Ministerie van Justitie en Veiligheid

Domeinarchitectuur Data- en gegevensdiensten

Architectuurkatern API Management

'just in time-, just enough architecture'

Versie 0.8

Datum	21-11-22
Status	Concept

Colofon

Afzendgegevens **Directie Informatievoorziening en Inkoop**

Turfmarkt 147
2511 DP Den Haag
Postbus 20301
2500 EH Den Haag
www.rijksoverheid.nl/JenV

[redacted]

[redacted]

Inhoud

Documenthistorie

Versiehistorie

Versie	Datum	Status	Beschrijving
0.1	22-9-2022	concept	Initiële versie
0.2	27-9-2022	concept	Verwerking review bemerkingen
0.3	28-9-2022	concept	Verwerking review bemerkingen
0.4	3-10-2022	concept	Verwerking review bemerkingen
0.5	18-10-2022	concept	Verwerking review bemerkingen
0.5	7-11-2022	concept	Verwerking review bemerkingen
0.6	7-11-2022	concept	Verwerking review bemerkingen AF
0.7	9-11-2022	concept	Verwerking review bemerkingen AF
0.8	21-11-2022	concept	Verwerking advies CISO-board

Goedkeuring

Naam	Datum	Versie
Architectuurforum	9-11-2022	0.7
CISO Board	17-11-2022	0.7
CTO Overleg		
CIO Raad		

1 Inleiding

Het beschikken over correcte en actuele informatie is essentieel in de uitvoering van de taken van de overheid. Omdat de beschikbare informatie verdeeld is over een groot aantal organisaties is de tijdige uitwisseling van informatie van cruciaal belang. Om de uitwisseling van informatie veilig en betrouwbaar te laten verlopen wordt binnen het Ministerie van Justitie en Veiligheid (JenV) gebruik gemaakt van de Justitie Berichten Service, afgekort JUBES.

JUBES is als centraal component binnen het ICT landschap van JenV sinds 2005 verantwoordelijk voor het routeren, controleren en beveiligen van de gestructureerde informatie-uitwisseling. De volledige set aan uitwisselingsstandaarden van de overheid, beschreven in de Digikoppeling documentatie, wordt door JUBES ondersteund.

Naast deze Digikoppeling standaarden is er de afgelopen jaren een sterke push op een 'nieuwe' manier om informatie beschikbaar te stellen en wel via REST API's. Daardoor is er inmiddels ook het Digikoppeling Rest API profiel. Mede dankzij de inspanningen van het Kennisplatform API's zien steeds meer partijen de meerwaarde in het gebruik van deze alternatieve wijze van ontsluiting. REST API's zijn overigens geen nieuwe technologie; hoewel ze binnen de overheid nog maar beperkt gebruikt worden zijn ze gemeengoed in de commerciële wereld. REST API (OpenAPI Specification, REST API Design Rules) komt daarbij ook voor in lijst verplichte standaarden van het Forum Standaardisatie.

Waar tot op heden binnen JenV vooral gebruik werd gemaakt van de asynchrone Digikoppeling ebMS standaard waarbij informatie van ketenpartner naar ketenpartner werd overgedragen, is er de laatste tijd meer en meer behoefte aan synchrone real-time bevragingen. Mede daarom neemt de vraag naar REST API's ook binnen de overheid toe.

Een andere stuwkracht achter het toegenomen gebruik van REST API is de eenvoud ervan en de kennis ervan bij de nieuwe generatie van developers t.o.v. ebMS.

Op dit moment zijn daarom diverse onderdelen van Justitie en Veiligheid bezig met het ontwikkelen van hun API voorzieningen en wordt nagegaan hoe de gemeenschappelijke diensten de opkomst van REST API ondersteunen dan wel zouden kunnen ondersteunen. Dit betekent ook dat er actuele vragen zijn rond het gebruik van REST API. Daarop gaat dit document in.

1.1 Probleemstelling

Binnen JenV, maar ook met partnerorganisaties buiten JenV, worden berichten nu de facto uitgewisseld via JUBES. Dit is een architectuurafspraak waaraan alle JenV-Onderdelen aangesloten op JustitieNet zich aan houden. De vraag is: "geldt dit naast de Digikoppeling eBMS en WUS profielen ook voor het Digikoppeling Rest API profiel, of breder REST API uitwisseling in het algemeen?". De onzekerheid hierover kan JenV-Onderdelen onderdelen hinderen in REST API (door)ontwikkelingen, maar ook de doorontwikkeling van gemeenschappelijke diensten.

Verder geldt dat bij een toename van REST API, er ook behoefte ontstaat aan overzicht, vindbaarheid en het eenvoudig en gecontroleerd ontsluiten van deze REST API's. Dat kan worden gerealiseerd met API management, het onderwerp van dit document. De vraag hier is niet alleen hoe je dat kan invullen met API management op organisatie niveau, maar ook hoe je daar JenV breed invulling aan kan geven.

1.2 Doel

Het doel van dit katern is voorgenoemde onzekerheid weg te nemen, en invulling te geven aan de vindbaarheid en het eenvoudig kunnen ontsluiten van REST API's binnen JenV, richting JenV partnerorganisaties en de buitenwereld, door heldere architectuur afspraken te maken binnen een 'just in time-, just enough architecture' benadering.

1.3 Doelgroep

De doelgroep van dit document zijn de architecten, ontwerpers en overige personen die te maken krijgen met vraagstukken rond de inrichting van REST API landschappen en diensten.

1.4 Rijkwijdte

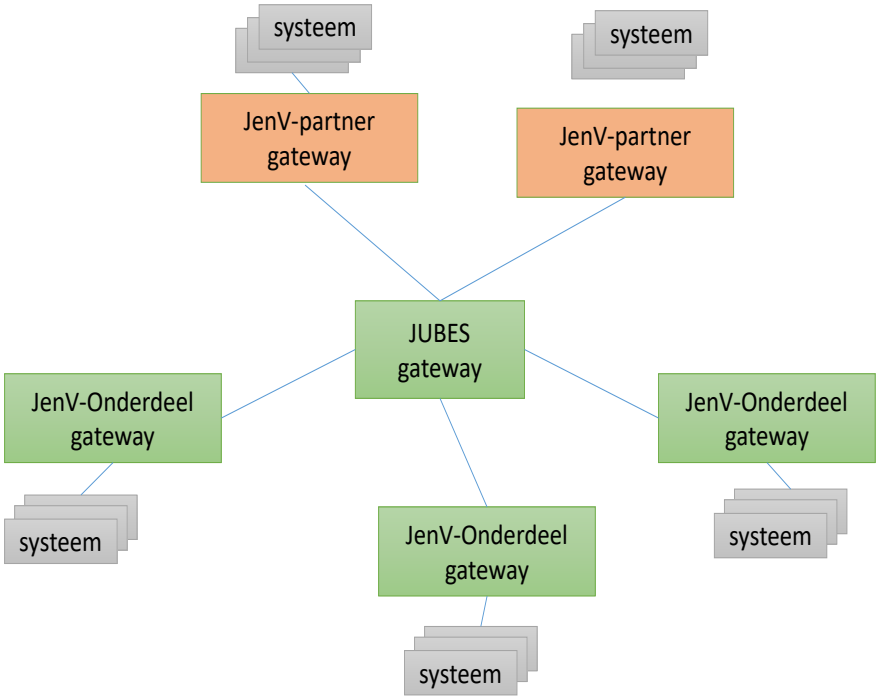
Dit document gaat in op de afhandeling van REST API verkeer al dan niet via gemeenschappelijke diensten. In de toekomst kan en zal het document waarschijnlijk worden uitgebreid als gevolg van de ‘just in time-, just enough architecture’ benadering.

1.5 Positionering

De ‘just in time, just enough’ architectuur is bedoeld om op tijd voldoende inhoudelijke richting te geven en daarover te kunnen besluiten. Kenmerkend is dat hoewel er veel meer over het onderwerp te schrijven of op te merken valt, wordt beperkt tot een kernvraagstuk. Zo is de benodigde snelheid te realiseren. Op deze wijze ontstaan losse katernen onder een domeinarchitectuur of zullen deze tezamen gaan vormen in de tijd. Dit katern maakt deel uit van domeinarchitectuur data- en gegevensdiensten.

2 Context

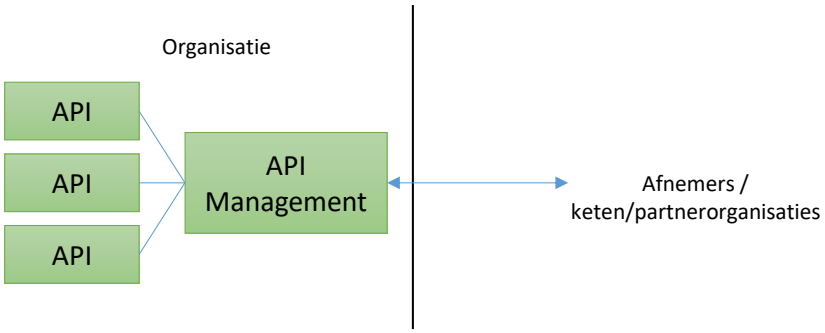
Op 11 april 2022 is de nieuwe versie van de Digikoppeling standaard formeel vastgesteld. Naast het REST API koppelvlak kent Digikoppeling WUS en ebMS koppelvlakken. De bulk van alle nu uitgewisselde berichten binnen, met en vanuit JenV zijn ebMS berichten. Deze worden uitgewisseld via het hub-spoke patroon zoals hieronder weergeven.



Figuur 1: Patroon WUS en ebMS berichtenuitwisseling via JUBES

Te zien is dat elke partij met andere partijen is verbonden via JUBES. JUBES vormt een centraal component in de berichten uitwisseling.

Het standaard patroon ten aanzien van REST API's is anders. Elke API is in principe op zich zelf te adresseren en te gebruiken via internet protocollen zonder tussenkomst van andere componenten. Wanneer er sprake is van meerdere API's dan wel een extra service en beveiligingslaag nodig is, wordt doorgaans een API Management oplossing ingezet. Via de API Management oplossing worden de API's dan ontsloten. Zie onderstaande figuur. De API Management oplossing zorgt dat op één plek o.a. toegang tot de REST API's kan worden verstrekt, informatie over deze API's kan worden gedeeld. Etc. (zie ook hoofdstuk 2.3.3 API Management).



Figuur 2: API Management patroon

2.1 REST API vs WUS en eBMS

Het Digikoppeling RESTful API profiel is van toepassing verklaard op best-effort[1] bevragingen/meldingen bij closed Government-naar-Government (G2G) verkeer. Zie onderstaande vergelijk met Digikoppling WUS en eBMS profielen.

Invulling	DK REST API profiel	DK WUS profiel	DK ebMS2 profiel
best-effort	V 1.0 of hoger	2W-be	osb-be
best-effort signed		2W-be-S	osb-be-s
best-effort signed/encrypted		2W-be-SE	osb-be-e
reliable			osb-rm
reliable signed			osb-rm-s
reliable signed en encrypted			osb-rm-e

Wanneer osb-rm gegarandeerde overdracht (once-and-only-once delivery) een vereiste is of indien aanvullende security maatregelen zoals osb-rm-s/e signing en encryptie nodig zijn dan zal voor WUS dan wel ebMS moeten worden gekozen. In alle andere gevallen kan ook voor REST API worden gekozen mits eventuele andere vereisten dat toestaan.

Merk op dat reliability vooral betrekking heeft op asynchroon[2] verkeer en van belang kan zijn in situaties waar sprake is instabiele en onbetrouwbare netwerkverbindingen[3]. Dit laatste komt vrijwel niet meer voor. REST API kent vaak een synchroon karakter en waar het een asynchroon karakter heeft wordt dit in de applicatie-laag afgevangen en niet in het protocol zelf zoals bij ebMS. Zie het volgende hoofdstuk voor hoe dit kan worden gedaan.

Merk op dat ook REST API signing en encryptie mogelijkheden kent. Deze zijn echter in de markt en ook binnen het Digikoppeling REST API profiel nog niet uitgekristalliseerd. Wel wordt er door het Kennisplatform API gewerkt aan deze aanvullende opties. Gegarandeerde overdrachtsaspecten zijn ook met REST API te realiseren, maar dan vanuit de applicatie laag (en niet binnen het protocol zelf). Zie ook het voorbeeld in hoofdstuk 2.2. Als probleem rond REST kan het grote aantal standaarden en methoden worden ervaren en het daarmee samenhangende uitzoekwerk.

2.2 Strategie, standaarden, specificaties en best practices

Doorgaans wordt rond REST API gebruik gemaakt van de volgende relevante bronnen, standaarden, specificaties en best practices:

a) De API Strategie voor de Nederlandse Overheid
b) De JenV API Visie en Strategie
c) De API Design Rules
d) Het NL-GOV profiel voor OAuth
e) De OpenAPI Specification
f) Het RESTful API profiel voor Digikoppeling
g) De OAuth 2.0 Security Best Current Practice
h) RFC 8725

Het aanhaken op de afspraken die overheidsbreed gemaakt worden, zie de API Strategie inclusief API Design Rules en Digikoppeling inclusief REST API profiel verhoogt de interoperabiliteit en vereenvoudigt de uitwisseling, ook over domeinen heen.

Daarnaast helpen API Design Rules ook om reliable aspecten toe te voegen bovenop het best effort karakter van REST zelf (vanuit de applicatielaag). Bijvoorbeeld door een POST op te volgen met een GET om daar de controle mee te kunnen doen. Dit type operaties moeten volgens de API

[1] Dit zijn uitwisselingen die geen faciliteiten voor betrouwbaarheid (ontvangstbevestigingen, duplicaateliminatie etc.) vereisen of waarbij dit in de applicatie-laag wordt afgevangen. Dit patroon wijkt af van “reliable messaging” waarbij asynchrone uitwisseling plaats vindt met ontvangst bevestigingen en duplicaateliminatie door de ontvangende message handler.

[2] Asynchroon verkeer betreft berichten waarop doorgaans niet direct een respons kan worden gegeven. Aan de kant van de ontvanger moet bijvoorbeeld eerst een verwerkingsproces worden doorlopen, waarvan de doorlooptijd op voorhand niet bekend is. In zo'n situatie is het wel van belang dat de verzender de zekerheid krijgt dat het bericht daadwerkelijk bij de beoogd ontvanger is aangekomen. Deze ‘reliability’ wordt standaard geboden door het ebMS2-protocol, maar kan ook door de applicatie-laag worden ingevuld.

[3] Asynchroon berichtenverkeer kan overigens ook waardevol zijn in situaties als het overbruggen van onderhoudwindows bij ketenpartners of bij verstoringen. De ‘retries’ zorgen er dan voor dat de berichten alsnog wordt afgeleverd.

Design Rules idempotent zijn, wat betekent dat indien een POST (andere operatie) meermaals wordt uitgevoerd dit niet zou mogen leiden tot een ander resultaat. Dus bij het uitblijven van een http OK zou de operatie gewoon nog een keer uitgevoerd moeten kunnen worden (retry).

2.3 Componenten in de keten

Om REST API's binnen JenV te kunnen ondersteunen zijn er verschillende componenten in de keten nodig, zowel centraal als decentaal:

Centraal:

1. API Routing Gateway, centrale voorziening voor routering en optioneel verificatie (van bijvoorbeeld tokens), breekt de end-to-end beveiliging niet.
2. API Catalog, centrale catalogus met gepubliceerde API's inclusief bijvoorbeeld gebruiksvoorwaarden, aanvraag proces, contactpersonen, e.d.
3. *API Management (optioneel), beheer en ontkoppeling van gepubliceerde API's, vormt een terminatie punt.*
    - *Hoewel voorzien wordt dat dit decentrale componenten zullen zijn bij de diverse organisatie onderdelen binnen JenV zou dit (mits er voldoende afname te verwachten is) ook als centrale component neergezet kunnen worden voor partijen die geen eigen oplossing willen (of kunnen) gaan neerzetten.*

Decentraal:

1. API Management (core component), beheer en ontkoppeling van gepubliceerde API's, vormt een terminatie punt. Decentrale voorziening aan de rand van het landschap van een service provider.
2. Developer Portal, publiceert API informatie en API verificatie en test mogelijkheden. Self service omgeving voor onder andere developers en architecten van service consumers. Direct gekoppeld aan de API Management voorziening.

2.3.1 API Routing Gateway

Centrale voorziening om REST API verkeer te kunnen routeren. De API Routing Gateway kent een optioneel karakter indien service consumers andere JenV onderdelen zijn (interne publicatie), maar een verplicht karakter bij service consumers buiten JenV (externe publicatie). De reden hiervoor is dat bij externe publicatie sprake is van een overgang tussen vertrouwenszones en hiervoor een protocol break in JUBIT vereist is.

2.3.2 API Catalogus

Een centrale voorziening waarop informatie over de binnen JenV beschikbare REST API's te vinden is. Het is daarmee een verwijsindex en een initieel punt waarop potentiële afnemers terecht komen. API specificaties en aanvullende documentatie worden hierbij opgenomen en tevens wordt doorverwezen naar de specifieke (de)centrale API voorzieningen waarop de REST API beschikbaar is.

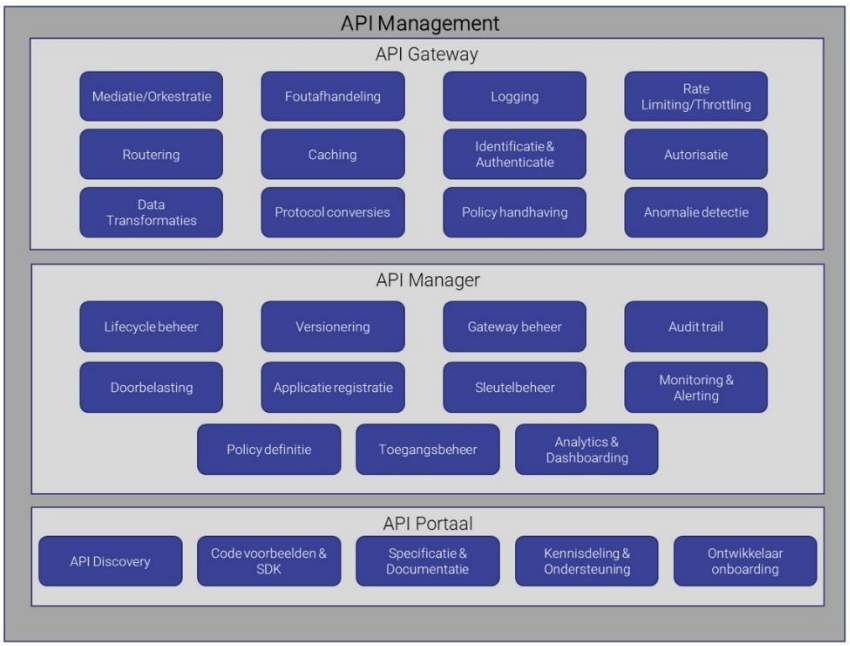
2.3.3 API Management

API Management kan als functie worden gezien en een API Management oplossing als middel om die functie in te vullen. In meer detail kan een API Management oplossing het volgende bieden:

- Generieke koppelwijze welke voorkomt dat per koppeling netwerk aanpassingen nodig zijn
- API repository (API catalog / Service Registry) functionaliteit
- Developer portal functionaliteit
- Beheer mogelijkheden voor doorontwikkeling van API's en het regelen van de versioning en het life cycle management
- Functionaliteit voor
    - Mediatie/orkestratie
    - Identificatie, authenticatie en autorisatie
    - Routering
    - Throttling
    - Flows
    - Fout afhandeling
    - Policies
    - Veilige toegang
        - OAuth

        - Content-based security
        - TLS-MA[4]
    - Fout afhandeling / anomalie detectie
    - Data transformaties
    - Logging, monitoring en alerting
- Selfservice functionaliteit zodat aanbieders van API's zelf het beheer kunnen doen op hun API's, waaronder deployment van API's, updates/upgrades, versiebeheer, toegangsregel-beheer, monitoring

Zie ook de beschrijving in de Nederlandse API Strategie waarin ook onderstaande figuur is opgenomen.



Figuur 3: API Management onderdelen en functies (bron: Nederlandse API Strategie)

2.3.4 *Developer portal*

Een goed developer portal neemt de doelgroep[5] mee door het API landschap en laat duidelijk zien welke dienstverlening en welke informatie via API's ontsloten wordt. Met daarbij de volledige API-beschrijving zichtbaar en met gekoppelde voorbeeldservices zodat ontwikkelaars direct en zonder tussenkomst van de service provider de verkenning kunnen aangaan.

Dit eerste deel van het 'onboarding' proces vindt daarmee plaats zonder tussenkomst van de service provider en biedt service consumers de vrijheid om naar eigen inzicht de te verkrijgen informatie te beoordelen waarbij na deze eerste beoordeling contact kan worden gezocht met de service provider over de verdere afhandeling. En dan niet ten aanzien van de technische specificaties van de service, die staan op het developer portal omschreven, maar wel ten aanzien van zaken als toegang, privacy aspecten, financiële aspecten en afspraken.

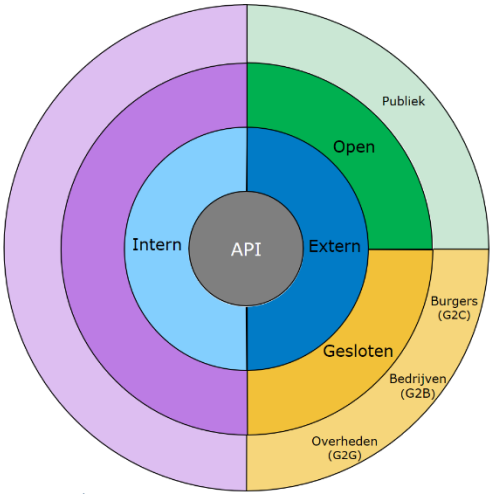
2.4 G2G, G2B, G2C

JenV heeft in de regel te maken met gevoelige gegevens (closed API's), maar er zal ook sprake kunnen zijn van open API's.

In de JenV context komt veelvuldig het government to government (G2G) patroon voor en het government to business (G2B) patroon. In mindere mate komt het government to citizen (G2C) patroon voor, maar dit volume kan groeien. Idealiter is de API management oplossing daarom geschikt voor al deze vormen.

[4] Mutual TLS (ook wel TLSMA) vereist naast een server certificaat en public key voor identificatie van die server (TLS) datzelfde van de client (TLS-MA)

[5] dit kunnen ontwerpers of ontwikkelaars zijn, maar ook business consultants of architecten



Figuur 4: API vormen, bron (Nederlandse API Strategie)

Merk op dat in relatie tot de API vormen die de Nederlandse API Strategie onderscheid, zie bovenstaande figuur, JenV ook een JenV interne variant kent naast een JenV-Onderdeel interne variant. Deze JenV-interne variant is ook van het type 'gesloten' en te typeren als G2G.

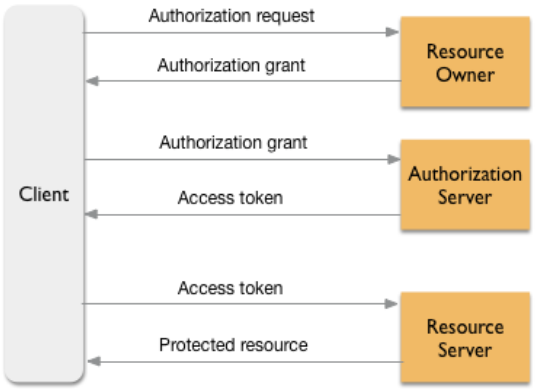
2.5 Authenticatie en autorisatie

Er zijn in hoofdlijnen drie opties om afnemers van REST API dienstverlening te kunnen autoriseren:

1. API keys, voor toegang tot niet of minder gevoelige data.
2. Mutual TLS (TLS-MA), identificatie op basis OIN (Organisatie Identificatie Nummer) en client certificaten. Dit is (voor nu) de standaard die gehanteerd wordt binnen het REST API profiel van Digikoppeling.
3. OAuth 2.0 en OIDC, dit zijn de standaarden die in de rest van de wereld normaliter voor de autorisatie op REST API's gebruikt worden.

Er is hier sprake van system2system koppelingen, gebruik het JenV-Trustframework bij het bepalen van de minimale vereisten voor het betrouwbaarheidsniveau (LAAG, SUBSTATIONEEL, HOOG) en maken van afspraken hierover.

Hieronder staan kort de begrippen in de OAuth context. Mede omdat we verwachten dat OAuth in de JenV-interne context en richting partnerorgansiaties (G2B), zie ook vorige hoofdstuk, steeds meer gebruikt zal worden.



Figuur 5: OAuth begrippen

Client	App of web App
Resource owner	Eindgebruiker, doorgaans de persoon die toegang tot een resource kan geven

Resource server	Denk aan services zoals een interne website, Facebook, Twitter etc. API Management is een resource server als token validatie nodig is
Authorization server	API Management kan deze rol invullen en is verantwoordelijk voor de validatie van authorization grants en de issuing van access tokens welke de app (client) toegang geeft tot de data van de user (resource owner) op de resource server
Authorization grant	Geeft de app (client) toestemming om een access token op te halen namens de user (resource owner)
Access token	Credential welk wordt gebruik om toegang te krijgen tot protected resources
Protected resource	Data van de data owner (op de resource server)

Bij het "client credential" grant type is de app (client) typisch ook de resource owner.

Vanuit de API Management voorziening zijn het valideren van een client en het genereren van tokens onafhankelijke functies. Een API Management voorziening kan beide afhandelen, de één of de ander afhandelen of beide niet afhandelen. API Management kunnen vaak externe IDP's gebruiken en externe token leveranciers, maar ook de equivalenten daarvan binnen de API Management oplossing zelf.

2.6 Grant types

Voor G2G/G2B zal doorgaans het grant type "OAuth 2.0 Client Credentials Grant Flow" worden toegepast. In het geval G2G conform het RESTful API profiel voor Digikoppeling, identificatie op basis OIN (Organisatie Identificatie Nummer) en certificaten. Voor G2C zal doorgaans het "OAuth 2.0 Authorization code flow (with PKCE)" grant type worden toegepast.

Opmerking: Het RESTful API profiel voor Digikoppeling wordt alleen bij G2G gebruikt, niet bij interne communicatie binnen JenV of communicatie met bedrijven, stichtingen en dergelijke, hoewel dit wel kan als beide partijen dat overeenkomen.

Het "OAuth 2.0 Implicit" grant type wordt (bij voorkeur) NIET toegepast gelet security concerns gerelateerd aan dit grant type. Idem voor de legacy "OAuth 2.0 Resource Owner password Credentials" grant.

Voor specifieke gevallen kan het "OAuth 2.0 Device Authorization Grant" type gebruikt worden. Voor de refresh van access tokens kan het "OAuth 2.0 Refresh Token" grant type worden toegepast.

2.7 JWT vs betekeningsloze tokens

JSON Web Token (JWT's) kunnen gebruikt worden in combinatie met bijvoorbeeld OAuth2 en worden toegepast voor verschillende doeleinden waaronder de overdracht van informatie, identiteiten en entitlements. OAuth daarentegen is autorisatie protocol. In JWT tokens kan zich informatie bevinden waarvan het onwenselijk is dat die inzichtelijk wordt voor personen of instanties waarvan dat niet de bedoeling is. Threats and Vulnerabilities geleerd aan JWT zijn o.a. beschreven in RFC 8725. Het inzichtelijk worden voor ongeautoriseerden kan worden voorkomen door TLS-MA te gebruiken, de JWT best current practices (RFC 8725) toe te passen en bijvoorkeur beide.

Een alternatief kan het gebruik van betekeningsloze tokens zijn. Dit kan echter aanzienlijke nadelen en beperkingen opleveren. Daarnaast is JWT min of meer de facto standaard om te gebruiken in deze context.

2.8 Risico's

De volgende generieke risico's rond REST API verkeer en API management worden onderkend. Om het document leesbaar en compact te houden, zijn hier tevens maatregelen opgenomen. Een aantal daarvan komen uit navolgende hoofdstuk, zie voor de achtergrond dat hoofdstuk.

Risico	Maatregel	Kans (na maatregel)	Impact (na maatregel)
Via REST API verkeer wordt malware ontvangen	Content scanning (Jubes External REST API Gateway)	Laag	Hoog
REST API verkeer (inhoud) kan onderweg worden gelezen door onbevoegden	TLS(-MA)	Laag	Hoog

Onbevoegden krijgen toegang tot de REST API	Authenticatie/autorisatie in combinatie met TLS(-MA)	Laag	Hoog
REST API zorgt voor versnippering landschap	API Catalog	Midden	Laag[6]
Implementatie en configuratiefouten	Toepassing de best practices genoemd in dit document, naast generieke best practices	Laag	Hoog
Uitlezen JWT tokens met informatie door onbevoegden	TLS-MA gebruiken, de JWT best current practices (RFC 8725) toe te passen, of betekeningsloze tokens toepassen	Laag	Hoog
Informatie is zichtbaar door Justid (scanning E2E dataverkeer)	Justid als trusted party inzetten en afspraken, contracten en regie daarop inrichten	Laag	Laag
Onbedoeld gebruik zwakke authenticatie methoden	Gebruik JenV Trustframework	Laag	Hoog

Dit is een generieke analyse van risico's in het kader van de architectuurrichting. Partijen die REST API's toe gaan passen en API management oplossing gaan realiseren zullen altijd ook zelf een risicoanalyse moeten doen (en al het andere vanuit de standaard processen zoals rond IB Quickscan, DPIA's, e.d.).

2.9 Best practices

API's worden gebruikt voor ontkoppeling en kan gebruikt worden bij service oriëntatie[7], hierbij kunnen de volgende best practices worden gehanteerd:

- Stel business functionaliteit beschikbaar via het REST resource model
- Optimaliseer het aanbieden van de API's zo dicht mogelijk bij de data
- Streef naar eenvoud en voorkom (ESB stijl) orkestratie in de API Management laag
- Doe authenticatie en autorisatie zo dicht mogelijk bij de data, laat dit doen door de data eigenaar/data verantwoordelijke dan wel een door hem/haar gedelegeerde partij

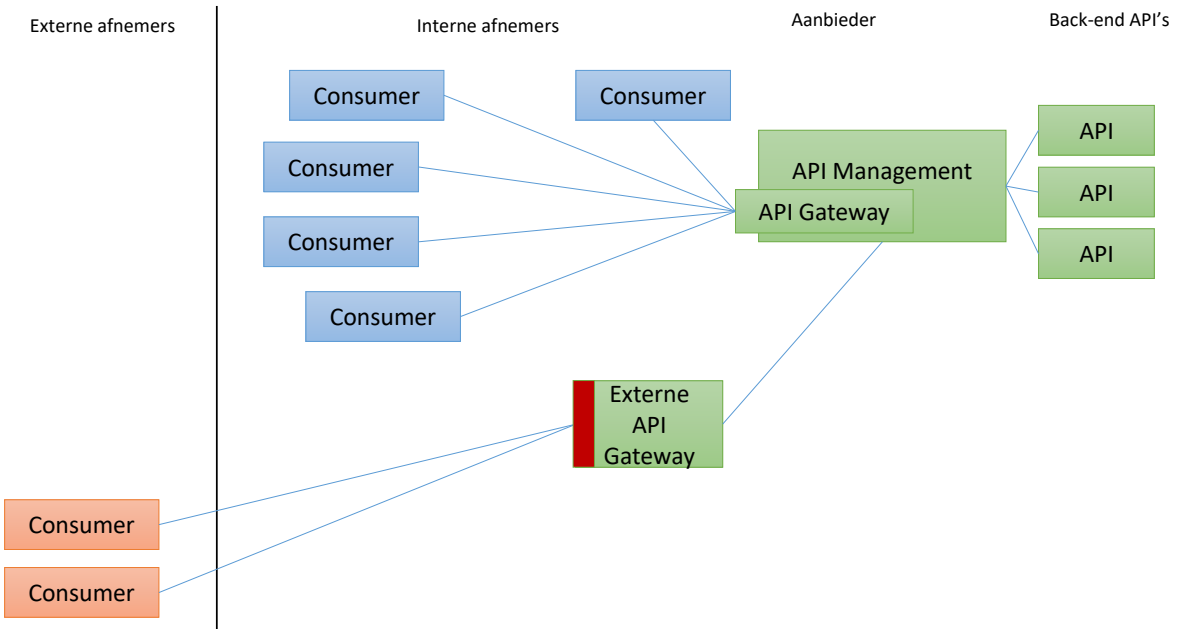
[6] Impact t.a.v. informatiebeveiligingsaspecten is laag, deze kan hoger zijn vanuit een functioneel oogpunt

[7] REST API (t.b.v. applicatie/service communicatie / uitwisseling gegevens) kan gebruikt worden bij service oriëntatie wat een architectuur en ontwerp benadering betreft, net als microservices dat eveneens een architectuur en ontwerp benadering betreft en waar REST API's eveneens een belangrijk rol kunnen spelen.

3 REST API koppelingen

3.1 Algemene patronen

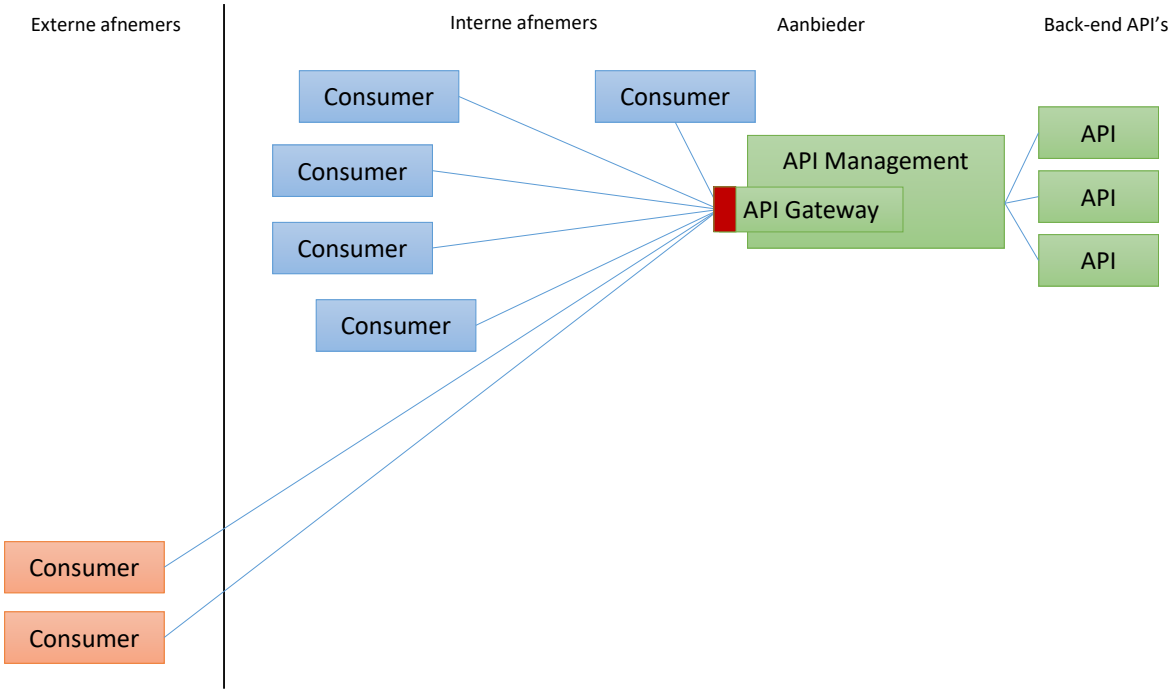
Een algemeen gangbare manier is om API's te ontsluiten via een API Management oplossing zoals hieronder geschetst.



Figuur 6: API Management met interne en externe gateway

Naar de JenV situatie vertaald worden JenV partijen aangesloten via JustitieNet op de interne API Gateway en externe partijen / partnerorganisaties op de externe API Gateway. Dat kan een manier zijn om onderscheid te kunnen maken tussen JenV interne en externe partijen. Zo kan onderscheid gemaakt worden in bijvoorbeeld security maatregelen.

Een andere optie is om alle partijen op een zelfde manier te benaderen, zoals te zien in onderstaande figuur.

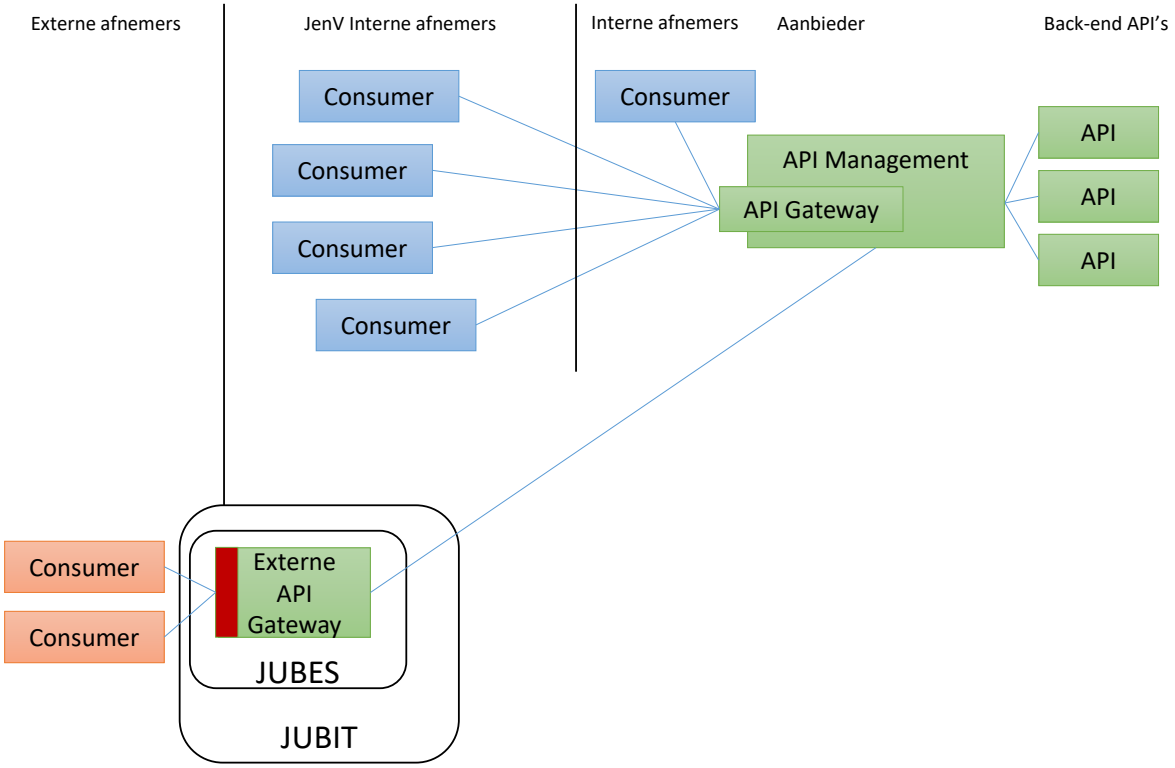


Figuur 7: API Management met een gateway voor zowel interne als externe partijen

Alle partijen worden op eenzelfde manier benaderd, namelijk als externe partijen. Dit kan leiden tot een eenvoudiger opzet en een eenduidigere afhandeling.

3.2 JenV situatie

Vertaald naar de JenV situatie komt er een dimensie bij. Interne afnemers zijn onder te verdelen in de interne afnemers binnen het JenV-Onderdeel zelf en interne afnemers vanuit JenV (andere JenV-Onderdelen aangesloten op Jusitienet). Verder is het zo dat JenV alle dataverkeer met externe partijen via Jubit laat verlopen en bij inkomend dataverkeer een inspectie doet of de inhoud veilig is. Voor berichtenverkeer wordt dat gedaan door JUBES. Dat ontstaat de volgende situatie, dus de eerst geschetste situatie in het vorige hoofdstuk met een interne en externe gateway.

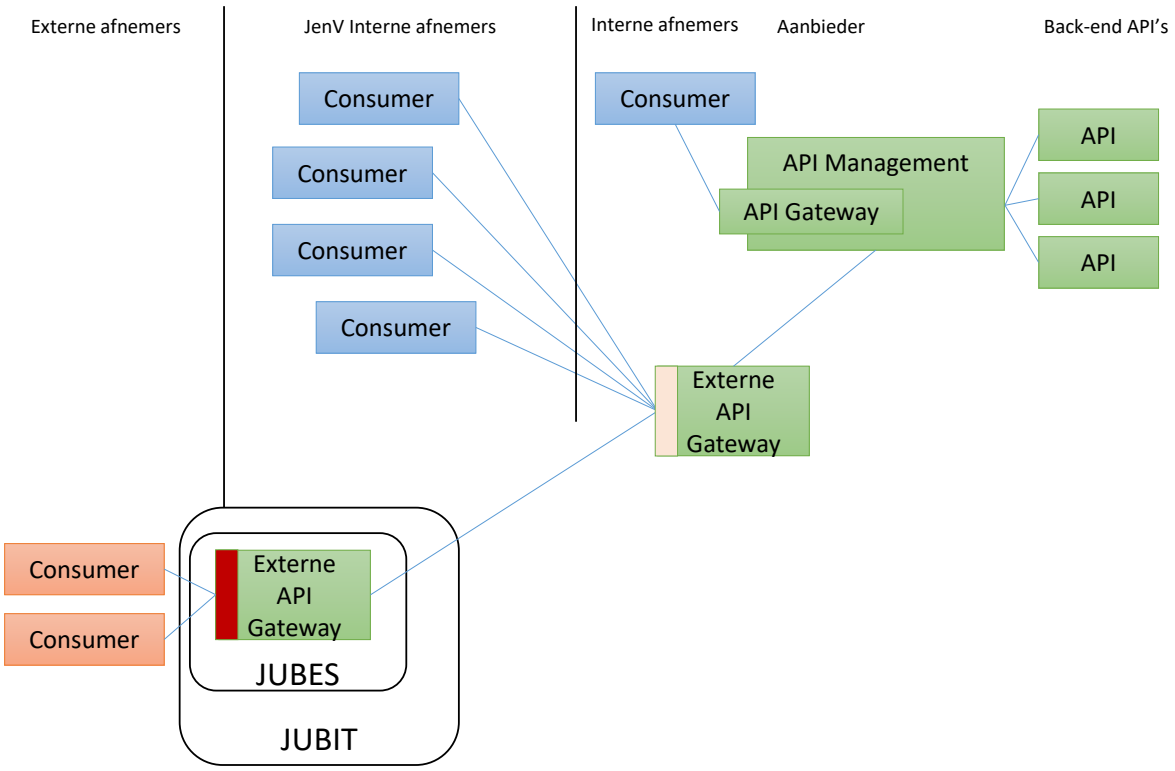


Figuur 8: JenV API Management landschap

De JUBES Externe Gateway ontkoppeld op TLS-MA[8] niveau. Inspecteert het berichtenverkeer op onder andere inhoud (malware e.d.) en protocollen op bedreigingen waaronder OSWAP aspecten. De JUBES Externe Gateway functioneert in passthough mode t.a.v. authenticatie, autorisatie en token afhandeling. Dat betekent dat deze aspecten door de API Management oplossing van het JenV-Onderdeel worden afgehandeld.

JenV-Onderdelen staat vrij om achter de JUBES Externe Gateway al dan niet een eigen Externe Gateway te plaatsen voor afhandeling van REST API verkeer met externe partijen en/of JenV interne partijen. Zie navolgende figuur.

[8] Mutual TLS (ook wel TLSMA) vereist naast een server certificaat en public key voor identificatie van die server (TLS) datzelfde van de client (TLS-MA)



Figuur 9: JenV API Management landschap met chained gateways

Wat deze benaderingen gemeen hebben is dat interne afnemers aansluiten op de API Management omgeving van het JenV-Onderdeel zonder tussenkomst van een JenV centrale API Management component (in tegenstelling tot de ebMS/WUS aanpak). Dit is een wenselijke situatie omdat:

1. Dit aansluit op de manier waarop de rest van wereld doorgaans met REST API's omgaat
2. De end2end connectie en daarmee encryptie niet doorbroken wordt
3. Een REST API serviceprovider zelfstandig kan vaststellen wie client is[9]
4. Extra componenten in de technische keten alleen worden ingezet als dat baten oplevert

Voor externe afnemers wordt wel een JUBES-component gebruikt, namelijk de JUBES Externe REST API Gateway. De decentrale API Management oplossing echter blijft bij het JenV-Onderdeel als ook het zelfstandig kunnen vaststellen wie client is (gelet de eerder benoemde pass through mode). Wel breekt de Externe Gateway de TLS-MA[10] (anders is inspectie niet mogelijk). Daarmee wordt punt 2 zoals hiervoor genoemd geraakt. Dit is echter geen probleem omdat:

a) Dit niet anders dan voor de huidige externe ebMS/WUS koppelingen, waar dit geen probleem is en er veel van zijn en al jaren in gebruik zijn.
b) De authenticatie, autorisatie en token afhandeling door het JenV-Onderdeel gedaan wordt (pass through mode)[11].
c) De terminatie door het JenV-Onderdeel wordt belegd bij de beheerder van JUBES wat niet afwijkt t.o.v. het beleggen van beheer van een API-Management oplossing bij een (externe) beheerpartij, maar wel (contractuele) afspraken kan vergen.

Merk op dat dit gaat over het aanbieden van API's door JenV-Onderdelen. Afnemen van API's aangeboden door externe partijen verloopt via de reguliere connectiviteitspaden en dus Jubit inclusief TLS interception. Daar waar TLS interception niet kan of mag plaatsvinden, bijvoorbeeld voor Stg. Confidentieel berichtenverkeer, kunnen mogelijk uitzonderingen op interception worden aangevraagd (exception proces).

[9] Door TLS-MA validatie en/of authenticatie/autorisatie (OAuth)

[10] Mutual TLS (ook wel TLSMA) vereist naast een server certificaat en public key voor identificatie van die server (TLS) datzelfde van de client (TLS-MA)

[11] Hier kan een issue ontstaan als de authenticatie en autorisatie uitsluitend op basis van de TLS-MA gebeurt. Omdat de TLS-MA wordt verbroken krijgt de service provider niet langer het client certificaat van de service consumer maar die van JUBES. Om dat op te lossen kan JUBES de oorspronkelijke certificaat gegevens (of zelfs het volledige certificaat) in een custom http-header mee geven.

3.3 Gateway vs Authorisation Server en Resource Server

Resource servers leveren de betreffende REST API's. De Authorisation Server, ook wel Token Server, genoemd levert de access tokens en wordt gebruikt om wel of geen toegang te verlenen tot de REST API. De API management oplossing kan een Authorisation Server bevatten of een externe Authorisation Server gebruiken. Via een Gateway component kan extern verkeer naar de interne REST API's worden gekanaliseerd en gecontroleerd.

Gateways kunnen interacteren met Authorisation Servers. De technische toegangsverlening afhandeling wordt dan feitelijk verplaatst naar de Gateway. We kiezen er expliciet voor dit niet toe te passen op de JenV externe Jubes REST API gateway. Dit zodat de verantwoordelijkheid voor de toegangsverlening bij het JenV-Onderdeel blijft liggen en om onnodige complexiteit te vermijden. De JenV externe Jubes REST API gateway kijkt daarmee alleen in de TLS(-MA) verkeerstroom en zet die daarna weer door met een TLS(-MA). Daarbij kan de gateway informatie over de originele client certifcates doorleveren.

3.4 End2end encryption

Er is een steeds sterkere trend naar end2end ongebroken connecties tussen aanbieders en afnemers. De TLS-MA break als gevolg van de JUBES Externe REST API Gateway kan vanuit die optiek als onwenselijk gezien worden tot voor bepaalde use cases mogelijk als onacceptabel. Er zijn hiervoor twee oplossingsrichtingen:

1. Gebruik van payload signing en encryption
2. Gebruik van een eigen Gateway binnen JUBIT en het zelf implementeren van de benodigde beveiligingsmaatregelen

<u>Payload signing en encryption</u>

Dit terrein is nog niet geheel ontgonnen voor REST API en voegt een potentieel onnodige complexiteit toe. TLS-MA inclusief sterke authenticatie/autorisatie (OAuth2) is voor veel use cases meer dan voldoende sterk. Toch zien we de tendens dat veel JenV-Onderdelen interesse hebben in payload signing en encryption. Daarmee is dan de end2end encryptie eis/wens in te vullen. Stel dat dit gemeengoed zou worden, dan voegt de beveiliging van de JUBES Externe REST API Gateway overigens weinig meer toe. De vercijferde payload is immers op dat punt niet te controleren. Dat zal het JenV-Onderdeel zelf moeten doen. Daarmee zou de vreemde situatie ontstaan dat er een centrale en decentrale component is, beide met dezelfde functie, en waarbij de centrale functie nauwelijks meerwaarde levert rond malware detectie (zien en tegenhouden van schadelijke content in berichten) omdat deze niet in de payload kan kijken.

<u>Beveiliging zelf regelen met eigen Gateway binnen JUBIT</u>

Ongebroken end2end encryptie betekent dat het dataverkeer onderweg niet ingezien kan worden en daarmee ook niet kan worden geïnspecteerd op schadelijke of risicovolle content. Dit heeft als gevolg dat de ontvangende organisatie zelf deze inspectie zal moeten uitvoeren. Een andere optie dat de ontvangende organisatie de expliciete keuze maakt geen payload content-inspectie te doen als sprake is van een vertrouwde partij en/of uitwisseling van (tekst)berichten waarin de kans op schadelijke of risicovolle content uiterst gering tot nihil is, en de ontvangende organisatie de restrisico's hiervan accepteert. Verder zal bij uitwisseling met externe partijen doorgaans sprake zijn van een zoneovergang[12] 'onvertrouwd–naar–vertrouwd' en het JenV-Onderdeel technische- en/of organisatorische maatregelen moeten treffen om een dergelijke overgang mogelijk te kunnen maken. Dat betekent ook dat de Gateway gepositioneerd zal moeten worden in JUBIT en binnen de JUBIT-hosting zone van het betreffende JenV-Onderdeel.

Belangrijk: Deze laatste optie valt binnen JenV vooralsnog niet binnen een als erkend toepasbaar architectuur patroon. Uitgangspunt is dat berichtenverkeer met externe partijen, ongeacht het protocol (WUS/ebMS/REST), via JUBES verloopt en daar wordt geïnspecteerd. De beveiliging zelf regelen met een eigen Gateway binnen JUBIT kan nu dus nog niet (tenzij als exceptie goedgekeurd, comply or explain principe). Waarbij opgemerkt dat JUBES is goedgekeurd voor gebruik van DepV verkeer en dat exceptie dus enkel aan de orde zou moeten zijn indien het te verwachten verkeer dit rubriceringsniveau overstijgt. Een comply or explain is vanuit zowel het JenV Architectuur Forum als CISO-board geadviseerd. Zij onderschrijven daarmee "*inspectie binnen tenzij*".

Concreet is te stellen:

[12] Zie het JenV zoneringsmodel en -architectuur voor meer informatie

Gebruik de JUBES externe Gateway voor inspectie (comply), tenzij er sprake is van een situatie die een 'explain' rechtvaardigt. Dan zijn er de volgende opties: a) Gebruik de Jubit3 reversed proxy met het het REST API profiel (ook een protocol break oplossing) b) Zorg zelf voor protocol en content inspectie

Voor veel use cases zal payload signing en encryption niet nodig zijn, er is immers al TLS(-MA)[13]. Payload signing en encryption zal daarom voor reguliere berichtenuitwisseling naar verwachting niet vaak worden toegepast. Als dit beeld echter gaat kantelen zullen JenV-Onderdelen sowieso zelf de beveiliging moeten gaan regelen. De payload kan dan immers centraal niet meer worden gecontroleerd op schadelijke elementen. Dat zal de waarde van de JUBES Externe REST API Gateway verwateren en voorgenoemde lijn (berichtenverkeer met externe partijen altijd via JUBES) mogelijk worden aangepast.

Advies: Anticipeer op de mogelijke toename van payload signing en encryption door in de API-Management oplossing capabilities te realiseren voor het zelf beveiligen van berichtenverkeer en het kunnen maken van zone-overgangen daarvoor. Haak hierbij aan op de ontwikkelingen in het Kennisplatform API's en in het Digikoppeling REST API profiel waar gewerkt aan het gestandaardiseerd kunnen toepassen van signing en encryption binnen dat profiel.

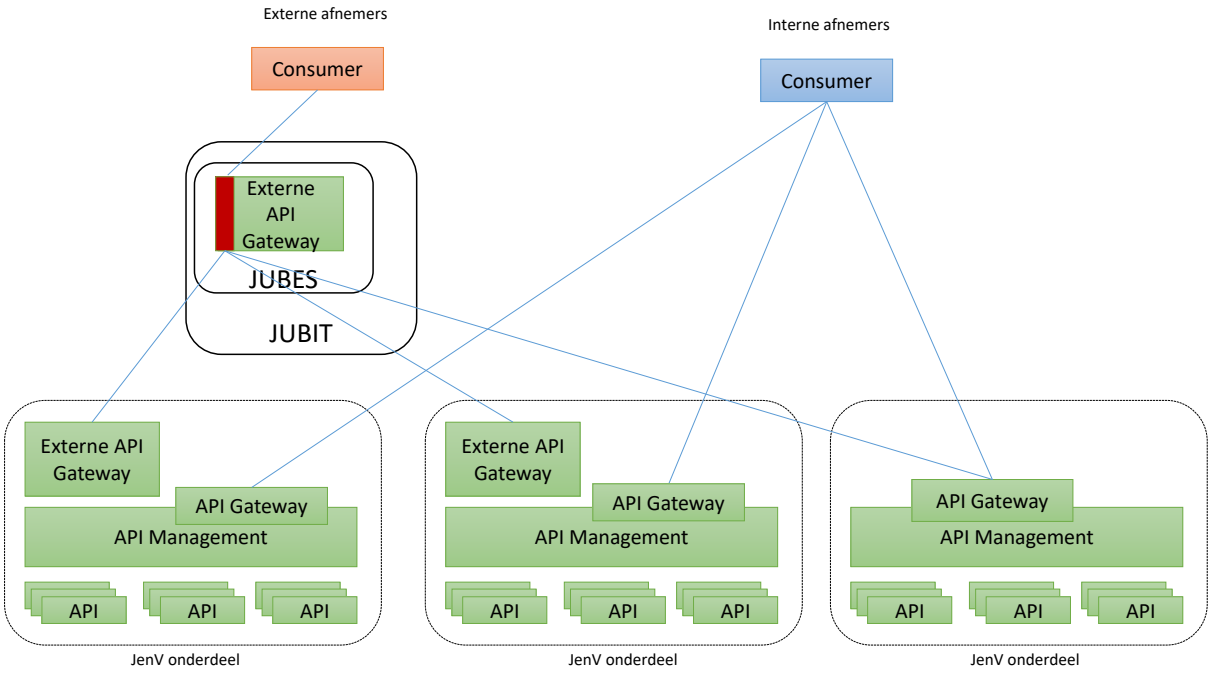
3.5 API Management in gedistribueerde landschappen

Er zijn partijen met REST API's in verschillende datacenters en/of clouds. In de regel geldt dat het wenselijk is een API Management oplossing dicht bij de REST API's te plaatsen. Partijen kunnen ervoor kiezen dit te doen en krijgen in dat geval meerdere API management oplossingen. Bijvoorbeeld één on-premises en één in de cloud. Deze kunnen dezelfde JenV externe Jubes REST API gateway gebruiken.

Het gebruik van meerdere API Management oplossingen leidt doorgaans ook tot meerdere Development Portals en API's Catalogs. Een organisatie zou ervoor kunnen kiezen een organisatie centrale API Catalog te realiseren. Zodat er één plaats is, waar alle via API Management ontsloten API's binnen de organisatie te vinden zijn.

3.6 Consequenties

Beredeneerd vanuit voorgaande ontstaat een REST API landschap zoals hieronder geschetst.



Figuur 10: API Management landschap JenV

[13] Dat weliswaar gebroken wordt op de externe gateway, maar dat is binnen het beveiligde domein. Verder wordt van de gateway naar de achterliggende bestemming opnieuw een TLS(-MA) verbinding opgezet.

Er ontstaan daarbij enkele problemen en ongemakken t.o.v. het WUS/eBMS model met een JUBES als centrale component:

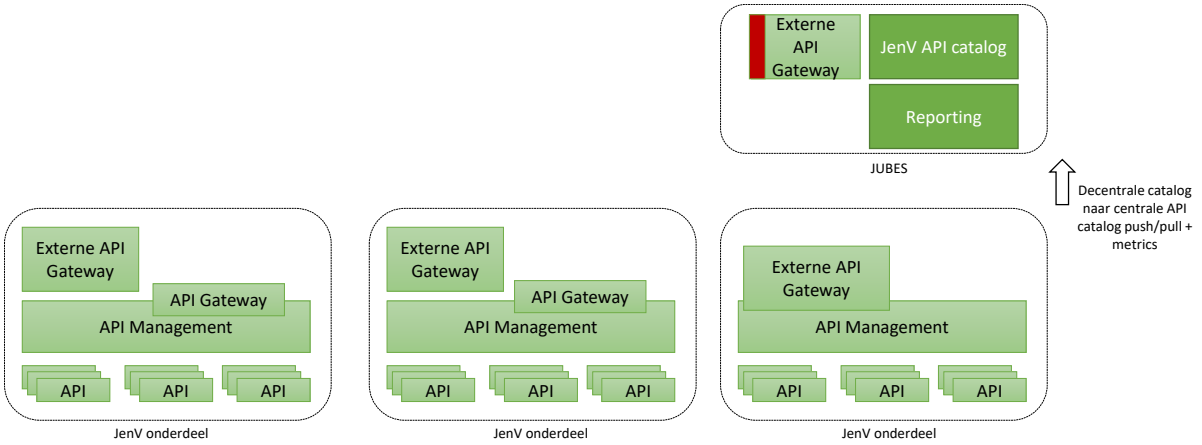
a) Er is geen centrale broker-partij waar organisaties terecht kunnen als ze connecties met anderen willen maken
b) Er ontstaat een versnippert landschap van API aanbieders, het vinden van API's wordt een zoektocht over de verschillende partijen heen
c) De ontzorging welke JUBES nu levert externe/partnerorganisaties op Internet of Diginetwerk mist als optie.
d) Er is geen JenV breed inzicht meer op berichtenuitwisseling zoals het aantal uitgewisselde berichten en het centraal overzicht op ketenconnecties verdwijnt.

a en b
Hiervoor zijn echter oplossingen. Probleem a+b is deels op te lossen voor een centrale JenV API Catalogus/Repository te realiseren als onderdeel van JUBES. De decentrale API Management oplossingen en de decentrale API Repositories kunnen hun inhoud synchroniseren met de JenV API Catalogus/Repository. Feit blijft dat partijen zelf meer moeten gaan doen op het moment dat JUBES er niet meer tussen zit (voor JenV intern verkeer).

c1
Probleem c is al (deels) opgelost. JUBES levert namelijk reeds een JUBES Externe REST API Gateway. Partijen hoeven daardoor niet zelf alle connecties, certificaten, routering en firewall issues op te lossen oplossen bij het koppelen met een nieuwe externe partij.

d
Probleem d is op te lossen door metrics te kunnen aanbieden in een vorm (een REST API!) die JUBES kan afnemen om daaruit rapportages te generen (indien dat wenselijk of nodig is dan wel baten oplevert).

Dan zou een volgende situatie ontstaan.



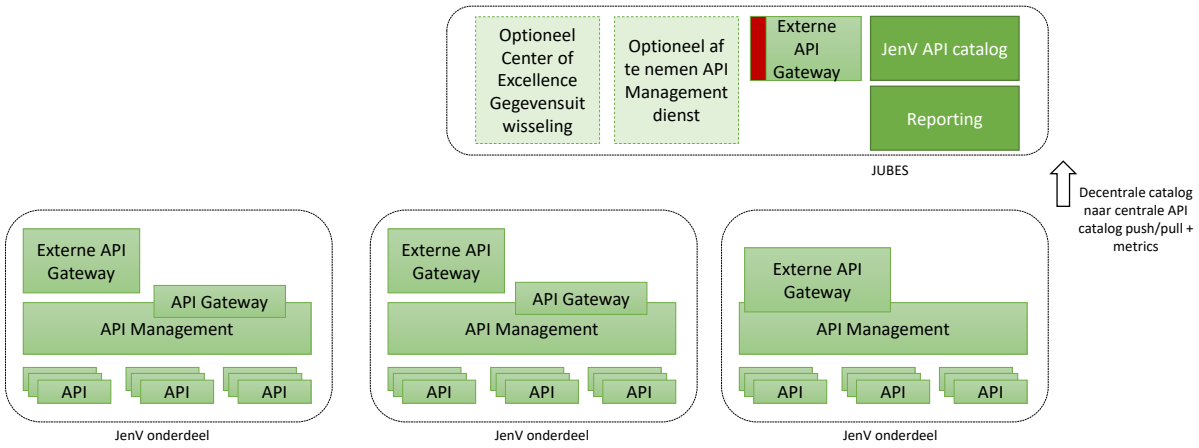
Figuur 11: JUBES REST API diensten

c2
Mogelijk zijn er echter ook JenV-Onderdelen die niet zelf een API Management oplossingen willen implementeren en onderhouden, maar deze als dienst af willen nemen. Deze zou geboden kunnen worden in de vorm van een gemeenschappelijke dienst bestaande uit een multi-tenant API Management oplossing, optioneel af te nemen door partijen die dat wensen. De behoefte hiernaar zou kunnen worden verkend en afhankelijk van de uitkomsten daarvan opgevolgd kunnen worden door een dienst ontwikkeltraject.

c3
Tot slot zien we dat veel organisaties met REST API aan de slag gaan of willen gaan. Kennisdeling en kennisopbouw kan voor JenV en JenV-Onderdelen aanzienlijke voordelen bieden. Het valt te daarom te overwegen een Center of Excellence Gegevensuitwisseling op te zetten. Wellicht zou dit overigens anders genoemd moeten worden, REST API's kunnen immers ook worden toegepast bij proces integratie en dergelijke waardoor de term gegevensuitwisseling feitelijk te smal is (Center

of Excellence API Management zou wellicht beter zijn). Deze functie zou gepositioneerd kunnen worden bij Justid welke als operationeel beheerpartij van JUBES deze rol al enigszins invult.

Het complete plaatje wordt dan als volgt.



Figuur 12: JUBES REST API diensten en optionele diensten

Een andere consequentie is dat (externe) partijen die meerdere REST API's bij meerdere JenV-Onderdelen voor één functionele bevraging moeten benaderen, dat afzonderlijk moeten doen en het resultaat zelf moeten combineren. Een andere benadering zou die zijn van een JenV 'shared resource server' die bevraagd kan worden en zelf het gecombineerde antwoord samenstelt en levert. Dat zou een andere centrale component (onderdeel gemeenschappelijke dienst) kunnen opleveren. De wenselijkheid hiervan en de technische mogelijkheden hier rondom zijn echter nog niet voldoende duidelijk.

3.7 Architectuurafspraken

Voorgaande levert de volgende architectuurafspraken op:

AA 1- JenV-onderdelen op Justitienet kunnen elkaars REST API omgeving en daarmee API's direct en zonder tussenkomst van JenV centrale componenten zoals JUBES benaderen.

AA 2- JenV-onderdelen ontsluiten REST API's naar externe- en partnerorganisaties via de JUBES Externe REST API Gateway.

AA 3- Bij end2end (payload) encryptie van REST API koppelingen dienen JenV-onderdelen invulling te geven aan zoneovergang- (bijvoorbeeld onvertrouwd-naar-vertrouwd) maatregelen waaronder potentieel content-scanning.

AA 4- JenV-onderdelen publiceren en synchroniseren hun relevante API-metadata naar de JenV API Catalog/Repository.

AA 5- JenV-onderdelen zorgen dat zij metrics voor afname door JUBES, en potentieel andere afnemers, kunnen publiceren, zodat daarmee waardevolle overzichten en inzichten kunnen worden gecreëerd rond REST API gebruik binnen JenV.

3.8 Dienstontwikkeling

Op benoemde architectuurafspraken kunnen de JenV-onderdelen hun (door)ontwikkeling van API-Management voorzieningen enten, maar kan ook een aanvang worden gemaakt met het ontwerp en de potentiele realisatie van de JUBES functionaliteiten. Denk aan de JenV API Catalog, API Repository en Reporting functie (de Gateway functionaliteit is reeds beschikbaar), maar potentieel ook een optionele API Management dienst en de opzet van een Center of Excellence Gegevensuitwisseling. Zie onderstaande tabel.

Ontwikkeling	Beschrijving	Kenmerk

JUBES functionaliteiten JenV API Catalog en Repository	Centrale plek om het bestaan van API's te publiceren en API's te vinden	Vereist
JUBES Reporting functie	Analytics- en dashboarding functie om geaggregeerde rapportages te kunnen maken van decentraal verwerkte gegevensuitwisseling	Optioneel, nadere verkenning en potentiele ontwikkeling indien wenselijk en funding beschikbaar gemaakt is
API Management as-a-service (optioneel af te nemen dienst)	Af te nemen API Management as-a-service dienst voor organisaties die niet zelf een API Management oplossing willen realiseren en beheren	Optioneel, nadere verkenning en potentiele ontwikkeling indien wenselijk en funding beschikbaar gemaakt is
'Shared resource server'	Voor enkelvoudige bevragen met achterliggend REST API's van verschillende JenV-Onderdelen	Optioneel, mogelijk nader te verkenning bij voldoende vraag en potentieel positieve business case
Center of Excellence Gegevensuitwisseling / API Management	Kennisdeling en kennisopbouw platform ter bevordering van de adoptie en kwalitatieve implementatie van REST API binnen JenV	Optioneel, nadere verkenning en potentiele opzet indien wenselijk en funding beschikbaar gemaakt is

Tot zo ver dit katern vanuit de 'just in time-, just enough' benadering. Toekomstige uitbreiding van het katern is denkbaar, bijvoorbeeld ten aanzien van het onderwerp signing, encryption, chaining, composite API's, scopes, authorization servers, shared resource servers, token exchange en/of dynamic client registration.