edustandaard

Edukoppeling

M2M gegevensuitwisseling binnen het onderwijs

OAuth 2.0 client credentials profiel

Edustandaard
Datum: 1 november 2026
Versie 1.0
Status: concept

Inhoudsopgave

1. Status van dit document 4
   1.1. Documenthistorie 4
2. Inleiding 5
   2.1. Doel van Edukoppeling 5
   2.2. Doelgroep 5
   2.3. Organisatorisch werkingsgebied van Edukoppeling 5
   2.4. Functioneel toepassingsgebied 5
   2.5. Positionering van Edukoppeling in het Edustandaard vijflagen model 6
   2.6. Positionering van dit document binnen Edukoppeling 6
   2.7. Notatiewijze voorschriften 7
   2.8. Leeswijzer 7
3. Toelichting OAuth 2.0 8
   3.1. Inleiding 8
   3.2. Client 9
      3.2.1. Clientregistratie 9
      3.2.2. Client lifecycle 10
      3.2.3. Client type 10
      3.2.4. Scopes 11
   3.3. Authorization Server 11
      3.3.1. Identificeren client 11
      3.3.2. Authenticeren client 11
      3.3.3. Autoriseren client 12
      3.3.4. Levering access token of foutmelding 12
   3.4. Resource Server 12
      3.4.1. Toegang tot protected resource 12
      3.4.2. API lifecycle 13
   3.5. Access token 13
   3.6. Transportbeveiliging (TLS) 13
   3.7. Technisch vertrouwensanker 14
4. Voorschriften OAuth client credentials grant 16
5. Toepassingsscenario's 24
   5.1. Probleemstelling - wat moet er met het profiel geregeld worden? 24
   5.2. Oplossing - wat biedt het profiel? 25
      5.2.1. Laag risicoprofiel 25
      5.2.2. Verhoogd risicoprofiel 25

5.2.3. Verhoogd risico met toepassing van overheidsstandaarden 25
6. Bijlage A: Begrippen 27
7. Bijlage B: Referenties 29

1. Status van dit document

Dit document is een conceptversie van het OAuth client credentials profiel (hierna OAuth-profiel) die ter vaststelling aan de architectuurraad zal worden aangeboden. De nieuwe Edukoppeling Architectuur versie 3.0 geeft meer context aan dit OAuth-profiel en wijkt sterk af van de vorige versie[1]. In de vorige versie onderkende we al RESTful API's, maar was nog sterk gericht op webservices en Digikoppeling.

Dit OAuth-profiel bevat voorschriften voor de beveiliging van API's op basis van token-based access. Deze beveiliging kan zowel toegepast worden binnen een REST-georiënteerde als een Event Driven Architecture (EDA). We introduceren een flexibel OAuth-profiel dat aansluit bij de actuele behoeften bij groeifondsprogramma's en ketens binnen het onderwijs. Bij de ontwikkeling van dit profiel is gestuurd op een balans tussen uniformiteit en flexibiliteit. Dit betekent dat een ketensamenwerking bij de implementatie van het OAuth-profiel een aantal keuzes moet maken die deels afhankelijk zijn van het risicoprofiel van de gegevensuitwisseling.

1.1. Documenthistorie

Versie	Status	Auteur	Datum	Opmerking
1.0	Concept	WG Edukoppeling	1 november 2026	• Openbare consultatieversie • Door werkgroep vastgestelde versie met laatste redactionele wijzigingen

[1] https://www.edustandaard.nl/app/uploads/2021/10/2021-02-10-Edukoppeling-Architectuur-2.0-definitief.pdf

2. Inleiding

2.1. Doel van Edukoppeling

Het doel van Edukoppeling is standaardisatie van de technische afspraken op het M2M-koppelvlak waardoor het ontwikkelen van ketensamenwerkingen[2] by design veilig en eenvoudiger wordt. De toepassing van Edukoppeling zorgt ervoor dat, op het niveau van de applicatielaag, vertrouwelijke gegevens[3] tijdens transport van de ene naar de andere organisatie niet ongeoorloofd kunnen worden ingezien of gemanipuleerd. De standaard gaat over de afhandeling van berichten (het transport) en niet over de inhoud van berichten.

2.2. Doelgroep

Dit document is bedoeld voor ICT-specialisten die betrokken zijn bij het ontwerpen en ontwikkelen van systeem-naar-systeem (M2M) koppelingen voor API's. Het gaat hier om werknemers (ontwikkelaars, architecten, projectmanagers, informatiemanagers etc.) werkzaam bij onderwijsgerelateerde organisaties, zowel in de publieke als private sector. De lezer van dit document willen wij vragen om zaken die ontbreken of onduidelijk zijn te melden bij de beheerorganisatie Edustandaard[4]. Edustandaard is een open platform waar partijen binnen het onderwijsveld bij elkaar komen om afspraken te maken. Hier vindt tevens de doorontwikkeling van de standaard plaats. Hiertoe is een werkgroep Edukoppeling[5] ingericht.

2.3. Organisatorisch werkingsgebied van Edukoppeling

Edukoppeling schrijft voor hoe onderwijsorganisaties, publieke uitvoeringsorganisaties, dienstverleners[6] en andere ketenpartners (binnen Nederland) gegevensuitwisselingen opzetten. Edukoppeling heeft als scope alle werkingsgebieden vallend onder alle onderwijssectoren. Zie de werkingsgebieden in ROSA[7]. Hieronder vallen dus alle door de overheid erkende onderwijsorganisaties binnen de sectoren po, vo, bve en ho. Daar waar behoefte is aan technische afspraken op het M2M-koppelvlak, volgens het functioneel toepassingsgebied van Edukoppeling, zijn ketensamenwerkingen binnen deze sectoren verantwoordelijk voor de toepassing ervan. De toepassing buiten het organisatorisch werkingsgebied is toegestaan.

2.4. Functioneel toepassingsgebied

Het functioneel toepassingsgebied van Edukoppeling is de geautomatiseerde uitwisseling van vertrouwelijke gegevens tussen informatiesystemen van onderwijsorganisaties en ketenpartners (onderling, met bedrijven of met de overheid).

---

[2] ROSA definieert een ketensamenwerking als een concrete invulling van een scenario, waarbij er sprake is van een vorm van gegevensuitwisseling (interactie) tussen twee of meer ketenpartners ter ondersteuning van het afhandelen een of meerdere ketenprocess(tapp)en.

[3] Ook wel het 'gesloten data'. Het gaat om gegevens waarvoor je geautoriseerd moet worden voor toegang. De gegevens zijn niet voor publiek hergebruik beschikbaar. Dit kan om verschillende redenen zijn, zie https://data.overheid.nl/gesloten-datasets

[4] https://www.edustandaard.nl/standaarden/afspraken/afspraak/edukoppeling/. Reageren kan via info@edustandaard.nl.

[5] Voor meer info over de Edukoppeling werkgroep, zie https://www.edustandaard.nl/standaard_werkgroepen/werkgroep-edukoppeling/

[6] Dienstverleners (ROSA: *Een organisatie die een dienst levert aan een organisatie of een natuurlijke persoon).*

[7] https://rosa.wikixl.nl/index.php/Werkingsgebieden

Deze uitwisseling vindt plaats in een machine-to-machine (M2M) context en verloopt via een point-to-point (P2P) verbinding tussen een confidential client[8] en een gesloten API.

Edukoppeling gaat over de afhandeling van berichten (het transport) en niet over de inhoud van berichten. Het is een functioneel technische standaard, maar zal ook aansluiting moeten vinden op kaders van andere architectuurlagen. Hoe Edukoppeling aansluit op bredere afspraken is aan ketensamenwerkingen[9] waarin de standaard als onderdeel van de afspraak of het afsprakenstelsel wordt gevat.

Edukoppeling regelt de volgende ketenfuncties: identificatie, authenticatie, autorisatie en routering, op de 'uitwisselingslaag'. Dit om de zorgen dat vertrouwelijke gegevens tijdens transport van de ene naar de andere organisatie, niet ongeoorloofd worden ingezien of gemanipuleerd.

2.5. Positionering van Edukoppeling in het Edustandaard vijflagen model

Het Edustandaard 5-lagenmodel[10] onderkent de volgende lagen:

1. Grondslagenlaag: borgt de juridische basis en beleidskaders waarbinnen gegevensuitwisseling is toegestaan;
2. Organisatorische laag: ketensamenwerking afspraken over wie welke rol heeft, welke gegevensdiensten, interfaces en interactiepatronen er zijn en welke gegevens onder welke condities uitgewisseld worden;
3. Informatielaag: semantiek, waaronder gegevensdefinities, informatiemodellen en de gebruikte identifiers voor rechtspersonen en natuurlijke personen;
4. Applicatielaag: API's en hun beveiligingsprofielen, berichtspecificaties, payload beveiliging, interactiepatronen en foutafhandeling;
5. IT-infrastructuurlaag: transportprotocollen en technische beveiligingsmechanismen zoals TLS.

Dit Edukoppeling-profiel heeft binnen het Edustandaard 5 lagen model met name betrekking op de applicatielaag (laag 4), levert een belangrijke ondersteuning aan de ketenfuncties identificatie, authenticatie, autorisatie en heeft een relatie met de IT-infrastructuurlaag (laag 5).

2.6. Positionering van dit document binnen Edukoppeling

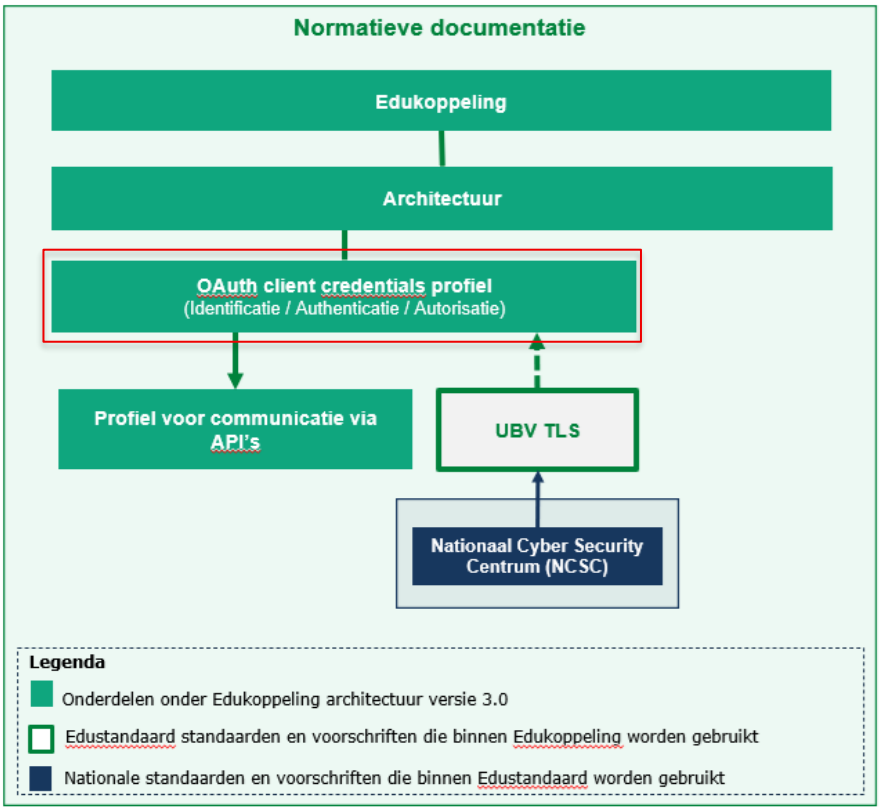
Edukoppeling bestaat uit verschillende documenten en is in beheer bij Edustandaard. Dit OAuth-profiel is één van de normatieve documenten binnen de Edukoppeling-afspraak. In Figuur 1 - Positionering van Oauth profiel binnen Edukoppeling is hiervan een schematische weergave opgenomen.

---

[8] Een confidential client is een client die draait in een omgeving waar vertrouwelijke gegevens (waaronder de credentials of cryptografisch sleutelmateriaal voor authenticatie) veilig bewaard kunnen worden (bijvoorbeeld server-side webapplicaties).

[9] Ketensamenwerkingen zijn bijvoorbeeld OKE, Edu-V, ROD ec. (zie ook: https://rosa-begrippenkader.wikixl.nl/index.php/Begrip:27a6accf-472d-4415-bc5b-1e9de17bf288#tab=Betekenis)

[10] AMIGO-methodiek-1.1.0-1.pdf en https://rosa.wikixl.nl/index.php/Interoperabiliteit_en_het_Edustandaard_lagenmodel#Opbouw_van_het_lagenmodel



Figuur 1 - Positionering van Oauth profiel binnen Edukoppeling

2.7. 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[11]. Deze gebruikt de volgende termen: "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL".

2.8. Leeswijzer

Hoofdstuk 3 bevat een toelichting op de OAuth 2.0 standaard en Edukoppeling voorschriften. In hoofdstuk 4 zijn de Edukoppeling voorschriften opgenomen. Dit zijn aanscherpingen op de onderliggende internationale open industriestandaarden rond OAuth 2.0 en heeft de client credentials grant (RFC6749) als basis. In hoofdstuk 5 worden scenario's beschreven die een aantal voorschriften in een bepaalde context plaatsen om ketensamenwerkingen te ondersteunen in de te maken keuzes.

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

3. Toelichting OAuth 2.0

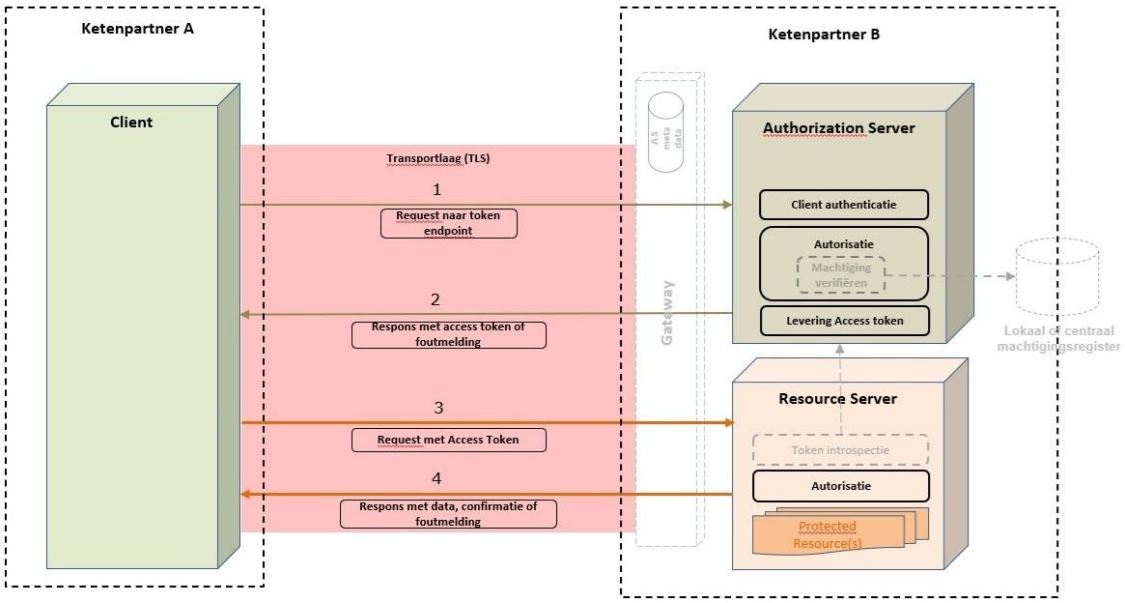
3.1. Inleiding

Dit hoofdstuk bevat een toelichting op de Edukoppeling voorschriften en hebben betrekking op de OAuth 2.0 client credentials grant. Met OAuth 2.0 client credentials wordt de mogelijkheid geboden om API's te beveiligen op de applicatielaag in een M2M context. Confidential clients krijgen alleen toegang tot de API's op basis van een valide access token dat ook de juiste rechten moet omvatten voor een succesvolle autorisatie. De grofmazige rechten worden gegeven op basis van zogenaamde scope(s) die in het access token worden opgenomen. In vorige versie van Edukoppeling boden we met mTLS de mogelijkheid om te autoriseren op het niveau van een organisatie. Met het OAuth 2.0 Access Token en de scopes claim hierin bieden we de mogelijkheid om clients (systemen) te autoriseren op het niveau van vooraf gedefinieerde functies en/of datagroepen. De autorisaties op data-niveau beschouwen we als fijnmazig.

De OAuth client credentials grant beschrijft de rol van Client, Authorization Server en Resource Server en de volgende interacties:

- Client-request naar Authorization Server token endpoint (1)[12]
- Response van Authorization Server naar client (2)
- Client-request naar Resource Server (3)[13]
- Response van Resource Server naar client (4)

De rollen worden respectievelijk in paragrafen **Fout! Verwijzingsbron niet gevonden.**, 3.3 en 3.4 verder toegelicht. Dit OAuth-profiel gaat ervan uit dat één organisatie controle heeft over de Authorization Server en Resource Server (zie ook Figuur 2).



Figuur 3 – OAuth 2.0 client credentials grant

[12] Zie Authenticeren client

[13] Zie Toegang tot protected resourceResource Server

3.2. Client

Een Client vraagt toegang tot één of meer protected resources die via een API beschikbaar worden gesteld. Binnen Edukoppeling vraagt de Client hiervoor een Access Token aan bij de Authorization Server. Bij een tokenaanvraag identificeert en authenticeert de Authorization Server de Client en beoordeelt deze of de gevraagde autorisatie, waaronder de aangevraagde scopes, mag worden verleend. Wanneer de aanvraag wordt toegestaan, verstrekt de Authorization Server een Access Token met een beperkte geldigheidsduur en de voor de interactie verleende autorisatie. De Client gebruikt dit Access Token vervolgens bij de aanroep van de API. De Resource Server valideert het token en controleert of de daarin verleende autorisatie toegang geeft tot de gevraagde protected resource. Alleen wanneer deze controles succesvol zijn, wordt de gevraagde interactie toegestaan.

3.2.1. Clientregistratie

De ketenpartner die verantwoordelijk is voor een Client zorgt ervoor dat deze voorafgaand aan de gegevensuitwisseling is geregistreerd binnen het vertrouwensdomein van de Authorization Server die Access Tokens[14] voor de betreffende API uitgeeft. De clientregistratie legt minimaal de unieke client-identiteit (client_id), de toegestane client-authenticatiemethode[15] en de daarvoor benodigde verificatiegegevens vast. Daarnaast kunnen de API's en scopes worden vastgelegd waarvoor de Client autorisatie mag aanvragen. Per clientregistratie wordt één van de binnen het Edukoppeling OAuth-profiel toegestane client-authenticatiemethoden vastgelegd. Hiermee is voor de Authorization Server eenduidig bepaald op welke wijze de betreffende Client zich bij een tokenaanvraag moet authenticeren.

Het kan zijn dat een ketensamenwerking voor toegang tot bepaalde API's het bestaan van een geldige machtiging voorschrijft. In dat geval moet de machtiging kunnen worden gerelateerd aan de onderwijsorganisatie waarvoor wordt gehandeld, de organisatie die de Client beheert en de API waarop de machtiging betrekking heeft. Via de clientregistratie kan vervolgens worden vastgesteld welke machtigingen er voor de Client bestaan.

Hoe ketenpartners clientregistratie inrichten valt buiten de scope van Edukoppeling, maar dit proces moet er wel voor zorgen dat o.a. wordt vastgelegd welke client bij welke organisatie hoort, welke authenticatiemethode is toegestaan, en welke verificatiegegevens (bijvoorbeeld sleutelmateriaal of geheim) wordt gebruikt.

Een ketensamenwerking[16] maakt binnen Edukoppeling de volgende keuzes:

1. Toegestane client-authenticatiemethoden
    a. Zie Authenticeren client
2. Scope(s)[17]
    a. De ketensamenwerking definieert de toe te passen scopes. De Authorization Server en Resource Server moeten deze ondersteunen.
3. De ketensamenwerking kiest het toe te passen technisch vertrouwensanker(s).
    a. Zie Technisch vertrouwensanker
4. Toepassing van een machtiging voor specifieke API’s

[14] Zie paragraaf Access token. Dit profiel ondersteunt het gebruik van bearer tokens.
[15] Zie voor opties paragraaf Authenticeren client
[16] Verschillende ketensamenwerkingen kunnen dus verschillende keuzes maken.
[17] Zie ook Edu-V, een API heeft specifieke scopes die binnen ketensamenwerking zijn vastgesteld.

Tijdens het registratieproces levert de client de volgende gegevens aan ten behoeve van de clientregistratie:

1. Identiteit organisatie die de client beheert (OIN)
2. Authenticatiemethode voor betreffende client (indien er sprake is van een keuze)
    a. De publieke sleutel indien private_key_jwt van toepassing is.
3. Tot welke API's de client toegang wil en met welke scope(s)
4. Namens welke onderwijsorganisaties de gegevensuitwisseling wordt uitgevoerd (voor API's waar een machtiging wordt toegepast.

De client wordt de volgende informatie geleverd:

1. Identiteit organisatie die de API beheert (OIN)
2. De identifier waar de client mee geïdentificeerd wordt door Authorization Server (client_id)
3. Bevestiging van te gebruiken API's en bevestiging van eventueel te gebruiken scopes en machtigingen
4. Bevestiging authenticatiemethode
    a. Client Password[18]
    b. private_key_jwt[19]
5. Verwijzing naar Authorization Server discovery endpoint + beleidsvoorwaarden
    a. OpenID Connect service discovery [OpenID.Discovery] endpoint

3.2.2. Client lifecycle

De gegevens van een client zijn niet altijd statisch die eenmalig worden geregistreerd. Tijdens de looptijd van een ketensamenwerking kunnen omstandigheden veranderen, waardoor de gegevens en afspraken die aan een clientregistratie gekoppeld zijn moeten worden aangepast. Voorbeelden zijn het toevoegen of intrekken van scopes of machtigingen (bevoegdheden), het gebruik van andere API's, of het overstappen op een andere client-authenticatievorm.

3.2.3. Client type

OAuth 2.0 onderscheidt twee typen clients, confidential en public clients. Deze zijn gebaseerd op hun vermogen om veilig te authenticeren bij de Authorization Server. Confidential clients beschikken over credentials en kunnen zich bij de Authorization Server authenticeren. Public clients hebben geen credentials en kunnen zich niet bij de Authorization Server authenticeren. Dit Edukoppeling OAuth-profiel is alleen van toepassing op confidential clients.

Een client wordt binnen dit Edukoppeling OAuth-profiel geacht het 'Client Secret' veilig te kunnen beheren en wordt vertrouwd op basis van het authenticatiemechanisme dat hiermee samenhangt. Bij de aanvraag van een access token mogen zij geen refresh token aanvragen of ontvangen.

[18] Zie RFC6749 'Client Password' https://www.rfc-editor.org/rfc/rfc6749.html#section-2.3.1 en betreft de HTTP Basic authentication scheme (RFC2617). In RFC 8414 - OAuth 2.0 Authorization Server Metadata wordt dit ook wel "client_secret_basic" genoemd

[19] Zie OpenID Client Authentication (Final: OpenID Connect Core 1.0 incorporating errata set 2)

3.2.4. Scopes

Scopes worden gebruikt om een client grofmazig[20] te kunnen autoriseren. Het zijn de bevoegdheden die de client bij registratie of op een later moment zijn toegekend. Scopes zijn autorisaties die de Authorization Server opneemt in het access token en de Resource Server gebruikt om de client toegang te geven tot bepaalde resources. Scopes zijn semantisch gekoppeld aan functionaliteit en/of datasoorten en worden binnen de Resource Server geïmplementeerd als autorisatieregels. Een ketensamenwerking definieert welke scopes voor een bepaalde resource van toepassing zijn.

3.3. Authorization Server

De Authorization Server is het centrale component binnen OAuth. Het bepaalt welke client onder welke voorwaarden toegang krijgt tot een API. De Authorization Server is verantwoordelijk voor het volgende:

1. identificeren client: uitgifte client_id;
2. authenticatie: op basis van minstens één van de methoden die de ketensamenwerking toestaat;
3. autorisatie: controleren of de door de client gevraagde scopes overeenkomen of een subset zijn van de scopes die zijn geregistreerd;
4. levering access token aan client.

3.3.1. Identificeren client

De Authorization Server koppelt bij registratie het OIN van de API-afnemer aan een client_id voor de betreffende client. Met de client_id kan de Authorization Server een client identificeren bij requests op het token endpoint. Het client_id wordt gebruikt om overige informatie aan de cliënt te koppelen, zoals authenticatiemethode en scopes.

3.3.2. Authenticeren client

De Authorization Server (token endpoint) is verantwoordelijk voor het authenticeren van de client. De vorm van clientauthenticatie komt overeen met de keuze die de Authorization Server aanbiedt (AS metadata) en is in lijn met de keuze van de ketensamenwerking.

Binnen dit Edukoppeling OAuth-profiel zijn twee client-authenticatiemethoden mogelijk, dit zijn basic authentication en de OIDC private_key_jwt methode. Deze worden niet als gelijkwaardig beschouwd, omdat ze verschillen in sterkte, beheerlast en het risico bij misbruik[21]. Het kan dus zijn dat Authorization Servers verschillende client-authenticatiemethoden ondersteunen en dat ook de eigenschappen van access tokens (bijvoorbeeld geldigheidsduur of claims) per ketensamenwerking kunnen verschillen. We accepteren deze variatie, omdat elke ketensamenwerking, en iedere ketenpartner daarbinnen, zelf verantwoordelijk is voor het maken van een passende keuze op basis van het risicoprofiel. Tegelijk streven we naar uniformiteit en interoperabiliteit door deze keuzes niet ad hoc te laten ontstaan, maar ze expliciet vast te leggen binnen dit Edukoppeling OAuth-profiel. Welke vormen van authenticatie en tokenuitgifte zijn toegestaan en hoe ze worden toegepast, wordt daarom centraal bepaald. De Authorization Server is daarmee het

[20] Zie Probleemstelling - wat moet er met het profiel geregeld worden?
[21] Zie Laag en verhoogd risicoprofiel in

Toepassingsscenario's

eenduidige referentiepunt: daar ligt vast welke methoden ondersteund worden, welke voorwaarden gelden en welke instellingen bij welke client of ketensamenwerking horen.

Nadat de client is geauthentiseerd kan de autorisatiebeslissing genomen worden of er een access token geleverd mag worden.

3.3.3. Autoriseren client

De Authorization Server controleert na authenticatie van de client of voor het token request de juist bevoegdheden bestaan. Dit betreft over het algemeen de controle van de geregistreerde scope(s), maar er kunnen ook machtigingen gecontroleerd worden. Een client kan bij het token endpoint scopes aanvragen die buiten zijn geregistreerde bevoegdheden vallen. Het token endpoint moet daarom bij elke token aanvraag de scope toetsen aan de scopes die voor die client zijn toegestaan. Scopes die niet zijn toegestaan worden geweigerd (of niet toegekend), zodat een client alleen een access token ontvangt met scopes waarvoor hij bevoegd is. Hetzelfde geldt voor eventueel toe te passen machtigingen.

De set met toegestane scopes (en machtigingen) is niet per definitie statisch. Na registratie kunnen scopes voor een client worden toegevoegd, gewijzigd of ingetrokken[22].

3.3.4. Levering access token of foutmelding

Na een positief autorisatiebesluit wordt een access token[23] gecreëerd. In het JWT access token worden claims als ‘scope’, ‘aud’ en ‘exp’ opgenomen. De JWT, als zelfbeschrijvend token[24], of een referentie (opaque token) wordt aan de client geleverd. Bij een negatief autorisatiebesluit wordt de client een foutmelding[25] conform RFC6749 geleverd.

3.4. Resource Server

De Resource Server is de OAuth rol die functionaliteit ondersteunt om vast te stellen dat de API-aanroep via een veilige verbinding plaatsvindt en dat het meegeleverde access token geldig is en voldoende rechten bevat. Het API-request wordt hierna pas inhoudelijk verwerkt.

3.4.1. Toegang tot protected resource

De Resource Server is verantwoordelijk voor het afdwingen van toegang op basis van de door de Authorization Server uitgegeven access tokens. Elk API-request moet een access token in de HTTP Authorization-header bevatten. Het access token kan een opaque of een JWT-token zijn. Bij een opaque token kan de Resource Server de geldigheid niet zelfstandig vaststellen. In dat geval gebruikt de Resource Server een token introspection endpoint van de Authorization Server om het token te laten verifiëren door de Authorization Server. Bij toepassing van een JWT access token controleert de Resource Server het token lokaal door de handtekening te verifiëren met de publieke sleutel van de Authorization Server. Daarnaast moet o.a. gecontroleerd worden of het token de juiste aud-claim bevat en het token nog niet verlopen is. Als het token ongeldig is moet het API-request worden afgewezen en wordt er een foutmelding gegeven conform RFC6750.

Er wordt aangenomen dat zowel de Authorization Server als de Resource Server onder de controle zijn van dezelfde organisatie. De Authorization Server kent[26] de Resource Server(s)

[22] Zie Client lifecycle
[23] https://www.rfc-editor.org/rfc/rfc6749#section-5.1
[24] Zie Access token
[25] https://www.rfc-editor.org/rfc/rfc6749#section-5.2
[26] Authorization Server kan de Resource Server identificeren (aud claim in access token).

en welke scopes er voor een Resource Server gelden. De Resource Server interpreteert de scope(s) in het access token, mogelijk op het niveau van endpoints.

Als het access token authorization_details bevatten dan kunnen de edu-from en edu-to claims geverifieerd worden bij het machtigingsregister. De Resource Server kan er ook op vertrouwen dat de Authorization Server de verificatie heeft uitgevoerd.

3.4.2. API lifecycle

Het is belangrijk dat organisaties tijdig geïnformeerd worden over (lifecycle) informatie rond de API. Zij moeten zich op eventuele wijzigingen kunnen voorbereiden.

3.5. Access token

Access tokens worden gebruikt om toegang te krijgen tot beveiligde resources. Een access token vertegenwoordigt de bevoegdheden die zijn toegekend aan een client. Dit Edukoppeling OAuth-profiel ondersteunt twee vormen van access tokens, een JWT (zelfbeschrijvend) en een opaque token (referentie).

JWT

Een JWT access token is een zelfbeschrijvend token (self-contained) met alle claims en metadata in de payload. Het wordt ondertekend door de Authorization Server. Omdat deze tokens (over het algemeen) alle informatie bevatten die de Resource Server nodig heeft is introspectie bij de Authorization Server niet nodig. Het intrekken van het access token is dus ook niet zinvol als de Resource Server na ontvangst geen controle bij Authorization Server uitvoert. Hiermee moet rekening worden bij het bepalen van de levensduur van het access token.

Opaque tokens en tokenintrospectie

Het gebruik van opaque tokens zorgt ervoor dat de client het access token niet kan inzien (zonder dat het access token versleuteld moet worden). Via het opaque token kan de Resource Server via een introspection-endpoint het bijbehorende access token controleren op geldigheid, scopes, cliëntrechten en attributen. Ook kan de Resource Server zo vaststellen dat in de tussentijd dat het opaque token is uitgegeven het access token niet is ingetrokken. Het nadeel van tokenintrospectie is dat het extra latency oplevert.

De access tokens die dit Edukoppeling Oauth-profiel ondersteunt zijn zogenaamde Bearer tokens[27]. Ze zijn eenvoudig te implementeren door brede ondersteuning. Een bearer token kan door iedereen die het token bezit worden gebruikt (bezit=toegang). Bearer-tokens kunnen alleen op TLS-beschermde kanalen worden gebruikt omdat het lekken ervan moet worden voorkomen. Een bearer token kan een JWT zijn (zelfbeschrijvend) of opaque (i.c.m. introspection). Toepassen als transportbeveiliging (TLS) voldoende bescherming biedt en er geen (of zeer beperkt) risico is dat het token binnen de levensduur gelekt kan worden.

3.6. Transportbeveiliging (TLS)

De OAuth 2.0 standaard vereist dat de Authorization Server en de Resource Server endpoints met Transport Layer Security (TLS) beveiligd moeten zijn. TLS zorgt ervoor dat

[27] Het profiel ondersteunt (nog) niet de proof-of-possession-tokens (zie RFC8705 en OAuth 2.0 Demonstrating Proof of Possession (DPoP) RFC9449).).

gegevens onderweg niet onderschept of gewijzigd kunnen worden. TLS zorgt voor integriteit, vertrouwelijkheid en authenticatie gedurende TLS sessie. De server authenticeert zich met behulp van een servercertificaat zodat de cliënt in staat is te controleren of de HTTPS-verbinding met de juiste server tot stand is gebracht. Hierbij wordt een PKI-certificaat toegepast. Deze certificaten, uitgegeven door vertrouwde autoriteiten (CA's[28]), verifiëren de identiteit van de server. De (root-)CA is het technische vertrouwensanker waar de ketensamenwerking voor kiest en een belangrijke basis voor het vertrouwen in de keten. Endpoints worden beveiligd op basis van het UBV TLS Basisprofiel[29]. De UBV TLS werkgroep is verantwoordelijk voor het volgen van de ontwikkelingen[30] rond TLS en het beheer van de standaard.

mTLS
Bij mutual TLS (mTLS) authenticeren zowel de client als de server zich tijdens het opzetten van de TLS-verbinding met een certificaat. Beide partijen valideren het aangeboden certificaat en de bijbehorende certificaatketen[31] tegen de door hen vertrouwde certificaatautoriteiten. De client en server bewijzen daarbij dat zij beschikken over de private sleutel die bij hun certificaat hoort. Wanneer de TLS-handshake succesvol is afgerond, ontstaat een wederzijds geauthentiseerde en versleutelde TLS-verbinding. HTTP-verkeer dat over deze verbinding plaatsvindt, wordt als HTTPS uitgevoerd. Bij de toepassing van mTLS kunnen PKIoverheid certificaten toegepast worden.

mTLS is dus een manier om clientauthenticatie in te richten. Deze versie van Edukoppeling ondersteunt echter niet clientauthenticatie op basis van mTLS. In de volgende paragraaf wordt toegelicht welke vormen van clientauthenticatie er wel ondersteund worden. mTLS mag echter wel bij de interactie tussen Client en Resource Server toegepast worden.

3.7. Technisch vertrouwensanker

Het technisch vertrouwensmodel[32] beschrijft hoe deelnemende ketenpartners elkaar en hun systemen identificeren en authenticeren. Het beschrijft hoe de identiteit van een Client, de betrouwbaarheid van de authenticatiemethode en de binding tussen de twee wordt geraliseerd. De Clientregistratie legt de relatie vast tussen de client_id, de organisatie die verantwoordelijk is voor de Client, de toegestane authenticatiemethode en de bijbehorende verificatiegegevens. De Edukoppeling Architectuur beperkt dit zich tot de volgende technische vertrouwensankers:

1. een vooraf geregistreerd gedeeld geheim;
2. een vooraf geregistreerde publieke sleutel;
3. een vertrouwde certificaatautoriteit (CA) binnen een Public Key Infrastructure (PKI).

Vooraf geregistreerd gedeeld geheim
Bij clientauthenticatie op basis van HTTP Basic authentication wordt er vertrouwt op een vooraf geregistreerde combinatie van client_id en client_secret. Binnen OAuth 2.0 betreft dit

[28] Zie Architectuur Technisch vertrouwensmodel
[29] Meer informatie via Werkgroep Uniforme Beveiligingsvoorschriften: https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/
[30] Bijvoorbeeld NCSC (Home | Nationaal Cyber Security Centrum)
[31] Indien van toepassing, zoals bij PKIoverheid certificaten
[32] Zie Edukoppeling Architectuur - Technisch vertrouwensmodel en vertrouwensankers

de Client Password[33] clientauthenticatiemethode. Het vertrouwen wordt dus primair geborgd door de betrouwbaarheid van het clientregistratieproces en het veilig beheer van het gedeelde geheim. Dit is eenvoudig te implementeren, maar kent een lager betrouwbaarheidsniveau: beide ketenpartners kennen hetzelfde geheim en bij uitlekken kan de client worden nagebootst. Deze optie past vooral bij een laag risicoprofiel en vraagt om strikte secret-rotatie, opslag en intrekking.

Vooraf geregistreerde publieke sleutel De publieke sleutel wordt bij clientregistratie vastgelegd. De Authorization Server gebruikt deze later om de ondertekende private_key_jwt te verifiëren. Hiermee heeft de cliënt bewezen dat deze over de bijbehorende private sleutel beschikt. Het vertrouwen in de publieke sleutel wordt hierbij niet afgeleid uit een certificaatketen of een vertrouwde certificaatautoriteit (CA), maar uit het proces waarmee de sleutel vooraf is gekoppeld en geregistreerd. De registratie vormt daarmee de basis voor het vertrouwen; de publieke sleutel is het cryptografische materiaal waarmee het bewijs wordt geverifieerd. Dit is sterker dan een gedeeld geheim, ondersteunt sleutelrotatie beter en voorkomt dat een shared secret bij beide ketenpartners aanwezig is.

Vertrouwde certificaatautoriteit (CA) binnen een Public Key Infrastructure
Ook hierbij zal de publieke sleutel bij clientregistratie worden vastgelegd en gebruikt de Authorization Server deze later om de ondertekende private_key_jwt te verifiëren. De Authorization Server moet de publieke sleutel die een client wil registreren (initiële registratie of rotatie) valideren tegen de binnen de ketensamenwerking geaccepteerde CA's. De sleutelinformatie wordt in de vorm van een jwk uitgewisseld. Bij toepassing van PKIoverheid[34] als CA wordt naast bezit van de private key ook de certificaatketen gevalideerd tegen een vertrouwde PKIoverheid hiërarchie. Het vertrouwen is dan op twee niveaus: door registratie van de client en publieke sleutel én door validatie van de certificaathiërarchie. Dit biedt een hoger betrouwbaarheidsniveau. Bij toepassing van PKIoverheid geldt verder dat de organisatie identiteit op basis van de eisen van PKIoverheid is vastgesteld. Deze variant past bij hogere risico's, formele ketens en gegevensuitwisseling met persoonsgegevens of wettelijke context.

[33] Zie RFC6749 'Client Password' https://www.rfc-editor.org/rfc/rfc6749.html#section-2.3.1 en betreft de HTTP Basic authentication scheme (RFC2617). In RFC 8414 - OAuth 2.0 Authorization Server Metadata wordt dit ook wel "client_secret_basic" genoemd
[34] PKIoverheid: https://www.pkioverheid.nl/

4. Voorschriften OAuth client credentials grant

Dit hoofdstuk bevat aanvullende voorschriften op de internationale open standaard OAuth 2.0[35]. Het vormt de basis voor een Edukoppeling OAuth client credentials profiel voor API's (hierna Edukoppeling OAuth-profiel). Het fundament van de OAuth 2.0 standaard zijn RFC6749 en RFC6750. Een overzicht van de IETF RFC's[36] die relevant zijn voor dit Edukoppeling OAuth2.0 profiel is opgenomen in '

[35] OAuth 2.0 — OAuth: https://oauth.net/2/
[36] IETF | RFCs: https://www.ietf.org/process/rfcs/

Bijlage B: Referenties'. De voorschriften in dit hoofdstuk zijn aanscherpingen op deze bovenliggende RFC's. Het geheel aan voorschriften biedt een Edukoppeling OAuth-profiel die kan worden toegepast wanneer een confidential client via een M2M/P2P verbinding interacteert met een gesloten API[37], waarbij de client vooraf bevoegdheden kunnen zijn toegekend op basis van scopes. Het Edukoppeling OAuth-profiel biedt op een aantal punten keuzemogelijkheden. Hiermee beogen we een profiel dat flexibel is en een ketensamenwerking keuzes biedt om voor ketenpartners niet onoverkomelijke drempels op te werpen. Er wordt voorzien in een minder betrouwbare vorm van clientauthenticatie voor scenario's met laag risicoprofiel. Er is een meer betrouwbare vorm voor scenario's met een verhoogd risicoprofiel, waarbij ook de optie wordt geboden om mandaatinfo door te geven en/of PKIoverheid certificaten toe te passen bij clientauthenticatie. Het is uiteindelijk aan een ketensamenwerking om te bepalen welke opties van toepassing zijn. Organisaties die participeren in deze ketensamenwerking passen toe wat een ketensamenwerking voorschrijft. Als zij in meerdere ketensamenwerkingen actief zijn dan kunnen zij dus te maken krijgen met verschillende beveiligingsmaatregelen. Verder gaat dit profiel ervan uit dat de Authorization Server en Resource Server elkaar vertrouwen en onder de controle staan van één organisatie. Interacties tussen deze twee rollen is geen onderdeel van dit Edukoppeling OAuth-profiel.

1. MUST: Ketenpartners die over een OIN kunnen beschikken worden binnen een ketensamenwerking op basis van dit OIN[38] geïdentificeerd.

2. MUST: Organisaties bepalen welke client-authenticatiemethode(n) ondersteund worden.
    a. MUST: De client-authenticatiemethode(n) die een Authorization Server moet ondersteunen wordt/worden door de ketensamenwerking bepaald.

3. MUST: Dit Edukoppeling OAuth-profiel is van toepassing bij de toepassing van confidential clients.

4. MUST: Dit Edukoppeling OAuth-profiel vereist het gebruik van Transport Layer Security (TLS) conform UBV TLS Basisprofiel[39] voor alle endpoints.

---

[37] Zie Functioneel toepassingsgebied
[38] Zie Digikoppeling OIN nummersystematiek https://logius-standaarden.github.io/Digikoppeling-Identificatie-en-Authenticatie/#identiteit-en-nummer
[39] Zie Transportbeveiliging (TLS)Access token
Access tokens worden gebruikt om toegang te krijgen tot beveiligde resources. Een access token vertegenwoordigt de bevoegdheden die zijn toegekend aan een client. Dit Edukoppeling OAuth-profiel ondersteunt twee vormen van access tokens, een JWT (zelfbeschrijvend) en een opaque token (referentie).

JWT
Een JWT access token is een zelfbeschrijvend token (self-contained) met alle claims en metadata in de payload. Het wordt ondertekend door de Authorization Server. Omdat deze tokens (over het algemeen) alle informatie bevatten die de Resource Server nodig heeft is introspectie bij de Authorization Server niet nodig. Het intrekken van het access token is dus ook niet zinvol als de Resource Server na ontvangst geen controle bij Authorization Server

uitvoert. Hiermee moet rekening worden bij het bepalen van de levensduur van het access token.

Opaque tokens en tokenintrospectie
Het gebruik van opaque tokens zorgt ervoor dat de client het access token niet kan inzien (zonder dat het access token versleuteld moet worden). Via het opaque token kan de Resource Server via een introspection-endpoint het bijbehorende access token controleren op geldigheid, scopes, cliëntrechten en attributen. Ook kan de Resource Server zo vaststellen dat in de tussentijd dat het opaque token is uitgegeven het access token niet is ingetrokken. Het nadeel van tokenintrospectie is dat het extra latency oplevert.

De access tokens die dit Edukoppeling Oauth-profiel ondersteunt zijn zogenaamde Bearer tokens. Ze zijn eenvoudig te implementeren door brede ondersteuning. Een bearer token kan door iedereen die het token bezit worden gebruikt (bezit=toegang). Bearer-tokens kunnen alleen op TLS-beschermde kanalen worden gebruikt omdat het lekken ervan moet worden voorkomen. Een bearer token kan een JWT zijn (zelfbeschrijvend) of opaque (i.c.m. introspection). Toepassen als transportbeveiliging (TLS) voldoende bescherming biedt en er geen (of zeer beperkt) risico is dat het token binnen de levensduur gelekt kan worden.
Transportbeveiliging (TLS)
De OAuth 2.0 standaard vereist dat de Authorization Server en de Resource Server endpoints met Transport Layer Security (TLS) beveiligd moeten zijn. TLS zorgt ervoor dat gegevens onderweg niet onderschept of gewijzigd kunnen worden. TLS zorgt voor integriteit, vertrouwelijkheid en authenticatie gedurende TLS sessie. De server authenticeert zich met behulp van een servercertificaat zodat de cliënt in staat is te controleren of de HTTPS-verbinding met de juiste server tot stand is gebracht. Hierbij wordt een PKI-certificaat toegepast. Deze certificaten, uitgegeven door vertrouwde autoriteiten (CA's), verifiëren de identiteit van de server. De (root-)CA is het technische vertrouwensanker waar de ketensamenwerking voor kiest en een belangrijke basis voor het vertrouwen in de keten. Endpoints worden beveiligd op basis van het UBV TLS Basisprofiel. De UBV TLS werkgroep is verantwoordelijk voor het volgen van de ontwikkelingen rond TLS en het beheer van de standaard.

mTLS
Bij mutual TLS (mTLS) authenticeren zowel de client als de server zich tijdens het opzetten van de TLS-verbinding met een certificaat. Beide partijen valideren het aangeboden certificaat en de bijbehorende certificaatketen tegen de door hen vertrouwde certificaatautoriteiten. De client en server bewijzen daarbij dat zij beschikken over de private sleutel die bij hun certificaat hoort. Wanneer de TLS-handshake succesvol is afgerond, ontstaat een wederzijds geauthentiseerde en versleutelde TLS-verbinding. HTTP-verkeer dat over deze verbinding plaatsvindt, wordt als HTTPS uitgevoerd. Bij de toepassing van mTLS kunnen PKIoverheid certificaten toegepast worden.

mTLS is dus een manier om clientauthenticatie in te richten. Deze versie van Edukoppeling ondersteunt echter niet clientauthenticatie op basis van mTLS. In de volgende paragraaf wordt toegelicht welke vormen van clientauthenticatie er wel ondersteund worden. mTLS mag echter wel bij de interactie tussen Client en Resource Server toegepast worden.

   a. MAY: Als technisch vertrouwensanker kunnen de PKI-certificaten van internationaal geaccepteerde public root CA's toegepast worden.
   b. MAY: Bij de interactie tussen Client en Resource Server mag mTLS toegepast worden.
      i. MAY: Bij toepassing van mTLS bij deze interactie kunnen PKIoverheid certificaten als technisch vertrouwensanker toegepast worden.

5. WOULD: Dit Edukoppeling OAuth-profiel definieert geen transportbeveiliging voor de payload anders dan TLS.
   a. Versleuteling van de payload is alleen noodzakelijk in scenario's waar (niet vertrouwde) tussenliggende componenten voor datalekken kunnen zorgen. Versleutelen is rekenintensief en het maakt het bovendien moeilijker voor beveiligingsmechanismen, zoals API-gateways, de payload te valideren en transformeren (indien nodig). Ondertekening van gegevens is alleen nodig indien er een risico is dat de integriteit van de gegevens aangetast kan worden of als onweerlegbare overdracht vereist wordt. Indien versleuteling of ondertekening van de payload toch wenselijk is kan een ketensamenwerking gebruik maken van payload-beveiliging zoals gedefinieerd in het Digikoppeling REST API profiel[40].

6. MUST: Dit Edukoppeling OAuth-profiel vereist de toepassing van de client credentials grant zoals gedefinieerd in RFC6749[41].
   a. Dit Edukoppeling OAuth-profiel gaat conform RFC6749 uit van de volgende rollen:
      i. Client
      ii. Authorization Server
      iii. Resource Server

7. MUST: Clients moeten zich authenticeren bij het token endpoint van de Authorization Server. De ketensamenwerking waarin de client participeert bepaalt welk van de onderstaande client-authenticatiemethoden een Authorization Server moet ondersteunen.
   a. MAY(standaard): Clients kunnen zich authenticeren op basis van Client Password[42]:
      i. MUST: Een wachtwoord moet een entropy van minimaal 256 bits (32 bytes) hebben.
      ii. MUST: Een wachtwoord moet veilig worden opgeslagen en mag niet in broncode of logs voorkomen.
      iii. SHOULD: Er zou binnen een ketensamenwerking een rotatiecyclus gedefinieerd moeten worden (bijvoorbeeld minstens elke 90 dagen), bij incidenten wordt er direct actie genomen.

Technisch vertrouwensanker
en https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/

[40] https://gitdocumentatie.logius.nl/publicatie/dk/restapi/3.0.1/#signing-encryptie-in-http-rest-context

[41] RFC6749 https://datatracker.ietf.org/doc/html/rfc6749#section-1.3.4

[42] Zie RFC6749 'Client Password' https://www.rfc-editor.org/rfc/rfc6749.html#section-2.3.1 en betreft de HTTP Basic authentication scheme (RFC2617). In RFC 8414 - OAuth 2.0 Authorization Server Metadata wordt dit ook wel "client_secret_basic" genoemd

iv. SHOULD: Authorization Server zou tijdelijk het gebruik van twee wachtwoorden moeten kunnen ondersteunen (t.b.v. rollover).

b. MAY(verhoogd risicoprofiel): Clients kunnen zich authenticeren op basis van de private_key_jwt methode (RFC7521, RFC 7523, OpenID)[43].

i. MUST: Clients die private_key_jwt toepassen MOETEN beschikken over een asymmetrisch sleutelpaar waarvan de publieke sleutel betrouwbaar is gekoppeld aan de geregistreerde Client. De ketensamenwerking bepaalt welk technisch vertrouwensmodel[44] hiervoor wordt toegepast.

ii. MUST: Clients moeten de private sleutel gebruiken voor het ondertekenen van de JWT.

- MUST: De JWT-clientassertion MOET worden ondertekend met een asymmetrisch ondertekenalgoritme conform RFC7518. Clients en Authorization Servers MOETEN RS256 (Rivest, Shamir, and Adleman (RSA) signature algorithm with a 256-bit hash) ondersteunen; Eventuele andere binnen een ketensamenwerking geaccepteerde asymmetrische ondertekenalgoritmen MOGEN worden toegepast;

iii. MUST: De organisatie die verantwoordelijk is voor de Client MOET ervoor zorgen dat de Authorization Server beschikt over actuele en betrouwbare verificatiegegevens van de Client. Bij registratie, vervanging of intrekking van een publieke sleutel MOET de koppeling tussen de sleutel en de geregistreerde client_id worden bijgewerkt.

iv. MAY: Een client kan in de JWT-header de jwk parameter opnemen zolang sleutelinformatie al bij de Authorization Server is geregistreerd (de jwk sleutel(s) is al gepind aan de betreffende client tijdens registratie). De jwk mag ook het vertrouwensanker (CA-hiërarchie) bevatten, maar ook deze mag niet afwijken van de reeds geregistreerde en door ketensamenwerking geaccepteerde vertrouwensanker(s). De Authorization Server mag de in de JWT opgenomen publieke sleutel (en hiërarchie) nooit vertrouwen enkel omdat de JWT-handtekening hiermee succesvol kan worden gevalideerd.

v. SHOULD: Als er meerdere sleutels in de JWK zitten dan MOET in de JWT de kid parameter meegeven worden

vi. MUST: JWT MOET een iss-claim bevatten (clientid van client die het token creëert)

vii. MUST: JWT MOET een sub-claim bevatten (clientid van client die het token creëert)

viii. MUST: JWT MOET een iat-claim bevatten met de tijd waarop het token was gecreëerd

ix. MUST: JWT MOET een exp-claim bevatten, na deze tijd mag het token niet geaccepteerd worden door de AS

x. MUST: JWT MOET een jti-claim bevatten en is uniek per request naar het token endpoint.

[43] Client Authentication (Final: OpenID Connect Core 1.0 incorporating errata set 2)
[44] Zie Technisch vertrouwensanker

xi. MUST: JWT MOET een aud-claim bevatten met de waarde van de ‘issuer identifier’[45] (de identifier van de Authorization Server[46]).
xii. MUST: de ondertekende JWT moet worden opgenomen in de ‘client_assertion‘ form-parameter in het request naar het token endpoint conform RFC6749

8. MUST: De Authorization Server moet het volgende controleren bij clientauthenticatie:
   a. Bij Client Password:
      i. op basis van de aangeboden client_id een geregistreerde, actieve Client en bijbehorende client_secret bepalen; hierbij MOET worden vastgesteld dat de Client Password voor deze Client als authenticatiemethode is toegestaan.
   b. Bij private_key_jwt methode:
      i. op basis van de aangeboden client_id en eventueel de JWT-header een geregistreerde, actieve Client en de bijbehorende vertrouwde publieke verificatiesleutel bepalen; hierbij MOET worden vastgesteld dat private_key_jwt voor deze Client als authenticatiemethode is toegestaan;
      ii. indien van toepassing de geldigheid van de certificaatketen (vertrouwensanker[47]) verifiëren;
      iii. bij toepassing van certificaatgebaseerd vertrouwen de intrekkingsstatus van de relevante certificaten controleren conform het PKI-beleid[48]; certificaten waarvan de geldigheid of intrekkingsstatus niet voldoende kan worden vastgesteld, MOGEN NIET voor clientauthenticatie worden geaccepteerd.
      iv. de digitale handtekening verifiëren met de vertrouwde publieke verificatiesleutel; alleen de voor de Client toegestane en door de ketensamenwerking geaccepteerde ondertekenalgoritmen[49] mogen worden gebruikt;
      v. vaststellen of de claims ‘iss’ en ‘sub’ overeenkomen met de geregistreerde client_id;
      vi. vaststellen of de claim ‘aud’ overeenkomt met de voor de Authorization Server toegestane audience[50] en dat de assertion binnen de geldigheidsperiode valt (claims ‘exp’, ‘iat’ en, indien aanwezig ‘nbf’);
      vii. controleren of de claim ‘jti’ aanwezig is en uniek is en dat de JWT-clientassertion niet eerder is geaccepteerd binnen de geldigheidsperiode;

9. MUST: Authorization Servers MOETEN controleren of de client over de gevraagde scopes beschikt.

---

[45] OpenID Foundation adviseerd om bij private_key_jwt de issuer identifier (AS Metadata spec RFC8414) als aud te gebruiken als mitigatie tegen misbruik. Zie 2025-01-22-private_key_jwt-aud-issues.pptx en draft-ietf-oauth-rfc7523bis-06 - Updates to OAuth 2.0 JSON Web Token (JWT) Client Authentication and Assertion-Based Authorization Grants. Zie besluit 2026-03-30 Verslag- Edustandaard Werkgroep Edukoppeling.docx
[46] Bijvoorbeeld de token-endpoint-URL van de Authorization Server
[47] Zie Technisch vertrouwensanker
[48] Bijvoorbeeld Certificate Revocation Lists (CRL) of het Online Certificate Status Protocol (OCSP),
[49] De JWT-clientassertion MOET worden ondertekend met een asymmetrisch ondertekenalgoritme conform RFC7518. Clients en Authorization Servers MOETEN RS256 (Rivest, Shamir, and Adleman (RSA) signature algorithm with a 256-bit hash) ondersteunen; Eventuele andere binnen een ketensamenwerking geaccepteerde asymmetrische ondertekenalgoritmen MOGEN worden toegepast
[50] Bijvoorbeeld token-endpoint-URL van de Authorization Server

   a. MAY: De ketensamenwerking bepaalt welke scopes de client in de betreffende rol kan gebruiken.
   b. SHOULD: De bevoegdheden die met access tokens samenhangen moeten beperkt worden tot wat echt nodig is (least privilege).

10. MUST: Authorization Servers MOETEN clients na een positieve authenticatie en autorisatiebesluit een Access Token[51] leveren van het type 'bearer'[52]. Er mag geen refresh-token[53] worden geleverd. Bij een negatief autorisatiebesluit wordt een foutmelding[54] conform RFC6749 geleverd.

11. MUST: Clients MOETEN bij het aanroepen van een beveiligde API het Access Token aanbieden via de HTTP Authorization request header[55], met gebruikmaking van het authenticatieschema Bearer, zoals gedefinieerd in RFC 6750[56]. Andere methoden voor het aanbieden van Access Tokens zijn niet toegestaan

12. MUST: Resource Servers MOETEN 'bearer'-tokens kunnen accepteren die via de HTTP Authorization request header met het authenticatieschema Bearer worden aangeboden, conform RFC 6750.

13. MUST: Resource Servers MOETEN de geldigheid van het Access Token controleren. Bij gebruik van een ondertekend JWT Access Token MOETEN zij de handtekening verifiëren met een vertrouwde publieke sleutel van de Authorization Server en minimaal de claims iss, aud en exp controleren. Bij gebruik van opaque Access Tokens MOETEN zij de geldigheid en relevante autorisatiegegevens vaststellen, bijvoorbeeld via token introspection bij Authorization Server conform RFC7662.

14. MUST: Resource Servers MOETEN voor iedere beveiligde protected resource controleren of de aan de Client verleende autorisatie voldoende is voor de gevraagde functionaliteit en de betreffende gegevens op basis van het Access Token.

15. SHOULD: De Resource Server zou conform RFC6750[57] een foutmelding moeten geven bij een invalide clientrequest.
    a. SHOULD: Resource Servers zouden een access tokens in een form-parameter of query-parameter niet moeten accepteren. Indien dit wel het geval is moet dit gezien worden als het ontbreken van het access token en zou er een foutmelding moeten worden gegeven conform RFC6750[58]

---

[51] https://www.rfc-editor.org/rfc/rfc6749#section-5.1
[52] Er wordt een bearer-token geleverd zoals gedefinieerd in RFC6750, zie https://www.rfc-editor.org/info/rfc6749/#section-7.1
[53] Zie RFC6749 (https://www.rfc-editor.org/info/rfc6749/#section-4.4.3
[54] https://www.rfc-editor.org/rfc/rfc6749#section-5.2
[55] Dit profiel verplicht het gebruik van de HTTP Authorization request header voor het aanbieden van Access Tokens aan Resource Servers, overeenkomstig RFC 6750, sectie 2.1. We sluiten aan op het OpenID iGov profiel https://openid.net/specs/openid-igov-oauth2-1_0.html#section-2.7.1. Het iGoV OAuth profiel heeft dit als "SHOULD" omdat ook andere (form-parameters) opties toegestaan worden. Edukoppeling staat geen alternatieve overdrachtsmethoden via URI-queryparameters of HTTP-requestbody's toe.
[56] Sectie 2.1: Authorization Request Header Field.
[57] https://datatracker.ietf.org/doc/html/rfc6750#section-3.1
[58] https://datatracker.ietf.org/doc/html/rfc6750#section-3.1

b. MAY: Resource Servers kunnen een scope-parameter in de foutmelding (RFC6750 'insufficient_scope') opnemen indien die nodig zijn in het access token om toegang tot de resource te verkrijgen.

16. SHOULD: Als een ketensamenwerking heeft besloten dat voor bepaalde API's een machtiging toegepast moet worden dan moet de Authorization Server voorafgaand aan de uitgifte van een Access Token vaststellen dat de client bevoegd is om binnen de gevraagde organisatiecontext te handelen. Hetzelfde geldt voor de Resource Server. Die moet de bevoegdheid kunnen controleren doordat de Authorization Server de RAR Authorization_details met de edu-to en edu-from in het Access Token heeft opgenomen.

    Belangrijke informatie rond een machtiging zijn identificerende gegevens van machtigingsverlener en gemachtigde en de client die namens gemachtigde handelt. Wat de bovenliggende grondslag is en het beheerproces rond de machtiging vallen buiten de scope van Edukoppeling. De Authorization Server moet op basis van de edu-to en edu-from parameters in het request naar het token endpoint kunnen vaststellen of de gegevensuitwisseling in de betreffende organisatiecontext toegestaan is. Er geldt hiervoor het volgende:

    a. MUST: De client wordt bij registratie of eventuele latere momenten geïnformeerd over welke API's het gebruik van machtiging vereisen.
    b. MUST: De referentie naar de machtiging moet worden doorgegeven via het OAuth 2.0 authorization_details mechanisme conform RFC 9396 (Rich Authorization Requests[59]).
    c. MUST: De referentie naar een machtiging moet worden opgenomen in het RAR authorization_details object in de body van het token request
    d. MUST: De authorization_details van RFC9396 moet een type-attribuut bevatten met waarde 'https://www.edustandaard.nl/standaard_afspraken/edukoppeling-transactiestandaard/authorization-details/v1/gemachtigde-gegevensuitwisseling'
    e. MUST: edu-from en edu-to moeten uitsluitend voorkomen binnen authorization_details object van type 'https://www.edustandaard.nl/standaard_afspraken/edukoppeling-transactiestandaard/authorization-details/v1/gemachtigde-gegevensuitwisseling'.
    f. MUST: De Authorization Server moet kenbaar maken dat authorization_details type 'https://www.edustandaard.nl/standaard_afspraken/edukoppeling-transactiestandaard/authorization-details/v1/gemachtigde-gegevensuitwisseling' wordt ondersteund.
    g. MUST: edu-from en edu-to moeten in deze versie gevuld worden met een OIN[60] met één van de volgende hoofdnummers:
        – 00000001 RSIN
        – 00000003 KvKnummer
        – 00000004 subnummer
        – 00000006 Logius OIN Hoofdnummer
        – 00000007 Instellingscode[61]
        – 00000008 Buitenlandse nummers

[59] RFC 9396 - OAuth 2.0 Rich Authorization Requests
[60] Zie https://gitdocumentatie.logius.nl/publicatie/dk/idauth/1.5.0/#bijlage-1-nummersystematiek-oin-en-hrn
[61] Op het moment van schrijven loopt er nog een wijzigingsproces bij Logius. In de huidige OIN-nummersystematiek staat nog het begrip 'BRIN'.

h. MUST: edu-from en edu-to moeten worden opgenomen in de authorization_details parameter in de body van het token request
   i. Voor clientauthenticatie bij het basis risicoprofiel (Client Password) geldt:

```
POST /token HTTP/1.1
Host: as.dd.nl
Authorization: Basic base64(client_id:client_secret)
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&
authorization_details=%5B%7B%22type%22%3A%22https%3A%2F%2
Fwww.edustandaard.nl%2Fstandaard_afspraken%2Fedukoppeling-
transactiestandaard%2Fauthorization-details%2Fv1%2Fgemachtigde-
gegevensuitwisseling%22%2C%22edu-
from%22%3A%22urn%3Aedukoppeling%3Aoin%3A0000000700025MB00
003%22%2C%22edu-
to%22%3A%22urn%3Aedukoppeling%3Aoin%3A0000000700025MB0000
3%22%7D%5D
```

   ii. Voor clientauthenticatie bij het verhoogd risicoprofiel (private_key_jwt methode) geldt:

```
POST /oauth2/token HTTP/1.1
Host: as.dd.nl
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&client_assertion_type=urn%3Aietf%3Aparam
s%3Aoauth%3Aclient-assertion-type%3Ajwt-
bearer&client_assertion=eyJhbG…
&authorization_details=%5B%7B%22type%22%3A%22https%3A%2F%2
Fwww.edustandaard.nl%2Fstandaard_afspraken%2Fedukoppeling-
transactiestandaard%2Fauthorization-details%2Fv1%2Fgemachtigde-
gegevensuitwisseling%22%2C%22edu-
from%22%3A%22urn%3Aedukoppeling%3Aoin%3A0000000700025MB000
03%22%2C%22edu-
to%22%3A%22urn%3Aedukoppeling%3Aoin%3A0000000700025MB00003
%22%7D%5D
```

i. MUST: De Authorization Server moet de ontvangen authorization_details valideren tegen de bovenliggende bevoegdheden voordat er een access token wordt geleverd.
   a. Indien authorization_details ontbreken of ongeldig zijn dan moet de Authorization Server het token request afwijzen.

j. MUST: Indien edu-to en edu-from succesvol zijn gevalideerd dan moet de Authorization Server de authorization_details en de edu-from en edu-to van het token request als claim opnemen in de in het Access Token.
   a. MUST: RFC 9396 definieert authorization_details expliciet als een JSON array van JSON objects. De RFC registreert bovendien authorization_details als JWT claim en beschrijft gebruik in JWT-based access tokens. We sluiten aan bij wat RFC 9396 definieert.

```
{
  "iss": "as.dd.nl",
```

```
  "sub": "my-client-id",
  "aud": "https://rs.dd.nl",
  "client_id": "my-client-id",
  "jti": "d3b07384-d113-4ec8-a152-09419149abcd",
  "exp": 1785522000,
  "iat": 1785518400,
  "authorization_details": [
    {
      "type":
"https://www.edustandaard.nl/standaard_afspraken/edukoppeling-
transactiestandaard/authorization-details/v1/gemachtigde-
gegevensuitwisseling",
      "edu-from": "urn:edukoppeling:oin:0000000700025MB00003",
      "edu-to": "urn:edukoppeling:oin:0000000700025MB00003"
    }
  ]
}
```


b. COULD: Nesting kan problemen geven bij resolvers die vooral platte claims verwachten. Mocht een ketensamenwerking met ketenpartners dit een probleem vinden dan is het toegestaan om edu-to en edu-from als extra claims op te nemen. Als er verschillen in zitten dan zijn de authorization_details leidend


```
{
  "iss": "as.dd.nl",
  "sub": "my-client-id",
  "aud": "https://rs.dd.nl",
  "client_id": "my-client-id",
  "jti": "d3b07384-d113-4ec8-a152-09419149abcd",
  "exp": 1785522000,
  "iat": 1785518400,


   "edu_from": "0000000700025MB00003",
   "edu_to": "0000000700025MB00003",

  "authorization_details": [
    {
      "type":
"https://www.edustandaard.nl/standaard_afspraken/edukoppeling-
transactiestandaard/authorization-details/v1/gemachtigde-
gegevensuitwisseling",
      "edu-from": "urn:edukoppeling:oin:0000000700025MB00003",
      "edu-to": "urn:edukoppeling:oin:0000000700025MB00003"
    }
  ]
}
```

5. Toepassingsscenario's

5.1. Probleemstelling - wat moet er met het profiel geregeld worden?

Om binnen een ketensamenwerking technisch vertrouwen tussen ketenpartners te realiseren, worden afspraken gemaakt over de wijze waarop transportbeveiliging en de identificatie en authenticatie van clients interoperabel worden ingericht. Deze afspraken vormen de basis voor een veilige uitwisseling van vertrouwelijke gegevens tussen ketenpartners. Het Edukoppeling OAuth-profiel geeft voor een belangrijk deel invulling aan deze afspraken. Het beoogt de gemeenschappelijke taal te zijn die voorkomt dat partijen bij elke koppeling opnieuw discussie voeren over logistieke details. Met het voorschrijven van TLS volgens de UBV TLS[62] afspraak zorgen we voor veilig transport van vertrouwelijke gegevens. Clients worden bij de Authorization Server geïdentificeerd aan de hand van een geregistreerde client_id. De clientregistratie legt tevens de relatie vast tussen de technische client en het OIN van de organisatie die verantwoordelijk is voor de client. Hiermee worden de technische identiteit van de client en de organisatorische identiteit van de ketenpartner expliciet met elkaar verbonden. Naast identificatie moet de client zich bij de Authorization Server authenticeren. Hiervoor wordt een binnen de clientregistratie vastgelegde authenticatiemethode toegepast, waarbij de client bewijst te beschikken over de bijbehorende credentials, zoals een secret of cryptografische sleutel. Het Edukoppeling OAuth-profiel beperkt de toegestane methoden voor clientauthenticatie, zodat ketenpartners deze op een interoperabele en voldoende veilige wijze kunnen ondersteunen. Een ketensamenwerking bepaalt binnen de kaders van het Edukoppeling OAuth-profiel welke van deze authenticatiemethoden worden ondersteund. Een deelnemende organisatie moet voor haar client ten minste één van de door de ketensamenwerking ondersteunde methoden kunnen toepassen. De Authorization Server registreert voor de client welke authenticatiemethode wordt gebruikt en de daarvoor benodigde verificatiegegevens.

Het Edukoppeling OAuth-profiel biedt ketensamenwerkingen een volgende stap in het gecontroleerd en interoperabel beveiligen van toegang tot API's. In bestaande ketensamenwerkingen worden hiervoor nog mechanismen toegepast zoals API-keys of eerdere Edukoppeling-profielen waarbij toegang onder meer wordt gebaseerd op whitelisting van OIN's. Deze mechanismen bieden beperkte mogelijkheden om toegang fijnmazig, contextafhankelijk en in de tijd begrensd te autoriseren. Met het Edukoppeling OAuth-profiel wordt toegang tot API's gebaseerd op Access Tokens. De Authorization Server authenticeert de client en bepaalt welke autorisatie voor de gevraagde interactie kan worden verleend. Daarbij kan gebruik worden gemaakt van de geregistreerde bevoegdheden van de client en de daaraan gekoppelde organisatie-identiteit, maar ook van aanvullende informatie uit bijvoorbeeld policies of machtigingen. Met scopes kan worden vastgelegd tot welke API of onderdelen en functies daarvan toegang wordt verleend. De verleende autorisatie wordt vastgelegd in een Access Token met een beperkte geldigheidsduur. Het token bevat daarmee alleen de autorisatie en context die voor de betreffende interactie nodig zijn. Naast scopes kan het token aanvullende autorisatiecontext bevatten, bijvoorbeeld in authorization_details met edu-from en edu-to. De Resource Server gebruikt deze informatie om bij de feitelijke API-aanroep te bepalen of toegang tot de gevraagde protected resource wordt toegestaan. De Authorization Server vormt hiermee een centraal beslis- en uitgiftepunt voor technische autorisatie, zonder dat alle onderliggende bevoegdheden noodzakelijk binnen de Authorization Server zelf worden beheerd. Bevoegdheden kunnen afkomstig zijn uit clientregistraties, autorisatiebeleid of andere gezaghebbende bronnen, zoals een machtigingsregister. Hierdoor kunnen autorisaties centraal en consistent worden toegepast,

[62] https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/

terwijl het beheer van de onderliggende bevoegdheden bij de daarvoor verantwoordelijke voorziening kan blijven.

5.2. Oplossing - wat biedt het profiel?

Ketensamenwerkingen binnen het onderwijs gebruiken veel API's die een verschillend risicoprofiel hebben. We belasten ketens niet onnodig met te zware maatregelen. We onderkennen echter wel dat regelgeving alleen maar toeneemt en dat er op termijn zwaardere eisen gaan gelden. We bieden ketensamenwerkingen dus de ruimte om van het lage risicoprofiel te migreren naar het verhoogd risicoprofiel. Verder wordt het overheidsdomein apart onderkend om in deze ketensamenwerking beter te kunnen aansluiten bij de vanuit de overheid opgelegde standaarden.

5.2.1. Laag risicoprofiel

Bij dit profiel mag de API geen of slechts beperkt gevoelige informatie ontsluiten. Er is een beperkt risico doordat er slechts een beperkte impact is bij misbruik. Voor zulke scenario's is de primaire behoefte dat ketenpartners uniform kunnen aansluiten met een lage implementatie- en beheerlast. Client-authenticatie bij het OAuth token-endpoint gebeurt in dit geval op basis van een gedeeld geheim (client-id en wachtwoord) via HTTP Basic-header. Het voordeel is vooral pragmatisch: vrijwel elke Authorization Server en client library ondersteunt dit. Hiermee is ook het registratieproces relatief eenvoudig. Met deze maatregel stappen we binnen ketensamenwerkingen af van een API-key. Hiermee wordt niet alleen bijgedragen aan interoperabiliteit en uitvoerbaarheid, maar wordt ook beveiliging verbetert ten opzichte van de toepassing van een API-key. Deze beveiligingsmaatregelen kunnen worden toegepast indien de impact van misbruik beperkt is.

5.2.2. Verhoogd risicoprofiel

Er geldt een verhoogd risicoprofiel als een API vertrouwelijke gegevens ontsluit en de impact bij misbruik hoog zal zijn. Er worden hogere eisen gesteld aan client-authenticatie. In plaats van authenticatie op basis van een wachtwoord wordt nu client-authenticatie uitgevoerd op basis van een ondertekening. De ondertekende JWT dient als bewijs dat de client over de private sleutel beschikt. De Authorization Server verifieert die handtekening met de geregistreerde publieke sleutel. De publieke sleutel wordt vertrouwd op basis van het gekozen technisch vertrouwensanker. Deze wordt bij voorkeur vooraf (tijdens client registratie) vastgelegd en gecontroleerd bij de Authorization Server. Deze beveiligingsmaatregelen passen bij ketensamenwerkingen waarin integriteit, traceerbaarheid en schade bij misbruik aanzienlijk zijn.

5.2.3. Verhoogd risico met toepassing van overheidsstandaarden

In de basis worden de beveiligingsmaatregelen van het verhoogd risicoprofiel toegepast, maar daarnaast worden ook overheidsstandaarden en voorzieningen toegepast, zoals PKIoverheid certificaten. Omdat dit profiel ook al de toepassing van het OIN voorschrijft krijgt men met PKIoverheid ook een sterke identificatie omdat de uitgifte van het certificaat gebonden is aan strikte controles door erkende aanbieders (TSP's) onder toezicht van Logius[63]. Het biedt hiermee hogere mate van betrouwbaarheid en sluit aan op afspraken binnen het overheidsdomein.

[63] PKIoverheid-certificaten bevatten het Organisatie-identificatienummer (OIN). De CA's die een PKIoverheid certificaat uitgeven verifiëren de identiteit van de organisatie in het handelsregister (https://www.logius.nl/onze-dienstverlening/toegang/pkioverheid)

6. Bijlage A: Begrippen

Begrip	Definitie	Bron
Certificaatautoriteit (CA)	Een certificaatautoriteit is in de cryptografie een entiteit die digitale certificaten verleent aan andere partijen. De bedoeling is dat het digitale certificaat bewijst dat de eigenaar daadwerkelijk degene is die hij beweert te zijn.	Edukoppeling
Client Secret	Een vertrouwelijke sleutel die alleen bekend is bij de client en afhankelijk van de vorm (symmetrisch (wachtwoord) of asymmetrisch (PKI)) ook bij de Authorization Server. Het wordt gebruikt om de client bij de Authorization Server te kunnen authenticeren via het token endpoint.	Edukoppeling
Shared Secret (gedeeld geheim)	Een vertrouwelijke gegeven dat bekend is bij zowel de Client als de Authorization Server en wordt gebruikt om de Client te authenticeren. Binnen OAuth 2.0 wordt hiervoor een client_secret toegepast. De Client bewijst bij authenticatie kennis van dit geheim, waarna de Authorization Server de aangeboden waarde verifieert aan de hand van de bij de Clientregistratie vastgelegde verificatiegegevens.	Edukoppeling
Confidential client	Een confidential client is een client die draait in een omgeving waar vertrouwelijke gegevens (waaronder credentials of cryptografisch sleutelmateriaal ten behoeve van clientauthenticatie) veilig bewaard kunnen worden (bijvoorbeeld server-side webapplicaties). Zie	RFC6749[64]
Public client	Een public client is een client die het client secret niet veilig kan beheren, bijvoorbeeld single-page web apps (SPA).	RFC6749[65]
RESTful API	Een RESTful API is een application programmable interface (API) die HTTP-methoden, zoals GET, POST, PUT, PATCH en DELETE, gebruikt om resources te beheren. Resources worden geïdentificeerd via URIs (Uniform Resource Identifiers) en worden doorgaans geretourneerd in JSON- of XML-formaat. We noemen	Edukoppeling

[64] https://datatracker.ietf.org/doc/html/rfc6749#section-2.1
[65] https://datatracker.ietf.org/doc/html/rfc6749#section-2.1

	ze RESTful omdat ze niet aan alle REST[66] principes[67] hoeven te voldoen.	
Vertrouwensanker (trust anchor)	Het startpunt van vertrouwen (bv. CA-root, pinned key, of whitelist).	Edukoppeling
Whitelisting	Whitelisting is een beveiligingsmaatregel waarbij alleen vooraf goedgekeurde toepassingen, bestanden, websites of gebruikers toegang krijgen tot een systeem of netwerk. Een whitelist is een (positieve[68]) lijst met identificerende gegevens waar bepaalde bevoegdheden mee samenhangen.	Edukoppeling

[66] https://en.wikipedia.org/wiki/REST
[67] Dit Edukoppeling OAuth-profiel gaat uit van vertrouwelijke gegevens waarbij ketenpartners weten wat ze van elke vragen binnen een bepaalde ketensamenwerking. Ondersteuning van HATEOAS lijkt niet noodzakelijk.
[68] In tegenstelling tot een 'blacklist' waar opgenomen identiteiten juist geblokkeerd moeten worden

7. Bijlage B: Referenties

- **RFC6749 OAuth 2.0 Authorization Framework (IETF PROPOSED STANDARD)** https://www.rfc-editor.org/info/rfc6749
- **RFC6750 OAuth 2.0 Authorization Framework: Bearer Token Usage (IETF PROPOSED STANDARD)** https://www.rfc-editor.org/info/rfc6750
- **RFC7519 JSON Web Token (IETF PROPOSED STANDARD)** https://www.rfc-editor.org/info/rfc7519
- **RFC7515 JSON Web Signature (JWS)** https://www.rfc-editor.org/rfc/rfc7515
- **RFC7523 JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants (IETF PROPOSED STANDARD)** https://www.rfc-editor.org/info/rfc7523
- **RFC7517 JSON Web Key (IETF PROPOSED STANDARD)** https://www.rfc-editor.org/rfc/rfc7517
- **RFC7518 JSON Web Algorithms (JWA) (IETF PROPOSED STANDARD)** https://www.rfc-editor.org/rfc/rfc7518
- **RFC7662 Token Introspection (IETF PROPOSED STANDARD)** https://www.rfc-editor.org/info/rfc7662
- **RFC8414 Authorization Server Metadata IETF PROPOSED STANDARD)** https://www.rfc-editor.org/info/rfc8414
- **RFC8705 Mutual-TLS Client Authentication (IETF PROPOSED STANDARD)** https://www.rfc-editor.org/info/rfc8705
- **RFC8707 Resource Indicators for OAuth 2.0 (IETF PROPOSED STANDARD)** https://www.rfc-editor.org/info/rfc8707
- **RFC9449 Demonstrating Proof of Possession (IETF PROPOSED STANDARD)** https://www.rfc-editor.org/info/rfc9449
- **RFC9068 JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens (IETF PROPOSED STANDARD)** https://www.rfc-editor.org/info/rfc9068
- **OpenID Connect Discovery 1.0 incorporating errata set 2 (OpenID Foundation[69] FINAL)** https://openid.net/specs/openid-connect-discovery-1_0.html
- **RFC5280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile** https://datatracker.ietf.org/doc/html/rfc5280

[69] Foundation - OpenID Foundation - OpenID Connect Core 1.0 incorporating errata set 2 OpenID connect is een identity layer bovenop het OAuth 2.0 protocol.

- **RFC9396 OAuth 2.0 Rich Authorization Requests Tokens (IETF PROPOSED STANDARD)** RFC 9396 - OAuth 2.0 Rich Authorization Requests