edustandaard

Edukoppeling

M2M gegevensuitwisseling binnen het onderwijs

OAuth 2.0 client credentials profiel voor RESTful API's

Edustandaard
Datum: 17 februari 2026
Status: concept

edustandaard

Inhoudsopgave

edustandaard

1. Status van dit document

Dit document is een conceptversie van het OAuth client credentials profiel voor RESTful API's (hierna OAuth-profiel). Het is gebaseerd op het OAuth client credentials werkdocument[1] met de toevoeging van de optie om mandaatinfo door te geven.

Er is momenteel nog geen RESTful API-architectuur die de context van dit OAuth-profiel nader toelicht. De toekomstige architectuur zal sterk afwijken van de vorige versie. In die versie onderkende we al RESTful API's, maar was nog sterk gericht op webservices en een Digikoppeling-indeling.

In de nieuwe opzet staan API's en token-based access centraal. We introduceren een flexibele architectuur die aansluit bij de actuele behoeften bij groeifondsen 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.

Met opmerkingen [ER1]: In deze versie van het OAuth-profiel is er momenteel nog geen standaardisatie doorgevoerd voor doorgifte van de publieke sleutel in de private_key_jwt header (zie H4 7.b.iii). We wachten nog op input van leden.

Verder ontbreekt ook het concept van een mandaat nog en hoe mandaatinformatie eventueel in de client credentials flow doorgegeven kan worden. Dit moet nog uitgewerkt worden nadat er input is gekomen vanuit DUO/OKE

1.1. Documenthistorie

Versie	Status	Auteur	Datum	Opmerking
0.9	Concept	E. Reinhoud (BES)	Februari 2026	

[1] 2026-02-03 Edukoppeling - OAuth client credentials profiel voor RESTful APIs.docx

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 by design veilig en eenvoudiger wordt. De toepassing van Edukoppeling zorgt ervoor dat, op het niveau van de applicatielaag, gesloten data tijdens transport van de ene naar de andere organisatie niet ongeoorloofd kan worden ingezien of gemanipuleerd. De standaard gaat over de afhandeling van berichten (het transport) en niet over de inhoud van berichten.

Met Edukoppeling worden voor ketensamenwerking een aantal ketenfuncties in de applicatielaag geregeld (zie Toepassingsscenario's).

2.2. Organisatorisch werkingsgebied van Edukoppeling

Edukoppeling schrijft voor hoe onderwijsorganisaties, publieke uitvoeringsorganisaties, dienstverleners[2] en andere ketenpartners gegevensuitwisselingen opzetten. Edukoppeling heeft als scope alle werkingsgebieden vallend onder alle onderwijssectoren. Zie de werkingsgebieden in ROSA[3]. 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.

2.3. Functioneel toepassingsgebied

Het functioneel toepassingsgebied van Edukoppeling is de geautomatiseerde uitwisseling van gesloten data[4] tussen informatiesystemen van onderwijsorganisaties en ketenpartners (onderling, met bedrijven of met de overheid). Deze uitwisseling betreft M2M point-to-point verbinding voor uitwisseling tussen een confidential client en een gesloten API, waarbij de toegang is geregeld met OAuth 2.0. De verschillende Oauth 2.0 client credentials rollen kunnen een computersysteem van een onderwijsorganisatie, een uitvoeringsinstantie, een overheidsorganisatie als een (commerciële) dienstverlener zijn.

2.4. Positionering van Edukoppeling in het Edustandaard vijflagen model

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

1. Grondslagenlaag: borgt de juridische basis en beleidskaders waarbinnen gegevensuitwisseling is toegestaan;

[2] Dienstverleners (ROSA: *Een organisatie die een dienst levert aan een organisatie of een natuurlijke persoon).*
[3] https://rosa.wikixl.nl/index.php/Werkingsgebieden
[4] 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
[5] AMIGO-methodiek-1.1.0-1.pdf en https://rosa.wikixl.nl/index.php/Interoperabiliteit_en_het_Edustandaard_lagenmodel#Opbouw_van_het_lagenmodel

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.

Het Edukoppeling OAuth2.0-profiel levert binnen het Edustandaard vijflagenmodel belangrijke ondersteuning aan de applicatielaag en heeft een relatie met de IT-infrastructuurlaag. De kaders van Edukoppeling worden opgenomen in het afsprakenstelsel van de betreffende ketensamenwerking[6], waarin ook de kaders van overige architectuurlagen[7] zijn gevat.

2.5. Compliance aan overheidsstandaarden

Edukoppeling is gebaseerd op internationale open standaarden en Edustandaard standaarden. De Edukoppeling profielen zijn op zichzelf staande documenten en waar relevant wordt verwezen naar de (volwassen) internationale open industriestandaarden en onderwijsstandaarden (bijvoorbeeld Edustandaard UBV TLS[8]). Edukoppeling wordt onder Edustandaard beheer doorontwikkeld. De werkgroep is verantwoordelijk voor het identificeren en oplossen van vraagstukken voor de Edustandaard Architectuurraad.

De vorige versies van Edukoppeling waren gebaseerd op de overheidsstandaard Digikoppeling[9]. We hebben de afgelopen tijd weer een aantal keer ervaren dat de doorontwikkeling rond deze standaard achter liep op waar binnen het onderwijs behoefte aan is. Dit was eigenlijk ook al de aanleiding in 2014 voor het creëren van de Edukoppeling-standaard. Bij de ontwikkeling van het Edukoppeling REST-profiel hebben we destijds aansluiting gevonden bij het Kennisplatform API's[10]. Meer recent hebben we ook voor de Edukoppeling OAuth Best practices gebruik gemaakt van het NL GOV Assurance profile for OAuth 2.0[11] van het Kennisplatform API's. Deze Edukoppeling OAuth Best practices gebruikte het NL GOV profiel als basis, maar doordat deze weer van het International Government Assurance Profile (iGov) for OAuth 2.0[12] profiel is afgeleid wordt het een uitdaging om traceerbaarheid te handhaven en goede leesbaarheid te behouden. We merken dat als we voorlopen, Digikoppeling ons soms volgt, maar men maakt soms daar ook weer (andere) grote stappen waar we als onderwijs (nog) niet op willen of zelfs kunnen aansluiten. Dit is in de huidige situatie aan de orde doordat het Digikoppeling REST profiel[13], dat eerst redelijk aansloot op het REST profiel van Edukoppeling, nu de Federated Service Connectivity (FSC) als onderdeel verplicht stelt, wat voor het onderwijs een brug te ver is en ook niet goed past.

[6] 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)
[7] Zie Positionering van Edukoppeling in het Edustandaard vijflagen model
[8] https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/
[9] Digikoppeling Overzicht Actuele Documentatie en Compliance 1.12.2
[10] Kennisplatform API's | developer.overheid.nl
[11] NL GOV Assurance profile for OAuth 2.0 v1.1.0
[12] https://openid.net/specs/openid-igov-oauth2-1_0-03.html
[13] Digikoppeling Koppelvlakstandaard REST-API 3.0.1

Vanaf deze versie van Edukoppeling[14] gaan we de directe afhankelijkheid van de Digikoppeling en Kennisplatform API's-Specificaties loslaten. Dit is dus een trendbreuk met wat we tot nu toe hebben gedaan, maar de achterliggende argumentatie wordt door de werkgroep herkend en onderschreven. We blijven wel de verschillende gremia en specificaties volgen. Periodiek wordt de werkgroep geïnformeerd over ontwikkelingen en we laten ons bij het opstellen van Edukoppeling-documentatie door betreffende bronnen inspireren, maar de Edukoppeling documenten zijn op zichzelf staande documenten. We gaan dus niet per thema aangeven waar wel of niet afgeweken wordt van overheidsstandaarden en waarom. We houden wel een overzicht bij welke overheidsstandaarden raakvlakken hebben met Edukoppeling[15]. Dit is onderdeel van het ontwikkelproces binnen de werkgroep.

2.6. 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[16]. Deze gebruikt de volgende termen: "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL".

2.7. Leeswijzer

Hoofdstuk 2 bevat een toelichting op de OAuth 2.0 standaard en Edukoppeling voorschriften. In hoofdstuk 3 zijn de Educkoppeling voorschriften opgenomen. Dit zijn aanscherpingen op de onderliggende internationale open industrie standarden rond OAuth 2.0 en heeft de client credentials grant (RFC6749) als basis.In hoofdstuk 4 worden scenario's beschreven die een aantal voorschriften in een bepaalde context plaatsen om ketensamenwerkingen te ondersteunen in de te maken keuzes.

[14] Op te stellen profielen en architectuur
[15] Zie **Fout! Verwijzingsbron niet gevonden.**
[16] https://tools.ietf.org/html/rfc2119

3. Toelichting voorschriften

3.1. Inleiding

Dit hoofdstuk bevat een toelichting op de OAuth 2.0 client credentials grant en de Edukoppeling voorschriften. Met OAuth 2.0 client credentials wordt de mogelijkheid geboden om RESTful API's te beveiligen op de applicatielaag. Confidential clients krijgen alleen toegang tot de RESTful 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.

3.1.1. Rollen

Binnen de OAuth client credentials grant zijn de volgende componenten relevant:

- Client;
- Authorization Server;
- Resource Server.

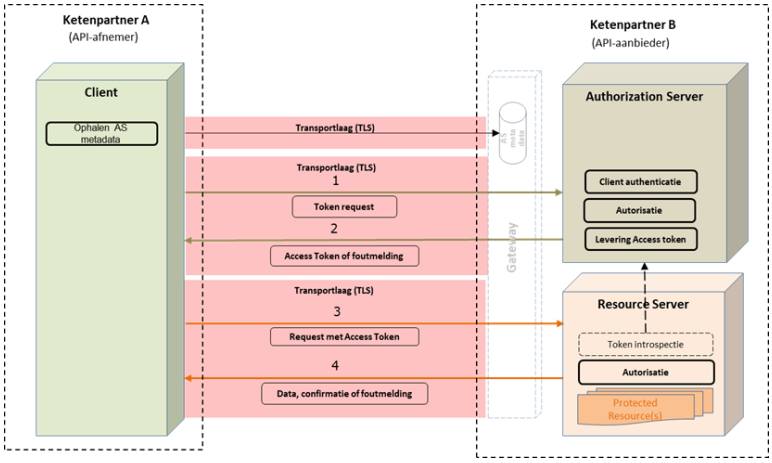
De rollen worden respectievelijk in paragrafen 3.2, 3.3 en 3.4 toegelicht.

3.1.2. Interacties

Deze componenten hebben de volgende interacties:

- Client request naar Authorization Server token endpoint (1)[17]
- Response van Authorization Server naar client (2)
- Client request naar Resource Server (3)[18]
- Response van Resource Server naar client (4)

De componenten en interacties worden weergegeven in Figuur 1.



Figuur 1 – OAuth 2.0 client credentials grant

[17] Zie Authenticeren client
[18] Zie Toegang tot resourceResource Server

3.2. Client

Een client is een systeem dat een organisatie als API-afnemer gebruikt om één of meer gegevensuitwisselingen uit te voeren binnen één of meerdere ketensamenwerkingen. Daarbij communiceert de client met één of meer API's van één of meerdere API-aanbieders. Om toegang te krijgen, moet de client zich kunnen authentiseren op basis van de methode(n) die de Authorization Server ondersteunt. De client identificeert zich daarbij met de unieke identifier die de Authorization Server voor de client heeft geregistreerd. Client-identificatie en -authenticatie vormen samen de technische basis voor het vertrouwen tussen ketenpartners. Ter ondersteuning hiervan moet de API-afnemer de client registreren bij de API-aanbieder, zodat de benodigde identificatiegegevens en authenticatiemiddelen (zoals secrets, sleutels of certificaten) kunnen worden vastgelegd en beheerd.

3.2.1. Clientregistratie

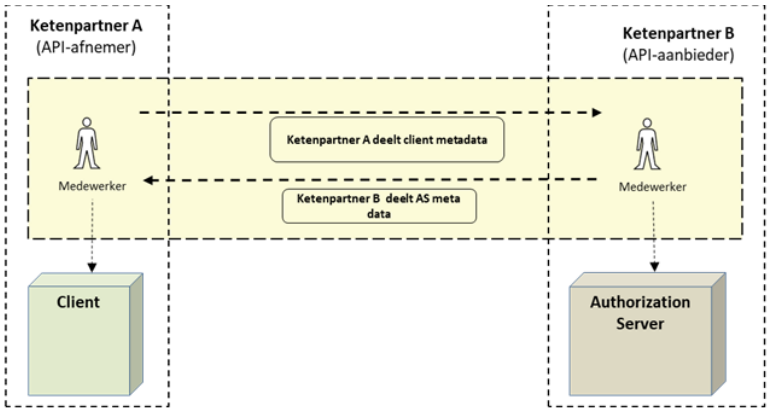
De ketenpartner die een client beheert (de API-afnemer) is ervoor verantwoordelijk dat deze client vóór deelname aan de gegevensuitwisseling is geregistreerd bij de API-aanbieder. Registratie bij de Authorization Server van de API-aanbieder is nodig omdat met de toepassing van OAuth 2.0 een client alleen toegang krijgt tot een API door deze een access token[19] aan te bieden dat door de Authorization Server aan de client is uitgegeven.

De registratie legt de technische en organisatorische basis voor vertrouwen in de keten. Tijdens dit proces worden de client-identiteit (client_id) en de bijbehorende authenticatiemethode[20] vastgelegd. Bij de registratie van de client wordt door de API-aanbieder vastgelegd welke authenticatiemethode de client toepast en is gekoppeld aan de client_id. Per client – API aanbiedercombinatie wordt er dus één client-authenticatiemethode toegepast.

Ook wordt vastgelegd welke bevoegdheden de client kan krijgen. Dit zijn de scopes die een ketensamenwerking of de betreffende API-aanbieder zelf voor de API('s) heeft gedefinieerd. Hierdoor kan de Authorization Server bij elke tokenaanvraag bij het token endpoint ook controleren of de client zich correct authentiseert en alleen tokens uitgeven met de rechten die passen bij de geregistreerde bevoegdheden.

Het registratieproces zelf is geen onderdeel van dit profiel, maar wel de metadata die nodig is voor de OAuth 2.0 client credentials grant.

[19] Zie paragraaf Access token. Dit profiel ondersteunt het gebruik van bearer tokens.
[20] Zie voor opties conform dit Edukoppeling Oauth-profiel paragraaf Authenticeren client



Figuur 2 – Uitwisseling van metadata tussen ketenpartijen t.b.v. registratie client.

Algemene afspraken die door een ketensamenwerking[21] moeten worden vastgesteld:

1. Toegestane client-authenticatiemethoden
    a. Zie Authenticeren client
2. Scope(s)[22]
    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

Specifieke afspraken van API-afnemer (client) naar API-aanbieder (Authorization Server):

1. Identiteit API-afnemer (OIN)
2. Authenticatiemethode voor betreffende client (indien er sprake is van een keuze)
    a. Publieke sleutel indien private_key_jwt van toepassing is.
3. Tot welke API's de client toegang wil en met welke scope(s)

Specifieke afspraken van API-aanbieder (Authorization Server) naar API-afnemer (client):

1. Identiteit API-aanbieder (OIN)
2. De identifier waar de client mee geïdentificeerd wordt door Authorization Server (client_id)
3. API's en bevestiging van te gebruiken scopes (bevoegdheden)
4. Bevestiging authenticatiemethode
    a. Wachtwoord (bij client_secret_basic)
    b. Bevestiging verwerking publieke sleutel (bij private_key_jwt)
5. Verwijzing naar Authorization Server discovery endpoint + beleidsvoorwaarden
    a. OpenID Connect service discovery [OpenID.Discovery] endpoint

[21] Verschillende ketensamenwerkingen kunnen dus verschillende keuzes maken.
[22] Zie ook Edu-V, een API heeft specifieke scopes die binnen ketensamenwerking zijn vastgesteld.

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, of de rol van de AP-afnemer in de keten verandert. Voorbeelden zijn het toevoegen of intrekken van scopes (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.

3.2.4. Scopes

Scopes worden gebruikt om een client grofmazig[23] 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

[23] Zie Probleemstelling - wat moet er met het profiel geregeld worden?

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 verschillende client-authenticatiemethoden mogelijk. Die worden niet als gelijkwaardig beschouwd, omdat ze verschillen in sterkte, beheerlast en het risico bij misbruik[24]. 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 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 aanvullende bevoegdheden gecontroleerd worden. Een client kan bij het token endpoint scopes aanvragen die buiten zijn geregistreerde bevoegdheden vallen. Het token endpoint moet daarom elke aangevraagde 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.

De set met toegestane scopes is niet per definitie statisch. Na registratie kunnen scopes voor een client worden toegevoegd, gewijzigd of ingetrokken[25].

3.3.4. Levering access token of foutmelding

Na een positief autorisatiebesluit wordt een access token[26] gecreëerd. In het JWT access token worden claims als ‘scope’, ‘aud’ en ‘exp’ opgenomen. De JWT, als zelfbeschrijvend token[27], of een referentie (opaque token) wordt aan de client geleverd. Bij een negatief autorisatiebesluit wordt de client een foutmelding[28] conform RFC6749 geleverd.

[24] Zie Toepassingsscenario’s
[25] Zie Client lifecycle
[26] https://www.rfc-editor.org/rfc/rfc6749#section-5.1
[27] Zie Access token
[28] https://www.rfc-editor.org/rfc/rfc6749#section-5.2

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 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 om het token te laten verifiëren door de Authorization Server. Bij een JWT access token (zelfbeschrijvend) 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 de API-aanbieder. De Authorization Server kent[29] de Resource Server(s) 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.

3.4.2. API lifecycle

Het is belangrijk API-afnemers en hun clients geïnformeerd worden over (lifecycle) informatie rond de API. Zij moeten zich op eventuele wijzigingen voor kunnen bereiden.

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

[29] Authorization Server kan de Resource Server identificeren (aud claim in access token).

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[30]. 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, maar definieert niet hoe TLS werkt. 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[31]), verifiëren de identiteit van de server. De (root-)CA is het technische

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

[31]
Toepassingsscenario's
Probleemstelling - wat moet er met het profiel geregeld worden?
Voor onderwijsorganisaties, sectorvoorzieningen en ketenpartners die binnen een ketensamenwerking met elkaar gesloten data uitwisselen is het van belang dat zij elkaar kunnen vertrouwen. Daarnaast is het wenselijk dat er een mate van standaardisatie is zodat de logistieke koppeling over ketensamenwerkingen heen interoperabel is. De afspraken binnen het Edukoppeling OAuth-profiel biedt een belangrijke basis voor vertrouwen en interoperabiliteit. Het beoogt de gemeenschappelijke taal te zijn die voorkomt dat partijen bij elke koppeling opnieuw discussie voeren over logistieke details.
Om binnen een ketensamenwerking vertrouwen tussen ketenpartijen te organiseren moeten er afspraken worden gemaakt over hoe veilig transport van de gesloten data en identificatie en authenticatie van clients op een interoperabele manier moet worden ingericht. Met de toepassing van TLS en OAuth 2.0 geven we invulling aan deze vraagstukken. TLS volgens de UBV TLS afspraak zorgt voor veilig transport van gesloten data. API-aanbieders identificeren clients van API-afnemers op basis van de door de Authorization Server uitgegeven client_id. Deze wordt gekoppeld aan het OIN van de API-afnemer. Authenticatie van de client wordt gedaan op basis van het door de Authorization Server uitgegeven authenticatiemiddel. Het Edukoppeling OAuth-profiel beperkt het aantal authenticatiemiddelen dat een ketensamenwerking mag voorschrijven. De API-aanbieder heeft slechts de keuze uit de door de ketensamenwerking geboden opties.
Het Edukoppeling OAuth-profiel gaat ketensamenwerkingen een duidelijke stap vooruit bieden ten opzichte van eerder manieren voor toegang tot API's. Ketensamenwerkingen

gebruiken nog steeds API-keys of een ouder Edukoppeling-profiel met whitelisting op basis van een OIN. Met het Edukoppeling OAuth-profiel maken we het mogelijk om toegang op tokens te baseren. De Authorization Server kan op het niveau van een client (en indirect organisatie-identiteit) bevoegdheden registreren voor een API en onderdelen binnen deze API (op basis van scopes). Een client heeft met het ontvangen access token dat een beperkte geldigheid heeft ook beperkte autorisaties passend bij de context. De Authorization Server zorg ervoor dat een API-aanbieder deze meer fijnmazige bevoegdheden centraal kan beheren.

Tegelijk is die fijnmazigheid niet onbeperkt. Scopes zijn meestal ontworpen als functionele rechten op API-niveau: ze geven aan welke handelingen (bijvoorbeeld lezen/schrijven) een client mag uitvoeren op een bepaalde set endpoints of functionaliteiten. Dat is “fijnmaziger” dan alleen een binaire vorm van toegang, maar het ondersteunt geen beheer van bevoegdheden op gegevensniveau.

Oplossing - wat biedt het profiel?
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.
Laag risicoprofiel
Ketensamenwerkingen binnen het onderwijs gebruiken veel API’s die een verschillend risicoprofiel hebben. De gesloten data die zij ontsluiten kunnen slechts beperkt gevoelige informatie bevatten en een beperkte impact bij misbruik hebben. Er is een laag risicoprofiel. Voor zulke scenario’s is de primaire behoefte dat ketenpartijen 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. Het voordeel is vooral pragmatisch: vrijwel elke Authorisation 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.
Verhoogd risicoprofiel
Er geldt een verhoogd risicoprofiel als een API gesloten data 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.
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. Het biedt hiermee hogere mate van betrouwbaarheid en sluit aan op afspraken binnen het overheidsdomein.

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[32]. De UBV TLS werkgroep is verantwoordelijk voor het volgen van de ontwikkelingen[33] rond TLS en het beheer van de standaard.

3.7. Technisch vertrouwensanker

Dit Edukoppeling OAuth-profiel is bedoeld voor M2M gegevensuitwisseling van gesloten data tussen ketenpartners. In dit type samenwerking moeten partijen op elkaars systemen kunnen vertrouwen en is een technisch vertrouwensanker noodzakelijk. Het zorgt ervoor dat ketenpartners elkaar eenduidig kunnen herkennen. Als technisch vertrouwensanker wordt er vaak gekozen voor een Public Key Infrastructure (PKI). PKI geeft ketenpartijen de mogelijkheid om elkaar en systemen cryptografisch te identificeren en authenticeren op basis van digitale certificaten. Het vertrouwen in certificaten is gebaseerd op een hiërarchie van certificaatautoriteiten (CA's) die de certificaten op basis van vooraf vastgestelde procedures uitgeven. De hiërarchie bestaat uit een root CA en eventuele intermediate CA's. Een root-CA wordt door de ketenpartners expliciet als betrouwbaar aangewezen en vormt het uitgangspunt voor het valideren van certificaten die door onderliggende (intermediate) CA's worden uitgegeven. Door uitsluitend certificaten te accepteren die herleidbaar zijn tot de afgesproken root-CA (en bijbehorende keten), kunnen systemen binnen de ketensamenwerking elkaar op een consistente en controleerbare manier vertrouwen.

Dit Edukoppeling OAuth-profiel ondersteunt het gebruik van een PKI technisch vertrouwensanker voor zowel TLS (beveiligde transportlaag) als voor het ondertekenen en verifiëren van JWT's. De ketensamenwerking kiest hiervoor één of meer opties en legt deze als onderdeel van de ketenafspraken vast:

1. Public trust: certificaten die herleidbaar zijn tot een algemeen vertrouwde publieke root-CA (publiek vertrouwde certificaten);
2. Private trust governmental: certificaten die herleidbaar zijn tot de private root-CA van PKIoverheid[34].

3.7.1. Private_key_jwt

Met de toepassing van de private_key_jwt authenticatiemethode wordt een hoger betrouwbaarheidsniveau gerealiseerd. De Authorization Server controleert niet alleen dat de JWT correct is ondertekend met de aangegeven publieke sleutel, maar verifieerd ook dat de gebruikte publieke sleutel kan worden vertrouwd. De publieke sleutel moet onderdeel zijn van een certificaat dat is uitgegeven door een vertrouwde CA (technisch vertrouwensanker). De JWT-header moet informatie bevatten waarmee de Authorization Server de publieke sleutel van de ondertekening kan vinden, maar kan ook die van CA's bevatten. Met de

[32] Meer informatie via Werkgroep Uniforme Beveiligingsvoorschriften: https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/
[33] Bijvoorbeeld NCSC (Home | Nationaal Cyber Security Centrum)
[34] PKIoverheid

publieke sleutel van de CA kan de Authorization Server vaststellen dat de publieke sleutel kan worden vertrouwd. Er worden drie varianten ondersteund:

1. jwk;
2. x5c;
3. x5u;
4. jku.

Met opmerkingen [ER2]: WORDT 1 MANIER! De werkgroepleden gaan hun voorkeur voor die ene optie nog kenbaar maken. Op dit punt wordt deze versie dus nog aangepast

jwk (JWK Set inline) – sleutelset in header

Een jwk in de JWT-header is alleen toegestaan wanneer de gebruikte publieke sleutel vooraf is geregistreerd. De jwk kan het vertrouwensanker bevatten, maar dan moet hierop, net als bij de sleutel voor ondertekening, validatie plaatsvinden tegen een vooraf ingerichte trust store. De jwk sleutel(s) is gepind aan de betreffende client tijdens registratie. 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.

Met opmerkingen [KG3]: Als de CA van te voren geregistreert is dan kun je de keten prima vertrouwen. DIt heeft onze voorkeur.. Kun je van certificaat wisselen zoalng je OIN en CA niet veranderen. Kun je ook makkelijk een nieuw vertifcaat infaseren die toch veilig is

Met opmerkingen [ER4R3]: Klopt, hier staat echter dat als het alleen in jwt is opgenomen dat dit dan niet moet worden vertrouwd. De keuze rond JWT params is apart punt wat besproken wordt. Op basis van komend overleg wordt dit document nog op punten aangepast

x5c (X.509 Certificate Chain) – certificaat (en hiërarchie) in JWT-header

Met x5c levert de client het certificaat direct mee in de JWT-header. Dit voorkomt ophalen via URL. Hiermee worden de tokens groter dan het opnemen van een (vooraf gevalideerde) referentie (jku/x5u). Om risico's te beperken wordt het vertrouwensanker voor het x5c certificaat bij clientregistratie geregistreerd en opgenomen in de truststore van de Authorization Server.

jku (JWK Set URL) – verwijzing naar een sleutelset

Met jku kan de client een URL meegeven waar een JWKS (JSON Web Key Set) staat. De Authorization Server kan die JWKS via de JWT-header ophalen en de sleutel selecteren op basis van kid in de JWT-header. Dit is functioneel handig (te behoeve van sleutelrotatie), maar introduceert ook risico omdat de Authorization Server een jku-URL uit de JWT-header haalt en accepteert. De opgegeven URL wordt dan "automatisch" vertrouwd en dat de aangeboden sleutel authentiek is, terwijl die URL (of de route ernaartoe) door een aanvaller kan worden gemanipuleerd. Deze optie is minder risicovol als de jku per client vooraf wordt geregistreerd. Het vertrouwensanker (root CA) moet bij gebruik van jku vooraf aan de uitwisseling in de truststore van de Authorization Server zijn opgenomen.

x5u (X.509 URL) – verwijzing naar een certificaat

x5u is vergelijkbaar met jku, maar verwijst naar een URL waar een X.509 certificaat (of hiërarchie) te halen is. Net als bij jku geldt dat het ophalen van het vertrouwensanker tijdens het request naar het token endpoint de risico's vergroot en niet wenselijk is. Als x5u gebruikt wordt dan moet de certificaten ook vooraf registreren en het ondertekencertificaat per client exact matchen (whitelist).

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 RESTful 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 'Bijlage C: 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 point-to-point 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. API-aanbieders en API-afnemers passen toe wat binnen de keuze van een ketensamenwerking valt. 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. Interacties tussen deze twee rollen is geen onderdeel van dit Edukoppeling OAuth-profiel.

1. MUST: Voor identificatie van ketenpartijen (API-aanbieder en API-afnemer) moet de Digikoppeling OIN nummersystematiek[38] worden toegepast.

2. MUST: API-providers communiceren aan API-afnemers welke client-authenticatiemethode(n) ondersteund worden.
   a. MUST: De client-authenticatiemethode(n) die een API-provider kan ondersteunen wordt/worden door de ketensamenwerking bepaald.

3. MUST: Dit Edukoppeling OAuth-profiel is alleen van toepassing in de context van confidential clients. De client moet over vertrouwelijke gegevens (client secret) kunnen beschikken.

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

Met opmerkingen [KG5]: Als je hier een MUST van maakt, dat maak je het clientid/secret level afhankelijk van pki overheid certificaten en dat willen we juist niet.

Moet je de lijst hier niet onderverdelen in
- altijd geldig
- alleen geldig voor niveau clientid/secret
- geldig voor next level met private_key_jwt
- geldig voor next/next level

Met opmerkingen [ER6R5]: Bespreken. In de vorige versie was identificatie op dit niveau een keuze voor de ketensamenwerking (keuze PKIO dan ook OIN). Er werd afgelopen overleg gesteld dat partijen altijd obv OIN worden geïdentificeerd.

[35] OAuth 2.0 — OAuth
[36] IETF | RFCs
[37] Zie Functioneel toepassingsgebied
[38] 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).

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.

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, maar definieert niet hoe TLS werkt. 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.

Technisch vertrouwensanker

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

   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. Versleuteling 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 HTTP basic authentication (RFC2617[42]) zoals beschreven bij Client Password[43] in RFC6749.
      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.
      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)[44].
      i. MUST: Clients moeten beschikken over een PKI-certificaat met een publiek-privaat sleutelpaar, uitgegeven door een partij die door de ketensamenwerking wordt vertrouwd (zie Technisch vertrouwensanker).

[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: HTTP Basic authentication scheme as defined in [RFC2617]
[43] https://www.rfc-editor.org/rfc/rfc6749.html#section-2.3.1
[44] Client Authentication (Final: OpenID Connect Core 1.0 incorporating errata set 2)

ii. MUST: Clients moeten de private sleutel gebruiken voor het ondertekenen van de JWT.
    - MUST: De ondertekenmethode moet ten minste RS256 (Rivest, Shamir, and Adleman (RSA) signature algorithm with a 256-bit hash) zijn en mag een andere asymmetrische ondertekenmethoden zijn conform de JSON Web Algorithms, RFC7518.

iii. MUST: Clients moeten in het token endpoint request de publieke sleutel (eindcertificaat) delen met de API-aanbieder / Authorization Server en kan op basis van vier manieren:
    - **jwk**:
        - MUST: de jwk keyset is vooraf bekend en geregistreerd voor de client
        - MUST: de sleutel moet worden gecontroleerd met de voor de client geregistreerde sleutel
        - SHOULD: bij meerdere keys per client moet een kid worden gebruikt
    - **x5c:**
        - MUST: De JWT-header bevat de x5c-claim die het X.509-certificaat van de ondertekening bevat
        - MUST: Het eerste certificaat in de lijst moet het certificaat zijn dat de publieke sleutel bevat die overeenkomt met de private sleutel waarmee de JWT is ondertekend.
        - MUST: Het vertrouwensanker (root CA) moet vooraf aan de uitwisseling in de truststore van de Authorization Server zijn opgenomen.
    - **jku:**
        - SHOULD: Bij toepassing hiervan zou de jku-URL vooraf bij Authorization Server geregistreerd moeten worden en zou in de JWT-header een kid-claim opgenomen moeten zijn om naar juiste sleutel te verwijzen.
        - MUST: Het vertrouwensanker (root CA) moet bij gebruik van jku vooraf aan de uitwisseling in de truststore van de Authorization Server zijn opgenomen.
    - **x5u:**
        - MUST: de JWT-header bevat de x5u-claim die verwijst naar een HTTPS-endpoint met een PEM-gecodeerd certificaat
        - MUST: de JWT-header bevat de claims x5t of x5t#S256 (RFC7515) om te controleren of het gedownloade certificaat niet is gemanipuleerd voordat het wordt gebruikt (t.b.v. whitelisting).

iv. MUST: JWT-header moet een jti-claim bevatten en is uniek per request naar het token endpoint.

v. MUST: de ondertekende JWT moet worden opgenomen in de ‘client_assertion‘ form-parameter in het request naar het token endpoint conform RFC6749

Met opmerkingen [ER7]: WORDT 1 MANIER! De werkgroepleden gaan hun voorkeur voor die ene optie nog kenbaar maken. Op dit punt wordt deze versie dus nog aangepast

8. MUST: De Authorization Server moet bij clientauthenticatie op basis van HTTP basic authentication controleren of voor de betreffende client het client secret klopt.

9. MUST: De Authorization Server moet bij clientauthenticatie op basis van de private_key_jwt methode de ondertekening van de JWT valideren
   a. MUST: De Authorization Server valideert vooraf het vertrouwensanker (hiërarchie root CA en eventuele intermediate CA's).
   b. MUST: Vertrouwensanker(s)[45] moeten vooraf aan de gegevensuitwisseling beschikbaar zijn voor de Authorization Server.

10. MUST: De Authorization Server controleert of de client over de gevraagde scopes beschikt.
    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).

11. MUST: De Authorization Server levert de client na een positief autorisatiebesluit een Access Token[46] in de vorm van een bearer token (RFC6750). Bij een negatief autorisatiebesluit wordt een foutmelding[47] conform RFC6749 geleverd. Er wordt geen refresh-token geleverd.
    a. MUST: Het Access Token wordt geleverd in de vorm van een opaque token of een JWT conform RFC7523.
    b. MUST NOT: Clients moeten niet om een refresh-token vragen en de Authorization Server moet clients geen refresh-token leveren[48]

12. MUST: Clients moeten de authorization request header gebruiken om het access token (bearer-token) te verzenden zoals gedefinieerd in RFC6750[49].

13. SHOULD: Resource Servers zouden conform RFC6750[50] een foutmelding moeten geven bij een invalide clientrequest.
    a. SHOULD NOT: 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[51]
    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.

[45] Zie **Fout! Verwijzingsbron niet gevonden.**
[46] https://www.rfc-editor.org/rfc/rfc6749#section-5.1
[47] https://www.rfc-editor.org/rfc/rfc6749#section-5.2
[48] Hoewel RFC6749 (https://www.rfc-editor.org/rfc/rfc6749#section-4.4.3) stelt dat dit type clients geen refresh token geleverd zou moeten worden, sluiten we hier aan op NL GOV OAuth (https://logius-standaarden.github.io/OAuth-NL-profiel/#direct-access-client) waarin gesteld wordt dat dit niet nodig is voor dergelijke clients.
[49] Er wordt op dit punt aangesloten op de nieuwe versie van het OpenID iGov profiel https://openid.net/specs/openid-igov-oauth2-1_0.html#section-2.7.1
[50] https://datatracker.ietf.org/doc/html/rfc6750#section-3.1
[51] https://datatracker.ietf.org/doc/html/rfc6750#section-3.1

5. Toepassingsscenario's

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

Voor onderwijsorganisaties, sectorvoorzieningen en ketenpartners die binnen een ketensamenwerking met elkaar gesloten data uitwisselen is het van belang dat zij elkaar kunnen vertrouwen. Daarnaast is het wenselijk dat er een mate van standaardisatie is zodat de logistieke koppeling over ketensamenwerkingen heen interoperabel is. De afspraken binnen het Edukoppeling OAuth-profiel biedt een belangrijke basis voor vertrouwen en interoperabiliteit. Het beoogt de gemeenschappelijke taal te zijn die voorkomt dat partijen bij elke koppeling opnieuw discussie voeren over logistieke details.

Om binnen een ketensamenwerking vertrouwen tussen ketenpartijen te organiseren moeten er afspraken worden gemaakt over hoe veilig transport van de gesloten data en identificatie en authenticatie van clients op een interoperabele manier moet worden ingericht. Met de toepassing van TLS en OAuth 2.0 geven we invulling aan deze vraagstukken. TLS volgens de UBV TLS[52] afspraak zorgt voor veilig transport van gesloten data. API-aanbieders identificeren clients van API-afnemers op basis van de door de Authorization Server uitgegeven client_id. Deze wordt gekoppeld aan het OIN van de API-afnemer. Authenticatie van de client wordt gedaan op basis van het door de Authorization Server uitgegeven authenticatiemiddel. Het Edukoppeling OAuth-profiel beperkt het aantal authenticatiemiddelen dat een ketensamenwerking mag voorschrijven. De API-aanbieder heeft slechts de keuze uit de door de ketensamenwerking geboden opties.

Het Edukoppeling OAuth-profiel gaat ketensamenwerkingen een duidelijke stap vooruit bieden ten opzichte van eerder manieren voor toegang tot API's. Ketensamenwerkingen gebruiken nog steeds API-keys of een ouder Edukoppeling-profiel met whitelisting op basis van een OIN. Met het Edukoppeling OAuth-profiel maken we het mogelijk om toegang op tokens te baseren. De Authorization Server kan op het niveau van een client (en indirect organisatie-identiteit) bevoegdheden registreren voor een API en onderdelen binnen deze API (op basis van scopes). Een client heeft met het ontvangen access token dat een beperkte geldigheid heeft ook beperkte autorisaties passend bij de context. De Authorization Server zorg ervoor dat een API-aanbieder deze meer fijnmazige bevoegdheden centraal kan beheren.

Tegelijk is die fijnmazigheid niet onbeperkt. Scopes zijn meestal ontworpen als functionele rechten op API-niveau: ze geven aan welke handelingen (bijvoorbeeld lezen/schrijven) een client mag uitvoeren op een bepaalde set endpoints of functionaliteiten. Dat is "fijnmaziger" dan alleen een binaire vorm van toegang, maar het ondersteunt geen beheer van bevoegdheden op gegevensniveau[53].

5.2. Oplossing - wat biedt het profiel?

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

[52] https://www.edustandaard.nl/standaard_afspraken/uniforme-beveiligingsvoorschriften/
[53] Een scope als student.read zegt doorgaans niet welke studentrecords, welke attributen, of welke categorieën gegevens binnen een record toegankelijk zijn

ketensamenwerking beter te kunnen aansluiten bij de vanuit de overheid opgelegde standaarden.

5.2.1. Laag risicoprofiel

Ketensamenwerkingen binnen het onderwijs gebruiken veel API's die een verschillend risicoprofiel hebben. De gesloten data die zij ontsluiten kunnen slechts beperkt gevoelige informatie bevatten en een beperkte impact bij misbruik hebben. Er is een laag risicoprofiel. Voor zulke scenario's is de primaire behoefte dat ketenpartijen 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. Het voordeel is vooral pragmatisch: vrijwel elke Authorisation 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 gesloten data 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[54]. Het biedt hiermee hogere mate van betrouwbaarheid en sluit aan op afspraken binnen het overheidsdomein.

[54] PKIoverheid-certificaten bevatten het Organisatie-identificatienummer (OIN). De CA's die een PKIoverheid certificaat uitgeven verifiëren de identiteit van de organisatie in het handelsregister (Logius | PKIoverheid)

6. Bijlage A: Begrippen

Met opmerkingen [BD8]: Nog nader af te stemmen met ROSA en NORA Begrippenkader. Veel begrippen hier hebben nu een betekenis die opgaat voor de heel specifieke context van deze specificatie. Bijvoorbeeld Verwerking en Data. We streven dan naar het gebruik van een specifieker begrip (dat afleidbaar is van het generiekere begrip) wat niet wegneemt dat we in het document wel de afgekorte versie kunnen gebruiken.

API-aanbieder (ook wel API provider): Ketenpartner die binnen een ketensamenwerking een RESTful API aanbiedt (protected resource binnen een OAuth 2.0 Resource Server). De RESTful API is beveiligd met de OAuth 2.0 client credentials grant.

API-afnemer (ook wel API provider): Ketenpartner die binnen een ketensamenwerking met een systeem (OAuth 2.0 client) een RESTful API afneemt.

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.

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.

Confidential client[55]: Een confidential client is een client die draait in een omgeving waar vertrouwelijke gegevens (waaronder het client secret) veilig bewaard kunnen worden (bijvoorbeeld server-side webapplicaties).

Public client[56]: Een public client is een client die het client secret niet veilig kan beheren, bijvoorbeeld single-page web apps (SPA).

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 ze RESTful omdat ze niet aan alle REST[57] principes[58] hoeven te voldoen.

Vertrouwensanker (trust anchor): het startpunt van vertrouwen (bv. CA-root, pinned key, of whitelist).

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[59]) lijst met identificerende gegevens waar bepaalde bevoegdheden mee samenhangen.

[55] https://datatracker.ietf.org/doc/html/rfc6749#section-2.1
[56] https://datatracker.ietf.org/doc/html/rfc6749#section-2.1
[57] REST - Wikipedia
[58] 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.
[59] In tegenstelling tot een 'blacklist' waar opgenomen identiteiten juist geblokkeerd moeten worden

7. Bijlage C: 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[60] 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

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