Edukoppeling

***M2M gegevensuitwisseling binnen het onderwijs***

*Mandated Data eXchange (MDX) Secure API OAuth profile*

Edustandaard

Datum: april 2023

Versie: 0.3

**Inhoudsopgave**

Inhoud

[1. Status van dit document 4](#_Toc131076725)

[1.1. Documenthistorie 4](#_Toc131076726)

[1.2. Overzicht actuele documentatie en compliance 5](#_Toc131076727)

[2. Inleiding 6](#_Toc131076728)

[2.1. Aanleiding voor het ontwikkelen van de Edukoppeling standaard 6](#_Toc131076729)

[2.2. MDX Secure API OAuth profile 6](#_Toc131076730)

[2.3. Doel en doelgroep 7](#_Toc131076731)

[2.4. Positionering binnen Edukoppeling Architectuur 7](#_Toc131076732)

[2.5. Functioneel toepassingsgebied 8](#_Toc131076733)

[2.6. Notatiewijze voorschriften 9](#_Toc131076734)

[2.7. Leeswijzer 9](#_Toc131076735)

[3. Uitgangspunten 10](#_Toc131076736)

[4. Voorschriften OAuth client credentials profiel 14](#_Toc131076737)

[4.1. MAY:Authorization Server metadata 14](#_Toc131076738)

[4.2. MAY: Client registratie 14](#_Toc131076739)

[4.3. MUST: Interacties 16](#_Toc131076740)

[4.3.1. Verzoek naar Token Endpoint (STAP #1) 16](#_Toc131076741)

[4.3.2. Verwerking v/h client verzoek bij het AS Token Endpoint 18](#_Toc131076742)

[4.3.3. Antwoord van Token Endpoint (STAP #2): verwerking succesvol 19](#_Toc131076743)

[4.3.4. Antwoord van Token Endpoint (STAP #2): foutmelding RFC6749 (§ 5.2) 19](#_Toc131076744)

[4.3.5. Interactie met protected resource (STAP #3) 20](#_Toc131076745)

[5. Voorschriften REST 21](#_Toc131076746)

[5.1. MUST: API Design conform het Digikoppeling REST profiel 21](#_Toc131076747)

[5.2. MUST: Routeringskenmerken zijn query parameters in het request naar het token endpoint van de Authorization Server. 23](#_Toc131076748)

[5.3. MAY: Berichtbeveiligingsvoorschriften 24](#_Toc131076749)

[5.4. MUST: Foutafhandeling m.b.t. het routeringskenmerk 24](#_Toc131076750)

[6. Overige voorschriften 26](#_Toc131076751)

[6.1. MAY: Gebruik van openbare internet 26](#_Toc131076752)

[6.2. MUST: Transportbeveiliging op basis van (m)TLS 26](#_Toc131076753)

[6.3. MUST: Identificatie en authenticatie van organisaties 26](#_Toc131076754)

[6.4. MAY: Kan worden toegepast voor zowel bevragingen als meldingen 28](#_Toc131076755)

[6.5. MUST: Toepassing van het MDX OSR protocol 28](#_Toc131076756)

[6.6. MUST: Foutafhandeling m.b.t. het routeringskenmerk 28](#_Toc131076757)

[7. Bijlage A: OAuth framework 29](#_Toc131076758)

[7.1. Inleiding 29](#_Toc131076759)

[7.2. OAuth client credentials profiel 29](#_Toc131076760)

[Stap 1. Request naar token endpoint 30](#_Toc131076761)

[Stap 2. Respons van token endpoint 30](#_Toc131076762)

[Stap 3. Resource-interactie 30](#_Toc131076763)

[8. Bijlage B: Overwegingen & acties 31](#_Toc131076764)

[8.1. Overwegingen 31](#_Toc131076765)

[8.2. Acties 32](#_Toc131076766)

[9. Bijlage C: Uitgangspunten discussiestuk versie 0.3 33](#_Toc131076767)

[10. Bijlage D: Bronnen 35](#_Toc131076768)

[11. Bijlage E: Begrippen 37](#_Toc131076769)

# Status van dit document

Dit document is een concept van de Edukoppeling MDX Secure API OAuth profile versie 1.0.

# Documenthistorie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versie** | **Status** | **Auteur** | **Datum** | **Opmerking** |
| 0.1 | Concept | E. Reinhoud (BES) | Januari 2023 | Initiële versie gebaseerd op uitgangspunten in het discussiestuk (versie 0.3). |
| 0.2 | Concept | E. Reinhoud (BES) | Februari 2023 | Gebaseerd op WG bijeenkomst 30 januari 2023. Zie document versie 0.2 |
| 0.3 | Concept | E. Reinhoud (BES) | April 2023 | * Deze versie verwijst niet meer naar het Edukoppeling REST-profiel, maar bevat zelf de voorschriften die deels overeenkomen met de voorschriften uit het REST-profiel (zie Voorschriften MDX OSR protocol en API Design Rules ). * De reden om niet meer voort te bouwen op het hele REST-profiel is omdat we in deze versie een duidelijk onderscheid maken in het koppelvlak naar het token endpoint van de AS (STAP #1) en het koppelvlak naar de protected resource (STAP#3). Het koppelvlak naar de protected resource sluit nu beter aan bij wat bij (RESTfull) platformen verwacht wordt. Dit betekent geen gebruik van mTLS, maar wel TLS conform het UBV TLS basisprofiel. * Het request naar het token endpoint van de AS voldoet wel aan alle eisen die we ook bij het REST en WUS profiel onderkennen. Dit betekent naast het toepassen van mTLS/PKIO/OIN ook het routeringskenmerk in de query string en toepassing van het MDX OSR protocol. * Verduidelijken van de uitgangspunten en opgenomen in eigen hoofdstuk (bijlage met uitgangspunten uit het discussiedocument zijn verwijderd). * In hoofdstuk 5 de overige voorschriften opgenomen. Dit zijn voor een groot deel de voorschriften die in de context van alle MDX-profielen (ook WUS/REST) gelden. Er zijn nog steeds generieke voorschriften maar deze verschillen wel per profiel (bijv. OAuth vereist nu niet altijd mTLS) * Aanpassing alle figuren * Verwerking overige opmerkingen * Herstructurering document |

# Overzicht actuele documentatie en compliance

Dit document (MDX Secure API OAuth profile) staat niet op zichzelf (zie Figuur 1). Het document “Overzicht actuele documentatie en compliance juni 2023[[1]](#footnote-2)” geeft een actueel overzicht van de verschillende normatieve en informatieve Edukoppeling documenten.

# Inleiding

# Aanleiding voor het ontwikkelen van de Edukoppeling standaard

De aanleiding voor de introductie van Edukoppeling in het onderwijsdomein is een steeds groter wordende stroom van geautomatiseerde machine-machine uitwisselingen in het onderwijs. Dit wordt veroorzaakt door vernieuwingen in het onderwijs zelf, in wetgeving en in de beschikbare techniek. In toenemende mate lopen de processen over organisaties heen, tussen onderwijsorganisaties (zowel op bestuursniveau van de onderwijsaanbieders, de “scholen”) onderling, tussen onderwijsorganisaties en overheidsorganisaties en tussen onderwijsorganisaties en private onderwijsgerelateerde organisaties. En vaak, als er iets nieuws komt, wordt er dan pas nagedacht over de benodigde wijze van koppelen. Als men niet oppast worden er evenveel verschillende soorten van koppelingen bedacht als er geautomatiseerde processen zijn. Dat is nadelig, omdat hiervoor veel kennis nodig is, dit onnodig veel en kostbaar onderhoudt vergt, dit de interoperabiliteit en aanpasbaarheid hindert. Met Edukoppeling verandert dat. Edukoppeling is een meervoudig inzetbare wijze van koppelen waarvan de ontwikkeling en het beheer gemeenschappelijk wordt aangepakt. Edukoppeling is tevens een open standaard, wat maakt dat partijen met een lage drempel kunnen deelnemen, wat gunstig is voor het onderwijs.

In het onderwijs is het normaal geworden dat onderwijsinstellingen veel van hun processen laten ondersteunen door zogeheten SaaS-diensten (diensten ‘in the cloud’)*.* Dit geldt voor onderwijskundige processen als ook voor hun administratieve processen. Het Edukoppeling Mandated Data eXchange[[2]](#footnote-3) (MDX) protocol en verwante profielen houden met deze ontwikkeling rekening. De diensten van leveranciers waar een onderwijsorganisatie gebruik van maakt beheren gegevens (administraties) en wisselen vaak namens de onderwijsorganisatie gegevens uit met ketenpartijen. De Edukoppeling Mandated Data eXchange profielen houden expliciet rekening met het uitwisselen van gegevens tussen verwerkers (bijvoorbeeld leveranciers) namens een eindorganisatie (bijvoorbeeld een onderwijsorganisatie).

# MDX Secure API OAuth profile

Dit document beschrijft het Edukoppeling MDX Secure API OAuth profile (verder aangeduid als OAuth-profiel) en is onderdeel van de Edukoppeling Architectuur. Het is een profiel dat overeenkomsten heeft met het MDX Secure API REST profiel (REST-profiel). Dit geldt met name rond een aantal belangrijke onderdelen van het koppelvlak met het token endpoint van de Authorization Server. Bij dit koppelvlak is het Mandated Data eXchange (MDX) protocol, mTLS/PKIo en het gebruik van het routeringskenmerk van toepassing. De verwerker die de client beheert zal het mandaat van de verwerker controleren die de combinatie van de protected resource en authorization server beheert en visa versa. Om de client te kunnen relateren aan de verwerker moet de client bij registratie gekoppeld worden aan het OIN van de verwerker. Om de Authorization Server te kunnen relateren aan de verwerker wordt de issuer identifier[[3]](#footnote-4) bij registratie gekoppeld worden aan het OIN van de verwerker.

Net als bij het REST-profiel geldt ook hier dat we er vanuit gaan dat het een synchrone point-to-point koppeling betreft en het routeringskenmerk in de respons is de inverse van het request[[4]](#footnote-5).

Toegang tot de protected resource is anders ingericht dan het koppelvlak met het token endpoint van de Authorization Server. Voor het koppelvlak met de protected resource hebben we toegang ingericht die goed aansluit bij gangbare implementaties. Toegang tot de protected resource kan alleen plaatsvinden met een geldig OAuth toegangstoken in het request. Er wordt bij de verbinding met de protected resource GEEN mTLS toegepast maar TLS en het toegangstoken is opgenomen in de HTTPS Authorization request header field. Zie voor meer details hoofdstuk 4.

# Doel en doelgroep

Het doel dat met dit profiel nagestreefd wordt is het op een generieke manier kunnen uitwisselen van gegevens binnen de onderwijssector. Het profiel ondersteunt zowel het scenario waarbij een Eindorganisatie zijn systeem zelf beheert in de eigen ICT-infrastructuur, als het scenario waarbij de Eindorganisatie deze als (SaaS-)dienst van een verwerker (leverancier) afneemt.

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. Edukoppeling is voor een groot deel compliant aan de overheidsstandaard Digikoppeling. De Edukoppeling-documentatie dient derhalve naast de Digikoppeling-documentatie gebruikt te worden.

De lezer van dit document willen wij vragen om zaken die ontbreken of onduidelijk zijn te melden bij de beheerorganisatie Edustandaard[[5]](#footnote-6).

# Positionering binnen Edukoppeling Architectuur

Het Edukoppeling OAuth-profiel is onderdeel van de Edukoppeling Architectuur. Het beschrijft welke aanvullende eisen er gelden rond het gebruik van de OAuth 2.0 client credentials profile. We maken hierbij gebruik van de internationale open standaarden zoals RFC6749 en RFC6750[[6]](#footnote-7).

Afbeelding met diagram

Automatisch gegenereerde beschrijving

Figuur 1 - Positionering van OAuth-profiel binnen de Edukoppeling Architectuur

# Functioneel toepassingsgebied

Het functionele toepassingsgebied van dit OAuth-profiel[[7]](#footnote-8) betreft M2M-gegevensuitwisseling via een point-to-point verbinding voor uitwisseling van vertrouwelijke gegevens via een gesloten API[[8]](#footnote-9). Er worden bevragingen (pull) en meldingen (push) op basis van een request-response uitwisselingspatroon ondersteund. De client is in deze context geen browser, maar een systeem (confidential client). De systemen worden beheerd door verwerkers en voeren de uitwisseling uit op basis van een mandaat van een eindorganisatie (het MDX OSR protocol is van toepassing). Het functionele toepassingsgebied komt vrijwel overeen met dat van het REST-profiel met dit verschil dat het gebruik van de authorization server het mogelijk maakt om toegang tot de protected resource volledig te baseren op een OAuth toegangstoken.

De gegevens kunnen op basis van de afspraken binnen dit profiel gerouteerd worden van verwerker naar eindorganisatie op basis van het routeringskenmerk in het request naar de Authorization Server. Het profiel kan ook worden toegepast indien de eindorganisatie ook zelf de rol van verwerker heeft.

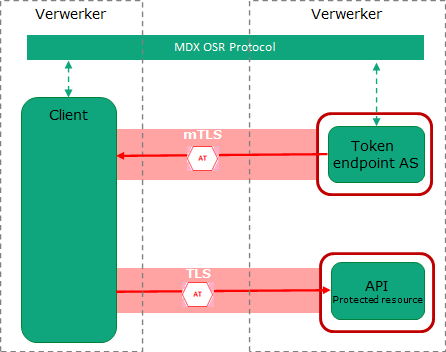
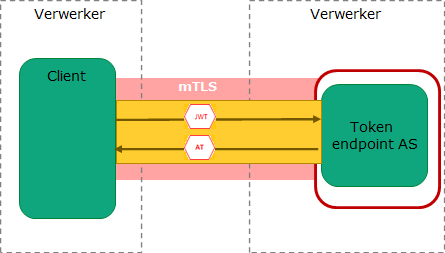
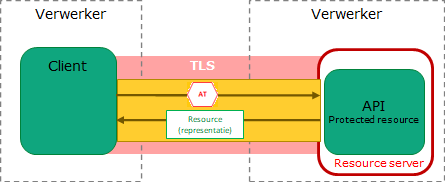
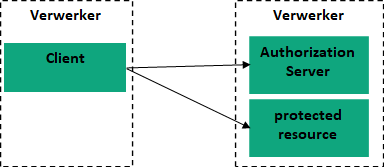
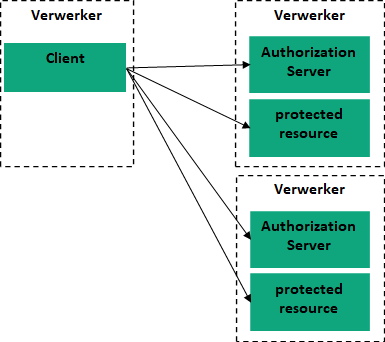
# Notatiewijze voorschriften

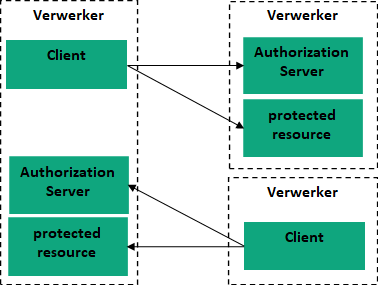
Voor elk voorschrift wordt aangegeven in welke mate hier invulling aan moet worden gegeven. Hiermee kunnen we duidelijk aangeven wat de grenzen van dit profiel zijn ten opzichte van de mogelijke externe bron(nen) waar het voorschrift eventueel van wordt overgenomen. We gebruiken hiervoor de notatiewijze van RFC2119[[9]](#footnote-10). Deze gebruikt de volgende termen: "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL".

# Leeswijzer

[todo]

# Uitgangspunten

1. Dit OAuth-profiel heeft vrijwel hetzelfde functionele toepassingsgebied als het REST-profiel. Het betreft M2M-gegevensuitwisseling via een point-to-point verbinding voor uitwisseling van vertrouwelijke gegevens via een gesloten API. De systemen (client, authorization server en resource server) worden beheerd door verwerkers en voeren de uitwisseling uit op basis van een mandaat van een eindorganisatie (het MDX OSR protocol is van toepassing). Er wordt van een synchrone point-to-point koppeling uitgegaan en er wordt impliciet gesteld dat het routeringskenmerk in de respons de inverse is van het request.
2. Het is gebaseerd op het OAuth client credentials profiel (zie Bijlage A: OAuth framework) en hiermee kan het functionele toepassingsgebied meer toegespitst worden op RESTful platformen. Een confidential client haalt bij het token endpoint van de Authorization Server een Access Token (AT) op en krijgt op basis hiervan toegang tot protected resource.
3. We gaan ervan uit dat een verwerker het beheer voert over een Authorization Server/Resource Server combinatie en een andere verwerker de client. Op basis van dit uitgangspunt worden de eisen rond het MDX OSR protocol, mTLS/PKIo en het routeringskenmerk alleen toegepast op het token endpoint koppelvlak van de authorization server. Een geldig Access Token van de Authorization Server wil dus (ook) zeggen dat de client op het niveau van Verwerker (OIN) geautoriseerd[[10]](#footnote-11) is en dat bij de uitwisseling met de resource server het MDX OSR protocol niet opgevolgd hoeft te worden.
4. Op het niveau van OAuth kan de client bij het request naar het token endpoint van de Authorization Server worden geautoriseerd op basis van een JWT token.
   1. Het request wordt beveiligd op basis mTLS (UBV TLS edukoppeling profiel)
   2. Het kan zijn dat de Authorization Server fijnmazige autorisaties ondersteunt met de optionele claims scope en/of edu\_org\_id[[11]](#footnote-12). De client neemt deze op in de JWT en de Authorization Server controleert of de betreffende rechten voor de client gelden.
   3. Na succesvolle autorisatie levert de Authorization Server een Access Token met een bepaalde geldigheidsduur (max 1 uur).
5. Bij het request naar de resource server wordt de client geautoriseerd op basis van het Access Token.
   1. Het request wordt beveiligd op basis van TLS (UBV TLS basisprofiel).
   2. Een fijnmazige autorisatie kan worden ingericht doordat het AT ook de scope en/of edu\_org\_id claim bevat.
   3. De toepassing van een fijnmazige autorisatie kan betekenen dat op het niveau van de protected resource (API) niet eenduidig is te bepalen welke gegevens met een bepaalde client in bepaalde context worden uitgewisseld.
6. Dit OAuth-profiel houdt dus rekening met scenario’s waarbij aanvullende randvoorwaarden kunnen gelden om te kunnen komen tot een meer fijnmazige autorisatie.
   1. edu\_org\_id: De client kan aangeven dat het een uitwisseling betreft in opdracht van een onderwijsorganisatie. Het gaat dan om een specifieke set leerlingen/medewerkers (betrokkenen) gegevens. De beheerder van de AS/RS combi heeft een onderwijsorganisatie-medewerker de data autorisatie laten registreren. Wanneer deze fijnmazige autorisatie geregistreerd wordt en hoe de client over de edu\_org\_id komt te beschikken bij de aanvraag voor een access token valt wel buiten de scope van dit OAuth-profiel. Partijen kiezen hier hun eigen inrichtingsvorm.
   2. scope: De client kan met de scope aangeven voor welke resource en/of actie een access token aangevraagd wordt. Wat de scope precies aangeeft (een actie, een resource, of een set attributen van een bepaalde resource of een combinatie van de drie) wordt niet voorgeschreven. Wel wordt het aanbevolen om de scope naming convention van dit OAuth-profiel toe te passen. Hiermee kan zowel de resource als de actie gedefinieerd worden. Bij de toepassing van de scope geldt wel dat de AS en RS hierover een eenduidig beeld moeten hebben.
   3. resource: Als de scope niet toegepast wordt of geen (logische) verwijzing naar een resource bevat dan kan deze met de “resource” claim expliciet aangegeven worden. Bij een authorization server die verschillende API’s ondersteunt kan hiermee expliciet doorgegeven worden voor welke API het Access Token gebruikt wordt.
7. Het Access Token is een bearer token. Een bearer token is een token dat de drager toegang geeft tot de protected resource. Een bearer token is portable en draagt bij aan de ontkoppeling van systemen. Het maakt de API-beveiliging stateless omdat (alle) informatie voor de autorisatievraag in het token zijn opgenomen. Dit kan echter ook als een nadeel gezien worden als het token niet goed beveiligd wordt. De drager is impliciet geautoriseerd en dat is iedereen die over een kopie van het token weet te beschikken. Ze moeten dus bij opslag en transport beveiligd worden. Bij transport bijvoorbeeld met behulp van Transport Layer Security (TLS). Een meer veilig type token is een proof-of-possession (PoP) token[[12]](#footnote-13). Hierbij is het token (cryptografisch) beveiligd doordat deze aan een specifieke drager is gebonden. De drager moet kunnen aantonen dat deze in bezit is van de “sleutel” van het “slot” dat in het token is opgenomen.
   1. Het gebruik van een bearer token betekent dat de rechten “bevroren” zijn gedurende de levensduur van het token. Mede hierom[[13]](#footnote-14) is het wenselijk om een access token een zo kort mogelijke levensduur te geven.
8. Het Access Token is zelfbeschrijvend[[14]](#footnote-15)
   1. Dit OAuth-profiel gebruikt o.a. RFC9068 voor het definiëren van het Access Token. Er wordt dus geen gebruik gemaakt van een token dat een referentie is naar authorisatie-informatie bij de Authorization Server. Het gebruik van een zelfbeschrijvend Access Token voorkomt dat de Resource Server een call naar de Authorization Server moet maken (introspection[[15]](#footnote-16)) om tot een autorisatiebesluit te komen.
9. Token revocation mag worden toegepast, maar valt verder buiten de specificaties van dit OAuth-profiel.
   1. Het zelfbeschrijvende bearer token heeft een korte levensduur (een uur of korter). Bij een korte levensduur lijkt token revocation niet noodzakelijk. We nemen aan dat een client niet binnen een uur besluit om een token toch niet te gebruiken, maar we sluiten dit niet uit.
10. Zoals aangegeven gaan we er vanuit dat één partij de resource server en de betreffende authorization server beheert. Een verwerker kan dus bij een bepaalde uitwisseling de combinatie van een OAuth authorization server en de resource server rol hebben of de client rol.
11. In verschillende ketens en zelfs binnen een bepaalde keten kunnen er meerdere Authorization Servers en Resource Servers combi’s zijn die door verschillende verwerkers worden beheerd. Een partij die (binnen een bepaalde ketensamenwerking) als client koppelingen heeft met API’s van verschillende partijen moet dus bij de AS van alle betreffende partijen een Access Token aanvragen voor de betreffende uitwisseling.



1. Een verwerker kan bij een bepaalde uitwisseling de rol van de Authorization Server en Resource Servers hebben en in een andere uitwisseling die van client.
2. De client wordt (handmatig) geregistreerd bij de Authorization Server (zie Client registratie).
3. Afbeelding met diagram

   Automatisch gegenereerde beschrijvingMet de komst van het OAuth-profiel wordt het nog urgenter om ook de architectuur op punten te herzien en of aan te vullen. Het doel is om Edukoppeling beter binnen de ROSA te kunnen duiden en beter aan te sluiten op actuele architectuurstijlen en beveiligingsstandaarden. Deze standaarden zijn ontwikkeld voor moderne infrastructuur, worden door vele platforms ondersteund en ontwikkelaars beschikken over de benodigde kennis. We borgen hiermee ook de interoperabiliteit en herbruikbaarheid van onze onderwijsstandaarden.
4. Net als het REST-profiel worden in dit OAuth-profiel ook de eisen uit het Digikoppeling REST API profiel overgenomen. Deze afhankelijkheden worden weergegeven in Figuur 1. Het gaat hierbij voornamelijk om principes rond API design die als onderdeel van het Kennisplatform API’s Design Rules[[16]](#footnote-17) zijn opgesteld. Deze afhankelijkheid wordt beschreven in hoofdstuk “Voorschriften MDX OSR protocol en API Design Rules”.

# Voorschriften OAuth client credentials profiel

# MAY:Authorization Server metadata

Om conform OAuth 2.0 te kunnen werken moeten de verschillende rollen eerst informatie uitwisselen binnen het registratieproces. Ter ondersteuning van het registratieproces worden hieronder wel een aantal gegevens benoemd waarmee aan dit Oauth-profiel kan worden voldaan.

1. issuer

De issuer identifier (URL) van de authorization server.

* 1. De issuer moet herleid kunnen worden naar het OIN van de verwerker die de Authorization Server beheert. De client kan op basis van deze gegevens samen met het te gebruiken routeringskenmerk binnen een bepaalde uitwisseling het mandaat controleren.

1. token\_endpoint

Het token endpoint (URL) waar de client een verzoek voor een AT heen stuurt.

1. jwks\_uri

De locatie (URL) van de JWK Set document. Deze bevat onder andere de (publieke) sleutel(s) voor het valideren van de door de AS geplaatste ondertekeningen.

* 1. Voor de publieke sleutel zijn in de JWK Set de volgende gegevens opgenomen:
     1. kid: De key ID van het sleutelpaar voor ondertekening
     2. kty: Het sleutel type
     3. alg: het (default) algoritme
  2. Voor de client is het Access Token niet relevant en moet deze niet verwerken of verifiëren, De jwks info is dus met name van belang voor de resource server. De communicatie tussen AS en RS is in principe buiten scope van dit profiel. Het is wel belangrijk dat de client kan vaststellen dat de AT van de AS afkomstig is. Dit kan op basis van de (m)TLS verbinding.

1. scopes\_supported

Een lijst met de scopes die de authorization server ondersteunt.

* 1. Het wordt aanbevolen dat als de authorization server scopes verwacht dat er ook een default scope gedefinieerd wordt. Deze kan dan toegepast worden als in het verzoek van de client de scope ontbreekt. Het alternatief is een foutmelding.

1. grant\_types\_supported

Een lijst (zie RFC7591) met de grant types die de authorization server ondersteunt.

* 1. Voor dit profiel moet de lijst de waarde “client\_credentials” bevatten.

1. token\_endpoint\_auth\_methods\_supported

Een lijst (zie RFC8414) van client authenticatie methoden die de authorization server ondersteunt.

* 1. Voor dit profiel moet de lijst de waarde "private\_key\_jwt" bevatten.

1. token\_endpoint\_auth\_signing\_alg\_values\_supported

Een lijst (zie RFC8414) van signing algoritmes die de authorization server ondersteunt.

* 1. Voor dit profiel moet de lijst de waarde "RS256" bevatten. In de lijst niet de waarde "none" opnemen.

# MAY: Client registratie

Voordat een client over een access token kan beschikken en toegang krijgt tot een API moet deze eerst bij de betreffende authorization server geregistreerd zijn. Dit Oauth-profiel gaat uit van een “handmatige” registratie van confidential clients. Binnen een bepaalde ketensamenwerking worden afspraken gemaakt welke gegevens er tussen verschillende clients en protected resources nodig zijn. Hieronder zijn een aantal gegevens opgenomen die bij de registratie van een client relevant worden geacht.

1. De client wordt bij de authorization server geregistreerd met de volgende informatie:
   1. client\_id
      1. De client krijgt bij registratie een identifier (client\_id) van de authorization server.
      2. Dit OAuth-profiel vereist dat de client\_id herleidbaar is naar de verwerker (OIN). De Authorization Server kan op basis van deze gegevens samen met het te gebruiken routeringskenmerk binnen een bepaalde uitwisseling het mandaat controleren.
   2. client\_name
   3. token\_endpoint\_auth\_method
      1. Dit OAuth-profiel vereist dat dit de waarde “private\_key\_jwt” bevat. De publieke sleutel voor verificatie van de JWT is beschikbaar via de jwks\_uri.
   4. grant\_types
      1. Dit OAuth-profiel vereist dat dit de waarde "client\_credentials” bevat.
   5. edu\_org\_id
      1. Als de AS een fijnmazige data-autorisatie ondersteunt dan kan deze al bij registratie van de client geregistreerd worden. Hoe deze autorisatie wordt ingericht valt buiten de scope van dit OAuth-profiel.
   6. scope(s)
      1. Als de AS scopes ondersteunt dan moet de client de te gebruiken scopes bij registratie aangeven.
   7. resource
      1. Het kan zijn dat de AS voor meerdere API’s Access Tokens uitgeeft en vereist dat een client aangeeft voor welke resource een Access Token wordt gevraagd. De client moet dan bij registratie aangeven voor welke API’s er toegang gevraagd gaat worden. Hiervoor ondersteunt dit OAuth-profiel een optioneel gegeven “resource” (conform RFC8707).
2. Dit OAuth-profiel vereist dat een client een public/private sleutelpaar heeft voor de private\_key\_jwt authenticatie methode bij het token endpoint van de authorization server.
   1. De client moet een PKIo of private CA certificaat voor het JWT gebruiken.
   2. De authorization server gebruikt de publieke sleutel van het client certificaat om de JWT in stap #1 te valideren[[17]](#footnote-18).
      1. De authorization server moet voor validatiedoeleinden over de publieke sleutel kunnen beschikken.
      2. Er moet in een proces worden voorzien dat certificaten (probleemloos) kunnen worden vervangen.
         1. Omdat certificaten over tijd kunnen wijzigen moet de client (meta)data rond de publieke sleutel(s) beschikbaar stellen. Dit moet via het registreren van een jwks\_uri met verwijzing naar het JSON Web Key (JWK) Set (RFC7517) document (zie voorbeeld hieronder).
         2. Als de client (issuer) een certificaat vervangt dan moet de publieke sleutel in de JSON Key Set met een nieuwe kid toegevoegd worden. Partijen moeten daarom regelmatig dit document updaten.

|  |
| --- |
| {  "keys": [{  "e": "AQAB",  "use": "sig",  "alg": "RS256",  "kty": "RSA",  "n": "oNqXxxWuX7LlovO5reRNau5f96K\_o3DJx-wK7lrjBmp0qKwNszbbp8MvfrlVs-oYXfj1rzqAeY6GJF5BETViDTT0i2fEz37J0HGAeTrO7Z5zI5Ure9Cb0lulLOZj1hF8piZzWW\_z\_set2NyhafoZ-IG1NSe61mqHu7mTjuHYST84igz-bPKhkJAVlmPPjHTO51hG9T\_roVIkjXnvgqd2dCaJ0ExT2bR96jcyausbkdDNfPtJdfSCAWYXGQnt0PmIysOHPtCkyFqv5ez8KXT7Q4CAYd7nxwfWNOFRHyLAyF\_\_cYEJlBEKGyJniSIPtkGBWrbXUQhKF6TEFa-RRRl8Dw==",  "kid": "1516918956\_1"  }, {  "e": "AQAB",  "use": "sig",  "alg": "RS256",  "kty": "RSA",  "n": "kMfHwTp2dIYybtvU-xzF2E3dRJBNBtNbb-d3-Rm6nRUraxnTwZ6Fr1YpFBd1pnWzLzdtMv7ofCd28nx-1mfYZ6qhqPWF1RpGe2vVOSTmcu-QpA9h-rouqRKlv3jvXPn623Z2U1Wml0EIxyIzD3WLu7NkWEKSIcBzeY1TctpO5FSU3EyyCX1UoIMuvYBP9tiZlc74yIZvky-qT8Ej3S8L0JqhvD583E\_uGMoLowguOl2yYr9zhubiqOxT3VsxvpJCu04TWmvf4XX34IQRyAHcPJFQ2QiBfLWvWyc6iP3JJYJvyapwc5vVEismryXnngyBX8NXHZaarMi6g5kTQi8itw==",  "kid": "1516918956\_0"  }]  } |

# MUST: Interacties

### Verzoek naar Token Endpoint (STAP #1)

Het verzoek voor een access token dat de client naar het token endpoint van de authorization server stuurt moet aan de volgende eisen voldoen:

1. Het verzoek moet een HTTP POST over mTLS zijn:
   1. UBV TLS Edukoppeling profiel
   2. authenticatie van de verwerker op basis de certificaat hiërarchie;
   3. identificatie (verwerker) op basis van OIN.

1. Het verzoek moet een routeringskenmerk bevatten (zie 5.2.2 “Routeringskenmerken zijn query parameters in het request naar het token endpoint van de Authorization Server”):
   1. Op basis van het OIN en het routeringskenmerk moet het mandaat gecontroleerd worden (zie 5.1.5 "Toepassing van het MDX OSR protocol”).
2. Het verzoek moet de **grant\_type** parameter bevatten met de waarde “client\_credentials”.
3. Het verzoek moet de **client\_assertion\_type** parameter bevatten met de waarde “urn:ietf:params:oauth:client-assertion-type:jwt-bearer”.
4. Het verzoek moet de **client\_assertion** parameter bevatten met de JWT voor client authenticatie.
   1. De client moet voor ieder verzoek van een bepaalde context[[18]](#footnote-19) een nieuwe JWT genereren.
   2. De client moet de JWT ondertekenen met de private sleutel van de client.
      1. Het client certificaat moet een PKIo of een certificaat van een private CA gebruiken voor de JWT.
      2. De JWT voor client authenticatie moet voldoen aan RFC7523 (Using JWTs for Client Authentication.
   3. De JWT payload bevat de volgende claims:
      1. iss: een unieke identifier van de uitgever van de verklaring.
         1. In het verzoek naar de authorization server token endpoint moet de iss claim gevuld worden met de "client\_id".
      2. sub: een unieke identifier van de entiteit die geauthenticeerd is.
         1. In het verzoek naar de authorization server token endpoint moet de sub claim gevuld worden met de "client\_id".
      3. aud: een unieke identifier van de ontvanger(s) van de verklaring
         1. In het verzoek naar de authorization server token endpoint moet de aud claim een waarde bevatten die de authorization server identificeert. Dit mag het token endpoint URL zijn.
      4. iat: een timestamp van het moment waarop de JWT is gecreëerd.
      5. exp: een timestamp van het moment waarop de JWT verloopt.
      6. jti: een unieke identifier voor de gecreëerde JWT.
      7. Optioneel:
         1. scope: De client mag een Access Token aanvragen met een specifieke scope. Het wordt aanbevolen om de scope conform de scope naming conventions van dit profiel toe te passen. Deze zorgt ervoor dat er een logische referentie naar de protected resource is i.c.m. uit te voeren actie(s).
            1. Het AT is betekenisvol voor AS en RS en betekenisloos voor de client. Op basis hiervan zou kunnen worden aangenomen dat de scope dus niet relevant is voor de client en dus nooit opgenomen hoeft te worden in dit verzoek. Het wordt echter aanbevolen om waar relevant een client meerdere scopes te geven en dat de client een specifieke scope in aanvraag opneemt. Dit is vanuit beveiliging wenselijk om zo aan het bearer token alleen essentiële autorisaties te kunnen koppelen.
         2. resource: De client mag een Access Token aanvragen voor een specifieke resource. Dit optionele claim is relevant als de Authorization Server tokens uitgeeft voor meerdere API’s en er niet op basis van de scope (conform naming convention) een bepaalde resource kan worden geïdentificeerd. Dit wordt ondersteund met de claim “resource” (conform RFC8707). De client neemt deze op in het JWT en is in het Access Token opgenomen in de “aud” claim.
         3. edu\_org\_id: De client mag een Access Token aanvragen in de context van een onderwijsorganisatie. De client moet in dat geval de edu\_org\_id in het verzoek opnemen. Hiermee kan een meer fijnmazige (data)autorisatie[[19]](#footnote-20) ondersteund worden die door een bepaalde onderwijsorganisatie voor een client is geregistreerd. Hoe een partij dit als onderdeel van de authorization server inricht en hoe de interactie met de onderwijsorganisatie (admin) plaatsvindt is buiten de scope van dit OAuth-profiel.
   4. De JWT header bevat de volgende (JSON Web Signature RFC7515) gegevens:
      1. typ: The "typ" (type) Header Parameter is used by JWS applications to declare the media type [IANA.MediaTypes] of this complete JWS.
      2. alg: The "alg" (algorithm) Header Parameter identifies the cryptographic algorithm used to secure the JWS.
      3. kid: The "kid" (key ID) Header Parameter is a hint indicating which key was used to secure the JWS.

|  |
| --- |
| {  "typ" : "JWT",  "alg" : "RS256"  "kid" : “1516918956\_1”  } |

Een voorbeeld van het verzoek (stap #1) wordt weergegeven in het onderstaande Figuur 4.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Figuur 2 - Stap 1: Verzoek van client aan authorization server token endpoint

### Verwerking v/h client verzoek bij het AS Token Endpoint

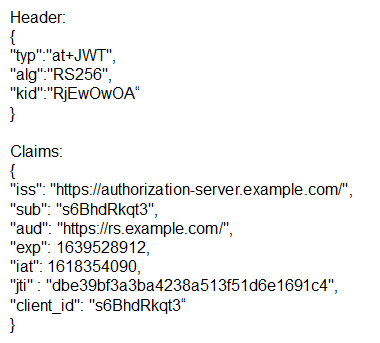
Het verzoek dat in stap #1 is opgesteld wordt door de authorization server gevalideerd.

1. De authorization server moet controleren of het verzoek via een mTLS verbinding verloopt (zie 5.1.2 "Transportbeveiliging op basis van (m)TLS”).
2. De authorization server moet controleren of het verzoek een querystring met het routeringskenmerk bevat (zie 5.2.2 “Routeringskenmerken zijn query parameters in het request naar het token endpoint van de Authorization Server”)
   1. Op basis van het OIN en het routeringskenmerk moet het mandaat gecontroleerd worden (zie 5.1.5 "Toepassing van het MDX OSR protocol”)..
3. De authorization server moet controleren of het request de volgende parameters bevat:
   1. grant\_type parameter met de waarde “client\_credentials”;
      1. Controleren of de authenticatie methode overeenkomt met de geregistreerde authenitcatiemethode (token\_endpoint\_auth\_method) voor deze client.
   2. client\_assertion\_type parameter bevatten met de waarde “urn:ietf:params:oauth:client-assertion-type:jwt-bearer”;
   3. client\_assertion met een JWT;
      1. De authorization server moet controleren of de JWT voldoet aan de eisen van de standaarden (RFC7519/RFC7523/RFC7521).
      2. De authorization server moet de waarden van de iss, sub, exp, aud and jti claims controleren en de ondertekening verifiëren.
         1. De authorization server moet controleren of de JWT is ondertekend met het certificaat dat voor deze client is geregistreerd (op basis van kid claim en jkws\_uri of cache).
   4. als de JWT een scope bevat dan moet de authorization server controleren of deze voor de client geregistreerd is;
      * 1. Als dit niet het geval is dan is het resultaat een AT op basis van een default scope of een foutmelding.

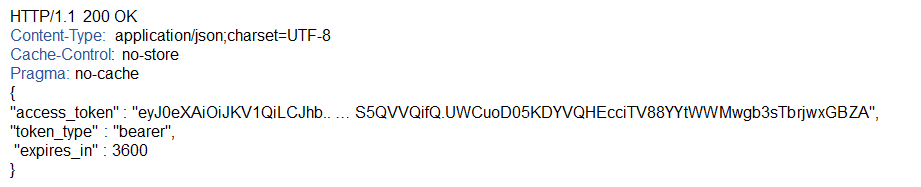
### Antwoord van Token Endpoint (STAP #2): verwerking succesvol

Als de autorization server request #1 succesvol heeft gevalideerd, MOET deze een HTTP 200 OK status code respons geven conform RFC6749 sectie 5.1.

1. Het respons media type mag application/at+jwt" zijn
2. Het access token wordt in de access\_token parameter opgenomen
   1. Het access token conformeert aan RFC9068.
      1. Als er niet met een “resource” claim wordt gewerkt en er wordt niet met een scope gewerkt die de AS kan gebruiken om een resource af te leiden, dan wordt de aud claim gevuld met een default resource waarde.
   2. Het access token moet ondertekend zijn volgens een algoritme uit het UBV TLS Edukoppeling profiel[[20]](#footnote-21).
3. De respons bevat geen refresh token



Figuur 3 - Niet base64-encoded en niet ondertekend Access Token



Figuur 4 - Respons van AS token endpoint

### Antwoord van Token Endpoint (STAP #2): foutmelding RFC6749 (§ 5.2)

Als de autorisatie service request #1 niet succesvol heeft gevalideerd, moet deze een HTTP 400 Bad Request status code respons geven. In de body is een foutmelding opgenomen conform paragraaf 5.2 van RFC6749.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Figuur 5 - HTTP 400 Bad Request status code met foutmelding

### Interactie met protected resource (STAP #3)

De client heeft van de authorization server een access token ontvangen. Het access token is voor de client betekenisloos en moet deze niet analyseren.

Het verzoek naar de protected resource moet aan de volgende eisen voldoen:

1. De verbinding is op basis van TLS (zie UBV TLS basisprofiel)
2. De client biedt het access token aan in de Authorization request header (RFC2617) met toepassing van de "Bearer" authentication scheme. Een client mag de RFC6750 form-parameter gebruiken als het gebruik van de Authorization request header niet mogelijk is.
   1. Zie RFC6750 paragraaf 2.1:

“*When sending the access token in the "Authorization" request header field defined by HTTP/1.1 [RFC2617], the client uses the "Bearer" authentication scheme to transmit the access token*.”

en

“*Clients SHOULD make authenticated requests with a bearer token using the "Authorization" request header field with the "Bearer" HTTP authorization scheme. Resource servers MUST support this method.” en “The "application/x-www-form-urlencoded" method SHOULD NOT be used except in application contexts where participating browsers do not have access to the Authorization request header field. Resource servers MAY support this method*.”



Figuur 6 - Access token in Authorization request header

De resource server moet het volgende controleren:

1. Er wordt aanbevolen om het Access Token in de authorization request header van het verzoek op te nemen (zie Figuur 6);
   1. Als dit niet kan mag het Access Token in de form-parameter opgenomen worden conform RFC6750;
2. Het Access token wordt gevalideerd conform RFC9068 (H4 Validating JWT Access Tokens).

Als het request in stap #3 niet succesvol geautoriseerd kan worden dan moet de resource server een HTTP 400 Bad Request status code respons geven. In de body is een foutmelding opgenomen conform paragraaf 3.1 van RFC6750.

# Voorschriften REST

Er zijn een aantal voorschriften die overeenkomsten hebben met het REST-profiel, dit zijn:

1. Het Edukoppeling OAuth-profiel stelt eisen aan API Design (protected resource).
2. Het Edukoppeling OAuth-profiel eist dat de routeringskenmerken zijn opgenomen als query parameters in het request naar het token endpoint van de Authorization Server.

### MUST: API Design conform het Digikoppeling REST profiel

API design moet conform de API Design Rules zijn van het Digikoppeling Restful API profiel[[21]](#footnote-22).

De API Design principes zijn ook een belangrijk onderdeel van het Edukoppeling MDX Secure API REST profiel versie 1.0 (hierna REST-profiel). De API Design principes zijn echter sinds publicatie van de 1.0 versie van het REST-profiel gewijzigd. Hieronder wordt aangegeven welke principes voor het huidige REST-profiel versie 1.0 gelden en welke voor dit OAuth-profiel[[22]](#footnote-23) gelden. Bij gewijzigde principes is het nummer rood gekleurd, bij nieuwe groen en bij ongewijzigd zwart. De principes met een grijze vulling zijn extensies[[23]](#footnote-24) op de Design Rules. Vanuit het Digikoppeling RESTful API profiel worden er per principe een kwalificatie toegekend. Deze zijn in de tabel hieronder overgenomen.

|  |  |  |
| --- | --- | --- |
| **#** | **MDX Secure API REST profile v1.0** | **MDX Secure API OAuth profile v0.3** |
| API-01 | Operations are Safe and/or Idempotent | Verplicht: [Adhere to HTTP safety and idempotency semantics for operations](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-01) |
| API02 | Do not maintain state at the server | Verplicht: [Do not maintain session state on the server](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-02) |
| API03 | Only apply default HTTP operations | Verplicht: [Only apply standard HTTP methods](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-03) |
| API04 | Define interfaces in Dutch unless there is an official English glossary available | Verplicht: [Define interfaces in Dutch unless there is an official English glossary available](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-04) |
| API05 | Use plural nouns to indicate resources | Verplicht: [Use nouns to name resources](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-05) |
| API-06 | Create relations of nested resources within the endpoint | Verplicht: [Use nested URIs for child resources](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-06) |
| API-09 | Implement custom representation if supported | *Vervallen* |
| API-10 | Implement operations that do not fit the CRUD model as sub-resources | Verplicht: [Model resource operations as a sub-resource or dedicated resource](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-10) |
| API-11 | Encrypt connections using TLS following the latest NCSC guidelines | *Niet van toepassing, TLS voorschriften zijn opgenomen in UBV TLS.* |
| API-12 | Allow access to an API only if an API key is provided | *Vervallen* |
| API-13 | Accept tokens as HTTP headers only | Verplicht: [Accept tokens as HTTP headers only](https://docs.geostandaarden.nl/api/def-hr-API-Strategie-ext-20211013/#api-13) |
| API-14 | OAuth 2.0 can be used for authorisation | *Vervallen* |
| API-15 | Use PKIoverheid certificates for access-restricted or purpose-limited API authentication | *Vervallen* |
| API-16 | Documentation conforms to OAS v3.0 or newer | Verplicht: [Use OpenAPI Specification for documentation](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-16) |
| API-17 | Publish documentation in Dutch unless there is existing documentation in English or there is an official English glossary available | Verplicht: [Publish documentation in Dutch unless there is existing documentation in English](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-17) |
| API-18 | Include a deprecation schedule when publishing API changes | Verplicht: [Include a deprecation schedule when publishing API changes](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-18) |
| API-19 | Allow for a (maximum) 1 year transition period to a new API version | Verplicht: [Schedule a fixed transition period for a new major API version](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-19) |
| API-20 | Include the major version number only in ihe URI | Verplicht: [Include the major version number in the URI](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-20) |
| API-21 | Inform users of a deprecated API actively | *Vervallen* |
| API-22 | JSON first - APIs receive and send JSON | *Vervallen* |
| API-23 | APIs may provide a JSON Schema | *Vervallen* |
| API-24 | Support content negotiation | *Vervallen* |
| API-25 | Check the Content-Type header settings | *Vervallen* |
| API-26 | Define field names in camelCase | *Vervallen* |
| API-27 | Disable pretty print | *Vervallen* |
| API-28 | Send a JSON-response without enclosing envelope | *Vervallen* |
| API-29 | Support JSON-encoded POST, PUT, and PATCH payloads | *Vervallen* |
| API-30 | Use query parameters corresponding to the queryable fields | *Vervallen* |
| API-31 | Use the query parameter sorteer to sort | *Vervallen* |
| API-32 | Use the query parameter zoek for full-text search | *Vervallen* |
| API-33 | Support both \* and ? wildcard characters for full-text search APIs | *Vervallen* |
| API-34 | Support GeoJSON for GEO APIs | *Vervallen* |
| API-35 | Include GeoJSON as part of the embedded resource in the JSON response | *Vervallen* |
| API-36 | Provide a POST endpoint for GEO queries | *Vervallen* |
| API-37 | Support mixed queries at POST endpoints | *Vervallen* |
| API-38 | Put results of a global spatial query in the relevant geometric context | *Vervallen* |
| API-39 | Use ETRS89 as the preferred coordinate reference system (CRS) | *Vervallen* |
| API-40 | Pass the coordinate reference system (CRS) of the request and the response in the headers | *Vervallen* |
| API-41 | Use content negotiation to serve different CRSs | *Vervallen* |
| API-42 | Use JSON+HAL with media type application/hal+json for pagination | *Vervallen* |
| API-43 | Apply caching to improve performance | *Vervallen* |
| API-44 | Apply rate limiting | *Vervallen* |
| API-45 | Provide rate limiting information | *Vervallen* |
| API-46 | Use default error handling | Aanbevolen: [Use default error handling](https://docs.geostandaarden.nl/api/def-hr-API-Strategie-ext-20211013#api-46) |
| API-47 | Use the required HTTP status codes | Aanbevolen: [Use the required HTTP status codes](https://docs.geostandaarden.nl/api/def-hr-API-Strategie-ext-20211013#api-47) |
| API-48 | Leave off trailing slashes from API endpoints | Verplicht: [Leave off trailing slashes from URIs](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-48) |
| API-49 | Use public API-keys | *Vervallen* |
| API-50 | Use CORS to control access | *Vervallen* |
| API-51 | Publish OAS at a base-URI in JSON-format | Verplicht: [Publish OAS document at a standard location in JSON-format](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-51) |
| API-52 | Use OAuth 2.0 for authorisation with rights delegation | *Vervallen* |
| API-53 | - | Verplicht: [Hide irrelevant implementation details](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-53) |
| API-54 | - | Verplicht: [Use plural nouns to name collection resources](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-54) |
| API-55 | - | Verplicht: [Publish a changelog for API changes between versions](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-55) |
| API-56 | - | Verplicht: [Adhere to the Semantic Versioning model when releasing API changes](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-56) |
| API-57 | - | Verplicht: [Return the full version number in a response header](https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-57) |
| API-58 | - | Verplicht: [No sensitive information in URIs](https://docs.geostandaarden.nl/api/def-hr-API-Strategie-ext-20211013/#api-58) |

Over het algemeen zijn de wijzigingen enkel van tekstuele aard en is er vrijwel geen impact. Een aantal nieuwe principes hebben wel impact.

### MUST: Routeringskenmerken zijn query parameters in het request naar het token endpoint van de Authorization Server.

Een belangrijk aspect van MDX-profielen 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 dit OAuth-profiel (en het MDX Secure API REST profile) 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 dezelfde (of verschillende) eindorganisatie gerouteerd kan worden.

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

<https://prod.leverancier.com/v1/service?edu-to=0000000700011BB00001&edu-from=0000000700011BB00000>

Het routeringskenmerk wordt geleverd bij het request van elke HTTP methode. Mede op basis van het routeringskenmerk (zie MDX OSR protocol) kan de mandatering gevalideerd worden. Als één of beide routeringskenmerken in het request ontbreken moet er een foutmelding (zie MUST: Foutafhandeling m.b.t. het routeringskenmerk) gegeven worden.

### MAY: Berichtbeveiligingsvoorschriften

Berichtbeveiliging naast transportbeveiliging is met name relevant in ketens met transparante intermediairs. Het ondertekenen van het bericht kan wenselijk zijn om onweerlegbaarheid[[24]](#footnote-25) te ondersteunen.

Voor dit OAuth-profiel zijn momenteel geen berichtbeveiligingsvoorschriften opgenomen en heeft hiermee in vergelijking tot het MDX Secure API WUS-profiel beperkte functionaliteit. Wel zijn er ontwikkelingen rond berichtbeveiliging voor RESTful uitwisselingen (zie Digikopppeling REST profiel[[25]](#footnote-26))

### MUST: Foutafhandeling m.b.t. het routeringskenmerk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |
| 403 | Forbidden | Mandatering | EK | Autorisatiefout   * TLS: OIN in certificaat niet geautoriseerd * TLS: OIN in certificaat niet gemandateerd om voor eindorganisatie (From routeringskenmerk) in context van deze API 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 |

# Overige voorschriften

Dit Oauth-profiel heeft een aantal overeenkomsten met de andere Edukoppeling MDX profielen. De voorschriften die overeenkomen met de andere MDX profielen worden in dit hoofdstuk beschreven. Het betreft het volgende:

1. Het Edukoppeling OAuth-profiel houdt expliciet rekening met gebruik van een openbaar netwerk (Internet).
2. Het Edukoppeling OAuth-profiel stelt eisen aan transportbeveiliging.
3. Het Edukoppeling OAuth-profiel stelt eisen aan identificatie en authenticatie van organisaties.
4. Het Edukoppeling OAuth-profiel kan worden toegepast voor zowel bevragingen als meldingen
5. Het Edukoppeling OAuth-profiel wordt in combinatie met het MDX OSR protocol gebruikt.
6. Het Edukoppeling OAuth-profiel stelt eisen aan foutafhandeling m.b.t. het routeringskenmerk.

### 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.

### MUST: Transportbeveiliging op basis van (m)TLS[[26]](#footnote-27)

De profielen maken voor transportbeveiliging gebruik van (m)TLS. De voorschriften hiervoor zijn opgenomen in de Edustandaard Uniforme Beveiligingsvoorschriften (UBV TLS[[27]](#footnote-28)):

* De beveiligingsvoorschriften voor mTLS zijn opgenomen in het UBV TLS Edukoppeling profiel.
* De beveiligingsvoorschriften voor TLS zijn opgenomen in het UBV TLS basisprofiel.

Er wordt verder aanbevolen om bedreigingen rond beschikbaarheid, integriteit en vertrouwelijkheid te beperken door het opvolgen van OWASP-richtlijnen[[28]](#footnote-29).

### MUST: Identificatie en authenticatie van organisaties

Identificatie

Binnen ketens waarbij een Edukoppeling MDX profiel wordt toegepast voor de gegevensuitwisseling, worden partijen geïdentificeerd op basis van hun unieke Organisatie Identificatie Nummer (OIN)[[29]](#footnote-30). Onderwijsorganisaties worden geïdentificeerd op basis van het BRIN. Binnen de OIN-systematiek betekent dit dat het OIN hoofdnummer een BRIN is en een prefix met de waarde “00000007”. Voor bedrijven wordt een systematiek gebruikt die gelijk is aan de OIN-systematiek en wordt ook wel HRN genoemd. Voor het HRN wordt alleen de prefix 00000001 (RSIN als hoofdnummer) of 00000003 (KvK nummer als hoofdnummer) gebruikt.

In de Edukoppeling Architectuur worden binnen de MDX profielen drie[[30]](#footnote-31) rollen onderscheiden, dit Oauth-profiel gaat uit van de volgende twee rollen:

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 vertrouwelijke gegevens verwerkt[[31]](#footnote-32).

De eindorganisatie binnen een Edukoppeling MDX profiel betreft vaak een onderwijsorganisatie, maar kan ook een bedrijf of ZBO betreffen. Binnen de gegevensuitwisseling wordt het OIN van de eindorganisatie opgenomen in het zogenaamde ‘TO’ en ‘FROM’ routeringskenmerken.

De verwerker binnen een Edukoppeling MDX profiel betreft vaak een bedrijf, maar kan ook een onderwijsorganisatie of een ZBO betreffen. Binnen de gegevensuitwisseling is het OIN (HRN) van de verwerker in het PKIoverheid-certificaat opgenomen dat wordt gebruikt bij de mTLS-verbinding.

Authenticatie verwerker

Binnen de Edukoppeling MDX profielen worden beide verwerkers geauthenticeerd door toepassing van een PKI-infrastructuur. De PKI-infrastructuur (PKIo) 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 van de verwerker met zijn (CA) certificaat. Het certificaat van een verwerker wordt gebruikt in de mTLS verbinding. De voorschriften rond mTLS zijn opgenomen in de Edustandaard UBV TLS standaard[[32]](#footnote-33).

Authenticatie verwerkers op basis van mTLS (bij Authorization Server Token Endpoint)

De MDX-profielen vereisen dat verwerkers elkaar kunnen authenticeren op basis van mTLS. Dit wordt toegepast conform het Edukoppeling profiel in de UBV TLS voorschriften. Het Edukoppeling profiel binnen de UBV TLS afspraak maakt deels gebruik van de Digikoppeling beveiligingsvoorschriften, zoals het gebruik van PKIoverheid certificaten (UBV-TLS-PKI-01/ DK-TLS001[[33]](#footnote-34)) 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[[34]](#footnote-35). Bij de uitgifte hoort ‘face-to-face’ controle: de houder neemt het certificaat persoonlijk in ontvangst. Het identificerend kenmerk wordt conform Digikoppeling OIN nummersystematiek bepaalt (zie identificatie en authenticatie[[35]](#footnote-36)). 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.

Dit OAuth-profiel vereist alleen mTLS bij het token endpoint koppelvlak van de Authorization Server. De verwerker die een OAuth client beheert wordt alleen bij het verzoek naar het Authorization Server Token Endpoint geauthenticeerd. Bij het koppelvlak met de Resource Server betreft het dezelfde verwerkers en is authenticatie van de verwerker als client niet meer nodig. Het OAuth access token dat de client levert voor autorisatie volstaat. Voor transportbeveiliging wordt het UBV TLS basisprofiel vereist.

### MAY: Kan worden toegepast voor zowel bevragingen als meldingen

De Edukoppeling MDX profielen ondersteunt standaard de synchrone bevraging (pull) en synchrone melding (push) transactiepatronen. Andere transactiepatronen kunnen echter ook worden ondersteund (zie Edukoppeling Architectuur).

Dit sluit ook aan bij de (actuele[[36]](#footnote-37)) versie van de Digikoppeling architectuur en de Digikoppeling WUS en REST profielen. In vorige versies van de Digikoppeling Architectuur werden specifieke profielen gekoppeld aan bevragingen (WUS/REST) en meldingen (ebMS). Dit voorschrift bleek in de praktijk niet meer goed bruikbaar. Vandaar dat met ingang van versie 2.0.0 deze relatie is komen te vervallen.

### MUST: Toepassing van het MDX OSR protocol

Onderwijsorganisaties maken in de context van MDX profielen gebruik van (SaaS) diensten van een leverancier. Het Onderwijs Service Register (OSR) onderkent deze situatie en ondersteunt tevens de functie om mandateringen te registreren en verifiëren. Het mandaat is de registratie dat een bepaalde verwerker (bijvoorbeeld SaaS-leverancier) namens een bepaalde eindorganisatie (bijvoorbeeld onderwijsorganisatie) door middel van een dienst via één of meer interfaces gegevens mag uitwisselen binnen een ketensamenwerking. Welke eisen er gelden rond het OSR zijn vastgelegd in het MDX OSR protocol.

### MUST: Foutafhandeling m.b.t. het routeringskenmerk

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 MDX profiel. Voor dit OAuth-profiel zijn deze gelijk aan die van het MDX Secure API REST profile.

|  |  |  |
| --- | --- | --- |
| **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 |

### 

# Bijlage A: OAuth framework

# Inleiding

Het OAuth framework is gebaseerd op RFC6749 die gepaard gaat met een Bearer Token Usage specificatie (RFC6750). Op basis van verschillende profielen worden verschillende scenario’s ondersteund. Zo geeft het code grant profiel invulling aan het privacy-by-design principe doordat het voorkomt dat systemen van verschillende partijen vertrouwelijke gegevens kunnen uitwisselen zonder dat de betrokkene hier inzage of controle over heeft.

Het is een stap in de ontwikkeling van het beveiligen van API's. Bij eerdere mechanismen zoals HTTPS Basic Authentication (gebruikersnaam & wachtwoord) en statische API-keys konden credentials verloren raken of door onbevoegde gebruikt worden. Ook was een fijnmazige (dynamisch) autorisatie niet goed in te richten. Met OAuth wordt er een stap gemaakt naar een tokens gebaseerde architectuur. Het biedt de mogelijkheid om (centraal) overzicht en controle te krijgen rond toegangsrechten. Het biedt ook eenvoud bij de ontwikkeling van applicaties doordat zij toegang kunnen krijgen tot gegevens op basis van een token.

Het OAuth raamwerk definieert 4 rollen:

* Een Resource Owner (RO) beheert de gegevens (protected resource(s)) die door de API worden ontsloten. De RO wordt beschouwd als de "eigenaar" van de gegevens.
* De Authorization Server (AS) geeft Access Tokens (AT) uit en kan deze ook weer intrekken.
* De client is bijvoorbeeld een applicatie of website die de gegevens opvraagt ​​namens de RO.
* De Resource Server (RS) is de service die de gegevens ontsluit en opslaat. Dit is feitelijk de API.

OAuth onderkent dus zowel de client als de resource owner. De client en resource owner hebben elk hun eigen credentials en kunnen op basis hiervan geïdentificeerd en geauthentiseerd worden. Hierdoor kan in het interactiemodel de resource owner in staat wordt gesteld om via de authorization server aan te geven tot welke protected resource(s) de client toegang krijgt. Het access token dat de client van de authorization server krijgt vertegenwoordigt de autorisatie die de resource owner de client heeft gegeven.

# OAuth client credentials profiel

Dit OAuth-profiel is gebaseerd op de client credentials grant type. Hierbij speelt de resource owner geen rol. Er geldt over het algemeen dat de client credentials (identificatie en authenticatie) in principe voldoende zijn voor toegang tot de API.

Bij de client credentials grant wordt uitgegaan van confidential clients (afgeschermde vertrouwde systemen) die de te gebruiken credentials goed kunnen beveiligen. Er kan namelijk worden gesteld dat de client op basis van de credentials impliciet geautoriseerd is en als de resource owner kan worden gezien.

|  |
| --- |
| Belangrijke onderdelen binnen het client credentials grant type zijn de client credentials en het access token. Er zijn verschillende manieren om hieraan invulling te geven. Voor authenticatie van de client kan bijvoorbeeld gebruik worden gemaakt van Basic Authentication, mTLS of een JWT. Voor het access token kan bijvoorbeeld worden gekozen voor een string (i.c.m. introspection) of een zelfbeschrijvende JWT. De keuzes hierin vormen het belangrijkste onderdeel van dit OAuth-profiel. |

Hieronder wordt het client credentials grant type schematisch weergegeven (zie Figuur 2).

## Afbeelding met tekst, schermafbeelding, klok Automatisch gegenereerde beschrijvingStap 1. Request naar token endpoint

Figuur 7 - OAuth client credentials profile (bron Kennisplatform API's | Geonovum)

De client stuurt een verzoek naar het token endpoint van de authorization server. Deze valideert het verzoek op basis van de client credentials.

* De validatie kan gebaseerd zijn op (aanvullende) out-of-band overeengekomen criteria.
* Het verzoek kan referenties naar specifieke resources bevatten als er meerdere protected resources zijn en het access token een fijnmazige autorisatie moet bevatten.

## Stap 2. Respons van token endpoint

Na succesvolle validatie levert de authorization server een toegangstoken die overeenkomt met de gevraagde rechten, of een subset hiervan. De client ontvangt het access token en moet deze meesturen bij het zenden van een verzoek naar de protected resource.

## Stap 3. Resource-interactie

Met het access token verzoekt de client namens zichzelf toegang tot de protected resource. De resource server waarop de protected resource wordt gehost valideert het access token voordat de client toegang krijgt. Er moet onder andere kunnen worden gevalideerd dat het Access token door de Authorization Server is uitgegeven en nog niet verlopen is. De levensduur moet zo kort mogelijk zijn, maar lang genoeg om tot werkbare interacties te komen. Als het access token valide is wordt toegang gegeven tot de protected resource. Indien dit niet het geval is wordt er een foutmelding gegeven conform RFC6749.

# Bijlage B: Overwegingen & acties

# Overwegingen

We moeten nog een eerste versie vaststellen, maar we onderkennen nu al een aantal punten die we nu nog niet meenemen, maar wel op de roadmap zetten voor een volgende versie. Het betreft het volgende:

1. OAuth versie 2.1
2. Routeringskenmerk o.b.v. een JWT
3. Scope naming notation convention
4. …

**OAuth versie 2.1**

Er is een nieuwe OAuth versie in ontwikkeling <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-v2-1-07>

*This specification replaces and obsoletes the OAuth 2.0 Authorization Framework described in RFC 6749 and the Bearer Token Usage in RFC 6750.*

Impact moet bepaald worden.

**Routeringskenmerk o.b.v. een JWT**

Bij het REST profiel hebben we destijds nog niet voor een JWT gekozen gezien de extra implementatie effort die dit introduceert. We hebben toen ervoor gekozen om het routeringskenmerk op te nemen in query string met het uitgangspunt dat het point-2-pooint verbindingen betreft (geen intermediairs) en de respons bevat impliciet een routeringskenmerk voor dezelfde eindorganisatie.

Voor dit OAuth-profiel hebben we eerder besloten om zowel in het request als de respons een routeringskenmerk in een JWT te definiëren. In deze versie van het OAuth-profiel wordt echter op dit punt afgeweken om beter op het huidige REST profiel te kunnen aansluiten. We gaan dus ook in dit OAuth-profiel uit van een point-2-point uitwisseling waarbij slechts een routeringskenmerk in het request volstaat. Hiermee hebben we een meer vereenvoudigde eerste versie van het OAuth-profiel.

Het opnemen van het routeringskenmerk in een JWT in request en respons waarbij de respons een andere eindorganisatie kan bevatten dan het request kunnen we in een toekomstige versie gaan ondersteunen (eventueel ook in het REST-profiel).

**Scope Naming Convention**

Willen we een naming convention specificeren voor scopes? Een scopes kan gebruikt worden als een grofmazige access control attribuut. De scope naming convention zou bijvoorbeeld een logische verwijzing naar de resource van een bepaald referentiecomponent kunnen zijn en dit combineren met de toegestane actie:

https://edukoppeling.edustandaard.nl/[RC]/[version]/scope/[ketensamenwerking].[action]

waarbij geldt:

* [RC] het referentiecomponent en resource identificeert e.g. las-student.
* [version] de versie van de service specificatie v1p0, v2p1, etc.
* [ketensamenwerking] de keten waarvoor de autorisatie geldt, bijv eindtoets
* [action] de toegestane actie, bijvoorbeeld:
* readonly for a set of endpoints that permit read only using the GET verb
* createpost for a set of endpoints that permit creation using the POST verb
* update for a set of endpoints that permit changing of an established resource
* delete for a set of endpoints that permit delete only
* all for access to all of endpoints supported for the version of the identified specification

# Acties

1. In overige documenten de volgende begrippen wijzigen

* SaaS-profiel wordt Mandated Data eXchange (MDX) Secure API profiel
* School en onderwijsinstelling wordt onderwijsorganisatie

1. Begrippen: <https://rosa.wikixl.nl/index.php/Speciaal:GegevensBekijken/Begrippen?_single&Begrippenset=Edukoppeling>

# Bijlage C: Uitgangspunten discussiestuk versie 0.3

1. SaaS-Context en SaaS profielen zijn begrippen die we in de communicatie kunnen blijven gebruiken, maar we veranderen de naam van de profielen in ‘Secure API-profielen’. We onderkennen de volgende Secure API profielen[[37]](#footnote-38):
   1. Secure API WUS(be, be-S & be-SE)-profiel
   2. Secure API REST-profiel
   3. Secure API OAuth-profiel
2. In de Edukoppeling documentatie en die van OSR hebben we het over een mandatering. Digikoppeling heeft het in een vergelijkbare procesafspraak (bevoegdheid intermediair/SAAS partij door 'machtigen' ) over een machtiging. In Edukoppeling blijven we het begrip mandatering hanteren.
3. De procesafspraken rond het OSR[[38]](#footnote-39) worden onderdeel van de normatieve voorschriften voor Edukoppeling. De procesafspraken worden beschreven in een Mandated Data eXchange Protocol (zie Figuur 1).
4. De Secure API profielen worden toegepast bij de uitwisseling van vertrouwelijke gegevens. Dit kunnen persoonsgegevens, maar ook bedrijfskritische gegevens zijn.
   1. Het gaat om vertrouwelijke gegevens en het juridisch kader (AVG) wordt dus niet expliciet binnen het Secure API Protocol opgenomen. De Edukoppeling Secure API Profielen voldoen wel aan algemeen geldende beveiligings- en privacy maatregelen die bij het verwerken van persoonsgegevens verwacht kunnen worden. Het Edustandaard Certificeringsschema geeft aan wanneer deze toegepast dienen te worden.
   2. We definiëren eigen Edukoppeling begrippen (eindorganisatie en verwerker) en sluiten dus niet direct aan op het juridisch (AVG) kader. Wel is het handig als we voor het begrip ‘verwerker’ nog een ander begrip te gaan gebruiken om verwarring te voorkomen. En moeten wellicht ook de definitie aanscherpen.
5. Het functionele toepassingsgebied van Secure API profielen blijft ongewijzigd.
6. We gebruiken (voorlopig[[39]](#footnote-40)) de internationale open standaard OAuth (RFC6749[[40]](#footnote-41) en RFC6750) als vertrekpunt voor de ontwikkeling van het OAuth profiel.
7. Onze use case (toegang tot bulk gegevens van meerdere betrokkenen) past het beste bij het OAuth client credentials en dit OAuth profiel zal als basis dienen. Het gaat om confidential clients die een Access Token krijgen op basis van hun credentials[[41]](#footnote-42).De client moet zich bij de AS registreren zodat identificatie mogelijk is om de verificatie van het mandaat bij het OSR uit te voeren en te kunnen bepalen of een Access Token geleverd mag worden.
8. In de context van het Secure API OAuth profiel wordt een meer specifieke rolaanduiding wenselijk. In de context van het OAuth profiel kunnen we namelijk spreken van een verwerker die een OAuth AS en RS beheert en een verwerker als client die de protected resource wil verwerken. De relatie tussen bestaande Edukoppeling rollen en nieuwe OAuth rollen wordt weergegeven in Figuur 3.
9. Verwerkers met de AS/RS combi zijn zelf verantwoordelijk voor het registreren van een client.
10. In OAuth (RFC6749) worden een aantal methoden voor client authenticatie[[42]](#footnote-43) onderkend. Een client wordt geauthentiseerd op basis van een JWT.
11. We adviseren het toepassen van beveiliging best practices, OAuth 2.0 threat model and security considerations [RFC6819]. Ook wordt aanbevolen om kennis te nemen van de OAuth 2.0 security best current practices [OAUTH-SBP] en JSON Web Tokens [JSONWT-BP].

# Bijlage D: Bronnen

**[RFC2617]**

[HTTP Authentication: Basic and Digest Access Authentication](https://datatracker.ietf.org/doc/html/rfc2617). J. Franks; P. Hallam-Baker; J. Hostetler; S. Lawrence; P. Leach; A. Luotonen; L. Stewart. IETF. June 1999. Draft Standard. URL: <https://datatracker.ietf.org/doc/html/rfc2617>

**[RFC6749]**

[The OAuth 2.0 Authorization Framework](https://datatracker.ietf.org/doc/html/rfc6749). D. Hardt, Ed.. IETF. October 2012. Proposed Standard. URL: <https://datatracker.ietf.org/doc/html/rfc6749>

**[RFC6750]**

[The OAuth 2.0 Authorization Framework: Bearer Token Usage](https://www.rfc-editor.org/rfc/rfc6750). M. Jones; D. Hardt. IETF. October 2012. Proposed Standard. URL: <https://www.rfc-editor.org/rfc/rfc6750>

**[RFC6819]**

[OAuth 2.0 Threat Model and Security Considerations](https://datatracker.ietf.org/doc/html/rfc6819). T. Lodderstedt, Ed.; M. McGloin; P. Hunt. IETF. January 2013. Informational. URL: [*https://datatracker.ietf.org/doc/html/rfc6819*](https://datatracker.ietf.org/doc/html/rfc6819)

**[RFC7518]**

JSON Web Algorithms (JWA)**, RS256**

**[RFC7519]**

[JSON Web Token (JWT)](https://www.rfc-editor.org/rfc/rfc7519)*. M. Jones; J. Bradley; N. Sakimura. IETF. May 2015. Proposed Standard. URL:*[*https://www.rfc-editor.org/rfc/rfc7519*](https://www.rfc-editor.org/rfc/rfc7519)

**[RFC7523]**

[*JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants*](https://www.rfc-editor.org/rfc/rfc7523). M. Jones; B. Campbell; C. Mortimore. IETF. May 2015. Proposed Standard. URL: [*https://www.rfc-editor.org/rfc/rfc7523*](https://www.rfc-editor.org/rfc/rfc7523)

**[RFC7591]**

[OAuth 2.0 Dynamic Client Registration Protocol](https://www.rfc-editor.org/rfc/rfc7591)*. J. Richer, Ed.; M. Jones; J. Bradley; M. Machulak; P. Hunt. IETF. July 2015. Proposed Standard. URL:* [*https://www.rfc-editor.org/rfc/rfc7591*](https://www.rfc-editor.org/rfc/rfc7591)

**[RFC7662]**

*OAuth 2.0 Token Introspection. J. Richer, Ed.. IETF. October 2015. Proposed Standard. URL:* [*https://www.rfc-editor.org/rfc/rfc7662*](https://www.rfc-editor.org/rfc/rfc7662)

**[RFC7800]**

*Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs). M. Jones; J. Bradley; H. Tschofenig. IETF. April 2016. Proposed Standard. URL:* [*https://www.rfc-editor.org/rfc/rfc7800*](https://www.rfc-editor.org/rfc/rfc7800)

**[RFC8414]**

[*OAuth 2.0 Authorization Server Metadata*](https://www.rfc-editor.org/rfc/rfc8414)*. M. Jones; N. Sakimura; J. Bradley. IETF. June 2018. Proposed Standard. URL:*[*https://www.rfc-editor.org/rfc/rfc8414*](https://www.rfc-editor.org/rfc/rfc8414)

**[ RFC9068]**

JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens, V. Bertocci, 2021-10-21 , Proposed Standard URL *https://datatracker.ietf.org/doc/rfc9068/*

# Bijlage E: Begrippen

Kandidaten om toe te voegen aan: <https://rosa.wikixl.nl/index.php/Speciaal:GegevensBekijken/Begrippen?_single&Begrippenset=Edukoppeling>

API: Application Programming Interface. Een API specificeert hoe softwarecomponenten met elkaar moeten communiceren. Het dataformaat is over het algemeen JSON of XML. API's kunnen elk type communicatieprotocol gebruiken.

authorization server (bron RFC6749): The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization.

bearer tokens (RFC6750)

: A security token with the property that any party in possession of the token (a "bearer") can use the token in any way that any other party in possession of it can. Using a bearer token does not require a bearer to prove possession of cryptographic key material (proof-of-possession).

Claim (1edTech): Piece of information asserted about an entity.

Client (RFC6749): An application making protected resource requests on behalf of the resource owner and with its authorization. The term "client" does not imply any particular implementation characteristics (e.g., whether the application executes on a server, a desktop, or other devices). (bron RFC6749)

confidential clients (RFC6749): Clients capable of maintaining the confidentiality of their credentials (e.g., client implemented on a secure server with restricted access to the client credentials), or capable of secure client authentication using other means. (bron RFC6749)

credentials (W3C VC): A credential is a set of one or more claims made by the same entity. Credentials might also include an identifier and metadata to describe properties of the credential, such as the issuer, the expiry date and time, a representative image, a public key to use for verification purposes, the revocation mechanism, and so on. The metadata might be signed by the issuer. A verifiable credential is a set of tamper-evident claims and metadata that cryptographically prove who issued it.

protected resource (access-restricted resource): De gegevens die via een API ontsloten worden

resource owner (bron RFC6749): An entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end-user.

resource server (bron RFC6749) : The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens.

verwerken:

1. Voor deze versie van de Architectuur moet er nog een nieuwe versie van het “Overzicht actuele documentatie en compliance” document opgesteld worden. Na publicatie worden de links opgenomen. [↑](#footnote-ref-2)
2. Voorheen werden de betreffende profielen ook wel SaaS-profielen genoemd. Omdat hier vaak discussie bij ontstond omdat het niet altijd een SaaS-dienst /SaaS-leverancier betrof is de nieuwe naam Mandated Data eXchange geïntroduceerd vanaf deze versie. Dit geldt ook voor alle andere stukken die na maart 2023 zijn gepubliceerd [↑](#footnote-ref-3)
3. Zie metadata Authorization Server [↑](#footnote-ref-4)
4. De edu-from parameter van het request is de edu-to parameter in de response, de edu-to parameter in het request is de edu-from parameter in de response. [↑](#footnote-ref-5)
5. <https://www.edustandaard.nl/standaarden/afspraken/afspraak/edukoppeling/>. Reageren kan via [info@edustandaard.nl](mailto:info@edustandaard.nl). [↑](#footnote-ref-6)
6. Momenteel wordt ook binnen het iGOV NL aan een client credentials profiel gewerkt. We volgen de ontwikkelingen en zullen daar (op termijn) mogelijk gebruik van maken. [↑](#footnote-ref-7)
7. Het Kennisplatform API’s heeft een OAuth code grant profiel ontwikkeld (<https://forumstandaardisatie.nl/open-standaarden/nl-gov-assurance-profile-oauth-20>) welke op de ptolu-lijst van Forum Standaardisatie is opgenomen. Dit betreft dan ook een iets ander scenario dan dit OAuth-profiel dat op basis van OAuth client credentials grant is gebaseerd. [↑](#footnote-ref-8)
8. Het Kennisplatform API’s heeft o.a. een API Strategie ontwikkeld waarin verschillende soorten API's worden onderkend (<https://geonovum.github.io/KP-APIs/API-strategie-algemeen/>). Open API's: voor ontsluiten van diensten zonder toegangsbeperking bijv. open data. Gesloten API's: voor ontsluiten van diensten met toegangsbeperking bijv. persoonsgegevens en vertrouwelijke gegevens of diensten voor specifieke partijen (access-restricted and purpose-limited API’s). [↑](#footnote-ref-9)
9. https://tools.ietf.org/html/rfc2119 [↑](#footnote-ref-10)
10. Een valide mandaat voor de Verwerker bij een bepaalde ketensamenwerking is een randvoorwaarde voor het leveren van een Access Token. [↑](#footnote-ref-11)
11. Op basis van de edu\_org\_id kunnen partijen (AS/client) eventueel ook elkaars autorisatie verifieren voor de betreffende onderwijsorganisatie. Voor deze versie van het OAuth-profiel is dit buiten scope en gaat uit van het MDX OSR protocol en de interacties op basis van de OAuth standaard (RFC6749/RFC6750). [↑](#footnote-ref-12)
12. Deze vorm van extra beveiliging wordt ondersteund met RFC8705 waar we helaas met PKIo certificaten geen gebruik van kunnen maken doordat deze standaard (nu) het gebruik van het subject.serial veld niet ondersteund. [↑](#footnote-ref-13)
13. Ook iun algemene zin is het vanuit beveiliging wenselijk om de levensduur zo kort mogelijk te houden. [↑](#footnote-ref-14)
14. Self-contained: having all that is needed in itself. Self-contained tokens have a data structure that contains metadata and claims to communicate the identity of the user or client over the wire. A popular format would be JSON Web Tokens (JWT). The recipient of a self-contained token can validate the token locally by checking the signature, expected issuer name and expected audience or scope (<https://support.pingidentity.com/s/question/0D51W00007qYzmrSAC/selfcontained-vs-reference-token>). [↑](#footnote-ref-15)
15. <https://www.rfc-editor.org/rfc/rfc7662> [↑](#footnote-ref-16)
16. Het Digikoppeling REST API Profiel maakt gebruik van de Kennisplatform REST-API Design Rules (zie https://publicatie.centrumvoorstandaarden.nl/api/adr/) [↑](#footnote-ref-17)
17. Met het gebruik van een asymmetrische ondertekening wordt ook key management van belang. [↑](#footnote-ref-18)
18. Als er het om gegevens van een bepaalde edu\_org\_id gaat. [↑](#footnote-ref-19)
19. Consent door onderwijsorganisatie voor uitwisseling op gegevensniveau. [↑](#footnote-ref-20)
20. Deze moeten nog uitgewerkt en opgenomen worden. Het idee is om de algoritmen van WUS en REST in lijn met elkaar te houden. Hier gaat het (voorlopig) om het ondertekenen van header infomatie. [↑](#footnote-ref-21)
21. <https://publicatie.centrumvoorstandaarden.nl/dk/restapi/#api-design-rules> [↑](#footnote-ref-22)
22. Hetzelfde geld voor het MDX Secure API REST profiel versie 1.1 [↑](#footnote-ref-23)
23. <https://publicatie.centrumvoorstandaarden.nl/dk/restapi/#afspraken-api-design-rules-extensies> [↑](#footnote-ref-24)
24. Onweerlegbaarheid in de zin van: de verzender van de request/response kan niet ontkennen het bericht verzonden te hebben wanneer deze voorzien is van de digitale handtekening van de afzender. Zie Digikoppeling REST profiel [↑](#footnote-ref-25)
25. <https://publicatie.centrumvoorstandaarden.nl/dk/restapi/#bijlage-gebruik-van-signing-encryptie-in-de-context-van-http-rest-api> [↑](#footnote-ref-26)
26. Het OAuth-profiel wijkt af van de andere MDX profielen doordat bij het koppelvlak met de Resource Server TLS wordt gebruikt. [↑](#footnote-ref-27)
27. Meer informatie via Werkgroep Uniforme Beveiligingsvoorschriften: <https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/> [↑](#footnote-ref-28)
28. <https://www.owasp.org/index.php/OWASP_API_Security_Project> [↑](#footnote-ref-29)
29. Zie voor details de OIN nummersystematiek in het Edukoppeling Identificatie en Authenticatie document. [↑](#footnote-ref-30)
30. We gaan in dit OAuth-profieol uit van een point-to-point verbinding tussen de verwerkers (geen logistieke dienstverlener). [↑](#footnote-ref-31)
31. Verzamelt, opslaat, berekeningen uitvoert, verstrekt en dergelijke. [↑](#footnote-ref-32)
32. <https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/> [↑](#footnote-ref-33)
33. <https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/> [↑](#footnote-ref-34)
34. <https://www.cs.ru.nl/E.Verheul/SIO2019/D2.3_final.pdf> [↑](#footnote-ref-35)
35. <https://www.logius.nl/diensten/digikoppeling/documentatie> [↑](#footnote-ref-36)
36. <https://publicatie.centrumvoorstandaarden.nl/dk/architectuur/> [↑](#footnote-ref-37)
37. Met Secure API duiden we profielen aan waarbij ook de machtiging aan een verwerker een rol speelt. Als we in de Architectuur meer profielen gaan ondersteunen, bijvoorbeeld API key, dan is het wellicht beter om per profiel met een beveiligingsniveau aanduiding te gaan werken. De WUS be-S & be-SE profielen bieden extra functionaliteit (integriteit en integriteit icm vertrouwelijkheid op data niveau) doordat de data in transport ondertekend of ondertekend en versleuteld kan worden. Het OAuth-profiel biedt extra beveiliging doordat de autorisatie technisch geborgd is. Bij de uitwerking van de architectuur kunnen wellicht nog verschillende beveiligingsniveaus aan de profielen toekennen. [↑](#footnote-ref-38)
38. Nader onderzoek moet nog uitwijzen of alle scenario’s door OSR ondersteund kunnen (gaan) worden. [↑](#footnote-ref-39)
39. Het Kennisplatform ontwikkelt nog OAuth profielen. Mochten we op een later moment kunnen aansluiten dan nemen we dat in overweging. [↑](#footnote-ref-40)
40. <https://datatracker.ietf.org/doc/html/rfc6749> (ook OAuth wordt doorontwikkeld: <https://datatracker.ietf.org/doc/draft-ietf-oauth-v2-1/>). [↑](#footnote-ref-41)
41. Dus niet op basis van toestemming door een gebruiker (Resource Owner). [↑](#footnote-ref-42)
42. <https://datatracker.ietf.org/doc/html/rfc6749#section-2.3> [↑](#footnote-ref-43)