Edukoppeling

M2M gegevensuitwisseling binnen het onderwijs

Secure API OAuth Client Credentials profielen

Edustandaard

Datum: juli 2023

Versie: 0.8

Inhoudsopgave

Inhoud

1. Status van dit document

Dit document is een conceptversie van Secure API OAuth profielen welke in samenwerking met het programma Edu-V[1] is opgesteld. Deze conceptversie sluit aan op keuzes die zijn gemaakt binnen het Edu-V afsprakenstelsel en kan worden toegepast bij de geplande proof of concepts[2]. De proof of concepts worden in Q1 2024 geëvalueerd en in de tussentijd zal de werkgroep Edukoppeling een nieuwe versie van de Edukoppeling Architectuur ontwikkelingen. Deze architectuur bevat generieke kaders die breed toepasbaar zijn en zal samen met de ervaringen uit de Edu-V keten gebruikt worden voor het ontwikkelen van nieuwe versies van Edukoppeling profielen.

1.1. 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	Gebaseerd op WG bijeenkomst maart 2023. Zie document versie 0.3
0.4	Concept	E. Reinhoud (BES)	Juni 2023	Gebaseerd op WG bijeenkomsten juni 2023 Zie document versie 0.4
0.5	Concept	E. Reinhoud (BES)	Juni 2023	Zie document versie 0.5
0.61	Concept	E. Reinhoud (BES)	Juli 2023	Zie document versie 0.61
0.9	Concept	E. Reinhoud (BES)	Juli 2023	Openbare consultatie versie op basis van versie 0.61

[1] Zie Afsprakenstelsel Edu-V - Confluence (atlassian.net)
[2] Zie Proof of concept handreiking leveranciers - Afsprakenstelsel Edu-V - Confluence (atlassian.net)

2. Inleiding

2.1. 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 de adoptie van Edukoppeling is daar verandering in aangebracht. 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.

2.2. Secure API OAuth client credentials profielen

In dit document wordt het OAuth client credentials profiel als basis gebruikt voor het komen tot specifieke OAuth-profielen die in ieder geval toepasbaar zijn binnen de leermiddelenketen voor de werkingsgebieden po, vo en mogelijk mbo die op basis van het in opbouw zijnde Edu-V afsprakenstelsel gaat worden ingericht. Deze profielen sluiten daarom ook aan op de verschillende gegevensclassificaties zoals gedefinieerd in Edu-V[3]. Uitbreiding van de toepassing naar andere contexten wordt komende tijd in de Edukoppeling werkgroep besproken. Samen met de ervaringen uit de Edu-V proof of concepts kan dit tot aanpassingen van de OAuth profielen leiden. Begin 2024 worden een nieuwe versie van de architectuur en nieuwe versies van de onderliggende profielen opgeleverd.

[3] https://edu-v.atlassian.net/wiki/spaces/AFSPRAKENS/pages/1998909/M2M+gegevensuitwisselingen#Classificatie-van-gegevenssoorten%3A-I%2C-II%2C-III-en-IV.

Classificatie	I.	II.	III.	IV.
Vertrouwelijkheid	Niet vertrouwelijk	Niet vertrouwelijk	**Vertrouwelijk**	Vertrouwelijk
Regie op gegevens-uitwisseling	Leverancier	**Onderwijs-organisatie**	Leverancier	**Onderwijs-organisatie**
Persoonsgegevens	Nee	Nee	Nee	**Ja**
Verwerkers-overeenkomst	Nee	Nee	Nee	**Ja**
Voorbeeld	Catalogus-informatie	SchoolVak SchoolPeriode	Normeringen	Onderwijs-deelnemers en –medewerkers Gebruikers-gegevens

Figuur 1 - Gegevensclassificaties zoals gedefinieerd in Edu-V

In de naam is daarom een referentie naar de gegevensclassificatie opgenomen. Het betreft de volgende profielen:

1. **Secure API OAuth client credentials profiel (GCI)**: Dit profiel wordt gebruikt bij gegevensclassificatie I. Het betreft uitwisseling van niet-vertrouwelijke gegevens door ketenpartij namens zichtzelf. Transportbeveiliging is op basis van TLS en naar het token endpoint wordt Basic Authentication (client password) vereist. Voor autorisatie is in de header van het request naar RS het OAuth Access Token opgenomen.

2. **Secure API OAuth client credentials profiel (GCII)**: Dit profiel wordt gebruikt bij gegevensclassificatie II. Het betreft de uitwisseling van niet-vertrouwelijke gegevens door een ketenpartij die dit namens een onderwijsorganisatie uitvoert. Transportbeveiliging is op basis van TLS en naar het token endpoint wordt Basic Authentication (client password) vereist. Voor autorisatie is in de header van het request naar RS het OAuth Access Token opgenomen. Daarnaast is in de query string een referentie die naar betreffende onderwijsorganisatie (eu_org_id) opgenomen om via consentmanagement te controleren of er door de betreffende onderwijsorganisatie consent is gegeven.

3. **Secure API OAuth client credentials profiel (GCIII)**: Dit profiel wordt gebruikt bij gegevensclassificatie III. Het betreft de uitwisseling van vertrouwelijke gegevens door ketenpartij namens zichtzelf. Transportbeveiliging is op basis van TLS, maar client authenticatie is op basis van mTLS en PKIo-certificaat. Echter, daarnaast wordt ook de toepassing van Basic Authentication (client password) toegestaan. Voor autorisatie is in de header van het request naar RS het OAuth Access Token opgenomen.

4. **Secure API OAuth client credentials profiel (GCIV)**: Dit profiel wordt gebruikt bij gegevensclassificatie IV. Het betreft de uitwisseling van vertrouwelijke gegevens door

een ketenpartij die dit namens een onderwijsorganisatie uitvoert. Transportbeveiliging is op basis van TLS, maar client authenticatie is op basis van mTLS en een PKIo-certificaat. Echter, daarnaast wordt ook de toepassing van Basic Authentication (client secret) toegestaan. Voor autorisatie is in de header van het request naar RS het OAuth Access Token opgenomen. Daarnaast is in de query string een referentie die naar betreffende onderwijsorganisatie (eu_org_id) opgenomen om via consentmanagement te controleren of er door de betreffende onderwijsorganisatie consent is gegeven voor betreffende client en of toestemming is gegeven door een bestuur van een onderwijsorganisatie (o.b.v. toestemming in de vorm van bijvoorbeeld een afgegeven mandaat of een afgesloten verwerkersovereenkomst).

Profiel	Data		API		Client Authenticatie		Autorisatie		
	OPEN	GESLOTEN (vertrouwelijk)	OPEN	GESLOTEN (toegangs beperking)	Client password	mTLS/ PKIo	Access Token	Access Token in combinatie met edu_org_id in query string	Access Token in combinatie met edu_org_id in query string en verificatie toestemming bestuur onderwijsorganisatie (o.b.v. mandaat, verwerkersovereenkomst, of …)
GCI	X			X	X		X		
GCII	X			X	X			X	
GCIII		X		X	(MAY)	X	X		
GCIV		X		X	(MAY)	X			X

2.3. Doel en doelgroep

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.

Deze OAuth[4] client credentials profielen zijn in eerste instantie bedoeld als ondersteuning aan de Edu-V proof-of-concepts en de daaropvolgende implementaties.

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

[4] 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.
[5] https://www.edustandaard.nl/standaarden/afspraken/afspraak/edukoppeling/. Reageren kan via info@edustandaard.nl. Vragen rond PKIo of OIN kunnen eventueel ook naar digikoppeling@logius.nl gestuurd worden.

2.4. 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[6]. Deze gebruikt de volgende termen: "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL".

2.5. Leeswijzer

In hoofdstuk 2 wordt de aanleiding, het doel en de doelgroep voor dit profiel beschreven. In de hoofdstukken 3 t/m 6 worden de verschillende OAuth client credentials profielen beschreven. Elk profiel is voor een specifieke gegevensclassificatie (GC). In het eerste profiel (GCI) worden de interacties (stappen 1 t/m 4) in detail beschreven. Hier wordt in de overige profielen (GCII t/m GC IV) zoveel mogelijk naar verwezen. In hoofdstuk 7 worden een aantal metadata gegevens weergegeven die o.a. bij de registratie van de client relevant zijn. Een belangrijk onderdeel voor het API design zijn de API Design Rules. De eisen hiervoor worden beschreven in hoofdstuk 8. Hierbij wordt verwezen naar het betreffende onderdeel van het Digikoppeling Restful API profiel.

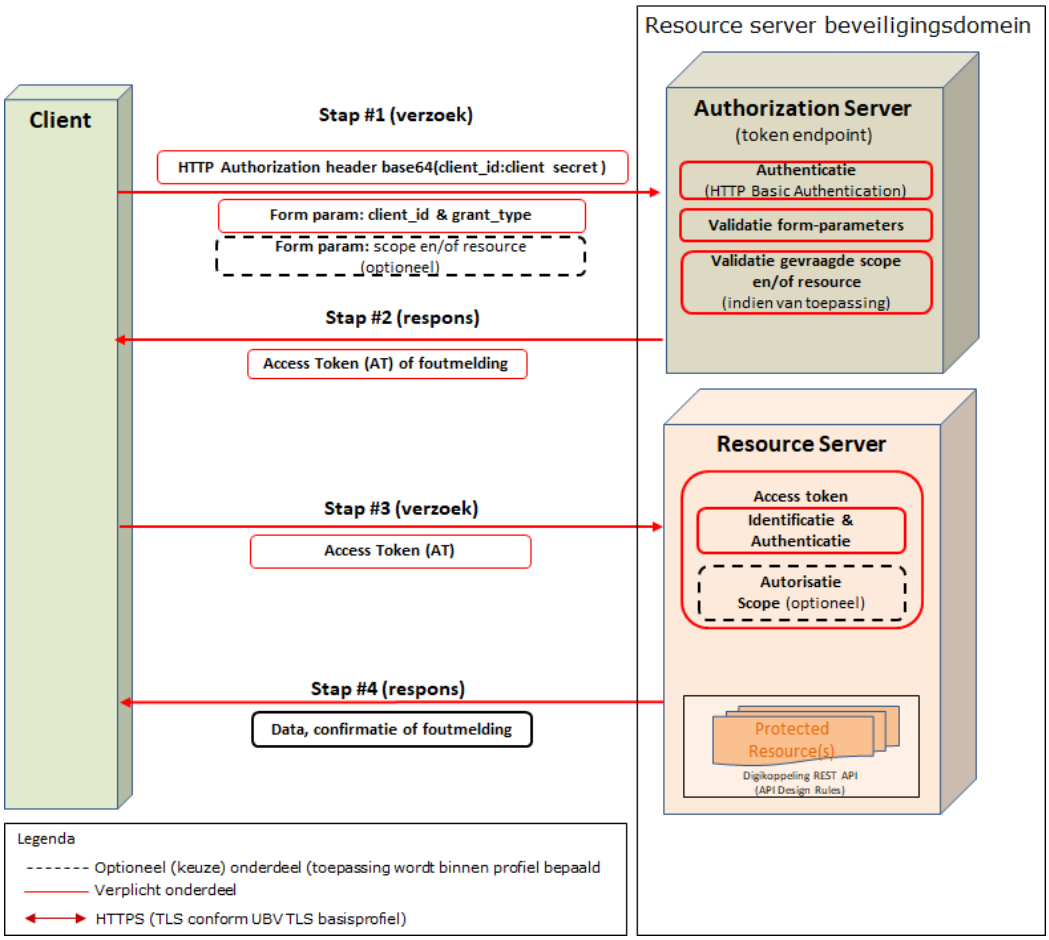
[6] https://tools.ietf.org/html/rfc2119

3. Secure API OAuth client credentials profiel (GCI)

3.1. Functioneel toepassingsgebied

Het functionele toepassingsgebied van dit OAuth client credentials profiel betreft M2M-gegevensuitwisseling via een point-to-point verbinding voor uitwisseling van open data via een gesloten API[7] namens de ketenpartijen zelf. Er worden bevragingen en meldingen op basis van een request-response[8] uitwisselingspatroon ondersteund.

De client is in deze context geen browser, maar een systeem (confidential client). Deze wordt bij de Authorization Server geauthentiseerd op basis van HTTP Basic Authentication scheme[9]. Autorisatie wordt bepaald op basis van een toegangstoken (OAuth Access Token).Transportbeveiliging wordt gerealiseerd met de toepassing van TLS.



Figuur 2 - Schematische weergave Secure API OAuth client credentials profiel (GCI)

[7] De Digikoppeling architectuur onderkent verschillende soorten API's (Digikoppeling Architectuur 2.0.2 (logius.nl)). Open API's (diensten) 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).
[8] Bijvoorbeeld pull (bijvoorbeeld HTTP Get) en push (bijvoorbeeld op basis van een HTTP Put)
[9] Zie RFC2617 https://datatracker.ietf.org/doc/html/rfc2617\

3.2. Verzoek naar Token Endpoint (STAP #1)

3.2.1. MUST: Basic Authentication

1) De client moet (must) in HTTP Authorization header een base64 encodering van client_id en client password op conform RFC2617 (zie ook RFC6749[10]).

3.2.2. MUST: HTTP POST request met form-parameters

2) Conform RFC6749 moet (must) er een POST request naar het token endpoint gestuurd worden.

3) In de request body moeten (must) de volgende form-parameters opgenomen zijn:
   a) **grant_type** met de waarde “client_credentials”.

4) In de request body mogen de volgende form-parameters opgenomen zijn:
   a) **client_id** met de waarde die de Authorization Server aan de client heeft gegeven bij registratie.

5) In de request body mogen (may) de volgende form-parameters opgenomen zijn:
   a) **scope**: de client mag een Access Token aanvragen met een specifieke scope;
   b) **resource**[11]: de client mag een Access Token aanvragen voor een specifieke resource.

Een voorbeeld van het verzoek (stap #1) wordt weergegeven in Figuur 3.

```
POST /token HTTP/1.1
Host: aserver.com
Authorization: Basic Y2xpZW50X2lkPTU1ZjlmNTU….TM1MWE1ODZiNzQ4NCZjbGllbnRfc2VjcmV0PXdyd0ZHRkRTRFNBRGFhc2RzYXNh
Content-Length: <length>
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&client_id=55f9f559-2496-49d4-b6c3-351a586b7484
```

Figuur 3 - Stap 1: Verzoek van client aan authorization server token endpoint

3.3. Antwoord van Token Endpoint (STAP #2)

1) Als de Authorization Server stap #1 succesvol heeft gevalideerd, moet (must) deze een HTTP 200 OK status code respons geven conform RFC6749[12].

2) Het antwoord mag (may) conform aan RFC9068 als media type “application/at+jwt" opnemen.

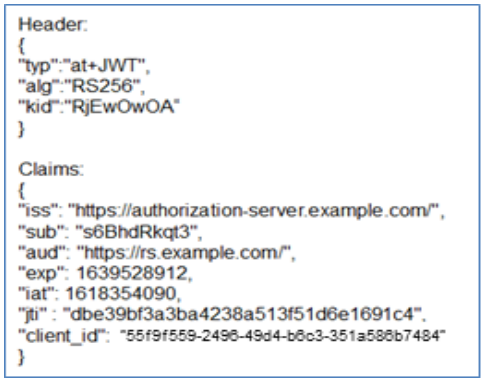
3) Het access token moet (must) conformeren aan RFC9068.
   a) Het access token moet (must) ondertekend zijn conform RFC7515
   b) De Authorization Server moet (must) de RS256 signature method (RFC7518) ondersteunen en zou (should) de PS256 signature method moeten ondersteunen.

[10] https://datatracker.ietf.org/doc/html/rfc6749#section-2.3.1
[11] Dit optionele claim is relevant als de Authorization Server tokens uitgeeft voor meerdere API’s en er niet op basis van de scope een bepaalde resource kan worden geïdentificeerd. Dit wordt ondersteund met de claim “resource” (conform RFC8707)
[12] https://datatracker.ietf.org/doc/html/rfc6749#section-5.1

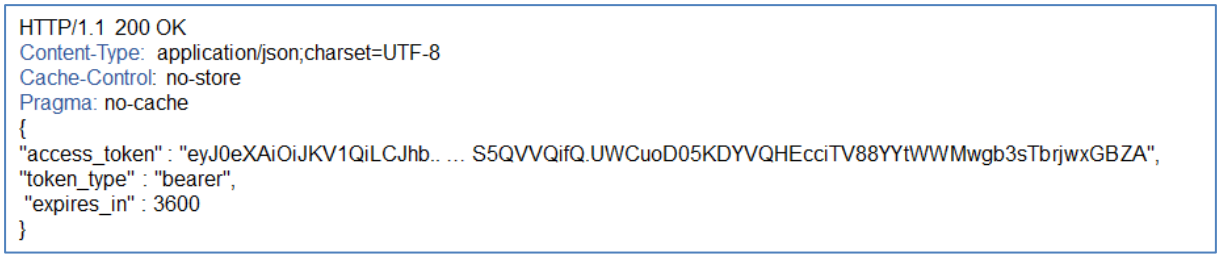
4) Het access token krijgt een korte levensduur (een uur of korter).
   a) Token revocation is niet van toepassing, maar we sluiten het gebruik van token revocation echter niet uit.
   b) Als er niet met een “resource” claim of een “scope” claim wordt gewerkt die de Authorization Server kan gebruiken om een protected resource te identificeren, dan wordt aangeraden om de “aud” claim te vullen met een default resource waarde.

5) De respons bevat het access token
   a) Het access token wordt in de respons opgenomen conform RFC6749[13].

6) De respons zou geen (should not) refresh token moeten bevatten[14].
   a) Er is geen interactie met een eindgebruiker (resource owner). Op basis van een client authenticatie aanroep kan een nieuw access token aangevraagd worden en maakt de functie van een refresh token overbodig.



Figuur 4 - Niet base64-encoded en niet ondertekend Access Token



Figuur 5 - Respons van AS token endpoint

3.3.1. Foutmelding - verzoek stap #1 bevat fout(en)

1) Als de Authorization Server de client niet succesvol heeft kunnen authenticeren dan moet (must) het verzoek worden afgewezen.

2) Als een scope en/of resource van toepassing is en de client heeft in het verzoek een scope en/of resource opgenomen waarvoor ze niet geautoriseerd zijn dan moet (must) het verzoek worden afgewezen.

[13] https://datatracker.ietf.org/doc/html/rfc6749#section-5.1
[14] RFC6749 4.4.3: “A refresh token SHOULD NOT be included.”

3) Als de Authorization Server het verzoek in stap #1 heeft afgewezen dan moet (must) deze een HTTP 400 Bad Request status code respons geven. In de body is een foutmelding opgenomen conform RFC6749[15].

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-
Cache-Control: no-store
Pragma: no-cache

{
"error" : "invalid_scope"
}
```

Figuur 6 – Voorbeeld foutmelding als de AS een scope vereist en deze niet voldoet: HTTP 400 Bad Request status code

3.4. Request naar Resource Server (stap #3)

De client heeft van de Authorization Server een Access Token ontvangen. Het Access Token is voor de client betekenisloos. Het verzoek naar de Resource Server moet aan de onderstaande eisen voldoen.

3.4.1. MUST: Access Token in header

1) De client biedt het access token aan in de Authorization request header met toepassing van de "Bearer" authentication scheme. Een client mag (may) de form-parameter gebruiken als het gebruik van de Authorization request header niet mogelijk is.
   a) Zie RFC6750[16]:
      "*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*."

```
GET /resource/1 HTTP/1.1
Host: provider.example.com
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhb. UWCuoD05KDYVQHEcciTV88YYtWWMwgb3sTbrjwxGBZA
```

Figuur 7 - Access token in Authorization request header

3.5. Antwoord van Resource Server (STAP #4)

De Resource Server verleent toegang aan clients als ze een geldig access_token (en indien van toepassing met juiste scope en/of resource) presenteren. Resource Servers vertrouwen erop dat de Authorization Server de client op de juiste manier authenticeert.

1) Als de Resource Server het request van stap #3 succesvol heeft gevalideerd, moet (must) deze een HTTP 200 OK status code respons geven.
2) In de respons wordt afhankelijk van de context de data of een confirmatie opgenomen.

[15] https://datatracker.ietf.org/doc/html/rfc6749#section-5.2
[16] https://datatracker.ietf.org/doc/html/rfc6750#section-2.1

3.5.1. Foutmelding - verzoek stap #3 bevat fout(en)

4. Als de Resource Server het request van stap #3 niet succesvol heeft kunnen valideren dan wordt er een foutmelding gegeven. Zie voor mogelijke opties in de respons de ADR extensies (zie API Design Rules (ADR)Client registratie en Authorization Server metadata

4.1. Confidential client registratie

De confidential clients worden niet dynamisch geregistreerd. Om interacties succesvol te laten verlopen moeten er bij de AS wel een aantal gegevens geregistreerd worden. Voor het eenduidig definiëren hiervan gebruiken we wel de definities van de dynamische registratie meta data standaard.

1) De client wordt bij de Authorization Server geregistreerd met de volgende informatie:
   a) client_id wordt bepaald door AS
      i) De client krijgt bij registratie een identifier (client_id) van de Authorization Server .
      ii) 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.
   b) client_name wordt bepaald door AS
   c) token_endpoint_auth_method
      i) Bij profiel GCI & GCII = “client_secret_basic”
      ii) Bij profiel GCIII & GCIV (mTLS) = “tls_client_auth”.
   d) grant_types = "client_credentials”.
   e) scope(s)
      i) Als de AS scopes ondersteunt dan kunnen deze bij registratie aangeven worden.
   f) resource
      i) Het kan zijn dat de AS voor meerdere API’s een Access Tokens kan uitgeven. Dit vereist dat er voor een client aangegeven wordt voor welke resource een Access Token kan worden gevraagd. Hiervoor ondersteunt dit OAuth-profiel een optioneel gegeven “resource” (conform RFC8707).

4.2. Authorization Server metadata

1) Issuer: De issuer identifier (URL) van de Authorization Server .
   a) De issuer moet herleid kunnen worden naar het OIN van de verwerker die de Authorization Server beheert.

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

3) scopes_supported
   Een lijst met de scopes die de Authorization Server ondersteunt.
   a) 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.
4) grant_types_supported

Een lijst (zie RFC7591) met de grant types die de Authorization Server ondersteunt.
a) Voor dit profiel moet de lijst de waarde “client_credentials” bevatten.

5) token_endpoint_auth_methods_supported
Een lijst (zie RFC8414) van client authenticatie methoden die de Authorization Server ondersteunt.
a) Voor profiel GCIII & GCIV moet de lijst de waarde “client_secret_basic” bevatten
b) Voor profiel GCIII & GCIV moet de lijst de waarde “tls_client_auth” bevatten.

1) API Design Rules (ADR)).

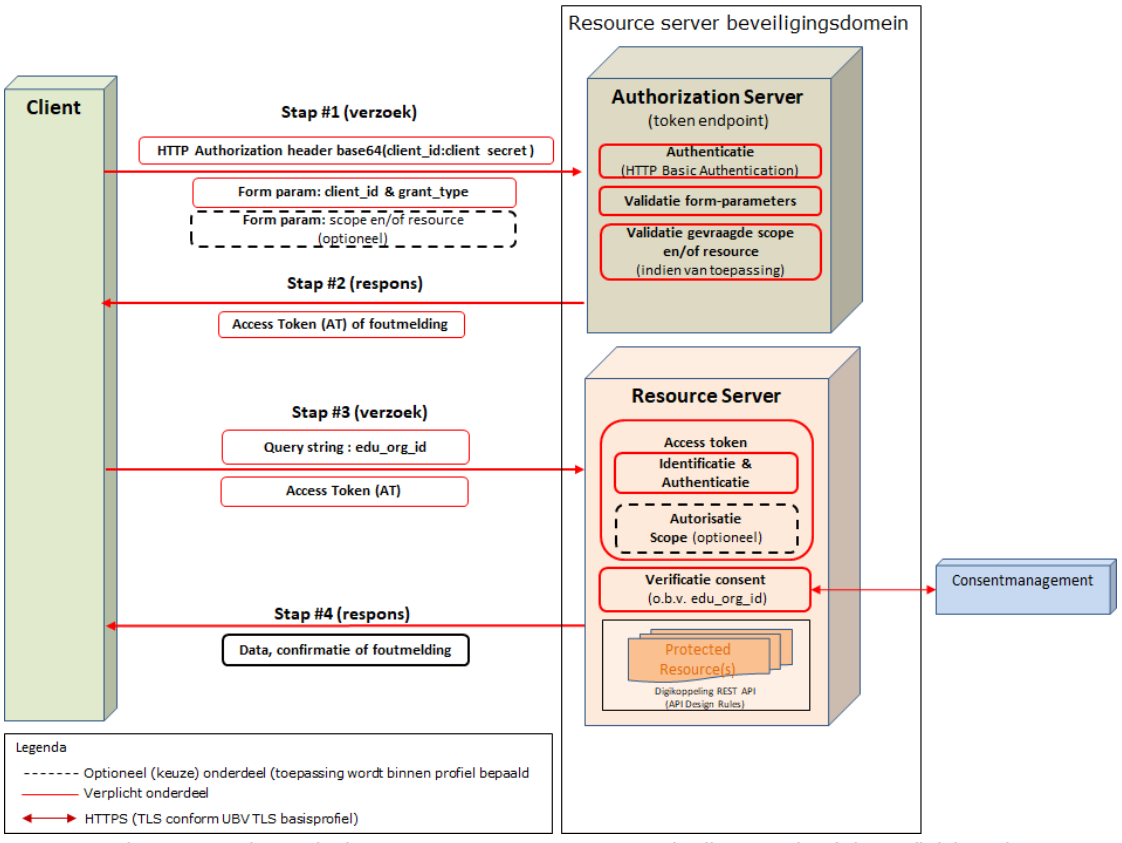
5. Secure API OAuth client credentials profiel (GCII)

5.1. Functioneel toepassingsgebied

Het functionele toepassingsgebied van dit OAuth client credentials profiel betreft M2M-gegevensuitwisseling via een point-to-point verbinding voor uitwisseling van open data via een gesloten API[17] namens een onderwijsorganisatie. Er worden bevragingen en meldingen op basis van een request-response[18] uitwisselingspatroon ondersteund.

De client is in deze context geen browser, maar een systeem (confidential client). Deze wordt bij de Authorization Server geauthentiseerd op basis van HTTP Basic authentication. Autorisatie wordt bepaald op basis van een toegangstoken (OAuth Access Token) in combinatie met een edu_org_id die via query string geleverd wordt. Transportbeveiliging wordt gerealiseerd met de toepassing van TLS.

Dit profiel is voor scenario's waarbij een uitwisseling namens een onderwijsorganisatie wordt uitgevoerd. Dit wordt technisch ondersteund met een referentie (edu_org_id) in de query string van het request naar de Resource Server. De autorisatie wordt geverifieerd door een consentmanagement component, Hoe deze is ingericht valt buiten de scope van dit profiel en dient binnen de betreffende keten zelf gedefinieerd te worden.



Figuur 8 - Schematische weergave Secure API OAuth client credentials profiel (GCII)

[17] De Digikoppeling architectuur onderkent verschillende soorten API's (Digikoppeling Architectuur 2.0.2 (logius.nl)). Open API's (diensten) 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).

[18] Bijvoorbeeld pull (bijvoorbeeld HTTP Get) en push (bijvoorbeeld op basis van een HTTP Put)

5.2. Verzoek naar Token Endpoint (STAP #1)

Conform Secure API OAuth client credentials profiel (GCI).

5.3. Antwoord van Token Endpoint (STAP #2)

Conform Secure API OAuth client credentials profiel (GCI).

5.4. Request naar Resource Server (stap #3)

Conform Secure API OAuth client credentials profiel (GCI) met de volgende toevoegingen:

5.4.1. MUST: edu_org_id in query string

1) Het verzoek van stap #3 moet (must) een edu_org_id bevatten. Deze is opgenomen in de query string van het request. Hiermee kan aangegeven namens welke onderwijsorganisatie de uitwisseling plaatsvindt.
2) Een Resource Server moet (must) (via consentmanagementlaag) controleren of er consent bestaat.
    a. De Resource Server verifieert of er een consent is op de uitwisseling van de gegevenssoort van de gevraagde protected resource.

5.5. Antwoord van Resource Server (STAP #4)

5.5.1. Foutmelding - verzoek stap #3 bevat fout(en)

1) Als de Resource Server het request van stap #3 niet succesvol heeft kunnen valideren dan moet (must) er een foutmelding gegeven. Zie voor mogelijke opties in de respons de ADR extensies (zie API Design Rules (ADR)).
2) Bij een fout met betrekking tot het edu_org_id moet (must) er geen toegang tot de protected resource worden gegeven. Als antwoord moet (must) één van de onderstaande foutmeldingen worden gegeven.

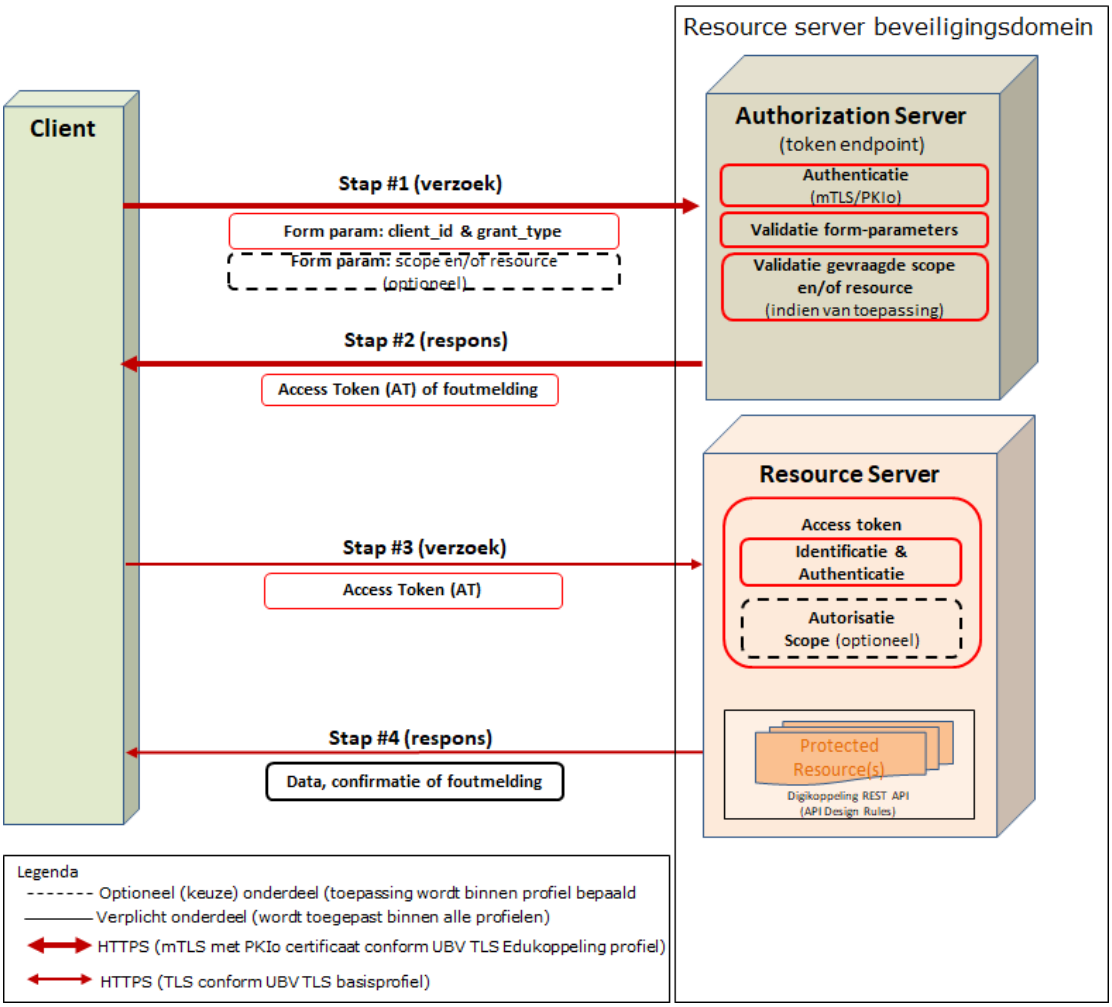
Status code	Omschrijving	Categorie	Domein	Toelichting
400	Bad Request	Syntax	EK	edu_org_id ontbreekt
400	Bad Request	Syntax	EK	edu_org_id is niet van het juiste formaat
403	Forbidden	Consent	EK	Autorisatiefout: geen consent voor edu_org_id gevonden

6. Secure API OAuth client credentials profiel (GCIII)

6.1. Functioneel toepassingsgebied

Het functionele toepassingsgebied van dit OAuth client credentials profiel betreft M2M-gegevensuitwisseling via een point-to-point verbinding voor uitwisseling van gesloten data via een gesloten API[19] namens de ketenpartijen zelf. Er worden bevragingen en meldingen op basis van een request-response[20] uitwisselingspatroon ondersteund.

De client is in deze context geen browser, maar een systeem (confidential client). Deze wordt bij de Authorization Server geauthentiseerd op basis van mTLS en een PKIo-certificaat. Autorisatie wordt bepaald op basis van een toegangstoken (OAuth Access Token). Transportbeveiliging wordt gerealiseerd met de toepassing van TLS.



Figuur 9 - Schematische weergave Secure API OAuth client credentials profiel (GCIII)

[19] De Digikoppeling architectuur onderkent verschillende soorten API's (Digikoppeling Architectuur 2.0.2 (logius.nl)). Open API's (diensten) 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).

[20] Bijvoorbeeld pull (bijvoorbeeld HTTP Get) en push (bijvoorbeeld op basis van een HTTP Put)

6.2. Verzoek naar Token Endpoint (STAP #1)

6.2.1. MUST: mTLS voor client authenticatie

1) Het verzoek moet (must) worden verstuurt op basis van mTLS. De client moet (must) hiervoor een PKIo-certificaat[21] met het OIN gebruiken. Op basis van het OIN kan de verwerker die ook verantwoordelijk is voor het verzoek van deze client geïdentificeerd worden. Hierbij gelden de voorschriften van het UBV TLS Edukoppeling profiel[22].

6.2.2. MAY: Basic Authentication voor client authenticatie

1) De client mag (may) in de HTTP Authorization header een base64 encodering van client_id en client password op conform RFC2617 (zie ook RFC6749[23]) opnemen.

6.2.3. MUST: HTTP POST request met form-parameters

Conform Secure API OAuth client credentials profiel (GCI).

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

```
POST /token HTTP/1.1
Host: aserver.com
Content-Length: <length>
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&client_id=55f9f559-2496-49d4-b6c3-351a586b7484
```

Figuur 10 - Stap 1: Verzoek van client aan authorization server token endpoint (authenticatie o.b.v. mTLS)

6.3. Antwoord van Token Endpoint (STAP #2)

Conform Secure API OAuth client credentials profiel (GCI).

6.4. Request naar Resource Server (stap #3)

Conform Secure API OAuth client credentials profiel (GCI).

6.5. Antwoord van Resource Server (STAP #4)

Conform Secure API OAuth client credentials profiel (GCI).

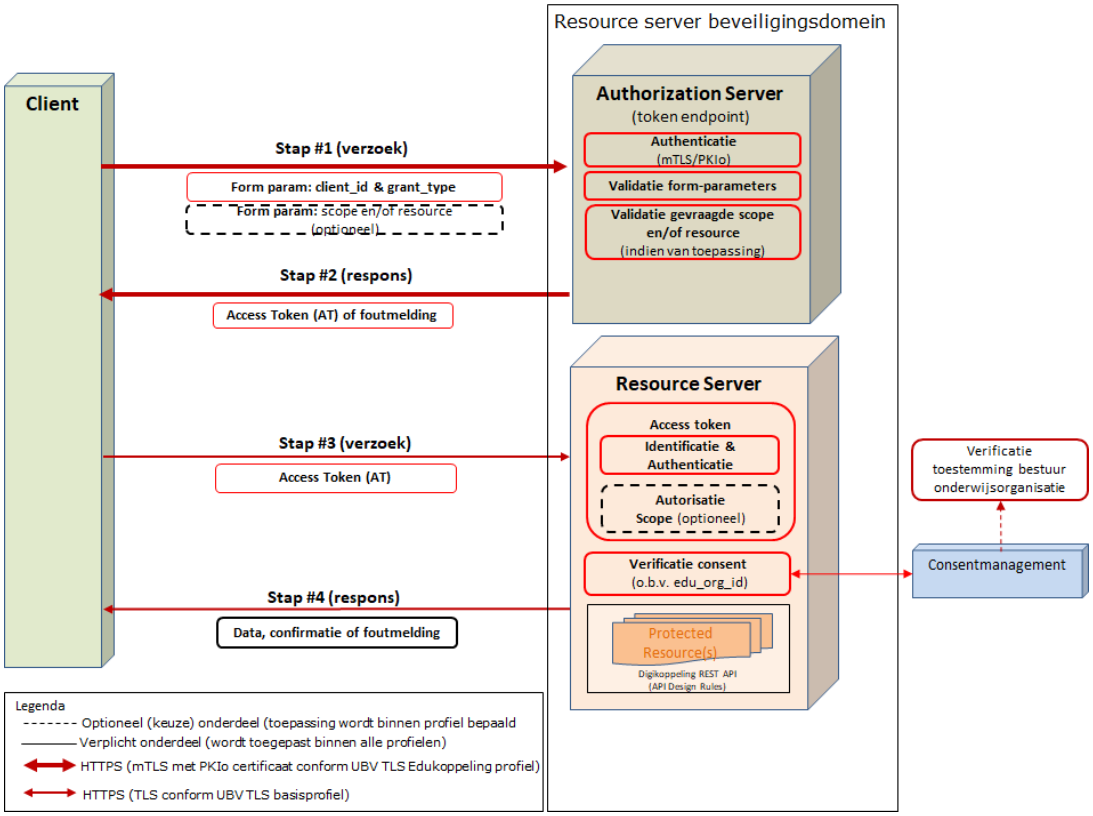
[21] Zie Digikoppeling Gebruik en Achtergrond Certificaten 1.6.3 (logius.nl)
[22] https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/
[23] https://datatracker.ietf.org/doc/html/rfc6749#section-2.3.1

7. Secure API OAuth client credentials profiel (GCIV)

7.1. Functioneel toepassingsgebied

Het functionele toepassingsgebied van dit OAuth client credentials profiel betreft M2M-gegevensuitwisseling via een point-to-point verbinding voor uitwisseling van gesloten data via een gesloten API[24] namens een onderwijsorganisatie. Er worden bevragingen en meldingen op basis van een request-response[25] uitwisselingspatroon ondersteund.

De client is in deze context geen browser, maar een systeem (confidential client). Deze wordt bij de Authorization Server geauthentiseerd op basis van mTLS en een PKIo-certificaat. Autorisatie wordt bepaald op basis van een toegangstoken (OAuth Access Token) en de verificatie van consent en of toestemming is gegeven door bestuur onderwijsorganisatie (o.b.v. mandaat, verwerkersovereenkomst, of …). Transportbeveiliging wordt gerealiseerd met de toepassing van TLS.



Figuur 11 - Schematische weergave Secure API OAuth client credentials profiel (GCIV)

7.2. Verzoek naar Token Endpoint (STAP #1)

Conform Secure API OAuth client credentials profiel (GCIII).

[24] De Digikoppeling architectuur onderkent verschillende soorten API's (Digikoppeling Architectuur 2.0.2 (logius.nl)). Open API's (diensten) 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).

[25] Bijvoorbeeld pull (bijvoorbeeld HTTP Get) en push (bijvoorbeeld op basis van een HTTP Put)

7.3. Antwoord van Token Endpoint (STAP #2)

Conform Secure API OAuth client credentials profiel (GCIII).

7.4. Request naar Resource Server (stap #3)

Conform Secure API OAuth client credentials profiel (GCIII) met de volgende toevoegingen:

7.4.1. MUST: edu_org_id in query string

1) Het verzoek van stap #3 moet (must) een edu_org_id bevatten. Deze is opgenomen in de query string van het request.
2) Een Resource Server moet (must) (via consentmanagementlaag) controleren of er consent en toestemming door bestuur van een onderwijsorganisatie bestaat.
    a. De Resource Server verifieert voor het request of er consent is gegeven voor de uitwisseling van de gegevenssoort van de gevraagde protected resource.
    b. Bij de registratie van het consent is gecontroleerd of toestemming is gegeven door een bestuur van een onderwijsorganisatie (o.b.v. mandaat, verwerkersovereenkomst, of …).

7.5. Antwoord van Resource Server (STAP #4)

Conform Secure API OAuth client credentials profiel (GCIII) met de volgende toevoegingen:

7.5.1. Foutmelding - verzoek stap #3 bevat fout(en)

8. Als de Resource Server het request van stap #3 niet succesvol heeft kunnen valideren dan moet (must) er een foutmelding worden gegeven. Zie voor mogelijke opties in de respons de ADR extensies (zie API Design Rules (ADR)Client registratie en Authorization Server metadata

8.1. Confidential client registratie

De confidential clients worden niet dynamisch geregistreerd. Om interacties succesvol te laten verlopen moeten er bij de AS wel een aantal gegevens geregistreerd worden. Voor het eenduidig definiëren hiervan gebruiken we wel de definities van de dynamische registratie meta data standaard.

2) De client wordt bij de Authorization Server geregistreerd met de volgende informatie:
    a) client_id wordt bepaald door AS
        i) De client krijgt bij registratie een identifier (client_id) van de Authorization Server .
        ii) 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.
    b) client_name wordt bepaald door AS
    c) token_endpoint_auth_method
        i) Bij profiel GCI & GCII = “client_secret_basic”
        ii) Bij profiel GCIII & GCIV (mTLS) = “tls_client_auth”.
    d) grant_types = "client_credentials”.
    e) scope(s)
        i) Als de AS scopes ondersteunt dan kunnen deze bij registratie aangeven worden.
    f) resource

i) Het kan zijn dat de AS voor meerdere API's een Access Tokens kan uitgeven. Dit vereist dat er voor een client aangegeven wordt voor welke resource een Access Token kan worden gevraagd. Hiervoor ondersteunt dit OAuth-profiel een optioneel gegeven "resource" (conform RFC8707).

8.2. Authorization Server metadata

6) Issuer: De issuer identifier (URL) van de Authorization Server .
   a) De issuer moet herleid kunnen worden naar het OIN van de verwerker die de Authorization Server beheert.

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

8) scopes_supported
   Een lijst met de scopes die de Authorization Server ondersteunt.
   a) 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.
9) grant_types_supported
   Een lijst (zie RFC7591) met de grant types die de Authorization Server ondersteunt.
   a) Voor dit profiel moet de lijst de waarde "client_credentials" bevatten.

10) token_endpoint_auth_methods_supported
    Een lijst (zie RFC8414) van client authenticatie methoden die de Authorization Server ondersteunt.
    a) Voor profiel GCIII & GCIV moet de lijst de waarde "client_secret_basic" bevatten
    b) Voor profiel GCIII & GCIV moet de lijst de waarde "tls_client_auth" bevatten.

1) API Design Rules (ADR)).
2) Bij een fout met betrekking tot het edu_org_id, de verificatie van het consent, of als er geen toestemming is gegeven door een bestuur van een onderwijsorganisatie (o.b.v. mandaat, verwerkersovereenkomst, of …) moet (must) er geen toegang tot de protected resource worden gegeven. Als antwoord moet (must) één van de onderstaande foutmeldingen worden gegeven.

Status code	Omschrijving	Categorie	Domein	Toelichting
400	Bad Request	Syntax	EK	edu_org_id ontbreekt
400	Bad Request	Syntax	EK	edu_org_id is niet van het juiste formaat
403	Forbidden	Consent	EK	Autorisatiefout: geen consent voor edu_org_id gevonden
403	Forbidden	Toestemming bestuur onderwijs-organisatie	EK	Autorisatiefout: geen toestemming gegeven door bestuur onderwijsorganisatie (o.b.v. mandaat, verwerkersovereenkomst, of …) voor betreffende edu_org_id

9. Client registratie en Authorization Server metadata

9.1. Confidential client registratie

De confidential clients worden niet dynamisch geregistreerd. Om interacties succesvol te laten verlopen moeten er bij de AS wel een aantal gegevens geregistreerd worden. Voor het eenduidig definiëren hiervan gebruiken we wel de definities van de dynamische registratie meta data standaard[26].

3) De client wordt bij de Authorization Server geregistreerd met de volgende informatie:
   a) client_id wordt bepaald door AS
      i) De client krijgt bij registratie een identifier (client_id) van de Authorization Server .
      ii) 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.
   b) client_name wordt bepaald door AS
   c) token_endpoint_auth_method
      i) Bij profiel GCI & GCII = “client_secret_basic”
      ii) Bij profiel GCIII & GCIV (mTLS) = “tls_client_auth”[27].
   d) grant_types = "client_credentials”.
   e) scope(s)

[26] OAuth 2.0 Dynamic Client Registration Protocol [RFC7591]
[27] Voor deze optie wordt gebruik gemaakt van de meta data definities in RFC 8705: OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens (rfc-editor.org)

   i) Als de AS scopes ondersteunt dan kunnen deze bij registratie aangeven worden.
f) resource
   i) Het kan zijn dat de AS voor meerdere API's een Access Tokens kan uitgeven. Dit vereist dat er voor een client aangegeven wordt voor welke resource een Access Token kan worden gevraagd. Hiervoor ondersteunt dit OAuth-profiel een optioneel gegeven "resource" (conform RFC8707).

9.2. Authorization Server metadata

11) Issuer: De issuer identifier (URL) van de Authorization Server .
   a) De issuer moet herleid kunnen worden naar het OIN van de verwerker die de Authorization Server beheert.

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

13) scopes_supported
   Een lijst met de scopes die de Authorization Server ondersteunt.
   a) 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.
14) grant_types_supported
   Een lijst (zie RFC7591) met de grant types die de Authorization Server ondersteunt.
   a) Voor dit profiel moet de lijst de waarde "client_credentials" bevatten.

15) token_endpoint_auth_methods_supported
   Een lijst (zie RFC8414) van client authenticatie methoden die de Authorization Server ondersteunt.
   a) Voor profiel GCIII & GCIV moet de lijst de waarde "client_secret_basic" bevatten
   b) Voor profiel GCIII & GCIV moet de lijst de waarde "tls_client_auth"[28] bevatten.

[28] Voor deze optie wordt gebruik gemaakt van de meta data definities in RFC 8705: OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens (rfc-editor.org)

10. API Design Rules (ADR)

API design moet conform de normatieve API Design Rules zijn van het Digikoppeling Restful API profiel[29].

API design mag conform de API Design Rules extensies zijn van het Digikoppeling Restful API profiel[30].

[29] https://gitdocumentatie.logius.nl/publicatie/dk/restapi/#regels
[30] https://gitdocumentatie.logius.nl/publicatie/dk/restapi/#afspraken-api-design-rules-extensies

11. Bijlage A: OAuth client credentials profiel

De OAuth-profielen in dit document zijn gebaseerd op het OAuth basismodel in dit hoofdstuk. Het basismodel maakt gebruik van de OAuth 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.

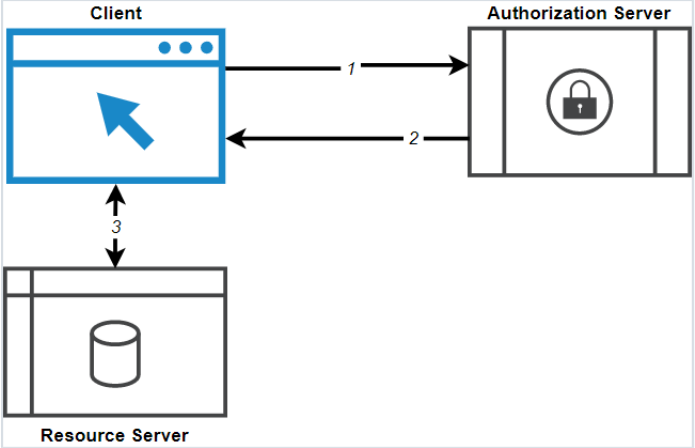
Bij de client credentials grant wordt uitgegaan van confidential clients (afgeschermde vertrouwde systemen) die de te gebruiken credentials goed kunnen beveiligen. Er kan namelijk worden gesteld dat de client op basis van de credentials impliciet geautoriseerd is en als de resource owner kan worden gezien.

Belangrijke onderdelen binnen het client credentials grant type zijn o.a. 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 referentie (i.c.m. introspection) of een zelfbeschrijvende JWT die alle info bevat waarmee de Resource server een autorisatiebeslissing kan uitvoeren. De profielen in dit document maken voor client authenticatie gebruik van mTLS / PKIo en het Access Token is op basis van een JWT.

Hieronder wordt het client credentials grant type schematisch weergegeven (zie Figuur 12).

Stap 1. Verzoek naar token endpoint
De client stuurt een verzoek naar het token endpoint van de Authorization Server. Deze valideert het verzoek op basis van de client credentials.

Stap 2. Verzoek van token endpoint
Na succesvolle authenticatie 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.



Figuur 12 - OAuth client credentials profile (bron Kennisplatform API's | Geonovum)

Stap 3. Resource-interactie
Met het access token verzoekt de client 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.

12. Bijlage B: Bronnen

[RFC2617]
HTTP Authentication: Basic and Digest Access Authentication. 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. 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. 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. T. Lodderstedt, Ed.; M. McGloin; P. Hunt. IETF. January 2013. Informational. URL: *https://datatracker.ietf.org/doc/html/rfc6819*

[RFC7515]
JSON Web Signature (JWS). M. Jones; J. Bradley; N. Sakimura. IETF. May 2015. Proposed Standard**.** URL**:** *https://www.rfc-editor.org/rfc/rfc7515*

[RFC7518]
JSON Web Algorithms (JWA), M. Jones. IETF. May 2015. Proposed Standard. URL: *https://www.rfc-editor.org/rfc/rfc7518*

[RFC7519]
JSON Web Token (JWT). *M. Jones; J. Bradley; N. Sakimura. IETF. May 2015. Proposed Standard. URL: https://www.rfc-editor.org/rfc/rfc7519*

[RFC7523]
JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants. M. Jones; B. Campbell; C. Mortimore. IETF. May 2015. Proposed Standard. URL: *https://www.rfc-editor.org/rfc/rfc7523*

[RFC7591]
OAuth 2.0 Dynamic Client Registration Protocol*. J. Richer, Ed.; M. Jones; J. Bradley; M. Machulak; P. Hunt. IETF. July 2015. Proposed Standard. URL: 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

[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

[RFC8414]

OAuth 2.0 Authorization Server Metadata. M. Jones; N. Sakimura; J. Bradley. IETF. June 2018. Proposed Standard. URL: 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/*

[RFC8707]

Resource Indicators for OAuth 2.0, Brian Campbell, John Bradley , Hannes Tschofenig, February 2020, Proposed Standard URL *https://datatracker.ietf.org/doc/html/rfc8707*