Edukoppeling

REST/SaaS-profiel

voor

M2M gegevensuitwisseling binnen het onderwijs

Edustandaard

Datum: Oktober 2020

Versie: 0.7

Status: Concept

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 ● Bij de beschrijving van de MoSCoW methode (H3) aangegeven wat de aanduiding ‘MUST NOT’ betekent. ● 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
0.5	Werkgroep Edukoppeling	Juni 2020	Na bespreking 0.4 in WG van 26 mei 2020: ● Keuze routeringskenmerk op basis van query parameters in het request (paragraaf 3.2) ● Bij de beschrijving van de MoSCoW methode (H3) verduidelijkt wat de aanduiding ‘WON’T’ betekent. ● Verduidelijking bij bijlage 5 dat de flowschema’s enkel informatief en nog in ontwikkeling zijn. ● Versie voor openbare consultatie en ROSA Scan
0.6	E. Reinhoud	september 2020	Na openbare consultatie:

			• Tekstuele aanpassingen, API Design Rules • Notatiewijze voorschriften niet meer conform MoSCoW methode, aanduidingen conform API strategie (RFC2119). Betreffende voorschriften hebben een nieuwe aanduiding met vergelijkbare betekenis • Gebruik openbare internet is niet verplicht (aanduiding gewijzigd naar 'MAY') • PKI, PKIo en identificatie OIN samengevoegd • Verschillende tekstuele aanpassingen om generieke voorschriften in lijn te brengen met WUS • Toepassing routeringskenmerk verduidelijkt en voorbeeld aangepast
0.7	Werkgroep Edukoppeling	Oktober 2020	• ROSA scan review verwerkt • Versie voor voorlopige vaststelling in Edustandaard Architectuurraad d.d. 29-10-2020

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 Designrules 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

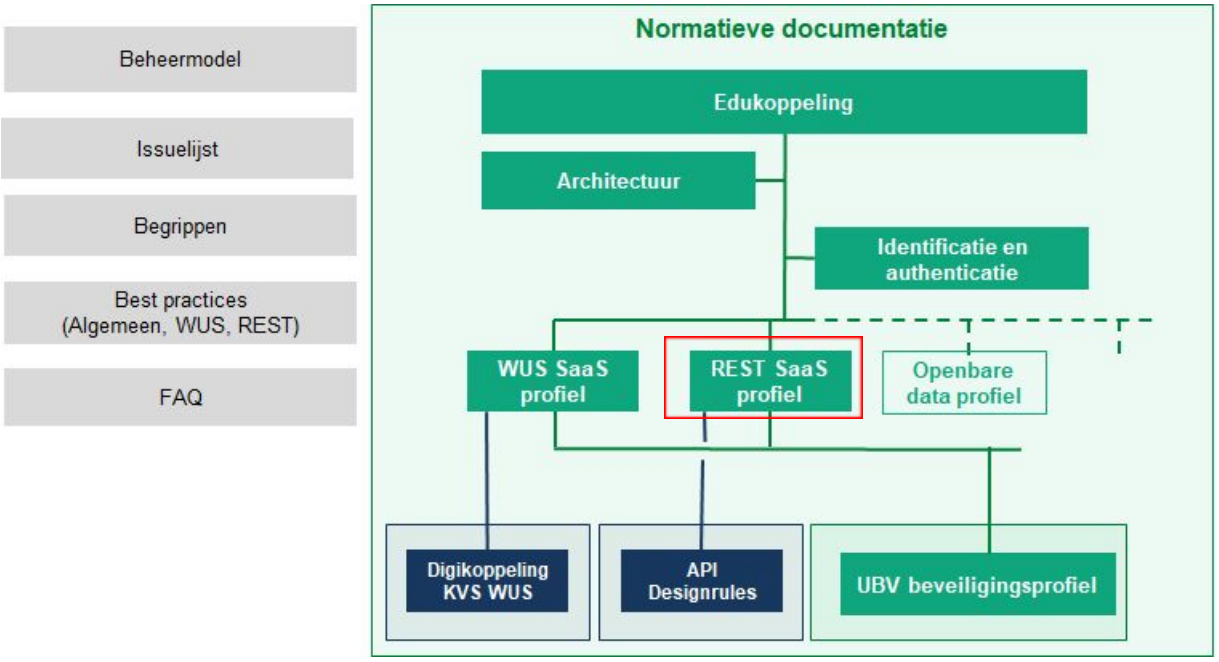
Dit document beschrijft de Edukoppeling REST/SaaS-profiel (verder aangeduid als REST-profiel) en is onderdeel van de Edukoppeling Architectuur. Het doel dat met dit profiel nagestreefd wordt is het op een generieke manier kunnen uitwisselen van gegevens binnen de onderwijssector. Daarbij wordt zowel 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. Voor de inhoudelijke invulling wordt aangesloten op een aantal bestaande RESTful standaarden en afspraken. Een belangrijk onderdeel hiervan wordt gevormd door de Kennisplatform API Design Rules.

[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 voor bevragingen (pull) en meldingen (push) op basis van een request-response uitwisselingspatroon. De gegevens kunnen op basis van de afspraken binnen dit profiel gerouteerd worden tussen een verwerker (SaaS-leverancier) en eindorganisatie. Het profiel kan ook worden toegepast 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 dan wordt het Edukoppeling WUS/SaaS-profiel toegepast. Bij gegevensuitwisseling op basis van XML heeft het de voorkeur om het WUS/SaaS-profiel toe te passen.

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.

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 van de Open Onderwijs API beperkt is tot de HO-sector verwachten we niet dat er onduidelijkheid is wanneer deze of het Edukoppeling REST/SaaS-profiel 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.

2.5 Notatiewijze voorschriften

Voor elk voorschrift wordt aangegeven in welke mate hier invulling aan moet worden gegeven. Hiermee kunnen we duidelijk aangeven wat de grenzen van dit profiel zijn t.o.v. de mogelijke externe bron(nen) waar het voorschrift eventueel van wordt overgenomen. We gebruiken hiervoor de notatiewijze van RFC2119[4]. Deze gebruikt de volgende termen: "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL".

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

[4] https://tools.ietf.org/html/rfc2119

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.

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[5] 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 REST-profiel wordt zoveel mogelijk aangesloten op de nationale afspraken, maar er worden binnen het onderwijs wel een aantal afwijkende voorschriften geformuleerd. Dit hoofdstuk beschrijft deze afwijkende voorschriften. Het REST-profiel conformeert zich aan de API Design Rules, maar wijkt op een aantal punten af, te weten:

1. Het Edukoppeling SaaS-profiel houdt expliciet rekening met gebruik van een openbaar netwerk (Internet).

2. Het Edukoppeling SaaS-profiel stelt eisen aan transportbeveiliging.

3. Het Edukoppeling SaaS-profiel stelt eisen aan identificatie van organisaties.

4. Het Edukoppeling SaaS-profiel kan worden toegepast voor zowel bevragingen als meldingen

5. Het Edukoppeling SaaS-profiel kan het Onderwijs Service Register gebruiken voor verificatie van de mandatering

6. Het Edukoppeling SaaS-profiel stelt aanvullende eisen bij fouten in het eindorganisatie routeringskenmerk.

7. Het Edukoppeling REST/SaaS-profiel stelt aanvullende eisen aan de query string om formele (bv onderwijsinstellingen) en administratieve partijen (bv SaaS-leveranciers) te kunnen onderscheiden.

Er zijn momenteel twee SaaS-profielen (WUS/SaaS-profiel en REST/SaaS-profiel). De eerste zes punten gelden voor beide SaaS-profielen (generieke voorschriften).

3.1 Generieke voorschriften SaaS-profielen

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

3.1.1 MAY: 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.

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

3.1.2 MUST: Transportbeveiligingsvoorschriften o.b.v. UBV

De transportbeveiligingsvoorschriften worden overgenomen van de Edustandaard Uniforme Beveiligingsvoorschriften (UBV[6]). Het UBV-document bevat voorschriften voor transportbeveiliging in verschillende contexten (H2M/M2M). Er is een specifiek Edukoppeling-profiel opgenomen, zodat partijen dit kunnen gebruiken voor een implementatie van een Edukoppeling REST/SaaS-profiel of WUS/SaaS-profiel. Het UBV/Edukoppeling-profiel bevat o.a. voorschriften rond:

- TLS versie
- Ciphers
- SNI
- Poortnummer
- mTLS
- PKIoverheid[7]

Er wordt verder aanbevolen om bedreigingen rond beschikbaarheid, integriteit en vertrouwelijkheid te beperken door het opvolgen van OWASP-richtlijnen[8].

3.1.3 MUST: Identificatie van organisaties op basis van OIN/HRN

3.1.3.1 Rollen

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.

3.1.3.2 Identificatie op basis van OIN/HRN

Deze eerder genoemde rollen worden op verschillende wijze geïdentificeerd. Elke partij die via Edukoppeling de gegevensuitwisseling inricht, worden geïdentificeerd op basis van het unieke Organisatie Identificatie Nummer (zie voor details de OIN nummersystematiek in het Edukoppeling Identificatie en Authenticatie document). Voor onderwijsinstellingen is een prefix van 00000007 gereserveerd. Voor rechtspersonen wordt dit ook wel een HRN genoemd (prefix 00000001 of 00000003).

- De eindorganisatie wordt geïdentificeerd middels zogenaamde ‘TO’ en ‘FROM’ routeringskenmerken. Beide routeringskenmerken zijn feitelijk het OIN van de eindorganisaties.
- De verwerker (SaaS-leverancier) wordt (aan beide kanten) geïdentificeerd door het OIN (HRN) dat in het PKIoverheid-certificaat is opgenomen dat wordt gebruikt bij de mTLS-verbinding.
- Een logistieke dienstverlener is een organisatie die faciliteert bij de verzending en ontvangst van berichten.[9]

---

[6] Meer informatie via Werkgroep Uniforme Beveiligingsvoorschriften: https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/

[7] In het UBV Edukoppeling profiel worden PKIoverheid (PKIo) certificaten voorgeschreven welke een OIN bevatten (zie voor meer details het Edukoppeling I&A document).

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

[9] De rol van verwerker en logistieke dienstverlener wordt vaak door dezelfde partij ondersteund

3.1.3.3 Authenticatie van verwerker op basis van UBV voorschriften (mTLS en PKIoverheid)

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 (Trust Service Providers) ondertekent het certificaat met zijn certificaat.

Een aantal UBV-transportbeveiligingsvoorschriften hebben een relatie met hoe identificatie en authenticatie geregeld wordt. De koppelvlakken die bij de gegevensuitwisseling gebruikt worden moeten voldoende beveiligd zijn. Dit houdt ook in dat er een bepaalde zekerheid is over de identiteit van de partij die bij de gegevensuitwisseling betrokken is. De UBV-transportbeveiligingsvoorschriften voor Edukoppeling sluiten hiervoor aan bij Digikoppeling en vereisen het gebruik van PKIoverheid certificaten (UBV-TLS-PKI-01/ DK-TLS001[10]) en de toepassing van mTLS (DK-TLS002). De certificaten worden uitgegeven door erkende Trust Service Providers (TSP's). Hierbij wordt het OIN/HRN vastgesteld door de TSP, op basis van het door de aanvrager opgegeven KvK-nummer, dat door de TSP wordt gecontroleerd. De PKI-overheidscertificaten zijn van het niveau STORK QAA 4[11]. Bij de uitgifte hoort 'face-to-face' controle: de houder neemt het certificaat persoonlijk in ontvangst. Het identificerend kenmerk wordt conform Digikoppeling OIN nummersystematiek bepaald (zie identificatie en authenticatie[12]). 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.

3.1.4 MAY: Kan worden 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.

3.1.5 MAY: Gebruik Serviceregister voor verificatie van de 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 Onderwijs Service Register (OSR) onderkent deze situatie en ondersteunt tevens de functie om mandateringen te registreren. Het mandaat is de registratie dat een bepaalde SaaS-leverancier (verwerker) namens een bepaalde onderwijsinstelling (eindorganisatie) door middel van een dienst in een bepaalde context via één of meer services gegevens mag uitwisselen met ketenpartijen.

[10] https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/
[11] https://www.cs.ru.nl/E.Verheul/SIO2019/D2.3_final.pdf
[12] https://www.logius.nl/diensten/digikoppeling/documentatie

3.1.6 MUST: Eindorganisatie routeringskenmerk foutmelding

Edukoppeling definieert een aantal categorieën voor foutmeldingen. Deze zijn opgenomen in architectuur. De hier opgenomen foutmeldingen hebben betrekking op de eindorganisatie routeringskenmerken. Hoe deze gecommuniceerd worden zijn uniek per SaaS-profiel.

Omschrijving	Categorie	Toelichting
To parameter ontbreekt	A (Syntax)	Ontvanger niet ingevuld
To parameter is geen OIN	A (Syntax)	Ontvanger parameter is geen valide OIN
From parameter ontbreekt	A (Syntax)	Afzender niet ingevuld
From parameter is geen OIN	A (Syntax)	Afzender parameter is geen valide OIN

3.2 Specifieke voorschriften REST/SaaS-profiel

3.2.1 MUST: Eindorganisatie routeringskenmerken zijn opgenomen als query parameters in het request

Een belangrijk aspect van het SaaS-profiel is het kunnen routeren naar een eindorganisatie. Bij de point-to-point (TLS) verbinding tussen de verwerker rollen moet er gerouteerd kunnen worden naar de eindorganisatie. De eindorganisaties worden middels een FROM en een TO routeringskenmerk gespecificeerd. Het kan zijn dat partijen gegevens uitwisselen voor de zelfde eindorganisatie. Ook al zijn dat de FROM en TO routeringskenmerk hetzelfde deze worden altijd gevuld.

In het REST/SaaS-profiel worden worden de FROM (edu-from) en TO (edu-to) routeringskenmerken als query parameters opgenomen als onderdeel van het request. Hierdoor is het voor de ontvanger direct duidelijk vanuit welke eindorganisatie het bericht verstuurd is, en voor welke het bedoeld is. Gezien het een synchrone point-to-point koppeling betreft (een bilaterale koppeling zonder intermediairs) wordt impliciet gesteld dat de edu-from parameter in het request de edu-to parameter in de response is en de edu-to parameter in het request de edu-from parameter in de response is. We ondersteunen dus de use case waar zowel client als server een verwerker zijn en er aan beide kanten naar eindorganisatie gerouteerd kan worden.

Een voorbeeld wordt hieronder weergegeven. Hierin communiceren twee partijen als verwerker gegevens naar dezelfde school (eindorganisatie).:

https://prod.lasleverancier.nl/v1/service?edu-to=00000007000011BB00001&edu-from=00000007000011BB00000

Het routeringskenmerk wordt geleverd bij het request van elke HTTP methode. Mede op basis van het routeringskenmerk kan bij een serviceregister de mandatering gevalideerd worden. Als één of beide routeringskenmerken in het request ontbreken moet er een foutmelding gegeven worden.

3.2.2 MAY: 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[13]. Het gaat met name om de API Design Rules[14] en en Extensies[15] die

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

[14] https://docs.geostandaarden.nl/api/API-Designrules/ (17-01-2020)

[15] https://geonovum.github.io/KP-APIs/API-strategie-extensies/

momenteel ook in behandeling is bij Forum Standaardisatie[16] om op de PTOLU-lijst te komen. De principes in deze documenten zijn nu dus nog niet formeel vastgesteld en nog in ontwikkeling.

Indien het REST/SaaS-profiel niet Fully Conformant is met de API Design rules of Extensies wordt bij de toelichting in Tabel 1 met “LET OP…” aangegeven wat de invulling voor dit profiel is. Bij de toelichting zijn soms een aantal zaken overgenomen uit de API strategie van Digitaal Stelsel Omgevingswet (DSO[17]).

Logius / Kennisplatform API’s - API Design Rules (Normatief)		
Niveau van conformance*	**Principe**	**Toelichting**
Fully Conformant	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.
Fully Conformant	API-02: Do not maintain state at the server	De client-toestand wordt volledig bijgehouden door de client zelf.
Fully Conformant	API-03: Only apply default HTTP operations[18]	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.
Fully Conformant	API-04: Define interfaces in Dutch unless there is an official English glossary	
Fully Conformant	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.

[16] https://www.forumstandaardisatie.nl/open-standaarden/in-behandeling

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

[18] GET: Indien er query parameters zijn met vertrouwelijke gegevens is het wenselijk aanvullende beveiligingsmaatregelen toe te passen, zoals adequaat versleutelen en/of voorkomen ongewenste logging dmv filter.

Fully Conformant	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).
Fully Conformant	API-09: Implement custom representation if supported	
Fully Conformant	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 vooraf wordt gegaan door een underscore, bijvoorbeeld: _zoek
Fully Conformant	API-16: Documentation conforms to OAS[19] v3.0 or newer	
Fully Conformant	API-17: Publish documentation in Dutch unless there is existing documentation in English or there is an official English glossary available	
Fully Conformant	API-18: Include a deprecation schedule when publishing API changes	
Irrelevant	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.
Fully Conformant	API-20: Include the major version number only in ihe URI	
Fully Conformant	API-48: Leave off trailing slashes from API endpoints	
Fully Conformant	API-51: Publish OAS at a base-URI in JSON-format	
Logius / Kennisplatform API’s - API Extensies (Informatief)		
Consistent	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.
Consistent	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.
Consistent	API-13: Accept tokens as HTTP headers only	LET OP: Het REST/SaaS-profiel maakt geen gebruik van Tokens en heeft hiervoor geen voorschriften.

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

Consistent	API-14: OAuth 2.0 can be used for authorisation[20]	LET OP: Het REST/SaaS-profiel maakt geen gebruik van OAuth en heeft hiervoor geen voorschriften.
Fully Conformant	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.
Fully Conformant	API-21: Inform users of a deprecated API actively	
Fully Conformant	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.*[21])
Fully Conformant	API-23: APIs may provide a JSON Schema	
Fully Conformant	API-24: Support content negotiation	LET OP: Als er XML gebruikt moet worden dan is het wenselijk het WUS/SaaS-profiel te gebruiken.
Fully Conformant	API-25: Check the Content-Type header settings	
Fully Conformant	API-26: Define field names in camelCase	
Fully Conformant	API-27: Disable pretty print	
Fully Conformant	API-28: Send a JSON-response without enclosing envelope	
Fully Conformant	API-29: Support JSON-encoded POST, PUT, and PATCH payloads	
Fully Conformant	API-30: Use query parameters corresponding to the queryable fields	Gebruik unieke query-parameters die gelijk zijn aan de velden waarop gefilterd kan worden.
Fully Conformant	API-31: Use the query parameter sorteer to sort	
Fully Conformant	API-32: Use the query parameter zoek for full-text search	
Fully Conformant	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
Irrelevant	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).
Irrelevant	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).
Irrelevant	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).

[20] Zie toelichting bijlage A

[21] Geen voorschrift vanuit overheidsbrede afspraak

Irrelevant	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).
Irrelevant	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).
Irrelevant	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)
Irrelevant	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)
Irrelevant	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)
Fully Conformant	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.
Fully Conformant	API-43: Apply caching to improve performance	
Fully Conformant	API-44: Apply rate limiting	
Fully Conformant	API-45: Provide rate limiting information	
Fully Conformant	API-46: Use default error handling[22]	API support the default error messages of the HTTP 400 and 500 status code ranges, including the parsable JSON representation (RFC-7807)
Fully Conformant	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.
Consistent	API-49: Use public API-keys	LET OP: Het REST/SaaS-profiel maakt geen gebruik van API-key's en heeft hier geen voorschriften voor, zie ook API-12
Irrelevant	API-50: Use CORS to control access	LET OP: Bij toepassing van het REST/SaaS-profiel is CORS niet relevant.
Consistent	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

Tabel 1 - Overzicht Kennisplatform API's principes en de mate van compliance hieraan binnen het Edukoppeling REST/SaaS- profiel

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

* Geeft aan in welke mate we de basis (bijvoorbeeld API Design rules) volgen. Zie bijlage Niveaus van conformance.

3.2.4 MUST: Foutafhandeling

HTTP status codes[23]: *API Designrules Extensions principe API-46: Use default error handling*

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.*

[23] 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

Hieronder wordt weergegeven hoe het REST/SaaS-profiel invulling geeft aan de foutafhandeling rond o.a. de routeringskenmerken.

HTTP status code	Omschrijving	Categorie	Domein	Toelichting
400	Bad Request	Syntax	EK	To routeringskenmerk ontbreekt
400	Bad Request	Syntax	EK	To routeringskenmerk is geen OIN
400	Bad Request	Syntax	EK	From routeringskenmerk ontbreekt

400	Bad Request	Syntax	EK	From routeringskenmerk is geen OIN
401	Unauthorized			Authenticatiefout · mTLS PKIo vereist · TLS invalid client certificate
403	Forbidden			Autorisatiefout · TLS: OIN in certificaat niet geautoriseerd · TLS: OIN in certificaat niet gemandateerd om voor school (From routeringskenmerk) in context van deze service gegevens uit te wisselen
404	Not Found			De server kan resource niet vinden. Het endpoint kan echter valide zijn. Een server kan deze code sturen ipv code 403 om het eventuele bestaan van de resources niet vrij te geven aan een niet geautoriseerde client

Bijlage A: Niveaus van conformance

Met de niveaus van conformance kunnen we aangeven in welke mate het REST/SaaS-profiel aansluit op de specificaties waar het deels op gebaseerd is (API Design rules en Extensies). Het is gebaseerd op de TOGAF-categorieën[24] die ook voor de ROSA Architectuurscan worden gebruikt. We hanteren hierbij de onderstaande invulling voor de gebruikte termen.

Irrelevant	Er is geen relatie tussen het REST/SaaS-profiel en het principe van de API Design rules of Extensies. Deze mate van conformance wordt bijvoorbeeld gebruikt als we stellen dat het principe niet binnen de context van het REST/SaaS-profiel wordt gebruikt.
Consistent	Er is overlap tussen het REST/SaaS-profiel en het principe van de API Design rules of Extensies. Binnen die overlap is het REST/SaaS-profiel conform het principe, de overlap is echter niet volledig, de scope van het principe wordt niet volledig overgenomen, en het REST/SaaS-profiel heeft onderdelen die een relatie hebben met het principe maar er niet door worden gedekt. Deze mate van conformance wordt bijvoorbeeld gebruikt als we in het REST/SaaS-profiel functioneel een andere invulling geven aan het principe, maar de aanvullende aspecten die het principe introduceert niet willen uitsluiten.
Compliant	Het REST/SaaS-profiel valt volledig binnen het principe van de API Design rules of Extensies. De scope van het principe gaat wel verder dan de scope van het betreffende functionele deel van het REST/SaaS-profiel.
Conformant	Het principe van de API Design rules of Extensies geeft slechts een beperkte dekking voor wat we op dat deel met het REST/SaaS-profiel willen bereiken.
Fully conformant	Het principe van de API Design rules of Extensies dekt het geheel van wat we op dat deel met het REST/SaaS-profiel willen bereiken. De scope van het principe valt samen met de scope van het betreffende functionele deel van het REST/SaaS-profiel.
Non-conformant	Er is (functionele) overlap tussen het principe van de API Design rules of Extensies en het REST/SaaS-profiel, en binnen die overlap is het REST/SaaS-profiel niet conform het principe van de API Design rules of Extensies.

[24] https://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/48_conformance.png