Edukoppeling

REST/SaaS-profiel

voor

M2M gegevensuitwisseling binnen het onderwijs

Edustandaard

Datum: mei 2020

Versie: 0.4

Status: Concept

edustandaard

Inhoudsopgave

1. Historie

Versie	Auteur	Datum	Opmerking
0.1	E. Reinhoud	december 2019	Initiële versie
0.2	E. Reinhoud	maart 2020	Na bespreking versie 0.1 in WG van 22 januari het volgende verwerkt: • Aandachtspunten verwijderd • Voorschriften indeling generiek (idem voor WUS) en specifiek. • Verschillende tekstuele wijzigingen
0.3	E. Reinhoud	april 2020	Na bespreking 0.2 in WG van 18 maart en online input: • Keuze routering obv HTTP header • Opdeling voorschriften in deel generiek (idem voor WUS) en specifiek voor REST • Opdeling voorschriften beveiliging in deel Transportbeveiliging (beheer UBV) en Berichtbeveiliging (beheer EK) • Volgorde en niveau van compliance van voorschriften overheidsbrede afspraken aangepast • Verschillende tekstuele aanvullingen / wijzigingen.
0.4	E. Reinhoud	Mei 2020	Na bespreking 0.3 in WG van 29 april: • Keuze routering aangepast obv voorlopige conclusie notie • Enkele aanpassingen op 3.2. Specifieke voorschriften REST/SaaS-profiel en definitie van kenmerk 'WON'T' aangepast. • Kennisplatform API's heeft principe API-11 aangepast, dit is overgenomen in dit profiel: 'Encrypt connections using TLS following the latest NCSC' dit wordt echter overschreven door voorschrift van UBV. • Bijlage Foutafhandeling aangepast

2. Inleiding

2.1. Aanleiding

Omdat gegevensuitwisseling meer en meer op basis van RESTful API's gerealiseerd wordt heeft de Architectuurraad gevraagd om een inventarisatie naar REST-standaarden uit te voeren om helder te krijgen hoe dit zich tot de WUS toepassingsgebieden verhoudt en wat nodig is voor een veilige en betrouwbare gegevensuitwisseling op basis van RESTful standaarden. De inventarisatie en de ontwikkeling van een aantal concept profielen hebben uiteindelijk geresulteerd in het REST/SaaS-profiel dat in dit document is uitgewerkt. Hierbij worden een beveiligde point-to-point koppeling en het kunnen routeren tussen de Edukoppeling rollen als belangrijke uitgangspunten gehanteerd. Op basis van het transportbeveiligingsprofiel (op termijn UBV-voorschriften) wordt de integriteit en veiligheid van de gegevens in transport geborgd. Hierbij wordt de identiteit van beide partijen bepaald door het OIN in het certificaat dat wordt gebruikt voor de beveiliging van het transport. Een ander uitgangspunt voor dit REST/SaaS-profiel is dat het goed moet aansluiten op standaarden die overheidsbreed voor RESTful uitwisselingen voorgeschreven worden. De producten van het kennisplatform API's worden daarom als basis gebruikt. De API Design rules en overige producten van het kennisplatform API's[1] worden in beheer genomen door Logius en zullen waarschijnlijk nog doorontwikkeld worden. We verwachten verder dat er in de nabije toekomst meerdere Edukoppeling REST-profielen zullen ontstaan voor andere contexten dan het SaaS-profiel.

2.2. Doel en doelgroep

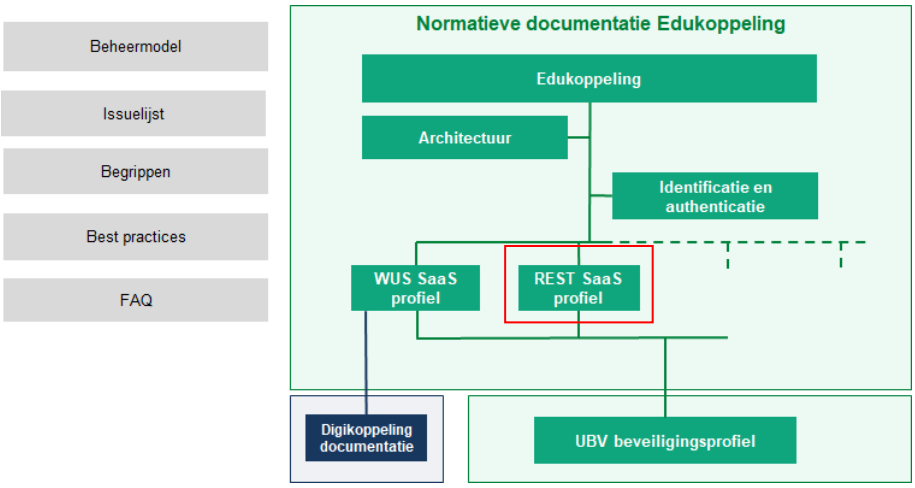
Dit document beschrijft de Edukoppeling REST/SaaS-profiel (verder aangeduid als REST-profiel) en is onderdeel van de Edukoppeling Architectuur. Het doel dat met dit profiel nagestreefd wordt is het op een generieke manier kunnen uitwisselen van gegevens binnen de onderwijssector. Daarbij wordt zowel het model waarbij een onderwijsinstelling zijn systeem zelf host, als waarbij de onderwijsinstelling deze via diensten afneemt van een SaaS-leverancier, ondersteund. Dit document definieert de kaders voor de profielen om dit te bereiken.

Dit document is bedoeld voor ICT-specialisten die betrokken zijn bij het ontwerpen en ontwikkelen van systeem-naar-systeem (M2M) koppelingen. Het gaat hier om werknemers (ontwikkelaars, architecten, projectmanagers, informatiemanagers etc.) werkzaam bij onderwijsgerelateerde organisaties, zowel in de publieke als private sector.

2.3. Positionering binnen Edukoppeling Architectuur

Het Edukoppeling REST/SaaS-profiel is onderdeel van de Edukoppeling Architectuur. In het volgende hoofdstuk wordt het REST/SaaS-profiel inhoudelijk beschreven.

[1] https://www.geonovum.nl/themas/kennisplatform-apis



Figuur 1- Positionering van REST/SaaS-profiel binnen de Edukoppeling Architectuur

2.4. Functioneel toepassingsgebied

Het functionele toepassingsgebied van het REST/SaaS-profiel betreft M2M-gegevensuitwisseling via een beveiligde point-to-point verbinding waarbij RESTful standaarden worden toegepast. Het wordt gebruikt voor bevragingen (pull) en meldingen (push) op basis van een request-response uitwisselingspatroon. De client is in deze context geen browser, maar een systeem (applicatie). Dit profiel heeft overlap met het functionele toepassingsgebied van het WUS/SaaS-profiel. De gegevens kunnen op basis van de afspraken binnen dit profiel gerouteerd worden tussen een verwerker en eindorganisatie. Dit laatste is geen verplichting indien de eindorganisatie ook de rol van verwerker (en logistieke dienstverlener) heeft. Als er sprake is van een transparante intermediair, of er is noodzaak voor onweerlegbaarheid, of gegevensuitwisseling op basis van XML dan wordt het Edukoppeling WUS-profiel toegepast.

Er is reeds een Edustandaard Open Onderwijs API (OOAPI[2]) afspraak. Het functionele toepassingsgebied is niet expliciet gedefinieerd, maar ondersteunt processen waar o.a. persoonsgegevens, faculteitsgegevens, onderwijsafdelingen, onderwijsplannen, cursusgroepen, cursussen, cursusresultaten, toetsresultaten, gebouwen, ruimtes, roostergegevens, nieuwskanalen en nieuwsitems uitgewisseld worden. Het is nu nog niet duidelijk of hierin de Edukoppeling rollen expliciet onderkend worden en of het kunnen routeren hiertussen ondersteund wordt. Als dit het geval is dan is het wenselijk dat hierin dezelfde keuzes worden gemaakt. Omdat het werkingsgebied beperkt is tot de HO-sector verwachten we niet dat er onduidelijkheid is wanneer welke afspraak toegepast moet worden.

Verder heeft ook de internationale standaard SCIM-standaard overlap met het functionele toepassingsgebied. Voor deze standaard kunnen we met zekerheid stellen dat de Edukoppeling rollen hierin niet expliciet onderkend worden en het kunnen routeren hiertussen.

[2] https://www.edustandaard.nl/standaard_afspraken/open-onderwijs-api/open-onderwijs-api/

Een organisatie heeft verschillende soorten API's[3]:

- Open API's: voor ontsluiten van diensten zonder toegangsbeperking bv open data.
- Gesloten API's: voor ontsluiten van diensten met toegangsbeperking bv persoonsgegevens en vertrouwelijke gegevens of diensten voor specifieke partijen.

Zoals eerder aangegeven is dit profiel bedoeld voor vertrouwelijke gegevensuitwisseling en betreft dus gesloten API's. In termen van het Kennisplatform API's wordt gesteld dat het REST/SaaS-profiel wordt toegepast bij access-restricted and purpose-limited API's.

[3] https://geonovum.github.io/KP-APIs/API-strategie-algemeen/

3. REST/SaaS-profiel

Binnen het onderwijs zijn er al vele ketenpartijen die al RESTful gegevens uitwisselen. Deze ketens maken nu zelfstandig keuzes hoe transport, logistiek en beveiliging ingericht moeten worden. Met dit profiel willen we hier meer standaardisatie in doorvoeren, waarbij we ook met name invulling willen geven aan het kunnen routeren tussen de rollen van het SaaS-profiel. We maken zoveel mogelijk gebruik van bestaande standaarden om hier invulling aan te geven.

Met dit profiel gaan we er vanuit dat er (nog) meerdere bronnen zijn voor dezelfde gegevens. Als gevolg hiervan stellen we dat we zowel te maken hebben met het bevragen (pull) van bepaalde bronnen, maar ook het synchroon houden van bronnen op andere locaties middels meldingen (push). We sluiten hierbij aan op de bedrijfstransactiepatronen zoals deze in de Edukoppeling Architectuur[4] zijn gedefinieerd. Verder kunnen partijen bij zowel een push als een pull bij het serviceregister verifiëren of voor de betreffende uitwisseling een ketenpartner gemandateerd is door de betreffende school.

Voor het definiëren van dit profiel wordt er zoveel mogelijk aangesloten bij bestaande (overheidsbrede) standaarden. We willen per voorschrift wel een bepaalde mate van ontkoppeling mogelijk maken en er is zodoende aangegeven hoe zwaar er aan een bepaald voorschrift gevolg moet worden gegeven. Hiermee kunnen we duidelijk aangeven wat de grenzen van dit profiel zijn t.o.v. de mogelijke externe bron waar het voorschrift van wordt overgenomen. Voorschriften zijn aangeduid met Must, Should, Could en Won't waarvoor de volgende definities gelden:

- M – Must have: De Must have eisen moeten gerealiseerd worden. Hier kan niet van afgeweken worden.
- S – Should have: Implementatie conform voorschrift tenzij dit niet mogelijk is én er een work-around beschikbaar is die een vergelijkbaar resultaat mogelijk maakt.
- C – Could have: Dit betreft eisen die gewenst zijn maar waar men vrij is een andere keuze te maken.
- W – Won't have (this time): Deze eisen worden in de context van dit profiel niet toegepast.

3.1. Generieke voorschriften SaaS-profielen

De generieke voorschriften zijn gelijk voor het WUS/SaaS-profiel en REST/SaaS-profiel.

3.1.1. Must: Transportbeveiligingsvoorschriften

De transportbeveiligingsvoorschriften worden voorgeschreven door de Edustandaard Uniforme Beveiligingsvoorschriften (UBV[5]). Het UBV document bevat voorschriften voor verschillende contexten (H2M/M2M). Er is een specifiek Edukoppeling profiel opgenomen zodat partijen dit kunnen gebruiken voor een REST/SaaS-profiel implementatie.

In het UBV Edukoppeling profiel worden PKIoverheid (PKIo) certificaten voorgeschreven welke een OIN bevatten. Het Programma van Eisen van PKIo zorgt ervoor dat er voldoende betrouwbaarheid is rond de identiteit doordat wordt gecontroleerd of de aanvrager de tekenbevoegde van een organisatie is. Hierbij wordt een OIN gecreëerd op basis van de OIN-systematiek en wordt opgenomen in het subject.serialNumber van het certificaat.

In de huidige versie van de API-strategie beveiliging extensie (15-07-2019) wordt als niet-normatieve eis (principe API-11) minimaal TLS 1.3 en een PKIoverheid certificaat voor access-restricted or purpose-limited API authentication voorgeschreven. De toepassing van het REST/SaaS-profiel is

[4] https://www.edustandaard.nl/app/uploads/2019/02/2019-01-31-Edukoppeling-Architectuur-1.2.2-definitief.pdf

[5] Meer informatie via Werkgroep Uniforme Beveiligingsvoorschriften: https://www.edustandaard.nl/standaard_werkgroepen/uniforme-beveiligingsvoorschriften-in-oprichting/

expliciet gericht op access-restricted and purpose-limited API authentication, maar zoals hierboven aangegeven geeft UBV[6] invulling aan de transportbeveiliging.

Er wordt verder aanbevolen om bedreigingen rond beschikbaarheid, integriteit en vertrouwelijkheid te beperken door het opvolgen van OWASP-richtlijnen[7].

3.1.2. Must: Gebruik van openbare internet

De partijen die deel uitmaken van de sector onderwijs maken nagenoeg zonder uitzondering gebruik van het openbare internet om gegevens met elkaar uit te wisselen. Edukoppeling bevat maatregelen om beveiligde gegevensuitwisseling over een dergelijk openbaar netwerk mogelijk te maken. Overigens kan Edukoppeling, net als Digikoppeling, ook toegepast worden in gesloten netwerken.

3.1.3. Must: Toegepast voor zowel bevragingen als meldingen

Voor betrouwbare gegevensoverdracht schrijft Edukoppeling een ander profiel voor dan Digikoppeling. Digikoppeling gebruikt hiervoor het ebMS-profiel. De onderwijssector wil geen complexe varianten introduceren die hetzelfde functionele doel hebben, maar biedt een architectuur die een end-to-end reliable interactieproces mogelijk maakt (in plaats van dit alleen op protocolniveau te regelen zoals Digikoppeling ebMS).

Betrouwbare gegevensoverdracht wordt vaak gekoppeld aan een melding; de initiator van de gegevensuitwisseling wil een andere partij informeren over een gegevenswijziging. De initiator verwacht niet direct een real-time resultaat, anders dan een bevestiging dat de gegevens zijn ontvangen. Op andere (business-)niveaus is het in deze context vaak wel gewenst dat de verwerking van de gegevens of aanverwante resultaten worden teruggekoppeld. Deze patronen kunnen zeer complex zijn en hiermee ook de standaarden die dit soort patronen ondersteunen (zoals ebMS). Er worden in de Edukoppeling Architectuur wel een aantal generieke bedrijfstransactiepatronen beschreven die (deels) kunnen bijdrage aan een betrouwbare gegevensoverdracht.

3.1.4. Must: PKI infrastructuur

De koppelvlakken die bij de gegevensuitwisseling gebruikt worden en de gegevens zelf tijdens transport moeten voldoende beveiligd zijn. Conform Digikoppeling wordt hiervoor met een PKI-infrastructuur en certificaten gewerkt. Deze certificaten worden uitgegeven door Trust Service Providers(TSP). Een TSP is verantwoordelijk voor het controleren van de identiteit van de aanvrager en het opnemen van het identificerend gegeven dat voor deze aanvrager in het certificaat opgenomen moet worden.

3.1.5. Must: PKI-Overheidscertificaten

Edukoppeling schrijft conform Digikoppeling het gebruik van PKIoverheid-certificaten voor. PKI-Overheidscertificaten zijn certificaten die worden uitgegeven in het kader van PKI-overheid van Logius. PKI-overheid certificaten hebben als root (mastercertificaat) ‘Staat der Nederlanden’ en zijn beveiligd naar de laatste stand van techniek. Zodra deze techniek niet meer voldoende is, zal er een nieuw type certificaat met een sterkere encryptiemethode gebruikt moeten worden. Uitgegeven certificaten hebben een beperkte geldigheidstermijn .

[6] Er worden door de UBV nog twee profielen ondersteund. Dit zijn Enkelzijdig TLS: wordt bijvoorbeeld toegepast bij de ontsluiting van Linked Open Data) en tweezijdig TLS: wordt bijvoorbeeld toegepast als de client geïdentificeerd moet worden. Bij dit profiel maakt de keten zelf keuzes rond identificatie van de client en kan dus per keten verschillen.

[7] https://www.owasp.org/index.php/OWASP_API_Security_Project

De certificaten worden uitgegeven door erkende TSP's. De PKI-overheidscertificaten zijn van het niveau STORK QAA 4[8]. Bij de uitgifte hoort 'face-to-face' controle: de houder neemt het certificaat persoonlijk in ontvangst. Het identificerend kenmerk wordt conform Digikoppeling systematiek bepaald (zie identificatie en authenticatie[9]). De TSP die het certificaat uitgeeft heeft de verantwoordelijkheid om de uniciteit van het subject te waarborgen en de identiteit te vermelden in het certificaat in het veld Subject.serialNumber.

Een Digikoppeling-certificaat is een specifiek PKI-overheidscertificaat. Bij de aanvraag hiervan moet men bij de TSP expliciet aangeven dat deze moet voldoen aan de specifieke Digikoppeling-eisen.

3.1.6. Must: Identificatie & Authenticatie

Identificatie

Met een PKI-infrastructuur kan de identificatie en authenticatie van organisaties geregeld worden. Elke partij die via Edukoppeling de gegevensuitwisseling inricht, worden geïdentificeerd op basis van het unieke Overheids Identificatie Nummer (OIN), zie nummersystematiek Digikoppeling en het Edukoppeling Identificatie en Authenticatie document voor details. Voor onderwijsinstellingen is een prefix van 00000007 gereserveerd. Marktpartijen zullen over het algemeen de HR-variant van de nummersystematiek toepassen (prefix 00000001 of 00000003). Hierbij worden de nummers vastgesteld door de TSP, op basis van het door de aanvrager opgegeven KvK-nummer, dat door de TSP wordt gecontroleerd.

In de Edukoppeling Architectuur worden bij de gegevensuitwisseling de volgende rollen onderscheiden:

1. De eindorganisatie is de organisatie die in het kader van zijn doelstellingen samenwerkt met een andere organisatie.
2. De verwerker is een organisatie die in opdracht van de eindorganisatie gegevens verzamelt, opslaat, berekeningen uitvoert, verstrekt en dergelijke.
3. Een logistieke dienstverlener is een organisatie die faciliteert bij de verzending en ontvangst van berichten

Deze rollen worden op verschillende wijze geïdentificeerd. De eindorganisatie wordt geïdentificeerd middels zogenaamde 'TO' en 'FROM' header attributen. Deze versie van het REST profiel beschrijft alleen directe koppelingen tussen verwerkers (die ook logistieke dienstverlener zijn). Inzet van transparante dienstverleners wordt (nog) niet ondersteund. De verwerker wordt (aan beide kanten) geïdentificeerd door het OIN dat in het PKIoverheid-certificaat is opgenomen dat wordt gebruikt bij de TLS-verbinding.

Authenticatie

Bij authenticatie wordt een aangegeven identiteit geverifieerd. De mate van betrouwbaarheid kan hierbij verschillen. Authenticatie levert als het ware de kwaliteit van de identificatie. De PKI-infrastructuur biedt een keten van vertrouwen (chain of trust); de identiteiten zijn met een vastgestelde mate van betrouwbaarheid opgenomen in de certificaten. De organisatie die de identiteit vaststelt (TSP) ondertekent het certificaat met zijn certificaat. Als het certificaat niet is ingetrokken of verlopen is, dan kan men op de inhoud van het 'root' certificaat van de TSP vertrouwen.
De PKI-certificaten kunnen worden gebruikt bij de tweezijdige TLS-verbinding en voor de ondertekening en versleuteling van berichten zoals dit ook in Digikoppeling wordt toegepast. Op basis

[8] https://www.cs.ru.nl/E.Verheul/SIO2019/D2.3_final.pdf
[9] https://www.logius.nl/diensten/digikoppeling/documentatie

van het certificaat en dus ook de identiteit dat hierbij betrokken is kan de identiteit geauthenticeerd worden.

3.1.7. Must: Gebruik Serviceregister

Mandatering

Onderwijsinstellingen hebben zelf services die ze willen registreren, maar het is vaak zo dat een onderwijsinstelling gebruikt maakt van de producten van een SaaS-leverancier. Hierdoor zijn het niet meer de services van de onderwijsinstellingen die geregistreerd worden, maar de services van de SaaS-leverancier. Het serviceregister onderkent deze situatie en ondersteunt tevens de functie om mandateringen te registreren. Het mandaat is de registratie dat een bepaalde SaaS-leverancier namens een bepaalde onderwijsinstelling door middel van een service in een bepaalde context gegevens mag uitwisselen met ketenpartijen.

3.2. Specifieke voorschriften REST/SaaS-profiel

3.2.1. Must: Routering o.b.v. HTTP header

Ook in het REST-profiel is de ondersteuning van het onderscheid tussen eindorganisatie en verwerker en het kunnen routeren van belang.

TODO

Met opmerkingen [ER1]: Besluiten of we voor verplichte logistieke info foutmeldingen definiëren of dat er alleen met HTTP status codes gewerkt wordt

Met opmerkingen [ER2]: Besluiten of query parameters (voorlopig) de beste optie zijn voor routeringskenmerk eindorganisatie.

Met opmerkingen [ER3]: Besluiten of we niet ook overige logistieke info willen opnemen, zoals messageid en relatesto. Afhankelijk van het bedrijfstransactiepatroon kan dit wenselijk zijn als dit niet al in payload opgenomen is.

3.2.2. Could: Berichtbeveiligingsvoorschriften

Voor dit REST/SaaS-profiel zijn momenteel geen berichtbeveiligingsvoorschriften opgenomen. Berichtbeveiliging naast transportbeveiliging is met name relevant in ketens met transparante intermediairs.

3.2.3. Should: Aansluiten op overheidsbrede afspraken rond REST

De basis van dit profiel wordt gevormd door de overheidsbrede afspraken die ontwikkelt zijn als onderdeel van de API-strategie[10] en Design rules extensies[11]. Voor de toelichting zijn een aantal zaken overgenomen van de API strategie van Digitaal Stelsel Omgevingswet (DSO[12]). De principes in deze documenten zijn nog niet formeel vastgesteld en nog in ontwikkeling, maar worden hieronder wel als voorschrift overgenomen. Indien we er in dit profiel op afwijkingen dan is dit expliciet aangegeven. Als we het volledig overnemen dan geldt een Must, alle andere kwalificaties betekenen dat we hier in bepaalde mate van afwijken.

[10] https://geonovum.github.io/KP-APIs/API-strategie-algemeen/ en https://geonovum.github.io/KP-APIs/API-strategie-extensies/ (15-07-2019)

[11] De extensies zijn nog in ontwikkeling, zie https://docs.geostandaarden.nl/api/API-Strategie-ext/

[12] https://aandeslagmetdeomgevingswet.nl/publis/library/219/dso_-_architectuur_-_api-strategie_-_2_0_vastgesteld.pdf

Logius / Kennisplatform API's - API Design rules (Normatief)		
Verplicht	**Principe**	**Toelichting**
MUST	API-01: Operations are Safe and/or Idempotent	Veilig (Safe) betekent in dit geval dat de semantiek is gedefinieerd als alleen-lezen. Dit is van belang als afnemers en tussenliggende systemen gebruik willen maken van caching. Daarnaast kan in een API-gateway een policy zijn ingesteld die slechts 'alleen-lezen' operaties doorlaat. Onder idempotent wordt verstaan dat meerdere identieke verzoeken exact hetzelfde effect hebben als één verzoek. Dit is van belang wanneer in het geval van bijvoorbeeld een falende verbinding, dezelfde berichten opnieuw worden aangeboden.
MUST	API-02: Do not maintain state at the server	De client-toestand wordt volledig bijgehouden door de client zelf.
MUST	API-03: Only apply default HTTP operations[13]	Een RESTful API is een application programming interface die de standaard HTTP-operaties GET, PUT, POST, PATCH en DELETE gebruikt. Dit is van groot belang omdat een API-gateway een policy kan hanteren die alleen specifieke operaties doorlaat.
MUST	API-04: Define interfaces in Dutch unless there is an official English glossary	
MUST	API-05: Use plural nouns to indicate resources	Namen van resources zijn zelfstandige naamwoorden en altijd in het meervoud, zoals verzoeken, activiteiten, locaties. Een uitzondering hierop is de situatie waarin een resource een zogenaamde singleton of een collectie met een kardinaliteit van n:1 of 1:1 betreft. Resource-namen zijn beperkt tot de alfanumerieke reeks en beginnen altijd met een letter.
MUST	API-06: Create relations of nested resources within the endpoint	Als een relatie alleen kan bestaan binnen een andere resource (geneste resource), wordt de relatie binnen het "endpoint" gecreëerd. De afhankelijke resource heeft geen eigen "endpoint". Beperk het aantal geneste sub-resources tot drie (drie niveaus diep).
MUST	API-09: Implement custom representation if supported	
MUST	API-10: Implement operations that do not fit the CRUD model as sub-resources	Acties die niet passen in het CRUD-model worden op de volgende manieren opgelost: • Behandel een actie als een sub-resource. • Alleen in uitzonderlijke gevallen wordt een actie met een eigen "endpoint" opgelost. In dat geval wordt gebruik gemaakt van een werkwoord in gebiedende wijs dat

[13] GET: Indien er query parameters zijn met vertrouwelijke gegevens dienen deze gepseudonimiseerd te zijn

		vooraf wordt gegaan door een underscore, bijvoorbeeld: _zoek
MUST	API-16: Documentation conforms to OAS[14] v3.0 or newer	
MUST	API-17: Publish documentation in Dutch unless there is existing documentation in English or there is an official English glossary available	
MUST	API-18: Include a deprecation schedule when publishing API changes	
WONT	API-19: Allow for a (maximum) 1 year transition period to a new API version	LET OP: De keten bepaald zelf wanneer er een nieuwe versie komt. T.a.v. dit REST profiel wordt er vanuit Edustandaard aangegeven als een nieuwe versie van het profiel komt en een oude versie uitgefaseerd gaat worden.
MUST	API-20: Include the major version number only in ihe URI	
MUST	API-48: Leave off trailing slashes from API endpoints	
MUST	API-51: Publish OAS at a base-URI in JSON-format	
Logius / Kennisplatform API's - API Extensies (Informatief)		
MUST	API-11: Encrypt connections using TLS following the latest NCSC guidelines	LET OP: Het REST/SaaS-profiel heeft eigen voorschriften voor Transportbeveiliging (zie generieke voorschriften UBV). Deze overschrijven dit principe van de API Extensie.
COULD	API-12: Allow access to an API only if an API key is provided	LET OP: Het REST/SaaS-profiel maakt geen gebruik van API-key's en heeft hier geen voorschriften voor. Identificatie is op basis van het OIN in het certificaat dat voor de TLS-verbinding gebruikt wordt.
COULD	API-13: Accept tokens as HTTP headers only	LET OP: Het REST/SaaS-profiel maakt geen gebruik van Tokens en heeft hiervoor geen voorschriften.
COULD	API-14: OAuth 2.0 can be used for authorisation[15]	LET OP: Het REST/SaaS-profiel maakt geen gebruik van OAuth en heeft hiervoor geen voorschriften.
MUST	API-15: Use PKIoverheid certificates for access-restricted or purpose-limited API authentication	LET OP: Het REST/SaaS-profiel heeft eigen voorschriften voor Transportbeveiliging (zie generieke voorschriften). Deze overschrijven dit principe van de API Extensie.
MUST	API-21: Inform users of a deprecated API actively	
MUST	API-22: JSON first - APIs receive and send JSON	API's ontvangen en versturen JSON. (*Bovendien worden alleen JSON-objecten toegepast, dus geen (naamloze) arrays of primitieve datatypes als "top-level" element. Het gebruik van alleen JSON-objecten als "top-level" element vergroot de uitbreidbaarheid.*[16])
COULD	API-23: APIs may provide a JSON Schema	

Met opmerkingen [ER4]: Besluiten wat de beste manier is om hiermee in het document om te gaan, toch een WON'T van maken omdat we naar UBV verwijzen?

[14] https://github.com/OAI/OpenAPI-Specification

[15] Zie toelichting bijlage A

[16] Geen voorschrift vanuit overheidsbrede afspraak

COULD	API-24: Support content negotiation	LET OP: Als er XML gebruikt moet worden dan is het wenselijk het WUS/SaaS-profiel te gebruiken.
MUST	API-25: Check the Content-Type header settings	
MUST	API-26: Define field names in camelCase	
MUST	API-27: Disable pretty print	
MUST	API-28: Send a JSON-response without enclosing envelope	
MUST	API-29: Support JSON-encoded POST, PUT, and PATCH payloads	
MUST	API-30: Use query parameters corresponding to the queryable fields	Gebruik unieke query-parameters die gelijk zijn aan de velden waarop gefilterd kan worden.
MUST	API-31: Use the query parameter sorteer to sort	
MUST	API-32: Use the query parameter zoek for full-text search	
MUST	API-33: Support both * and ? wildcard characters for full-text search APIs	API's die vrije-tekst zoeken ondersteunen kunnen overweg met twee soorten wildcard karakters: * Komt overeen met nul of meer (niet-spatie) karakters ? Komt precies overeen met één (niet-spatie) karakter
WON'T	API-34: Support GeoJSON for GEO APIs	LET OP: Het REST/SaaS-profiel is niet bedoeld voor het uitwisselen van GEO-informatie (GeoJSON / RFC-7946).
WON'T	API-35: Include GeoJSON as part of the embedded resource in the JSON response	LET OP: Het REST/SaaS-profiel is niet bedoeld voor het uitwisselen van GEO-informatie (GeoJSON / RFC-7946).
WON'T	API-36: Provide a POST endpoint for GEO queries	LET OP: Het REST/SaaS-profiel is niet bedoeld voor het uitwisselen van GEO-informatie (GeoJSON / RFC-7946).
WON'T	API-37: Support mixed queries at POST endpoints	LET OP: Het REST/SaaS-profiel is niet bedoeld voor het uitwisselen van GEO-informatie (GeoJSON / RFC-7946).
WON'T	API-38: Put results of a global spatial query in the relevant geometric context	LET OP: Het REST/SaaS-profiel is niet bedoeld voor het uitwisselen van GEO-informatie (GeoJSON / RFC-7946).
WON'T	API-39: Use ETRS89 as the preferred coordinate reference system (CRS)	LET OP: Het REST/SaaS-profiel is niet bedoeld voor het uitwisselen van coördinaatreferentiesysteem (CRS / GeoJSON)
WON'T	API-40: Pass the coordinate reference system (CRS) of the request and the response in the headers	LET OP: Het REST/SaaS-profiel is niet bedoeld voor het uitwisselen van coördinaatreferentiesysteem (CRS / GeoJSON)
WON'T	API-41: Use content negotiation to serve different CRSs	LET OP: Het REST/SaaS-profiel is niet bedoeld voor het uitwisselen van coördinaatreferentiesysteem (CRS / GeoJSON)
MUST	API-42: Use JSON+HAL with media type application/hal+json for pagination	Voor het opnemen van hyperlinks in JSON biedt de Hypertext Application Language (HAL) een set conventies. Voor het gebruik van deze conventies in JSON dient het volgende mediatype gebruikt te worden: application/hal+json HAL is ontworpen voor het bouwen van API's waarin clients door resources

		navigeren door (hyper)links te volgen. De HAL-representatie voor JSON dient te worden gebruikt indien gerelateerde resources (relaties) worden teruggegeven als hyperlinks.
WON'T	API-43: Apply caching to improve performance	We gaan er vanuit dat bij de M2M uitwisselingen (ook bevragingen) caching via HTTP headers niet gebruikt wordt
MUST	API-44: Apply rate limiting	
MUST	API-45: Provide rate limiting information	
MUST	API-46: Use default error handling[17]	API support the default error messages of the HTTP 400 and 500 status code ranges, including the parsable JSON representation (RFC-7807)
MUST	API-47: Use the required HTTP status codes	API's should at least support the following HTTP status codes: 200, 201, 204, 304, 400, 401, 403, 404, 405, 406, 409, 410, 415, 422, 429, 500, and 503.
COULD	API-49: Use public API-key's	LET OP: Het REST/SaaS-profiel maakt geen gebruik van API-key's en heeft hier geen voorschriften voor, zie ook API-12
WON'T	API-50: Use CORS to control access	LET OP: Bij toepassing van het REST/SaaS-profiel is CORS niet relevant.
COULD	API-52: Use OAuth 2.0 for authorisation with rights delegation	LET OP: Het REST/SaaS-profiel maakt geen gebruik van OAuth en heeft hier geen voorschriften voor, zie ook API-14

[17] https://tools.ietf.org/html/rfc7807

4. Bijlage A: OAuth

De Logius/Kennisplatform Design Rules stellen: Use OAuth 2.0 for authorisation (API-14). Als invulling hiervoor is er een OAuth profiel[18] in ontwikkeling. Deze zal na een expertconsultatie door Forum Standaardisatie op ptolu-lijst gezet worden. Als functioneel toepassingsgebied wordt het volgende voorgesteld:

"*Het gebruik van OAuth 2.0 is verplicht voor applicaties waarbij gebruikers (resource owner) toestemming geven (impliciet of expliciet) aan een dienst (van een derde) om namens hem toegang te krijgen tot specifieke gegevens via een RESTful API. Het gaat dan om een RESTful API waar de resource owner recht tot toegang heeft*".

Het feit dat er nu nog geen OAuth profiel is vastgesteld door Forum Standaardisatie en het concept een ander functioneel toepassingsgebied heeft dan dit REST/SaaS-profiel stellen we voor om OAuth nog geen verplicht onderdeel te maken van dit REST/SaaS-profiel. Dit wordt wel gezien als een mogelijk volgend profiel.

[18] https://geonovum.github.io/KP-APIs-OAuthNL/#dutch-government-assurance-profile-for-oauth-2-0

5. Bijlage B: Foutafhandeling

Voor het REST/SaaS-profiel sluiten we voor een groot deel aan op overheidsbrede afspraken. Dit zijn momenteel met name de API-strategie[19] en Design rules extensies[20]. Hierin wordt ook beschreven welke fouten wanneer gecommuniceerd moeten worden.

Principe API-46: Use default error handling

16.1 HTTP status codes[21]

HTTP defines a range of default status codes for APIs. These assist users to of the APIs to handle errors.

Operation	CRUD	Full collection (e.g. `/resource`) specific item (e.g. `/resource/\<id>`)
`POST`	Create	201 (Created), HTTP header `Location` with the URI to the new resource (`/resource/\<id>`) 405 (Method Not Allowed), 409 (Conflict) in case the resource already exists
`GET`	Read *	200 (OK), list of resources. Use paging, filtering and sorting to ease the handling of large collections 200 (OK) single resource, 404 (Not Found) if the ID does not exist or is invalid
`PUT`	Update	405 (Method Not Allowed), except for the purpose to modify or replace every resource in a collection 409 in case a modification is not possible due to the current state of an instance 200 (OK) or 204 (No Content), 404 (Not Found) if the ID does not exist or is invalid
`PATCH`	Update	405 (Method Not Allowed), except for the purpose to replace the full collection. 409 if a modification is not possible due to the current state of an instance. 200 (OK) or 204 (No Content), 404 (Not Found) if the ID does not exist or is invalid
`DELETE`	Delete	405 (Method Not Allowed), except for the purpose to remove the full collection. 200 (OK) or 404 (Not Found) if the ID does not exist or is invalid

Figuur 1 – Overzicht van HTTP operaties en primaire HTTP statuscodes

* *De GET method wordt voor de Read functie gebruikt. In een uitwisseling waar persoonsgegevens worden gebruikt in het request (bijvoorbeeld id = PGN) kunnen aanvullende beveiligingsmaatregelen toegepast worden, zoals het adequaat versleutelen van de persoonsgegevens en/of het voorkomen dat persoonsgegevens in ongewenste logs komen door het toepassen van een filter.*

[19] https://geonovum.github.io/KP-APIs/API-strategie-algemeen/ en https://geonovum.github.io/KP-APIs/API-strategie-extensies/ (15-07-2019)

[20] De extensies zijn nog in ontwikkeling, zie https://docs.geostandaarden.nl/api/API-Strategie-ext/

[21] https://geonovum.github.io/KP-APIs/API-strategie-extensies/#error-handling

HTTP status code	Description
200 OK	Response to a successful GET, PUT, patch or DELETE. Also suitable for POST that does not result in a creation
201 Created	Response to a POST that results in a creation. Should be combined with a location header that points to the location of the newly created resource
204 No Content	Response to a successful request that does not return content (e.g. a DELETE)
304 Not Modified	If HTTP caching headers are applied
400 Bad Request	The request is invalid, e.g. in case the request (body) cannot be interpreted
401 Unauthorized	If no or invalid authentication credentials are supplied. Also useful to display an authentication window if the API is used in a Web browser
403 Forbidden	Response to a successful authentication, but the verified users is not authorised to access the resource
404 Not Found	Response to a request for a non-existing resource
405 Method Not Allowed	Response to an HTTP method that is not allowed for the authenticated user
406 Not Acceptable	Reponse to an unsupported format request (part of content negotiation)
409 Conflict	The request cannot be handled due to a conflict with the current state of the resource
410 Gone	Indicates the resource is no longer available at the requested endpoint. Useful top level response to requests for previous API versions
412 Precondition Failed	The preconditions supplied in one or more fields in the request header have been omitted or failed upon validation by the server
415 Unsupported Media Type	If the wrong content type is supplied as part of the request
422 Unprocessable Entity	Response to a request (body) that is correct but that cannot be handled by the server
429 Too Many Requests	Response if the rate limit has been exceeded.
500 Internal Server Error	If an unexpected error occured and server cannot respond.
503 Service Unavailable	If an API is not available (e.g. due to planned maintenance)

Figuur 2 – Overzicht relevante HTTP statuscodes

5.1. Authorisation[22]

Dit REST/SaaS-profiel realiseert de Identificatie en Authenticatie van de client (verwerker rol) in de transportlaag op basis van tweezijdige TLS en PKI.

Met opmerkingen [ER5]: Besluiten of we de flows overnemen. Zijn op zich wel wenselijk maar hebben een bredere context dan het Edukoppeling SaaS profiel.

De API Design rules gaan ook uit van het mogelijk gebruik van tokens in request header[23]. De onderstaande flow schema's beschrijven de gewenste foutafhandeling.

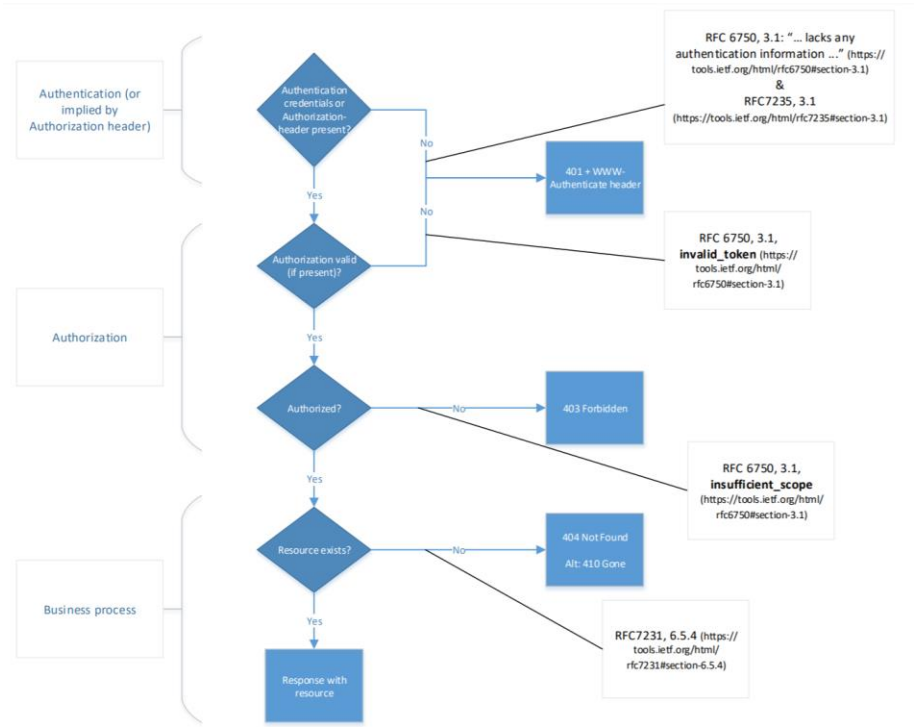
Authorisation errors

In a production environment as little information as possible is to be disclosed. Apply the following rules for returning the status error code 401 Unauthorized, 403 Forbidden, and 404 Not Found.

[22] https://geonovum.github.io/KP-APIs/API-strategie-extensies/#authorisation
[23] https://tools.ietf.org/html/rfc6750#section-3

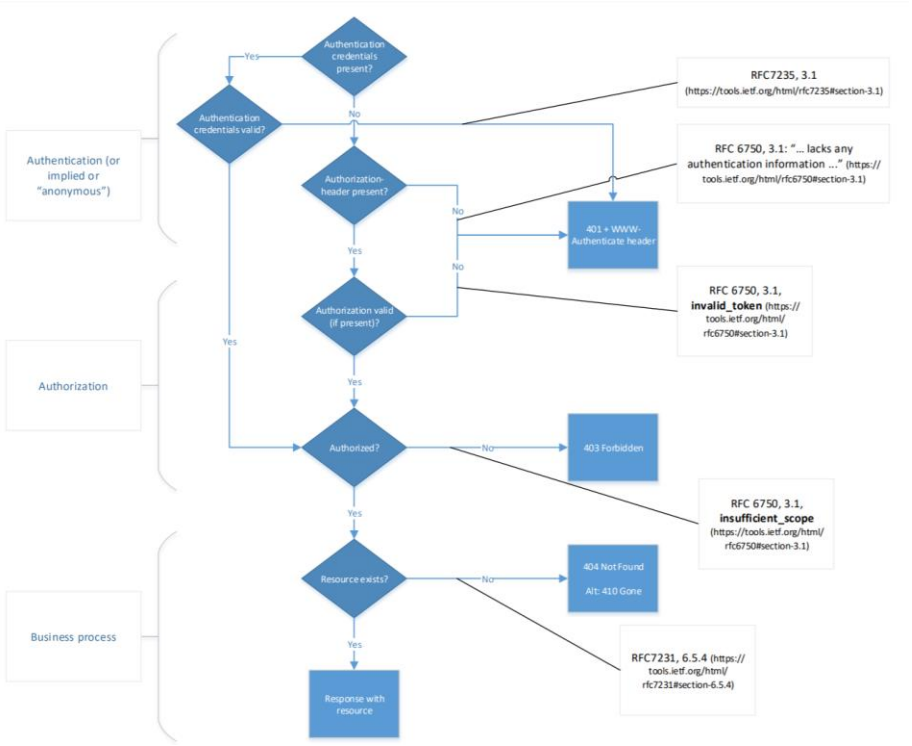
Implicit authentication

When authentication is implicit or when just the presense of an Authorization header (API-Key) is enough for authentication: use the flow chart in figure 1 to determine the correct error code.



Explicit authentication

When authentication is explicit, that is the authentication credentials are actively verfied when present use the flow chart in figure 2 to determine the correct error codes.



Explicit authentication while matching client authorization CNF

When authentication is explicit and there is a check wether the provided authorization confirmation claim (CNF) matches the credentials provided for authentication use the flow chart in figure 3 to estblish the correct error codes.