkele meer inhoudelijke thema's die Logius overstijgen verder uit te werken: In ieder geval: "API's, doelbinding en toestemmingen", mogelijk ook "grenzen en mogelijkheden van federatieve inrichting van API-gewijze dienstverlening" (ter onderbouwing en validatie van Common Ground strategie).

6.3 Neem via Digikoppeling rol in standaardisatie API-gewijze dienstverlening

Voor de geloofwaardigheid van de dubbelslag is voortgang in het standaardisatieproces belangrijk. Omdat conform het "eat you own dogfood" adagium interne integratie verbonden is met bevraging vanuit de buitenwereld heeft opnemen van standaarden voor REST API's binnen de Digikoppeling familie de voorkeur. Bovendien blijft de waarde van de "merknaam" Digikoppeling als afspraak voor interoperabiliteit voor gegevensuitwisseling binnen de overheid daarmee behouden.

Dit vereist dat Logius een nog pro-actievere rol gaat spelen in het kennisplatform API's en een middellange termijn strategie opstelt voor doorontwikkeling van DigiD voor authenticatie en autorisatie met apps en API's.

6.4 Voer in 2019 pilots uit per voorziening

Maak gebruik van de ruimte en (terechte!) aandacht die er momenteel is voor innovatie om verdere pilots te doen met API-gewijze dienstverlening. Het domein MijnOverheid, DigiD. Stelseldiensten is hiervoor het meest voor de hand liggend. Gedacht kan worden aan vier parallel lopende pilotprojecten:

- Enkele MijnOverheid API's waarmee gegevens die de burger nu binnen MijnOverheid kan zien via een API bevraagd kunnen worden. Dit vereist toepassing van het Oauth en Open ID connect profiel.
- Pilot rond leveren en controleren van Oauth en Open ID connect tokens gekoppeld aan DigiD
- Beproeving van hoog volume, event gedreven berichtenverkeer op basis van API standaarden als doorontwikkeling van Digilevering. Deze beproeving omvat ook het beoordelen van toepasbaarheid van het Estse X-Road en het in kaart brengen van benodigde opschaling van NLX (van Common Ground).
- Dwars op deze drie een pilot rond API-management waarbij actief beproeft wordt hoe een developer portal gefaciliteerd wordt en waarbij bekeken wordt hoe dit binnen de – in verandering zijnde – Logius organisatie kan worden ingepast.

Op basis van deze pilots zal een beter beeld ontstaan van de impact van API-gewijze dienstverlening op het geheel van de Logius voorzieningen.

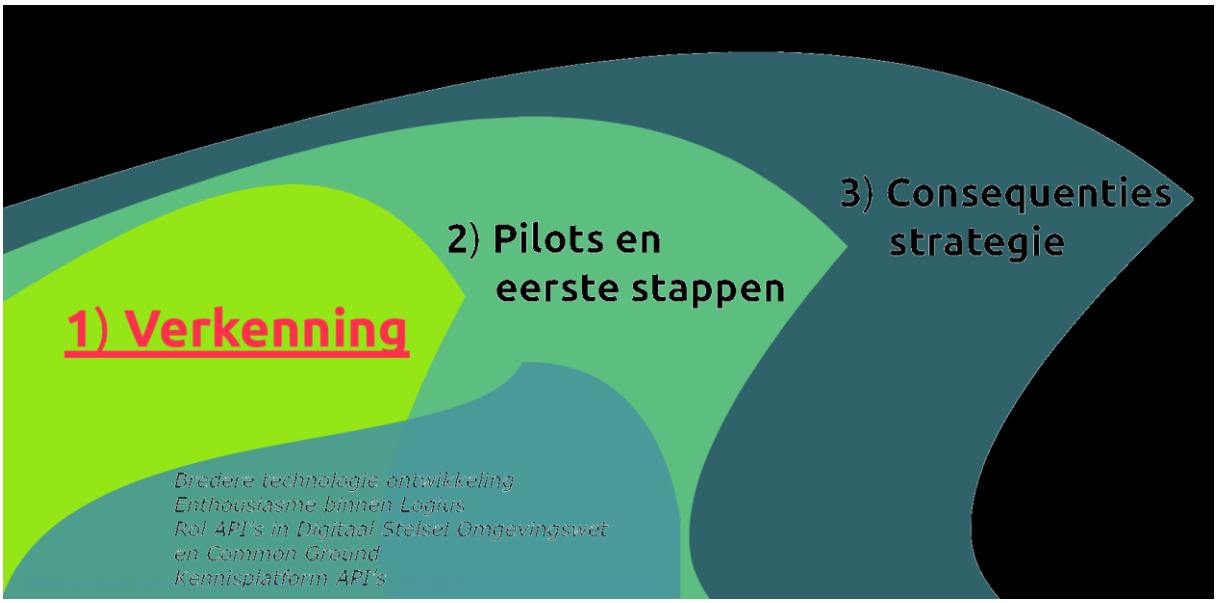
Bijlage A. Opdracht, vraagstelling en aanpak

Aanleiding voor de verkenning aangaande API's

"API's" vormen zowel de voorkeurstechnologie voor digitale dienstverlening als voor integratie van systemen. API gebaseerde *platforms* vormen de ICT van belangrijke innovaties in dienstverlening buiten de overheid, maar ook in de ontwikkelrichting van bijvoorbeeld het Digitaal Stelsel Omgevingswet en strategische ICT-plannen van afnemers van Logius (b.v. UWV en Kadaster). Als uitvoerder van de Generieke Basisinfrastructuur, specifieker de stelselvoorzieningen, wil Logius team Stelseldiensten deze ontwikkeling verkennen. Op basis van deze verkenning wil Logius samen met de opdrachtgever voor de stelselvoorzieningen, het ministerie van BZK, bepalen welke impact deze technologie heeft op de stelselvoorzieningen en de verdere doorontwikkeling van vergelijkbare basisinfrastructuur voor de Nederlandse digitale overheid. Het bepalen van deze impact vereist een inzicht in de technologie en meer nog in de gevolgen ervan voor de taken en dienstverlening van Logius.

Context van de verkenning

De verkenning staat in de context van de bredere interesse voor API-gewijze dienstverlening binnen de Nederlandse overheid. Deze krijgt onder andere vorm in het kennisplatform API's waarin Logius participeert. Het is de bedoeling met lopende pilots door te gaan en in 2019 nog verdere pilots te realiseren.



Vraagstelling van Logius

Logius heeft Capgemini gevraagd om in kaart te brengen wat de impact van *een API-gewijze benadering van de Basisinfrastructuur* op haar dienstverlening zou zijn.

Onderdelen van die vraagstelling zijn:

- Een definitie van API-gewijze benadering (zie § 1.2) en marktscan (zie hoofdstuk 0).
- Bepaling van het *toepassingsgebied* van API's in de context van de huidige en toekomstige Basisinfrastructuur (stelselvoorzieningen, basisregistraties en aanverwante diensten binnen de overheid waarvan inrichting als generieke digitale infrastructuur voor de hand ligt). Zie hoofdstuk 3.
- Welk effect zou het toepassen van de API-gewijze benadering hebben op *voorzieningen, diensten, processen en ketens* van deze Basisinfrastructuur en op welke aspecten levert dit een verbetering op ten opzichte van de bestaande dienstverlening? Zie hoofdstuk 5.
- Het in kaart brengen van *behoeften* en verwachtingen van afnemers van Logius, uitvoeringsorganisaties van de overheid ten aanzien van API-gewijze dienstverlening op het toepassingsgebied. Zie hoofdstuk 4.
- Wat zou er nodig zijn bestaande diensten door te ontwikkelen tot API-diensten? In welke *sectoren*, bij welke Logius-afnemers en voor welke diensten is dit met name *kansrijk*? Zie advies, in het bijzonder 6.4 en **Fout! Verwijzingsbron niet gevonden.** en Bijlage D.

De verkenning wordt geschreven voor een niet-ICT doelgroep die daarmee inzicht wil verkrijgen in de organisatorische gevolgen en de effecten op de dienstverlening van Logius, van de basisregistraties waarmee Logius het stelsel vormt en van de afnemers die de stelselvoorzieningen en andere Basisinfrastructuur benutten in hun eigen taakuitvoering en *afhandelingsregistraties*. Gerelateerd aan bovenstaande vraagstelling is er een aantal aanvullende specifieke vragen, zie **Fout! Verwijzingsbron niet gevonden.**.

Aanpak

Voor de verkenning is zowel binnen Logius als bij Logius afnemers met een groot aantal betrokkenen gesproken. Stukken aangaande actuele ontwikkelingen en voorbeelden uit deze gesprekken zijn in deze verkenning verwerkt (zie voetnoten). De hoofdlijn van de verkenning is in interactieve workshops met begeleidingsteam van Logius Stelseldiensten en in het reguliere overleg met BZK en bij API-management betrokken afnemers getoetst. Tenslotte zijn de conclusies op 14 januari 2019 besproken in het Logius MT en heeft een presentatie plaatsgevonden bij BZK DIO op 20 december 2018. De tekst van de verkenning is gebaseerd op expertise van het team, aangevuld met specialisten van Capgemini. Vervolgens is deze tekst door deze experts gereviewed en door het begeleidingsteam gereviewed. Van de eindversie is een versie voor verspreiding buiten Logius gemaakt (deze versie), naast de versie die de voor Logius intern relevante adviezen bevat.

Betrokken afnemers

Voor deze verkenning is gesproken met verschillende organisaties:

- Gemeente Amsterdam
- Ministerie van Onderwijs, Cultuur en Wetenschap
- Ministerie van Binnenlandse Zaken en Koninkrijksrelaties
- BKWI
- CBS
- Geonovum
- Digitaal Stelsel Omgevingswet
- VNG
- UWV

Bijlage B. Eigenschappen van REST

De afkorting REST staat voor REpresentational State Transfer. Het betreft enkele strikte regels om eenvoud te bereiken. Het richt zich op integratie in de context van allerlei devices die allerlei internetdiensten benutten. Net zoals het voor een mens makkelijk is meerdere stukjes informatie van een aantal webpagina's te bekijken en te verwerken, is het voor een ontwikkelaar van bijvoorbeeld een app makkelijk meerdere REST gebaseerde API's te benutten in zijn app. Daarnaast past REST in de langere termijn technologie ontwikkeling naar kleinere korrelgrootte, deze lange termijn trend verbindt REST met zaken als containerisering van de infrastructuur en microservices (zie verderop).

REST-regels voor eenvoud van API's betreffen:

Stateless is dat de server "vergeetachtig" is. De server, de voorziening die de eindgebruiker aanroept, onthoudt niks over de vraag die de ene stelt of die de andere stelt. Iedere vraag wordt hetzelfde behandeld, er wordt niet onthouden of deze gerelateerd is aan een eerdere vraag van dezelfde afnemer. De server geeft het antwoord en vergeet dan alles. Dat maakt het goed afhandelen van antwoord voor de afnemer ingewikkelder. Maar het grote voordeel van deze eenvoud is dat het niet uitmaakt of de ene server de vraag afhandelt of de ander. De afhandeling kan gedaan worden door heel veel servers naast elkaar die niet onderling hoeven te communiceren. Stateless betekent dat het heel schaalbaar is en dat heel grote volumes haalbaar zijn. REST is hierdoor ook een protocol met weinig overhead in de communicatie en dus met erg kleine omvang van berichten.

Duidelijke scheiding client (afnemer, vrager) **en server** (aanbiedende voorziening). Zolang de interface (de API specificatie) niet wijzigt, kunnen afnemer en aanbieder onafhankelijk van elkaar wijzigingen doorvoeren.

Cacheable en layered system zijn twee verdere regels die schaalbaarheid bevorderen. Net als webpagina's geven antwoorden aan of deze dichter bij de afnemer tijdelijk opgeslagen kunnen worden (caching). Gelaagdheid betekent dat er tussenliggende servers kunnen zijn (voor schaalbaarheid, veiligheid etc.) maar dat de afnemer het verschil niet ziet tussen een directe verbinding en een via tussenliggende servers. De afnemer hoeft hier geen weet van te hebben en de aanbieder kan hierin flexibel zijn.

Tenslotte kan de server als optie programmeercode naar de client sturen voor lokale processing, bijvoorbeeld JavaScript. Dit wordt veelvuldig toegepast in moderne rijke user interfaces.

Voor uniformering van de interface benut REST het basisprotocol dat ook websites benutten, namelijk HTTP. Dit is strikt opdracht - antwoord. Voor webpagina's geïmplementeerd met webbrowser en webserver. De interactie wordt opgesplitst in basale opdrachten als „haal op wat staat op deze URL", „schrijf dit veld naar die URL" met een antwoord in XML of JSON. **Door het gebruik van het HTTP basisprotocol is REST eenvoudig, zeer breed toepasbaar en erg efficiënt.** Alle dingen die opgevraagd of gewijzigd kunnen worden krijgen hun eigen hyperlink ofwel URL (of eigenlijk de veralgemenisering daarvan een Uniform Resource Indicator (URI)). De antwoorden bevatten alle informatie die nodig is om ze te verwerken of aan eindgebruikers te tonen.

Ondanks de eenvoud zijn daar complexe interacties mee te realiseren. Het antwoord kan hyperlinks (URI's) bevatten die de volgende stap aanduiden in een proces. Dan wordt gesproken van Hypermedia as the Engine of Application State (HATEOAS).

Deze regels maken REST „**loosely coupled**" en geoptimaliseerd voor performance, schaalbaarheid, eenvoud, flexibele aanpasbaarheid en hergebruik.

Verder lezen over deze regels, zie bijvoorbeeld https://www.manning.com/books/irresistible-apis.

REST is niet gericht op formele validatie en het doorgeven van berichten over een lange keten met bewijsbare betrouwbaarheid en encryptie. Dergelijke eisen passen beter bij SOAP/WSDL en/of ebMS, de beide protocollen die nu binnen Digikoppeling voor de Nederlandse overheid gestandaardiseerd zijn. Zie Discussiedocument RESTful APIs Versie 1.1 – juni 2016. Dit neemt niet weg dat er rondom REST innovaties plaatvinden die dezelfde kant op gaan. De voordelen van eenvoud en schaalbaarheid wegen immers ook wanneer deze zwaardere eisen gelden.

Bijlage C. Gerealiseerde API's Logius

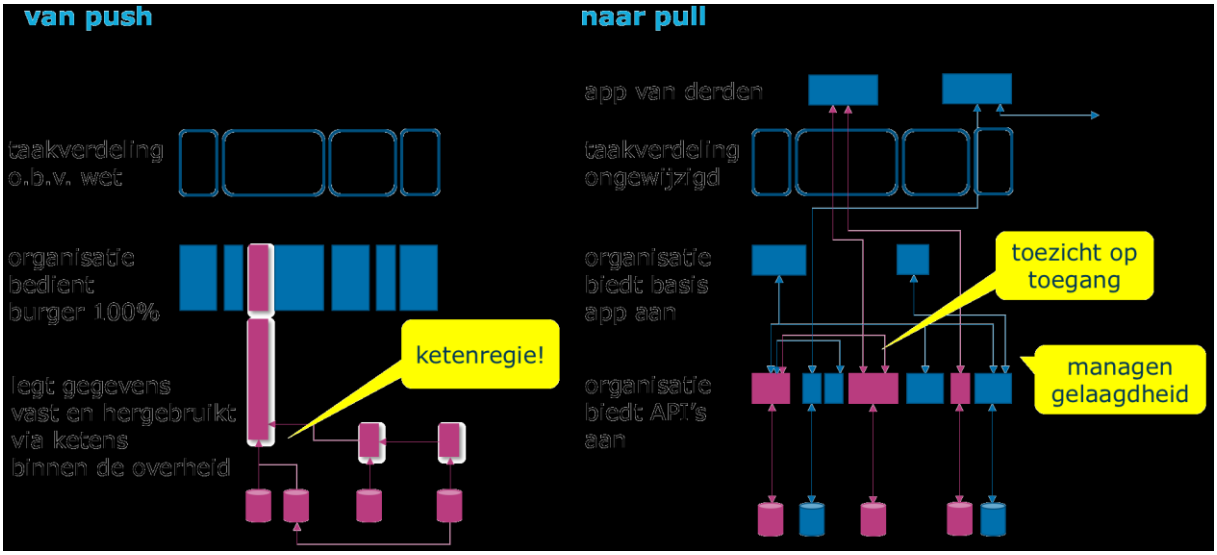
Afdeling	API	Genoemd door
Machtigingen	API voor gegevensuitwisseling	Frank Zwart
Mijn Overheid	API voor Berichtenbox	Marc Bonekamp
Stelseldiensten	API voor OIN register (+ structuur voor RDF-endpoint voor Stelselcatalogus) Gaat eind december in productie op portaal.digikoppeling.nl/register/api.	Pieter Hering

Op haar eigen website biedt Logius een basis overzicht van API ontwikkelingen binnen de overheid: https://www.logius.nl/diensten/overheidsgegevensnl/overzichten[52].

[52] Opmerkelijk aan dit overzicht is dat er naast verwijzingen naar andere overheden ook één verwijzing naar een marktpartij tussen staat!

Bijlage D. Architectuur impact API's

API-gewijze dienstverlening kan tot andere manier van uitvoeren van overheidstaken leiden. Omdat dit op middellange termijn grote invloed heeft op de GDI en op de inspanningen van Logius afnemers, wordt dit als een eerste schets besproken. Dit is uitdrukkelijk bedoeld als startschets voor verder gesprek met architecten.



Bovenstaand schema geeft links het huidige paradigma voor gedeeltelijk gedigitaliseerde uitvoering van overheidstaken en rechts de situatie van een API-overheid. Van boven naar beneden, beginnend bij de huidige situatie (links):

Op basis van wettelijke taken is de uitvoering van overheidsdiensten als belastingheffing, verlenen van bijstand, verlenen van vergunningen, van subsidies e.d. toebedeeld aan overheidsorganisaties.

Deze organisaties realiseren vervolgens portals en applicaties voor burgers voor de digitale indiening en afhandeling van bijbehorende processen. Er is in de Nederlandse situatie slechts een beperkt aantal generieke voorzieningen als MijnOverheid.nl. Ook voor deze voorzieningen is sprake van een wettelijke taak (of is deze in aantocht) en technologisch zijn het soortgelijke losstaande voorzieningen met een afbakening gebaseerd op wettelijke taak. De paars gearceerde kolom is bijvoorbeeld de applicatie voor bepaalde toeslagen die gegevens over inkomen e.d. ophaalt uit een achterliggende keten.

"Onder" deze voorzieningen bevindt zich zonder uitzondering een complex landschap van back office voorzieningen, veelal legacy. Tussen deze back offices zijn in de afgelopen decennia omvangrijke en complexe ketens gerealiseerd. In deze ketens worden gegevens gekopieerd en via berichtenverkeer doorgegeven. Ketenregie en omvangrijke programma's voor gegevenskwaliteit zijn daarbij noodzakelijk. Deze ketens en voorzieningen zijn "organisch gegroeid". Er is geen sprake van een overall plan en ook geen sprake van een ontwerp dat in hoge mate vanuit de burger (of bedrijven) is uitgewerkt.

Vervolgens naar rechts: In de API situatie is de toebedeling op basis van wettelijke taak niet anders.

De primaire focus van uitvoerders met een wettelijke taak is niet om een portal of andere vorm van gebruikersapplicatie aan te bieden om het geheel van die taak mee af te handelen. Deze focus is verschoven naar het aanbieden van API's voor onderliggende kleinere

processtappen. Vervolgens worden wel taakspecifieke gebruikersapplicaties (of apps) gemaakt, maar deze benutten de eigen API's (eat your own dog food). Daarnaast zijn er andere partijen (en dat kan buiten de overheid liggen) die dezelfde API's benutten voor specifieke apps.

De integratie vindt plaats door meerdere API's te benutten. Deze integratie is veel wendbaarder en kan vanuit burger ontworpen zijn, bijvoorbeeld op basis van een levensgebeurtenis. Omdat API's kleinere functionele brokken betreffen is er geen noodzaak aan complexe ketens in de back office.

Uiteraard heeft deze API wereld haar eigen uitdagingen en complexiteit. De getekende laag API's is een verregaande versimpeling, in werkelijkheid zal sprake zijn van een gelaagdheid. Het toezicht op gebruik van API's door derden is in overheidscontext ook zeker een belangrijke nieuwe complexiteit.

Dat neemt niet weg dat deze paradigmaverschuiving op termijn kan leiden tot heel andersoortige GDI, met minder behoefte aan reliable messaging en systemen voor het grootschalig doorgeven van basisgegevens tussen voorzieningen.

Het vereist ook een andere blik op bedrijfsfuncties. ICT is niet langer een facilitaire bedrijfsfunctie maar een onlosmakelijk aspect van primaire bedrijfsfuncties.

Bijlage E. Begrippen en afkortingenlijst

API | Application Programming Interface is een goed gedocumenteerde interface om (derden) digitaal toegang te geven tot stuk-jes van uw taakuitvoering en dit op een manier die eenvoudig te gebruiken is en enorm goed schaalbaar tot internetvolumes.

API-economie. Een economie waarin grootschalige dienstverlening plaatsvindt op basis van business modellen en afzetkanalen die gebaseerd zijn op API's waarmee realisatie van steeds nieuwe toepassingen door derden gestimuleerd wordt. (Vrij naar Gartner The API Economy: Turning Your Business Into a Platform (or Your Platform Into a Business), Gartner G00280448)

API-gewijze dienstverlening. Producten, diensten en stukken eigen taakuitvoering in de vorm van API's openstellen voor afnemers waarbij de API's als volwaardige producten gemanaged worden en een developer community een belangrijk aspect van de relatie met deze afnemers vormt. (Definitie op basis van § 1.3 t/m 1.5)

API life cycle De productlevenscyclus die de aanbieder van de API's doorloopt, van planning, ontwerp, realisatie publicatie, operationeel houden, monitoring, onderhouden, versiebeheer tot uitfasering.

API-managementplatform: De ICT-voorzieningen die nodig zijn om de gehele levenscyclus van API's te managen en API gewijze dienstverlening te leveren, onder te verdelen in voorzieningen voor ontwikkelaars en voor schaalbaar en veilig operationeel gebruik.

API-strategie. Een bedrijfsstrategie gericht op het stap voor stap groeien naar API-gewijze dienstverlening die zowel de benodigde enterprise architectuur als organisatie en cultuurverandering omvat.

Community Een via internet gefaciliteerde groep van medewerkers van afnemers die – in dit geval – geïnteresseerd zijn in de API's die worden aangeboden, deze benutten in hun eigen producten, er feedback op geven, verwachtingen hebben van de prestaties en van

de service die de community hen biedt. Kenmerkend is dat community leden zowel met elkaar, in het openbaar als met de aanbieder communiceren.

Containerisering. Ontwikkeling naar de situatie waarin applicaties in virtuele omgevinkjes draaien die volledig softwarematig geconfigureerd zijn en beheerd kunnen worden, zeer hoge schaalbaarheid mogelijk maken, zeer vaak deployen (devops) ondersteunen en zeer sterk onafhankelijk zijn van onderliggende infrastructuur.

Digital enterprise Een onderneming (organisatie) die processen en diensten zoveel mogelijk op basis van verregaande digitalisering heeft ingericht.

Externalisatie. Het aan derden aanbieden van stukjes van de eigen taakuitvoering.

IAM Identity Access Management

Loosely coupled: losjes gekoppeld applicaties of componenten betekent dat de ene component geen kennis heeft van de andere component. Hierdoor kan een component anders geïmplementeerd worden zonder afhankelijkheden met andere componenten, zolang de interface ongewijzigd blijft.

Microservices. Een software architectuur stijl gericht op fijnmazige modulariteit en protocollen met weinig overhead voor de interne opbouw van applicaties. De voorkeursstijl voor nieuw te realiseren applicaties waarmee API-gewijze dienstverlening en / of API gebaseerde integratie plaatsvindt.

PaaS | Platform as a service Als Cloud dienst aangeboden faciliteit waarmee een afnemer zowel zelf geproduceerde als aangeschafte applicaties in gebruik kan nemen (deployen) op een via internet (of privaat) geleverde cloud infrastructuur, inclusief benodigde tools en ontwikkelstraatfaciliteiten, waarbij de afnemer wel controle heeft over benodigde configuraties maar de onderliggende cloud infrastructuur niet hoeft te managen (dus zonder zorg over netwerk, servers, operating systemen, opslag e.d.) (Naar NIST Special Publication 500-292)

PSD2 De Europese Payment Services Directive 2 (PSD2) is nieuwe wetgeving die banken dwingt bankrekeningen openstellen via API-gewijze dienstverlening.

REST | RESTful API: API's die opgezet zijn op basis van RESTful principes. Deze principes vormen geen dichtgetimmerde standaard maar een architectuur. Zie Bijlage B

web-API: API's die toegankelijk zijn via het internet.