edustandaard

Globale Architectuurschets (GAS)

REST-API onderzoek

Referentie:	20181128
Status:	definitief
Versie:	1.0
Datum:	27 november 2018
Auteur:	A.J. Sloos J.W. van Veen

archixl

Nijverheidsweg-Noord 60-27
3812 PM Amersfoort
T 033-258 5545
F 084-229 8880
E info@archixl.nl
I http://www.archixl.nl

Colofon

Project	REST-API onderzoek
Versienummer	1.0
Projectmanager	Brian Dommisse
Contactpersoon GAS	Tony Sloos / Jan Willem van Veen

Goedkeuring

Naam	Rol	Datum	Versie
Stuurgroep			1.0

Gehanteerde brondocumenten

Brondocument
ROSA - CERTIFICERINGSSCHEMA 2018 - Versiebeschrijving en definities
ROSA - CERTIFICERINGSSCHEMA 2018 - Algemene beschrijving
ROSA - CERTIFICERINGSSCHEMA 2018 - Proces
ROSA - CERTIFICERINGSSCHEMA 2018 - Classificatie
ROSA - CERTIFICERINGSSCHEMA 2018 - Toetsingskader
ROSA - CERTIFICERINGSSCHEMA 2018 - Toezicht
DSO - Architectuur - API-strategie - VASTGESTELD
DSO - Architectuur - URI-strategie - VASTGESTELD
DSO – Overzicht standaarden digitaal stelsel omgevingswet 20180709
Dzone Refcard (260) REST API security
eIDAS-Regulation-for-Dummies-ebook
Programma van Eisen Onderwijs serviceregister (OSR) - V1.0, 23-3-2018
OSR Nadere uitwerking – V0.1 (work in progress) augustus 2018

Betrokkenen bij de GAS

Naam	Functie/Rol
Jan Willem van Veen	**Eindredacteur, enterprise architect**
Tony Sloos	**API expert / architect**

Versie informatie

versie	datum	auteur(s)	doorgevoerde aanpassingen
0.1	12-09-2018	A.J. Sloos	Werkdocument
0.2	19-09-2019	A.J. Sloos	1e review met Marc t.b.v. GAS-raamwerk
0.3	21-09-2019	A.J. Sloos	Inhoudelijke invulling GAS
0.4	01-10-2019	A.J. Sloos	Samenvatting toegevoegd
0.45	04-10-2019	A.J. Sloos	Aanvullen protocollen en interactiepatronen
0.46	04-10-2019	J.W. van Veen	Review / redactie
0.5	17-10-2019	A.J. Sloos	Verwerking terugkoppeling Edukoppeling werkgroep
0.6	26-10-2019	A.J. Sloos	Verwerking reviews
0.7	30-10-2019	A.J. Sloos	Verwerking aanvullende input Marc en Gerald
0.8	31-10-2019	A.J. Sloos	Globale uitwerking OSR case
0.9	31-10-2019	J.W. van Veen	Review / redactie
1.0	21-11-2019	A.J. Sloos	Verwerking terugkoppeling uit interne bespreking

Versie: Werken aan nieuwe versie x.1 etc. eventueel met x.11 etc.; x.0 Definitieve versie, kleine revisies x.0x.
Status: In voorbereiding; In afstemming; In besluitvorming; Definitief.

Review informatie

Reviewer	0.1	0.2	0.3	0.4	0.46	0.9	Opmerking
Gerald Groot-Roessink					☺		Opmerkingen verwerkt
Marc Fleischeuers					☺		Opmerkingen verwerkt
Gerald Groot-Roessink						☺	Opmerkingen verwerkt
Marc Fleischeuers						☺	Opmerkingen verwerkt

☺ Akkoord met de inhoud
☺ Input of commentaar geleverd
☹ Geen reactie ontvangen

Inhoudsopgave

Samenvatting

Gegevensuitwisseling verandert voortdurend en de vraag naar efficiënte koppelingen tussen systemen neemt toe. In toenemende mate wordt gevraagd om een snelle en eenvoudige manier van uitwisselen van informatie, bijvoorbeeld met mobiele toepassingen. De wens is om binnen Edukoppeling het concept API's te introduceren en het uitwisselen van informatie volgens het REST architectuurprincipe mogelijk te maken. Om inzicht te geven in de haalbaarheid van deze wens is in deze GAS een verkenning uitgevoerd aan de hand van drie vragen:

1. Wat zijn relevante REST-standaarden?
2. Wat zijn de concrete maatregelen die passen binnen het toetsingskader van ROSA?
3. Wat zijn de criteria voor het toepassen van WUS, ebMS en/of REST-API's voor veel voorkomende integratiepatronen?

Ad 1.
De overzichten in Tabel 1 en Tabel 2 laten zien welke standaarden in deze verkenning relevant zijn bevonden, inclusief de standaarden die kunnen worden beschouwd als kandidaat voor een onderwijsstandaard.

Naam standaard	Versie	Beschrijving	Toepassing	Bron		
HTTPS	RFC2818, RFC6797	HTTP beveiligd met SSL/TLS	Beveiligde berichtuitwisseling	IETF	⊕	=
Open API Specification	3.0	Specificatiekstandaard voor API's	Specificatie/documentatie	OpenAPI Initiative	⊕	≥
TLS	1.3 (1.2 of hoger)	Translaag encryptie	Beveiliging	IETF	⊕	≥
Ades Baseline Profiles	2.x of hoger	Geavanceerde en gekwalificeerde digitale handtekeningen	Beveiliging	ETSI	⊕	=
DNSSEC	RFC 4033, RFC4034, RFC4035	Domeinnaambeveiliging	Beveiliging	IETF	⊕	=
IP V4/V6	RFC 4033, RFC4034, RFC4035	Internetprotocol versie 4 en 6	Netwerk	IETF	⊕	=
SAML	2.0	Standaard voor het uitwisselen van beveiligingsinformatie	Beveiliging	OASIS	⊕	≥
NEN-ISO/IEC 27001	NEN-ISO/IEC 27001:2013	Informatiebeveiligingsrichtlijn	Beveiliging	NEN	⊕	=
NEN-ISO/IEC 27002	NEN-ISO/IEC 27002:2013	Informatiebeveiligingsrichtlijn	Beveiliging	NEN	⊕	=
E-Portfolio NL	NEN 2035:2014 nl	Onderwijs en ontwikkeling	Gegevensdomein	NEN	⊕	=
NL LOM	1.0	Vindbaarheid van leermaterialen	Gegevensdomein	EduStandaard	⊕	=
SKOS	SKOS W3C Recommendation 18 August 2009	Linked data en begrippenlijsten	Linked-data	W3C	⊕	=
HTTP	V1.1 / RFC 7230	Hypertext transportprotocol	Berichtuitwisseling	IETF	●	=
JSON	RFC 7159	JavaScript Object Notatie standaard	Berichtuitwisseling	IETF	●	≥
JSON schema	json-schema.org (draft-07 of hoger)	Schemastandaard voor JSON structuurvalidatie	Berichtvalidatie	IETF	●	≥
JWT	RFC 7519	JSON Web Token standaard	Beveiliging	IETF	●	≥
OAuth	2.0 / RFC6749	Autorisatiestandaard	Beveiliging	IETF	●	≥
Problem Details for HTTP APIs	RFC 7807	Standaard voor retourneren foutmeldingen	Berichtuitwisseling	IETF	●	≥
URI	RFC 3986	Resource locator standaard: URI, URN, URL	Bronidentificatie	IETF	●	≥
JWS	RFC 7515	Standaard voor digitale handtekening in JSON-structuur	Beveiliging	IETF	●	~
Cross-Origin Resource Sharing	16 januari 2014	Standaard voor realiseren CORS-policy	Beveiliging	W3C	●	≥
Bearer token	RFC 6750	Standaard voor token-gebaseerde toegang	Beveiliging	IETF	●	≥
OpenID Connect	1.0	Autorisatiestandaard gebaseerd op Oauth	Beveiliging	OpenID	●	≥
PKIoverheid	zie TSP's	Digitale certificaten Nederlandse overheid	Beveiliging	Logius	●	=
OAuth 2.0 Dynamics client registration protocol	RFC 7591	Standaard voor het registeren van clients	Beveiliging	IETF	●	≥
OAuth 2.0 Token introspection	RFC 7662	Standaard voor het controleren van tokens	Beveiliging	IETF	●	≥
SAML as OAuth client grant	RFC 7522	Standaard voor SAML bearer assertion	Beveiliging	IETF	●	≥

~ In ontwikkeling
≥ Stabiele doorontwikkeling
= Stabiel

⊕ Pas toe of leg uit
● Vereist REST-API
○ Gewenst REST-API
◊ Optioneel

Tabel 1 - Overzicht relevante standaarden (deel 1/2)

Het overzicht in Tabel 1 laat zien welke standaarden door het Forum Standaardisatie op pas-toe-of-leg-uit lijst zijn gezet en welke standaarden voor REST-API's vereist zijn (gekoppeld aan BIV).

Daarnaast zijn in Tabel 2 verschillende standaarden gewenst of interessant voor REST-API's in de context van specifieke toepassingen, bijvoorbeeld linked-data, geo-data of open data.

Naam standaard	Versie	Beschrijving	Toepassing	Bron		
XML schema	Second Edition	Schemastandaard voor XML structuurvalidatie	Berichtvalidatie	W3C	○	≥
JOSE	draft-ietf-jose-cookbook-08	Standaard voor tekenen en versleutelen van berichten	Beveiliging	IETF	○	~
JWA	RFC 7518	Standaard voor registeren van algoritmes voor JWS en JWE	Beveiliging	IETF	○	≥
JWE	RFC 7516	Standaard voor versleuteling in JSON-structuur	Beveiliging	IETF	○	~
Forwarded HTTP Extension	RFC 7239	Standaard voor exentsie-headers zoals X-Api-key	Berichtuitwisseling	IETF	○	=
JWK	RFC 7517	Standaard voor sleuteldata in JSON-structuur	Berichtuitwisseling	IETF	○	≥
JSON HAL	draft-kelly-json-hal-08	Standard conventie via Hypermedia in REST/JSON API's	Berichtuitwisseling	IETF	○	~
JSON-LD	1.0, 16-01-2014	Standaard voor linked-data in JSON-structuur	Linked-data	IETF	○	≥
SHA-2	ISO/IEC 10118- 3:2016	Authenticatie en integriteitscontrole	Beveiliging	ISO	○	=
SCIM	RFC 7642, 7643 en 7644	Uitwisseling identiteitsinformatie	Beveiliging	IETF	◊	=
X509	RFC5280 en update RFC6818	Authenticatie (PKI Certificaten)	Beveiliging	IETF	◊	=
ETSI TS 119 312	v1.1.1 (2014-11)	Digitale handtekening	Beveiliging	ETSI	◊	=
GeoJSON	RFC 7946	Geografische gegevensstructuren	Geo-data	IETF	◊	≥
DCAT	Recommendation 16-01-2014	Beschrijven van datasets	Linked-data	W3C	◊	≥
OWL	OWL 2	Beschrijvingstaal semantisch web	Linked-data	W3C	◊	≥
RDF	1.1	Resource Description Framework	Linked-data	W3C	◊	=
LDP	1.0 of hoger	Linked Data Platform	Linked-data	W3C	◊	≥
CKAN (JSON API)	2.8 of hoger	Hulpmiddel voor open data websites	Open data	W3C	◊	≥
OData	4.0	Bevraging van REST APIs	Open data	OASIS	◊	=

~	In ontwikkeling
≥	Stabiele doorontwikkeling
=	Stabiel

⊕	Pas toe of leg uit
●	Vereist REST-API
○	Gewenst REST-API
◊	Optioneel

Tabel 2 - Overzicht relevante standaarden (deel 2/2)

Ad 2.

Een belangrijk principe in ROSA zegt dat voor vertrouwelijke gegevensuitwisseling Edukoppeling gebruikt dient te worden. Het is daarom belangrijk dat ook met REST API's alle maatregelen op het vlak van betrouwbaarheid, integriteit en vertrouwelijkheid voor handen zijn. De overzichten in Tabel 3 en Tabel 4 laten zien welke standaarden/technieken in aanmerking komen voor het realiseren van benodigde beveiligingsmaatregelen. Hierop zijn de drie niveaus uit het BIV-classificatiesysteem van ROSA reeds toegepast.

			Niveau 1			Niveau 2			Niveau 3			
#	**Maatregel**	**Standaard/techniek**	**B**	**I**	**V**	**B**	**I**	**V**	**B**	**I**	**V**	**Categorie**
M01	Berichtvalidatie	JSON schema/XML schema		●			●			●		Berichthygiëne
M02	Aangeleverde JSON altijd correct enoderen	JSON serialisatie		●			●			●		Berichthygiëne
M03	Aangeleverde XML altijd correct enoderen	XML serialisatie		●			●			●		Berichthygiëne
M04	Content- en response-type afdwingen	Validatie van content-type en accept headers		●			●			●		Berichthygiëne
M05	Bericht met digitale handtekening	JOSE/JWS/JWT		○			○			●		Encryptie en ondertekening
M06	Berichtversleuteling	JOSE/JWE/JWT			○			○			●	Encryptie en ondertekening
M07	Transportversleuteling (o.b.v. TLS tunnel)	HTTPS (HTTP niet toestaan)			●			●			●	Encryptie en ondertekening
M08	Gebruikersnaam en wachtwoord	HTTP basic-profiel			●			X			X	Gebruikerauthenticatie
M09	Multi-factor authenticatie	MFA hardtoken (RSA, Vasco)			○			●			●	Gebruikerauthenticatie
M10	Multi-factor authenticatie	MFA softtoken			○			●			X	Gebruikerauthenticatie
M11	Multi-factor authenticatie	MFA SMS			○			●			X	Gebruikerauthenticatie
M12	Token-gebaseerde authenticatie	HTTP Bearer-profiel, HTTP MAC-profiel			○			●			●	Gebruikerauthenticatie
M13	Rolgebaseerd autorisatie	OAuth2, SAML 2.0 (RBAC), JWT (rol in payload)			●			●			●	Gebruikerautorisatie
M14	Attribuut gebaseerde autorisatie	OAuth2, SAML 2.0 (ABAC), JWT (pseudo-id)			○			○			●	Gebruikerautorisatie
M15	Gedelegeerde autorisatie	OAuth2, OpenID connect, JWT			○			○			○	Gebruikerautorisatie
M16	Single sign-on (SSO)	SAML 2.0, OAuth2, OpenID connect, JWT			○			○			○	Gebruikerautorisatie
M17	Beperk toepassingsbereik van API-keys	API-key verbonden met aanroeper, domein, …			○			●			●	Gebruikerautorisatie
M18	Bescherm geprivilegieerde acties en informatie	Whitelist toegestane HTTP methoden		●			●			●		Gebruikerautorisatie
M19	Gecontroleerde cross-domain toegang	CORS-headers		○			●			●		Gebruikerautorisatie

X	niet toegstaan
○	optioneel
●	verplicht

Tabel 3 - Overzicht van beveiligingsmaatregelen en standaarden/technieken (deel 1/2)

			Niveau 1			Niveau 2			Niveau 3			
#	Maatregel	Standaard/techniek	B	I	V	B	I	V	B	I	V	Categorie
M20	Archivering berichten	Berichtarchief		○			○			○		Governance
M21	Horizontaal schalen (clusteren)	API gateway: balancer + autoscaler	○			●			●			Governance
M22	Afknijpen # verzoeken per dag (volume)	API gateway: throttling	○			○			●			Governance
M23	Afknijpen # verzoeken per seconde (frequentie)	API gateway: rate limiting	○			●			●			Governance
M24	Gestandaardiseerde foutinformatie	RFC 7807 / Problem Details for HTTP APIs		●			●			●		Governance
M25	Voorkomen van datalekken in GET	Gevoelige data alleen in request-headers			●			●			●	Governance
M26	Voorkomen van datalekken in POST/PUT	Gevoelige data alleen in request-headers/body			●			●			●	Governance
M27	Enkelzijdige authenticatie (digitaal certificaat)	TLS 1.2 of hoger						○			X	Systeemauthenticatie
M28	Tweezijdige authenticatie (digitaal certificaat)	TLS 1.2 of hoger			○			●			●	Systeemauthenticatie

X niet toegestaan
○ optioneel
● verplicht

Tabel 4 - Overzicht van beveiligingsmaatregelen en standaarden/technieken (deel 2/2)

JSON Web Token (JWT) wordt gebruikt in combinatie met JOSE (headers), JWS en JWE om handtekeningen en encryptie te ondersteunen op basis van JSON.

De drie niveaus volgen de indeling van ROSA. Zowel op Europees als nationaal niveau (Afsprakenstelsel Elektronische Toegangsdiensten = ETD[1]) wordt steeds meer gewerkt met vier of meer niveaus. Hieronder is in Tabel 5 weergegeven hoe deze niveaus zich verhouden tot het certificeringschema van ROSA.

ETD	eIDAS	Certifceringschema (ROSA)
1	Niet-bestaand	Niet-bestaand
2	Laag	Laag
2+	Laag	Laag
3	Substantieel	Middel
4	Hoog	Hoog

Tabel 5 – Relatie tussen verschillende classificatie methoden

Het is raadzaam om te onderzoeken hoe de Edustandaard het beste kan aansluiten op deze ontwikkeling.

Ad 3.
Een schema voor een classificering van integratiepatronen het toepassen van of REST-API's en/of WUS, ebMS al dan niet in combinatie met grote berichten (GB), is weergeven in Tabel 6.

Categorie	Patroon	REST	WUS	ebMS	GB
Synchroon	Fire-and-forget	●	●	-	-
Synchroon	Request-response	●	●		
Synchroon	Batch (bulk data)	1		●	○
Asynchroon	Fire-and-forget	-		●	
Asynchroon	Request-delayed-response	2		●	
Asynchroon	Batch (bulk data)	3	-	●	○
Asynchroon	Publish-Subscribe	4	-	●	

Tabel 6 – Classificering van integratiepatronen

[1] https://www.eherkenning.nl/inloggen-met-eherkenning/betrouwbaarheidsniveaus/

De Digikoppeling standaarden ebMS en GB zijn hierin voor de volledigheid meegenomen. Omdat deze in Edukoppeling niet worden gebruikt, zijn ze verder buiten beschouwing gelaten.

#	Toelichting
1	**Synchrone batch (bulk data)** API's zijn niet ontworpen voor grote bestanden, dat wil zeggen batchverwerking voor het ophalen of opsturen van batches met gegevens. API's zijn gericht op status-loze, meestal synchrone, web-achtige verzoeken voor individuele discrete gegevenstransacties. Het verwerken van batches kan echter worden bereikt door meerdere aanroepen naar dezelfde API te bundelen. Dit helpt bij het bereiken van atomiciteit van transacties en helpt in het geval van fouten bij het herstellen daarvan.
2	**Request-delayed-response** In sommige gevallen kan het passend worden geacht om met een API asynchroon een verzoek te versturen en op later moment geïnformeerd te worden over het resultaat. Dit kan op twee manieren worden gerealiseerd: a. De response wordt op een later moment via een aparte resource van dezelfde API beschikbaar gemaakt. De aanroeper moet dan feitelijk "pollen" om het resultaat terug te krijgen. b. De response wordt op een later moment via een terugroepmechanisme naar een andere API afgeleverd bij de aanroeper.
3	**Asynchrone batch (bulk data)** In sommige gevallen kan het passend worden geacht om een asynchrone batches te ondersteunen met een API. Een voorbeeld hiervan kan een bulkcreatie op basis van een batch vanuit een legacy-toepassing zijn. In een dergelijk scenario heeft het de voorkeur om de batch op te knippen in losse aanroepen (POST naar API). Dit lijkt wellicht veel onnodige overhead te veroorzaken, maar het opknippen in kleine transacties maakt het mogelijk om de verwerking te parallelliseren en individuele fouten afzonderlijk af te handelen en de aanroeper getailleerd te informeren over de voorgang en status van de transactie. Een grote batch moet in geen enkel geval synchroon worden verwerkt, want grote batches zullen de onderliggende transportfuncties (http) verstoren, resulteren in time-outs en ander onvoorspelbaar gedrag.
4	**Publish-Subscribe** Dit is een berichtenpatroon waarbij aanbieders berichten naar afnemers (abonnees) sturen. In dit patroon spelen drie componenten een rol: • De aanbieder: publiceert bericht naar een distribiteur • De abonnee: abonneert zich op een categorie berichten • De distribiteur: ontvangt berichten van aanbieder en onderhoudt abonnementen van alle geregistreerde afnemers. Hier is geen API-standaard voor, maar er zijn talloze voorbeelden te vinden van API's voor RESTful implementaties.

De toepasbaarheid van of REST-API's en WUS in relatie tot een aantal specifieke eigenschappen, die van toepassing kunnen zijn op verschillende interactiepatronen, is weergeven in Tabel 7.

Eigenschap	REST	WUS
Transactioneel	5	●
Onweerlegbaar	6	●
Gegarandeerd	7	●

Tabel 7 – Interactie-eigenschappen

#	Toelichting
5	**Transactionele verwerking** Bij het afhandelen van transacties is het belangrijk om rekening te houden met alle foutscenario's en de bijbehorende herstelaspecten. Het is essentieel om toegang te hebben tot de transactievoortgang alsmede de mogelijkheid om analyses uit te voeren op de hoofdoorzaken (root-cause). Om dit te bereiken moeten alle transacties die via een API tot stand komen worden gelogd met nauwkeurige tijdstempels, zodat via monitoring de voortgang van de transactie kan worden gevisualiseerd. Ook dienen transactie-ID's te

#	Toelichting
	worden gebruikt in alle transactionele API-aanroepen om te zorgen dat de transactie van begin tot einde traceerbaar is.
6	**Onweerlegbare verwerking** Om onweerlegbaarheid met API's goed te kunnen ondersteunen, is van belangrijk ervoor te zorgen dat alle afnemers een unieke identiteit hebben en dat er een uitgebreide audit-log wordt bijgehouden van alle API-verzoeken en antwoorden. Digitale handtekeningen zijn hierbij noodzakelijk om niet alleen de authenticiteit en integriteit te garanderen, maar ook om onweerlegbaarheid verder te ondersteunen.
7	**Gegarandeerde verwerking** Betrouwbare of gegarandeerde berichtaflevering wordt niet aangeboden in de REST-laag. Dat betekent dus dat de verantwoordelijkheid ligt bij de afzender van het bericht. Wanneer garandeerde aflevering moet worden ondersteund, is een patroon nodig dat bovenop REST kan worden geïmplementeerd. De aanroeper kan bijvoorbeeld "pollen" om een bevestiging te krijgen of er is een terugroepmechanisme nodig. Een andere aanpak is om de aanroeper in staat te stellen te bevestigen dat het oorspronkelijke verzoek is verwerkt.

1 Doel en inhoud

1.1 Doelen GAS

Dit document is de Globale Architectuurschets (GAS) voor het REST-API onderzoek. De GAS richt zich op de (architectuur)kaders die op dit specifieke project van toepassing zijn en op de consequenties van deze kaders voor de beoogde verandering.

De GAS inventariseert de voor het project geldende kaders, principes, richtlijnen, standaarden en normen en *schetst* de implicaties van deze kaders voor de oplossing die het project gaat realiseren. De GAS geeft globaal inzicht in de context en afbakening van de oplossing en in de samenhang met bestaande en te realiseren diensten, voorzieningen en bouwstenen.
De GAS beschouwt, van de gewenste verandering, de architectuur op hoofdlijnen: het is niet de gedetailleerde 'solution'-architectuur, deze wordt in projecten opgesteld en uitgewerkt. De GAS is kaderstellend voor de uitwerking binnen projecten.

De GAS dient twee hoofddoelen:

- het ondersteunen van besluitvorming voorafgaand aan projecten/programma (gericht op adviseurs en beslissers), en
- het, voor ontwerpers en ontwikkelaars, benoemen van kaders die in het project gevolgd moeten worden bij het ontwikkelen en realiseren van de voorgestelde oplossing.

1.2 Validatie, acceptatie, bijstelling en borging

Opdrachtgever voor het opstellen van de GAS is: Kennisnet (Brian Dommisse).
Opdrachtnemer voor het opstellen van de GAS is ArchiXL (Jan Willem van Veen en Tony Sloos).
Goedkeuring voor de GAS wordt verleend door de stuurgroep.
De goedgekeurde GAS wordt ter acceptatie aangeboden aan de opdrachtgever.

1.3 Gehanteerde begrippen

Begrip	Definitie
API	Application Programming Interface *Een API kan worden gezien als een gebruikersinterface voor softwareontwikkelaars.*
BIR	Baseline Informatiebeveiliging Rijksdienst *De BIR is een gemeenschappelijk normenkader voor de beveiliging van de informatie(systemen). De BIR 2017 is volledig gebaseerd op de internationale norm ISO/IEC 27002. Het concretiseert een aantal eisen tot verplichte operationele afspraken (rijksmaatregelen), om een eenduidig minimumniveau van beveiliging voor gegevens te garanderen. De standaard basisbeveiligingsniveaus (BBN's) met bijbehorende beveiligingseisen maken risicomanagement eenvoudiger*.
EAR	Enterprise Architectuur Rijk
ebMS	ebMS is een koppelvlakstandaard voor asynchroon berichtenverkeer.
GAS	Globale Architectuurschets
NORA	Nederlandse Overheid Referentie Architectuur
REST	REpresentational State Transfer *REST is in tegenstelling tot SOAP geen standaard. Het is een architectuurpatroon dat gebruik maakt van gevestigde internetstandaarden. De achterliggende principes zijn eenvoudig te begrijpen en maken de instap laagdrempelig.*
ROSA	Referentie Onderwijs Sector Architectuur *In de NORA Gebruikersraad wordt de ROSA vertegenwoordigd door het ministerie van OCW als stelselverantwoordelijke.*
WUS	WUS is een verzamelnaam voor koppelvlakstandaarden die zijn bedoeld voor synchroon berichtenverkeer: **W**SDL, **U**DDI en **S**OAP

2 De context

2.1 Doelstellingen en businessdrivers

In de werkgroep Edukoppeling is geconstateerd dat de wijze van gegevensuitwisseling voortdurend verandert. Er zijn steeds meer programma's en Apps die vragen om een snelle en eenvoudige manier van uitwisselen van informatie, bijvoorbeeld met applicaties op (mobiele) devices. Dit vraagt om efficiënte koppelingen tussen systemen. Deze koppelingen tussen systemen worden vaak via API's gerealiseerd en zijn gebaseerd op het REST architectuurprincipe.

2.2 Probleemstelling

Het REST principe voor informatie-uitwisseling wordt met regelmaat tegenover de WUS standaarden geplaatst. Waarbij REST staat voor snelheid, gebruiksvriendelijkheid en innovatie en WUS voor formaliteit, betrouwbaarheid en complexiteit. Opmerkingen die dan langskomen zijn dat RESTful API's de rol van standaarden vervangen of dat WUS en ebMS 'ouderwets' zijn. Maar is dat zo? In deze GAS wordt hierin duidelijkheid geschept.

De GAS geeft duidelijke door middel van:

- een inventarisatie van de REST-standaarden en hun werkingsgebied;
- het aanmerken van standaarden die als kandidaat voor een onderwijsstandaard kunnen worden beschouwd.

2.3 Scope

De scope van de opdracht en het op te leveren advies (in deze GAS) focust zich op:

- Een overzicht van relevante REST-standaarden (diegene die kunnen worden beschouwd als kandidaat voor een onderwijsstandaard);
- Concrete maatregelen gericht op interactiepatronen op basis van REST API's;
- Een schema voor classificering van soorten integratiepatronen met heldere criteria voor het toepassen van WUS, ebMS en/of REST-API's.

2.4 Relevante eisen en –beperkingen

Het certificeringsschema informatiebeveiliging en privacy ROSA wordt als startpunt gehanteerd voor concrete maatregelen gericht op interactiepatronen op basis van REST API's. In het toetsingskader van ROSA wordt de beschikbaarheid, integriteit en vertrouwelijkheid van informatie en systemen uitgewerkt tot concrete maatregelen. Ook bekend als de BIV-indeling:

- **Beschikbaarheid** (Continuïteit)
  De mate waarin beheersmaatregelen de beschikbaarheid en ongestoorde voortgang van de ICT-dienstverlening waarborgen.
- **Integriteit** (Betrouwbaarheid)
  De mate waarin de beheersmaatregelen (organisatie, processen en technologie) de juistheid, volledigheid en tijdigheid van de IT-dienstverlening waarborgen.
- **Vertrouwelijkheid** (Exclusiviteit)
  De mate waarin uitsluitend geautoriseerde personen, programmatuur of apparatuur gebruik kunnen maken van de gegevens of programmatuur, al dan niet gereguleerd door (geautomatiseerde) procedures en/of technische maatregelen.

ROSA kent ook een classificatiesysteem waarin de maatregelen zijn vastgesteld voor drie afzonderlijke niveaus.

	Beschikbaarheid	**Integriteit**	**Vertrouwelijkheid**
Niveau 1 (laag)[2]	Onbelangrijk	Onbelangrijk	Informatie is niet vertrouwelijk
Niveau 2 (middel)	Belangrijk	Beschermd	Informatie is vertrouwelijk
Niveau 3 (hoog)	Noodzakelijk	Noodzakelijk	Informatie is geheim

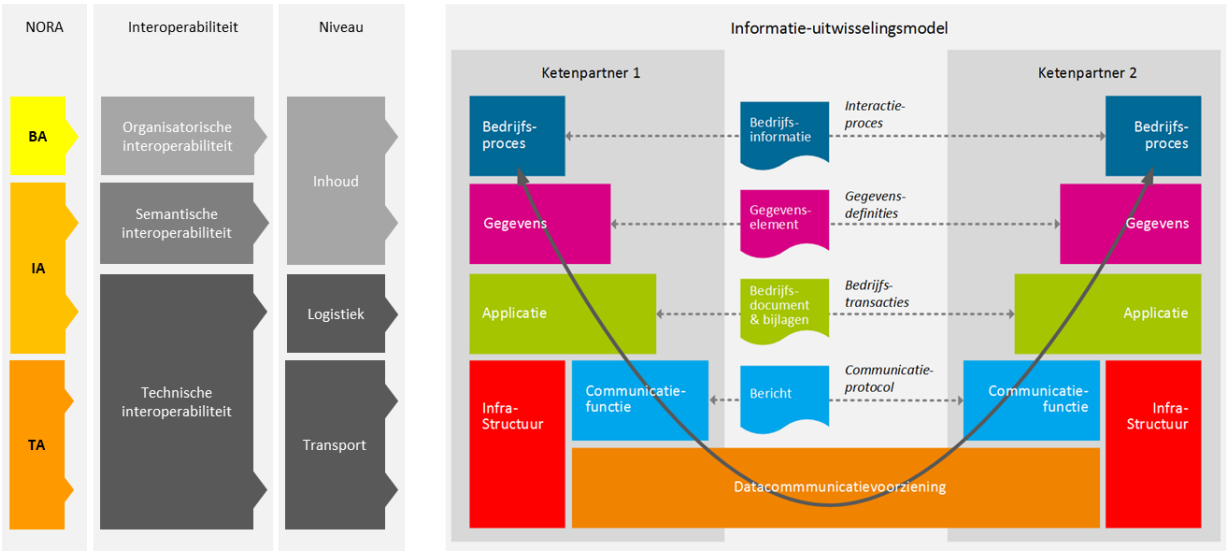
Het juiste niveau kan eenvoudig worden afgeleid uit een reeks vragen over beschikbaarheid, integriteit en vertrouwelijkheid.

[2] Toch geldt er een basisbeveiliging die elke ICT-toepassing op orde zou moeten hebben.

3 Uitgangspunten, principes, inrichtingsconcepten en kaders

3.1 Uitgangspunten

Voor de beschouwing van RESTful integratiepatronen, standaarden en maatregelen wordt gebruik gemaakt van het 5-laags informatie-uitwisselingsmodel. Dit is een model dat binnen de overheid als referentiemodel wordt toegepast.

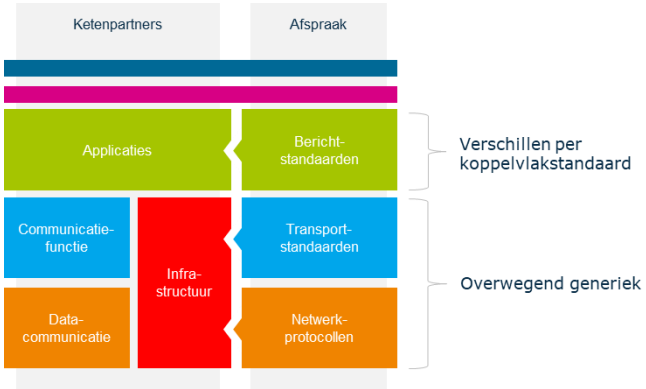


Figuur 1 - Informatie-uitwisselingsmodel

De indeling van de GAS volgt de architectuurlagen die in het model in Figuur 1 vanuit NORA worden onderscheiden. Per laag worden vervolgens de relevante aspecten van de informatie-uitwisseling geadresseerd. Dit gebeurt globaal als volgt:

BA	Hoofstuk 4 Bedrijfsarchitectuur	Focust zich op de organisatorische interoperabiliteit en de (potentieel) betrokken bedrijfsprocessen.
IA	Hoofstuk 5 Informatiearchitectuur	Focust zich op de semantische interoperabiliteit en een deel van de technische interoperabiliteit. Aspecten als inhoud (gegevens) en logistiek (tussen applicaties) spelen hierin de hoofdrol.
TA	Hoofdstuk 6 Technische architectuur	Focust zich op de technische interoperabiliteit. Dit is vooral gericht op transport via de technische infrastructuur.

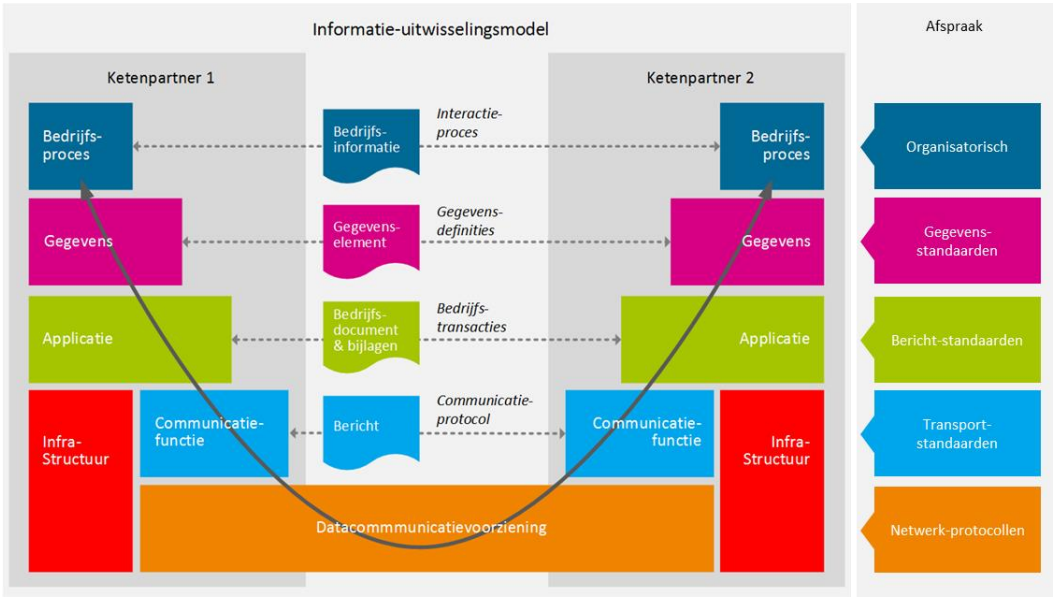
RESTful API's bouwen net als WUS en ebMS voort op dezelfde onderliggende communicatiefuncties en IP gebaseerde datacommunicatievoorzieningen. Dit betekent dat de technische verschillen tussen de koppelvlakstandaarden vooral tot uiting komen in de informatiearchitectuur.



Figuur 2 - Afspraken op de onderste drie lagen

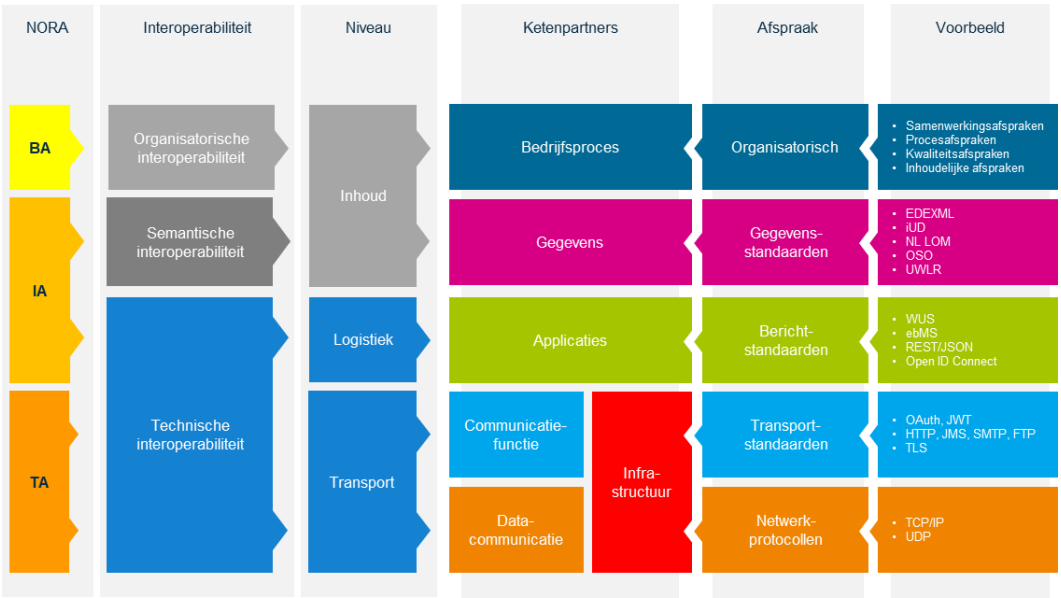
3.2 Architectuurprincipes

Kennisnet maakt bij het inrichten van haar werkprocessen en informatiehuishouding gebruik van open standaarden (waarbij de inhoud van de lijst met overheidsstandaarden van het landelijke Forum Standaardisatie leidend is) en sluit aan op de referentiearchitecturen van de overheid waaronder NORA, EAR en ROSA. In Figuur 3 is aan de hand van de 5 lagen van het informatie-uitwisselingsmodel weergegeven dat voor verschillende aspecten sprake is van een andere categorie afspraken. Deze kunnen variëren van een technisch protocol of standaard tot een procesafspraak.



Figuur 3 - Informatie-uitwisseling en afspraken

Met behulp van het informatie-uitwisselingsmodel kunnen de afspraken, waarvan er in Figuur 4 een aantal voorbeelden worden genoemd, logisch worden gegroepeerd en vanuit een bijpassend architectuurperspectief worden belicht.



Figuur 4 - Groepering van afspraken volgens de architectuurlagen in NORA

De standaarden die als voorbeeld op een specifieke laag zijn geplaatst kunnen in sommige gevallen ook op een andere laag voorkomen. Veel van deze standaarden kennen namelijk zelf ook een gelaagdheid. De plaatsing in de laag moet over de hele breedte worden beschouwd. Wanneer een standaard bijvoorbeeld meer over de inhoud gaat dan over de logistiek, dan hoort hij meer thuis in

de categorie gegevens dan in de categorie berichten. Het is soms lastig om een standaard in een specifiek hokje te plaatsen.

3.3 Generieke digitale infrastructuur (GDI)

De GDI van de overheid bestaat uit standaarden, producten en voorzieningen die gezamenlijk worden gebruikt door meerdere overheden, vele publieke organisaties en in een aantal gevallen door private partijen. Het betreft een onmisbaar deel van de digitale basisvoorzieningen waarmee organisaties hun primaire processen inrichten en is naar zijn aard niet organisatie-, sector- of domein specifiek.

4 Bedrijfsarchitectuur (BA)

4.1 Inleiding

Dit deel van de architectuur focust zich op de organisatorische aspecten van de koppelvlakstandaarden, waaronder:

- Verschillende rollen en actoren
- Interoperabiliteit en de (potentieel) betrokken bedrijfsprocessen
- Toezicht en controle op beveiliging en privacy

4.2 Impact (globaal)

Er zijn verschillende soorten risico's die van invloed zijn op API's. Dit omvat bedreigingen van de beschikbaarheid, evenals vertrouwelijkheid en integriteit. Veel bedreigingen kunnen worden beperkt door "secure coding" technieken, bijvoorbeeld op basis van OWASP-richtlijnen. Wanneer gevoelige en niet openbare gegevens beschikbaar worden gesteld, is het noodzakelijk om:

- Vroeg in de levenscyclus van de API-ontwikkeling een dreigingsanalyse uit te voeren
- Zodra een API is ontwikkeld en gepubliceerd (testbaar is) een penetratietest uit te voeren

Wanneer binnen een organisatie het aantal API's dat beschikbaar wordt gesteld toeneemt, zal ook de behoefte toenemen om naast toezicht en controle op beveiliging en privacy, beleid strikter af te dwingen door de generieke functies te verschuiven naar de infrastructuur (zie 6.4).

4.3 Organisatorische interoperabiliteit

Om de interoperabiliteit tussen organisaties op lange termijn te waarborgen is het noodzakelijk beleid te maken voor het doorontwikkelen en beheren van:

- Standaarden
- Functionele koppelvlakken (één of meer API's plus standaarden)
- Aansluitvoorwaarden

4.4 Inhoud (bedrijfsobjecten)

Bij het gebruik van REST API's voor gegevensuitwisseling en het leveren van diensten, al dan niet via derden is feitelijk sprake van een ecosysteem. Hierin kunnen de volgende bedrijfsobjecten worden onderkend:

- API-specificaties
- API-keys (toegang tot API)
- Beveiligings- en privacybeleid
- Toelichtingen, documentatie, FAQ's
- Identiteiten (personen, organisaties, systemen)
- Gegevens

API-keys dienen en net als identiteiten centraal te worden uitgegeven en beheerd maar bij voorkeur ook aan een centrale poort te worden gecontroleerd. Een REST API mag geen toestand (state) bijhouden. In plaats daarvan wordt elk verzoek voorzien van een token. In verband met het voorkomen van misbruik dient ook voor niet geauthentiseerd gebruik altijd een token in de vorm van een API-key te worden toegepast. Een aangewezen component om deze taak te vervullen is een zogenaamde API gateway. De positionering in weergegeven in Figuur 5 - Referentiearchitectuur API ecosysteem.

4.5 Bedrijfsprocessen (API voortbrengingsproces)

Bij het gebruik van REST API's voor gegevensuitwisseling en het leveren van diensten, al dan niet via derden is feitelijk sprake van een ecosysteem. Hierin kunnen de volgende rollen en bedrijfsprocessen worden onderkend:

Externe ontwikkelaar	Interne ontwikkelaar	Beheerder
• API zoeken • API toegang aanvragen • API vragen stellen • API gebruiken • API fouten rapporteren	• API ontwikkelen • API documenteren • API publiceren • API vragen beantwoorden	• API monitoren • API portaal beheren • Configuratie en standaarden beheren • API fouten afhandelen

Naast de bovenstaande rollen is er ook een beveiligingsautoriteit die specifiek beleid opstelt en beheert voor alle aspecten rondom beveiliging en privacy.

4.6 Standaarden

Dit heeft vooral betrekking op organisatorische afspraken, zoals:

- Samenwerkingsafspraken
- Procesafspraken
- Kwaliteitsafspraken
- Inhoudelijke afspraken

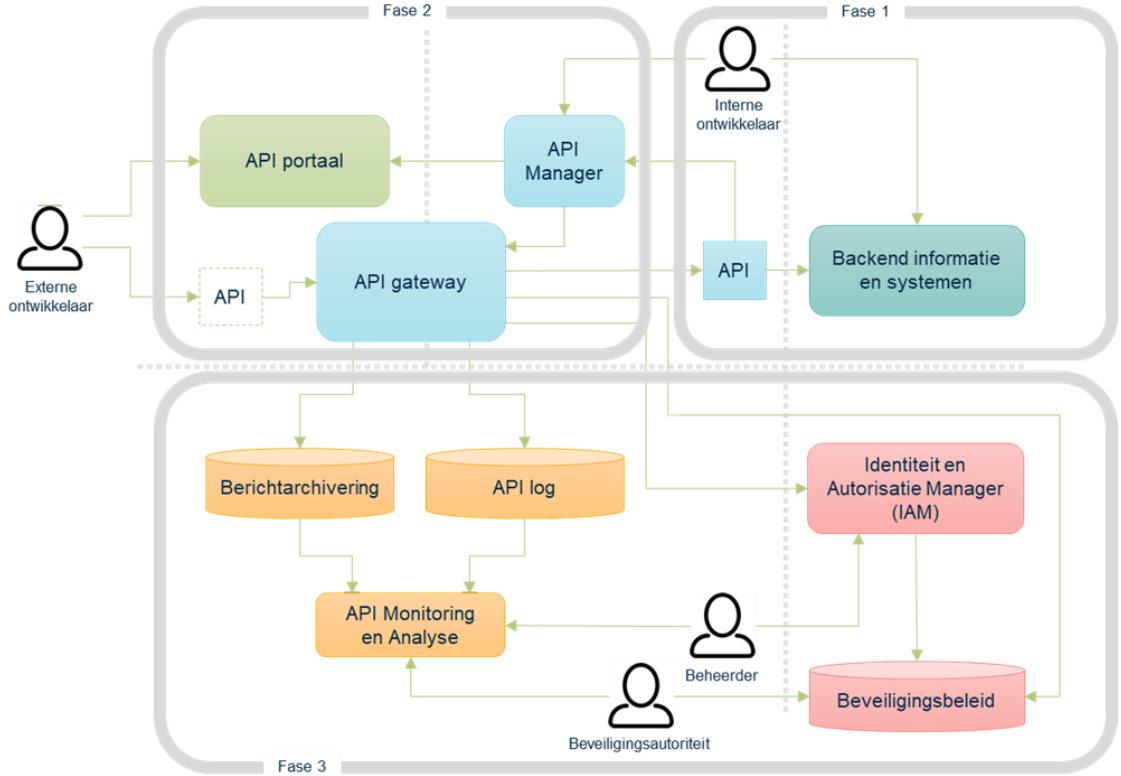
4.7 Betrokken architecturen

NORA, ROSA, EAR

4.8 Architectuurkaders

Projecten moeten gebruik maken van:

1. Referentiearchitectuur inclusief mogelijke fasering (Figuur 5)
2. API-strategie
3. URI-strategie



Figuur 5 - Referentiearchitectuur API ecosysteem

5 Informatiearchitectuur (IA)

5.1 Inleiding

Dit deel van de architectuur focust zich op de semantische interoperabiliteit en een deel van de technische interoperabiliteit. Aspecten als inhoud (gegevensdefinities en -elementen) en logistiek (tussen applicaties) spelen hierin de hoofdrol.

5.2 Impact (globaal)

Bij het ontwerpen en implementeren van RESTful API's wordt vaak gekozen voor JavaScript Object Notation (JSON) als het primaire uitwisselingsformaat. Zeker als de drempel voor ontwikkelaars zo laag mogelijk gemaakt moeten worden is dat noodzakelijk. In veel gevallen moeten bestaande standaarden dan worden omgezet, bijvoorbeeld van een XML-syntax naar een JSON-syntax. Een dergelijke omslag wordt door de Vereniging van Nederlandse Gemeenten (VNG) ook voorbereid voor de berichtenstandaard StUF.

Bij het ontwerpen en implementeren van RESTful API's dient ook serieus aandacht te worden besteed aan specifieke beveiligingsaspecten. De beveiliging van OAuth-berichten is bijvoorbeeld afhankelijk van de uitwisseling van tokens tussen de betrokken applicaties en API-backend systemen. Er is altijd een dreiging dat tokens illegaal worden verkregen, waardoor de vertrouwelijkheid en integriteit van de berichtinhoud of de integriteit van de afzender van het token verloren gaat. Dit risico geldt ook voor de overdracht van API-keys. In de onderstaande tabel worden de belangrijkste bedreigingen en mogelijke mitigatiestrategieën beschreven:

Dreiging	Maatregel (best practices)	Maatregel ROSA

Token vervaardiging of modificatie (nep-tokens en man-in-the-middle aanvallen)	• Digitaal ondertekenen van tokens (bijvoorbeeld JWS met JWT) of het toevoegen van een Message Authentication Code (MAC)
	• Gebruik TLS 1.2 met een encryptie die DHE[3] of ECDHE[4] bevat
	• De afnemer moet valideren:
	  ○ De TLS-certificaatketen
	  ○ De certificaat intrekkingslijst (CRL[5])
		M05, M12, M27, M28

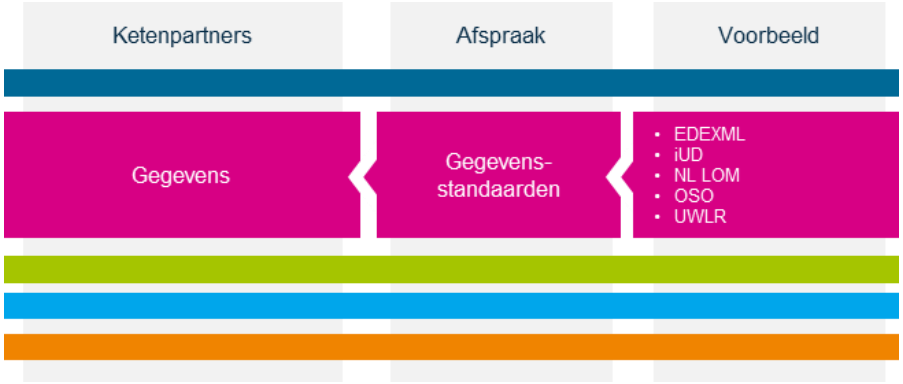
Token openbaarmaking (man-in-the-middle aanvallen) De toegangstoken wordt in klare tekst doorgegeven zonder hashing, ondertekening of versleuteling.	• Met behulp van de "audience-header" kunnen de afnemers, aanbieders en machtigingsserver samen zorgen[6] dat het token alleen kan worden gebruikt op de bronservers die door de afnemer zijn aangevraagd en worden herkend door de machtigingsserver • Wordt ook geadresseerd met de parameter "state" in de kop	M12, M17
Token omleidingen Zorg ervoor dat de machtigingsserver en aanbieders "gekoppeld" zijn en dat het token alleen hierin kan worden gebruikt tussen de opgegeven servers.	• Beperk de levensduur van het token (bijvoorbeeld 10 minuten)	M12
Token-replay (een bestaand token worden gekopieerd) Bijvoorbeeld als een verversingstoken of autorisatiecode wordt gekopieerd en in een ander verzoek opnieuw wordt gebruikt.	• Gebruik ondertekende verzoeken samen met nonce en timestamps • Valideer TLS-certificaatketen bij toegang tot aanbieder/bron	M05, M12, M27, M28

De generieke maatregelen rondom beschikbaarheid, integriteit en vertrouwelijkheid van informatie en systemen, zoals uitgewerkt in ROSA, blijven onverminderd van toepassing.

[3] Diffie Hellman Exchange protocol is een cryptografisch protocol waarmee over een onbeveiligd communicatiekanaal een geheime encryptiesleutel kan worden uitgewisseld
[4] Elliptic Curve Diffie Hellman Exchange protocol is een nieuwere en snellere variant DHE
[5] Lokaal opgeslagen in een bestand of LDAP-server
[6] Het ondertekenen van tokens is ook van toepassing op token doorverwijzingen

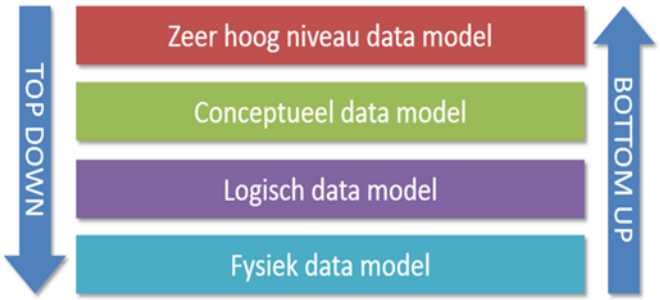
5.3 Semantische interoperabiliteit

De semantische interoperabiliteit heeft vooral betrekking op de afspraken die bij Edustandaard of bij derden in beheer zijn of komen. Enkele voorbeelden van dit soort afspraken en de relatieve positie in het vijflaags referentiemodel zijn weergeven in Figuur 6.



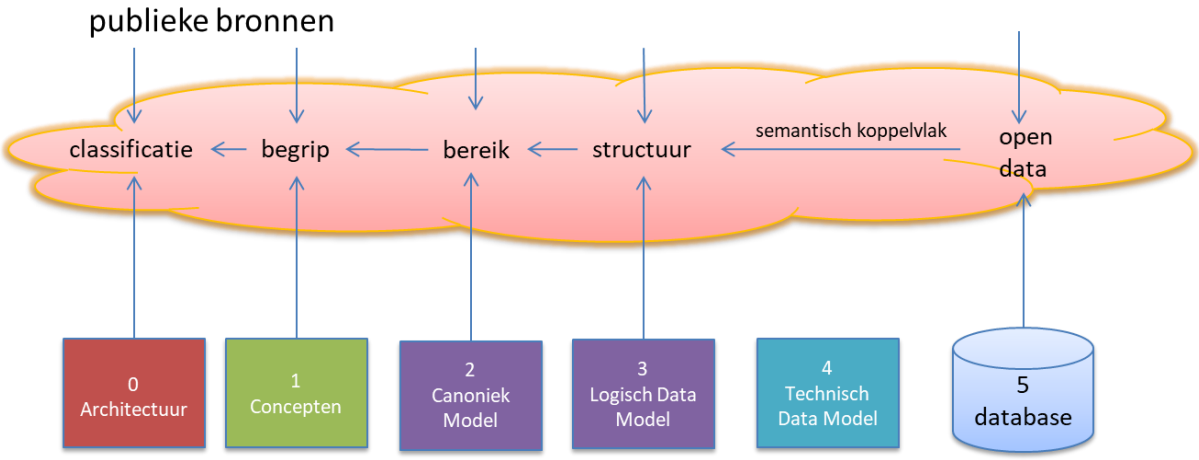
Figuur 6 - Positionering van semantische afspraken/standaarden

Op de gegevenslaag kan nog verder ingezoomd worden door naar de gegevensmodelering te kijken. Dit kan op verschillende abstractieniveaus zoals weergegeven in Figuur 7.



Figuur 7 - Gegevensmodellering op verschillende abstractieniveaus

Een conceptueel datamodel vormt in de praktijk een goede basis voor het vinden van zogenaamde resources (zelfstandige naamwoorden) in de context van een RESTful API-ontwerp. Met de inbreng van DUO zijn de genoemde niveaus op vijf lagen van een voortbrengingsproces geplot. Bij de toelichting is aangegeven welke standaarden o.a. relevant zijn voor het werken met API's en linked-data.



Figuur 8 - Gegevensontwikkelstraat

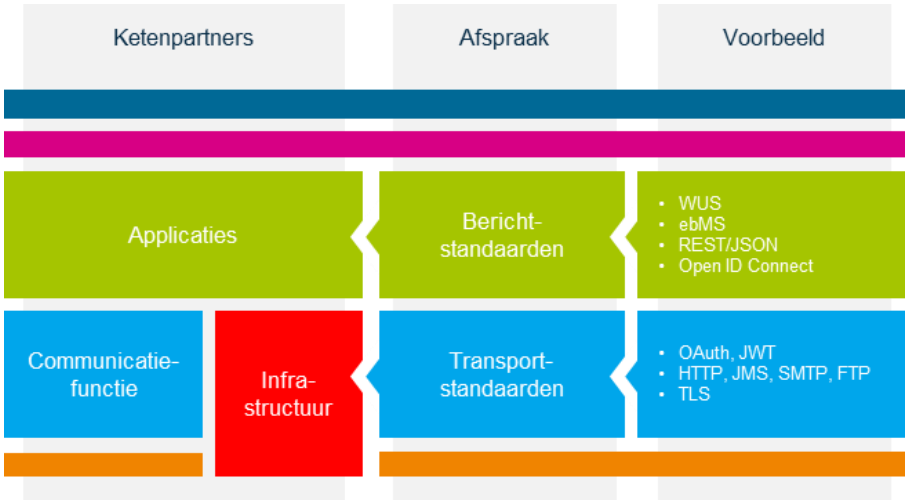
Een goed voorbeeld van een linked-data implementatie is het Gegevenswoordenboek DUO, welke beschikbaar is via https://lod.duo.nl.

Laag	Toelichting en relevante standaarden
0 - KOI	Kernmodel Onderwijs Informatie, Enterprise Architectuur
1 - CIM	Gegevenswoordenboek: SKOS (DCAT, DQV), pas-toe-leg-uit
2 - CDM	Gegevenswoordenboek: UML → RDFS, JSON-LD, OWL, SHACL
3 - PDM	Gegevenscatalogus: UML → RDFS, OWL, SHACL
4 - TDM	Product-schema's: XSD, SQL-DDL, JSON-Schema
5 - DMBS	RDBMS, Triple Store of anders

5.4 Technische interoperabiliteit

5.4.1 Logistiek

De technische interoperabiliteit heeft vooral betrekking op de afspraken die een rol spelen in berichtstandaarden en de uitwisseling van gegevens tussen applicaties (logistiek). Enkele voorbeelden van dit soort afspraken en de relatieve positie in het vijflaags referentiemodel zijn weergeven in Figuur 9.



Figuur 9 - Positionering technische afspraken/standaarden

5.4.2 Applicaties

Integriteit en vertrouwelijkheid kan bij het gebruik van API's voor een groot deel in de communicatiefuncties (zie 6.4) worden geborgd. Ondanks dat TLS het bericht tijdens het transport beschermt, is dit alleen van toepassing tussen de eindpunten van de verbinding tussen twee componenten.

Dus als tussenliggende componenten niet volledig onder de controle van dezelfde aanbieder vallen, is berichtversleuteling of een handtekening in applicaties noodzakelijk:

- Als de inhoud alleen zichtbaar moet zijn voor specifieke afnemers, is het noodzakelijk om het bericht te versleutelen;
- Wanneer alleen de inhoud van een specifieke bron moet worden gegarandeerd, is het alleen noodzakelijk het bericht te voorzien van een handtekening.

Door het gebruik van berichtencryptie kan een "JSON-payload" geheel of gedeeltelijk alleen door de afnemer worden gelezen. Dit is noodzakelijk wanneer de inhoud die door de API wordt gebruikt gevoelig is en het bericht meerdere tussenpunten passeert.

ⓘ	Versleuteling in applicaties is alleen noodzakelijk wanneer van wege niet vertrouwde (tussenliggende) componenten gegevensbescherming absoluut is vereist. Versleutelen is rekenintensief en het maakt het bovendien moeilijker voor beveiligingsmechanismen, zoals API-gateways, de berichtinhoud te valideren en transformeren.

Wanneer alleen de integriteit van het bericht moet worden gewaarborgd, is het beter om het bericht te voorzien van een handtekening. Er zijn veel bestaande manieren om berichtinhoud te versleutelen of te tekenen. Het is van belang dat de implementatie voldoet aan de standaardalgoritmen die zijn vastgelegd in NZISM (HMAC-algoritmen).

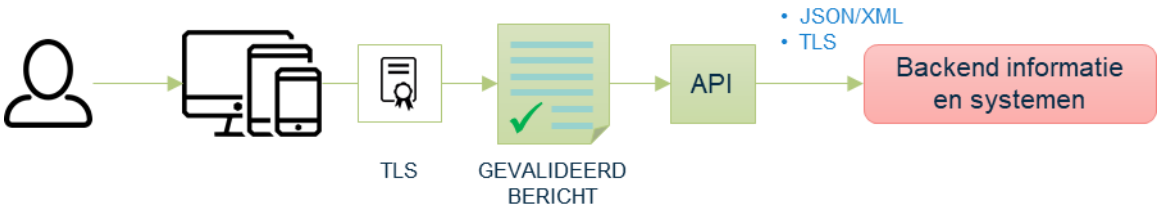
5.5 Standaarden

5.5.1 Gegevensstandaarden

Bestaande gegevensstandaarden blijven van toepassing. Dat wil zeggen, de concepten en de semantiek blijven onveranderd. Het is in veel gevallen wel noodzakelijk om over te stappen naar een de syntax van het uitwisselingsformaat, zoals de JavaScript Object Notation (JSON).

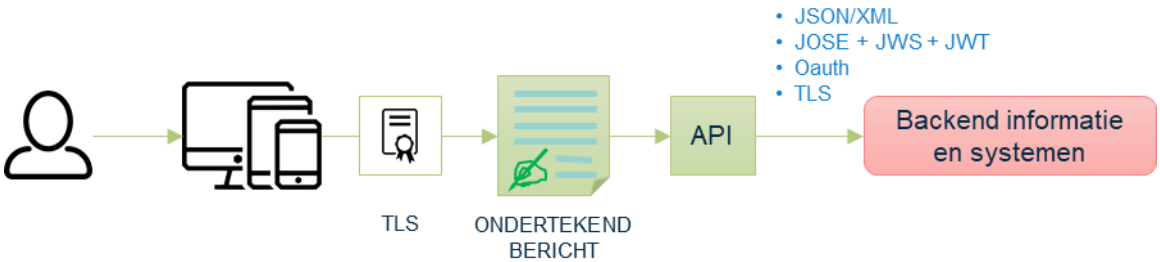
5.5.2 Berichtenstandaarden

Bij het ontwerpen en implementeren van RESTful API's wordt vaak gekozen voor JSON als het primaire uitwisselingsformaat. Bestaande standaarden die gebruik maken een XML-syntax kunnen over het algemeen snel worden omgezet naar JSON. Wanneer dit niet gewenst is, kan ook gekozen worden om binnen een REST/JSON API een extra content-type toe te staan. Zo kan naast JSON als primair formaat bijvoorbeeld ook XML uitgewisseld worden.

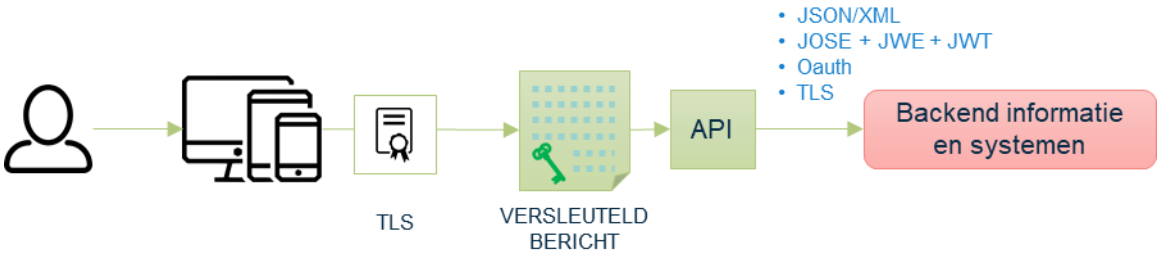


In beide gevallen gaat het om platte tekst met een voorgeschreven syntax en gegevensstructuur. Zowel voor XML als JSON geldt dat de gegevensstructuur in een zogenaamd schema kan worden vastgelegd. In de hele keten kan daarmee het bericht worden gevalideerd.

In aanvulling daarop kan het noodzakelijk zijn om naast de structuur ook de integriteit van het hele bericht, dus inclusief de gegevens te controleren. In dat geval kan door de verzender ook een handtekening worden meegestuurd. Een bericht kan worden getekend en verzonden met behulp van de JSON Web Signature (JWS) standaard. JWS objecten worden geserialiseerd en verstuurd als JSON Web Token (JWT). Naast het bericht worden ook zogenaamde Javascript Object Signing and Encryption (JOSE) headers meegegeven.

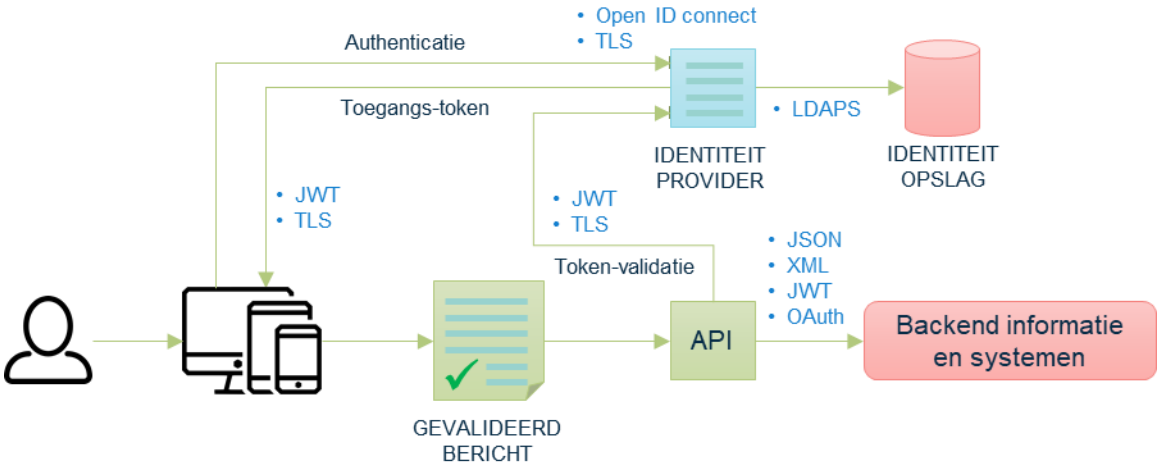


Een bericht kan worden versleuteld en verzonden met behulp van de JSON Web Encryption (JWE) standaard. JWE objecten worden geserialiseerd en verstuurd als JSON Web Token (JWT). Naast het bericht worden ook zogenaamde Javascript Object Signing and Encryption (JOSE) headers meegegeven.



5.5.3 Authenticatie- en autorisatiestandaarden

Autorisatie en authenticatie zijn intrinsiek gekoppeld binnen het OAuth-framework dat wordt beschouwd als synoniem voor het beveiligen van API's. OAuth biedt een uitbreidbare benadering voor beveiliging van API's en gaat verder dan standaard verificatie- en autorisatiemechanismen. Op basis van beveiligingstokens kan het namelijk ook worden gebruikt voor gedelegeerde autorisatie, bijvoorbeeld om een mobiele applicatie te autoriseren om namens de gebruiker te handelen.



OpenID Connect is een eenvoudige identiteitslaag bovenop het Oauth-protocol. Hiermee kunnen applicaties de identiteit van de eindgebruiker verifiëren op basis van de verificatie die wordt uitgevoerd door een autorisatieserver. Applicaties kunnen hiermee ook basisprofielinformatie over de eindgebruiker verkrijgen op een interoperabele (JSON, JWT) en REST-achtige manier.

5.6 Betrokken architecturen

EAR, NORA en ROSA.

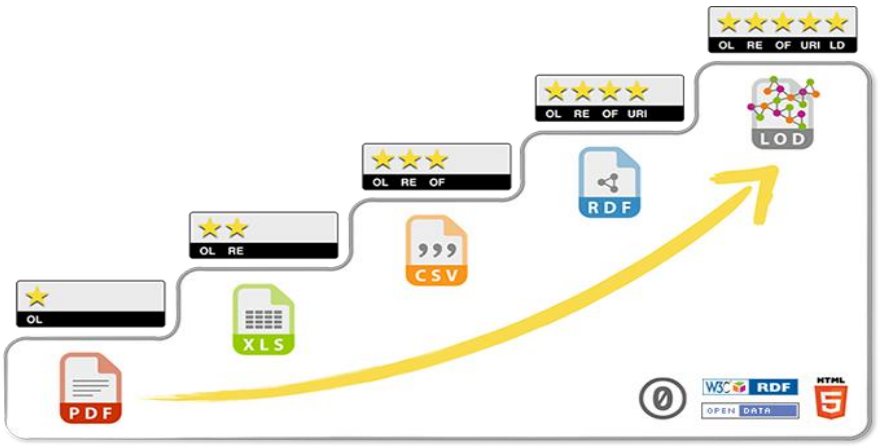
5.7 Architectuurkaders

Projecten moeten gebruik maken van het centrale gegevenswoordenboek voor uniformiteit (meta-) gegevenselementen. Zowel het informatiemodel als de uitwisselingen tussen applicaties of tussen applicatie en gebruiker worden gemodelleerd op basis van één centraal gegevenswoordenboek.

Bij voorkeur wordt dit gegevenswoordenboek ontsloten als linked open data, conform 5 sterren model[7] van Tim Berners-Lee. D.w.z. dat alle metadata via een URI beschikbaar is en ontsloten kan worden.

Een goed voorbeeld hiervan is het Gegevenswoordenboek DUO, beschikbaar via https://lod.duo.nl/

[7] https://5stardata.info/en/



Voor een globaal beeld van de referentiearchitectuur wordt op dit moment verwezen naar Figuur 5 in hoofdstuk 4 en voor de technische architectuur naar hoofdstuk 6.

De referentiearchitectuur geeft een globaal eindbeeld van een ecosysteem met API's. De invulling hiervan is gekoppeld aan een bepaald volwassenheidniveau. Voor een gedegen uitwerking die past in de beoogde context is het noodzakelijk om bij de juiste mensen in meer detail informatie op te halen.

6 Technische architectuur (TA)

6.1 Inleiding

Dit deel van de architectuur focust zich op de technische interoperabiliteit. Dit is vooral gericht op de aspecten die te maken hebben met het transport via de technische infrastructuur.

6.2 Impact (globaal)

Bij het beschikbaar stellen van API's is er sprake van specifieke dreigingen. Dit zijn dezelfde soort dreigingen die bij websites voorkomen en met een reeks standaard technische maatregelen kunnen worden opgelost:

Dreiging	Maatregel (OWASP)	Maatregel (ROSA)
Blootstelling van onjuiste API-methoden voor toegang tot services	• Bescherm en beperk (white list) de toegestane HTTP-methoden (GET, HEAD, POST, PUT en DELETE) • Valideer methode(s) in combinatie met sessietoken of API-key • Hanteer standaard gastoegang voor alle blootgestelde API's	M17, M18
Denial Of Service-aanvallen	• Beperk de aanroep-frequentie, bij voorkeur gekoppeld aan een SLA • Beperk het aanroep-volume, bij voorkeur gekoppeld aan een SLA	M22, M23
Schadelijke invoer, injectie-aanvallen en fuzzing	• Valideer alle input • Valideer het inkomende content-type: applicatie/json • Valideer JSON-inhoud en XML-inhoud (schema en indeling) • Scan bijlagen • Retourneer geldige en zinvolle HTTP-codes • Valideer het respons-type	M01, M02, M03, M04, M24
Cross-Site Request Forgery	• Gebruik tokens met status- en nonce-parameters	M12
Cross-Site Scripting Attacks	• Valideer alle input	M01, M02

6.3 Technische interoperabiliteit

6.3.1 Transport

In de meeste gevallen is het alleen de server die zich authentiseert met een certificaat. Hierbij wordt gebruik gemaakt van enkelzijdig TLS en wordt de integriteit de vertrouwelijkheid van het transport van berichten van en naar API gegarandeerd. Er zijn echter ook scenario's waarin ook de server authenticatie op basis van een certificaat van de client vereist. De server vraagt in dat geval om het client-certificaat tijdens de TLS-handshake via het netwerk.

Wederzijdse verificatie met TLS-certificaten is zeer geschikt voor systeem-tot-systeem communicatie. Daarmee zijn de beveiligingsreferenties die tussen de twee entiteiten worden uitgewisseld veel eenvoudiger te beheren.

6.4 Infrastructuur(diensten)

Wanneer vanuit beleid beveiliging en privacy strikter moet worden afgedwongen, kunnen de generieke functies die hierin een rol spelen worden verschoven naar de infrastructuur.

Zo heeft het bijvoorbeeld de voorkeur om API-keys centraal uit te gegeven en te beheren maar ook aan een centrale poort te controleren. Een aangewezen component om deze taak te vervullen is een zogenaamde API gateway.

6.4.1 Communicatiefuncties

Een API-gateways (zie positionering in Figuur 5 - Referentiearchitectuur API ecosysteem) kan generieke bescherming bieden tegen de typische API-kwetsbaarheden en bedreigingen. Meestal hebben deze betrekking op:

- Frequentie en volume van aanroepen beperken om Denial of Service-aanvallen te voorkomen door centraal te controleren op quota gekoppeld aan een API-key;
- Berichtanalyses uitvoeren om HTTP-aanvallen te blokkeren; parameteraanvallen zoals cross-site scripting, SQL-injectie, opdrachtinjectie en cross-site verzoekvervalsing te voorkomen;
- Controleren en beheersen van de berichtuitwisseling via de API, afgestemd op de vereiste toegangsrechten en het beveiligingsbeleid;
- Het verschaffen (indien nodig) van toegangscontrole tot API-functionaliteit (white listing).

6.4.2 Communicatievoorzieningen

In het kader van beschikbaarheid is het noodzakelijk om specifieke dreigingen, zoals een DDOS-aanval, die kunnen leiden tot API-downtime te beperken. In de infrastructuur kan dit door de zogenaamde "blootstelling" van API's in het basisontwerp van de communicatievoorzieningen te beschermen.

Beschikbaarheid omvat ook schaalbaarheid om aan de vraag te voldoen en ervoor te zorgen dat de hostingomgeving stabiel is. De beschikbaarheidsniveaus zijn onderdeel van de hardware- en softwarestacks die de hosting van API-diensten ondersteunen. Er zijn geen specifieke normen voor beschikbaarheid in relatie tot API's. De beschikbaarheid van API's kan het beste worden geplaatst in hetzelfde kader als bedrijfscontinuïteit en noodherstel. Dit soort standaarden leveren voldoende aanknopingspunten voor een adequate aanpak voor risicobeoordeling en om de beschikbaarheidsvereisten te definiëren.

6.5 Standaarden

6.5.1 Transportstandaarden

REST is een architectuurstijl die primair gebruik maakt van HTTP als transportprotocol. In de praktijk zal altijd gebruik worden gemaakt van de versleutelde variant, beter bekend als HTTPS. Door alleen gebruik te maken van TLS 1.2 of hoger worden API's op een effectieve en betrouwbare manier via HTTPS beschikbaar gesteld.

6.5.2 Netwerkprotocollen

RESTful API's bouwen net als WUS en ebMS voort op dezelfde onderliggende communicatiefuncties en IP gebaseerde datacommunicatievoorzieningen gebaseerd op de netwerkprotocollen IPv4 en IPv6.

6.6 Betrokken architecturen

EAR, GDI, PKI-overheid

6.7 Architectuurkaders

Integriteit en vertrouwelijkheid kan bij het gebruik van API's voor een groot deel in de infrastructuur worden geborgd op basis van de volgende kaders:

- Alle communicatie naar of van een API moet verlopen via TLS 1.2 of hoger. Andere versies van TLS en SSL moeten worden uitgeschakeld.
- De afnemer moet de TLS-certificaatketen valideren bij het gebruikt van beveiligde resources, waaronder de Certificate Revocation List (CRL) controleren.

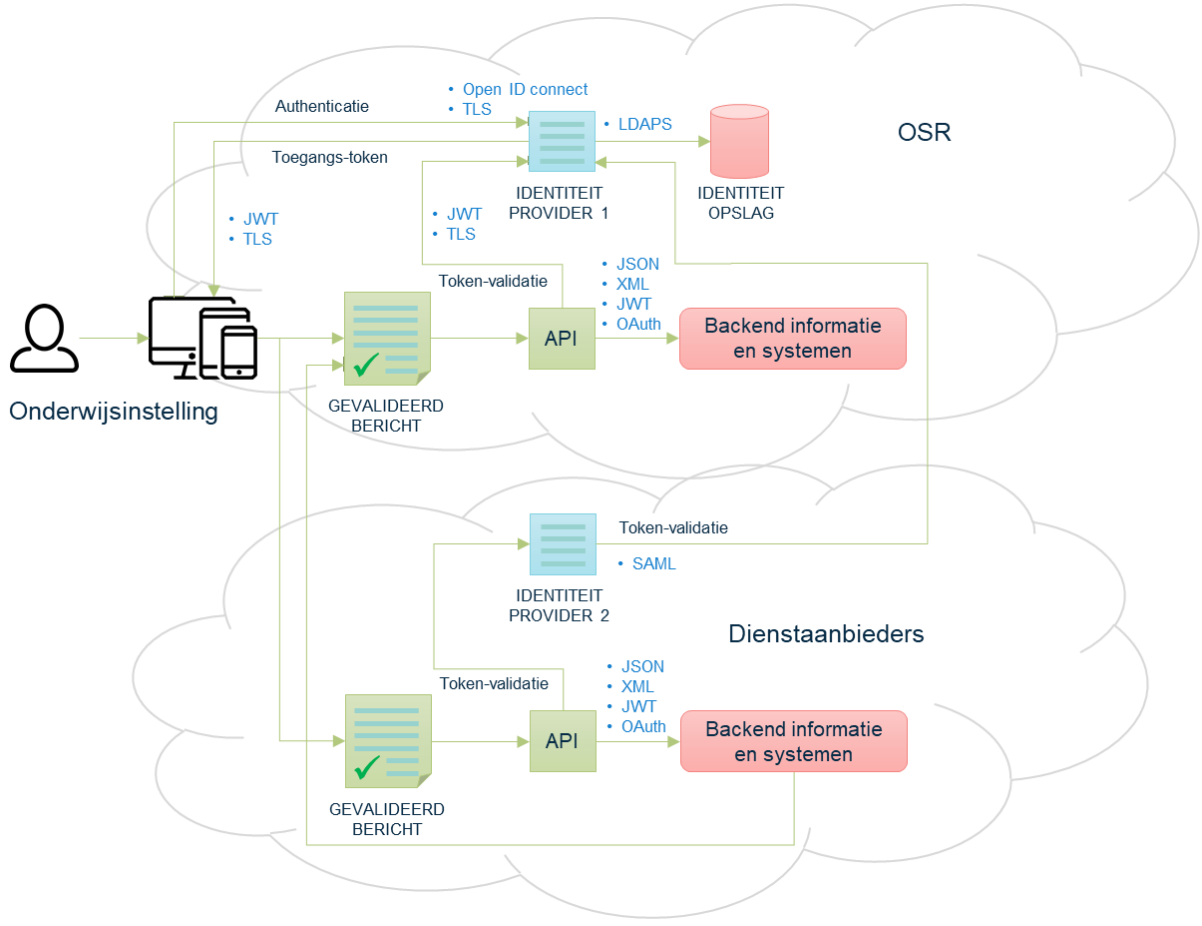
Appendix A - CASUS: Onderwijs Service Register (OSR)

Het serviceregister zal een telefoonboek-functie voor de onderwijsketen leveren. Een gebruiker zoals een onderwijsinstelling of leverancier van diensten voor een onderwijsinstelling kan het register bevragen voor kenmerken en technische details van diensten.

De onderstaande oplossing is fictief. Dit zou een oplossingsrichting kunnen zijn, maar voor een goede uitwerking van interactiepatronen tussen dienstaanbieders is binnen dit REST-API onderzoek onvoldoende tijd beschikbaar. Voor een gedegen uitwerking is het noodzakelijk om bij de juiste mensen in meer detail informatie op te halen.

De registerfunctie is op basis van een RESTful API eenvoudig te realiseren. Alle bevragingen van het serviceregister kunnen via verschillende resources binnen één OSR API worden afgehandeld. Onderhoud van de informatie in het register kan ook via verschillende resources binnen één OSR API worden afgehandeld. Bevragingen en mutaties (onderhoud) kunnen ook samen in één API worden gecombineerd, maar als de beveiligingscontext anders is, is het beter om de functionaliteit in afzonderlijke API's onder te brengen.

Bij wijze van voorbeeld, wil dienstaanbieder A een bericht sturen naar dienstaanbieder B, namens onderwijsinstelling X. A vraagt het OSR wat het afleveradres (URI) van de dienst is van B. Hierbij wordt er vanuit gegaan dat een beheerder bij een onderwijsinstelling heeft bepaald wie zijn dienstverleners zijn. Het serviceregister kan de actuele afleveradressen van de aanbieders door de hele keten beschikbaar stellen. Dienstaanbieder B kan op zijn beurt gegevens over A opvragen en nagaan of school X inderdaad wel een relatie heeft met A.
In Figuur 10 wordt een globaal beeld geschetst waarin de dienstaanbieders A en B op basis van een OSR-API namens een onderwijsinstelling X berichten kunnen uitwisselen. Authenticatie en het doorgeven van identiteit op basis van tokens (identity propagation) speelt hierin een belangrijke rol bij het geautoriseerd kunnen routeren en herleiden van de berichten.



Figuur 10 – Positionering van OSR met aanbieders en afnemers