

////////////////////////////////////

CLIENT CREDENTIALS VOOR PARTNERS ONDERWIJS & VORMING

24.10.2023

////////////////////////////////////

INHOUD

1	Doel en scope	3
2	Gebruikte technologieën	4
2.1	OAuth 2.0	4
2.2	JWKS	5
2.2.1	JWKS genereren	5
3	Client credentials stroom.....	8
3.1	Identificatie van de client	8
3.1.1	JWKS ter beschikking stellen	8
3.1.2	Een client ID verkrijgen	12
3.2	Een access token aanvragen	13
3.2.1	Een JWT voor authenticatie	13
3.2.2	Verzoek bij de authenticatieserver	15
3.3	Een partner API aanroepen met access token	16
3.4	Overzicht	17
4	Documenthistoriek	18

1 DOEL EN SCOPE

De partner API's van het beleidsdomein Onderwijs & Vorming (O&V) zijn beveiligd. Voordat je gebruik maakt van één van onze partner API's, moet je het OAuth 2.0-protocol implementeren en JWKS sleutelparen genereren. Deze handleiding biedt je de nodige handvaten om deze beveiliging op te zetten.

Partner API's worden enkel voor communicatie tussen verschillende toepassingen aangeboden. De scope van deze client credentials is dus beperkt tot server to server (s2s) communicatie. CORS wordt bijgevolg niet ondersteund, aangezien een partner API niet vanuit een browser opgeroepen wordt.

Om toegang te krijgen tot een partner API moeten volgende stappen doorlopen worden:

1. Identificatie van de client bij O&V
2. Aanvraag van een access token door de client
3. Aanroep van de partner API met het access token

Deze stappen worden in detail besproken in 3 Client credentials stroom.

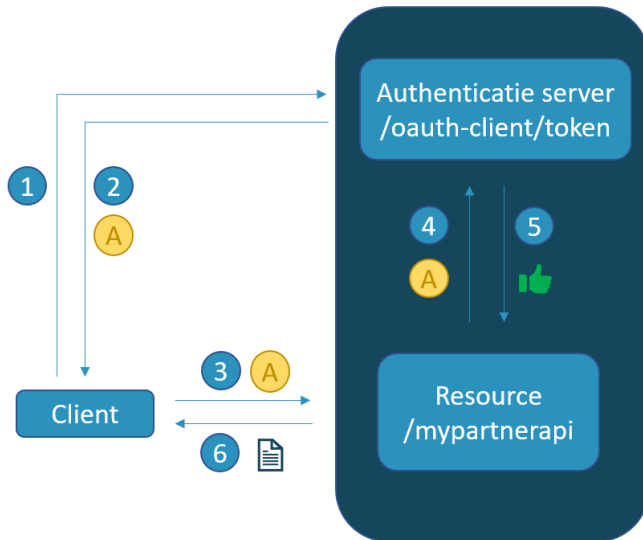
////////////////////////////////////

2 GEBRUIKTE TECHNOLOGIEËN

2.1 OAUTH 2.0

OAuth 2.0 is een beveiligingsprotocol waarmee een persoon zijn toegangsrechten aan een resource kan delegeren. Met dit protocol kan een toepassing namens iemand handelen zonder de geheime informatie van die persoon te moeten verstrekken. Dit protocol wordt gebruikt om het API verkeer tussen toepassingen te beveiligen.

De client haalt een access token op bij een authenticatieserver. Daarmee krijgt ze toegang tot een beschermde resource op naam van de eigenaar van de resource.



Figuur 1 OAuth 2.0 protocol

De **eigenaar van de resource** laat een toepassing in zijn naam handelen. Deze eigenaar kan een persoon zijn, maar kan ook een organisatie, overheidsdienst ... zijn.

De **resource** is iets waarvan we de toegang willen beschermen en waarvoor we kiezen om de toegangsrechten te delegeren. Een resource kan verschillende vormen aannemen, maar over het algemeen is het een API die verschillende toegangsrechten kan hebben (lezen, schrijven...).

Een **token** is een object dat is verkregen bij de authenticatieserver dat toelaat om te bewijzen dat je toegang hebt gekregen tot de resource. OAuth definieert twee verschillende soorten tokens: access tokens en refresh tokens. Voor de beveiliging van de O&V API's beschouwen we enkel het **access token**.

////////////////////////////////////

De **client** is de toepassing die toegang heeft tot de beschermde resource. Deze toepassing kan bijvoorbeeld een applicatie op een server zijn, of een javascript- of native toepassing.

Met de **authenticatieserver** kan een access token afgeleverd worden dat de client kan gebruiken om op te treden in plaats van de eigenaar van de resource. De authenticatieserver zal daartoe de eigenaar authenticeren en verifiëren of die akkoord is.

2.2 JWKS

In de eenvoudigste vorm van OAuth zal een client zich bij de authenticatieserver authenticeren met een client ID (ofwel API key) en een client secret. O&V vereist van haar partners dat zij zich authenticeren met een gehandtekend **JSON Web Token (JWT)** cf. [rfc7523](#).

Om dit JWT te handtekenen, dient de client een **JSON Web Key set (JWKS)** cf. [rfc7517](#) te genereren. Deze JWKS bestaat uit sleutelparen van telkens een private en een publieke sleutel. De client genereert en handtekent een JWT met één van zijn private sleutels uit de vooraf gegenereerde sleutelparen.

2.2.1 JWKS genereren

Er bestaan verschillende tools om JWKS te genereren. In onderstaand voorbeeld wordt de website mkjwk.org ter illustratie gebruikt. Dit volstaat om JWKS voor testdoeleinden aan te maken.

Voor productieve JWKS gebruik je liever JWKS die lokaal gegenereerd worden en niet via een website. Dit kan bv. met de [command line versie van mkjwk](#).

2.2.1.1 JWKS genereren met mkjwk.org

1. Ga naar <https://mkjwk.org>.
 - a. Selecteer RSA
 - b. Key Size: 2048
 - c. Key Use: Signature
 - d. Algorithm: RS256 (RSASSA-PKCS-V1_5 using SHA-256)
 - e. Key ID: SHA-256 (Kies "Specify" indien je zelf Key ID met logische naam aan de sleutel wil geven)
 - f. Show X.509: No

Key Size	Key Use	Algorithm	Key ID	Show X.509	Generate
2048	Signature	RS256 (RSA)	SHA-256	No	

Figuur 2 Gebruikersinterface van mkjwk.org: gegevens voor het genereren van een JWK

Enkel je publieke JWKS bezorg je aan O&V. Deze publieke JWK sets host je op een publiek bereikbare URL. Deze URL kan je zelf hosten, maar het is eveneens toegelaten ze te hosten bij een externe leverancier (bv. Google Cloud Storage, Amazon S3, ...).

Bezorg deze URL('s) met JWKS aan het Onderwijs API Management (onderwijs.api@vlaanderen.be).

3.1.1.2.1 Voorwaarden voor URL

1. De URL is openbaar. Iedereen die over deze URL beschikt kan de publieke JWKS raadplegen. Er wordt dus ook niet aan IP adres-whitelisting gedaan.
2. De URL kan niet eenvoudig geraden worden.
3. Indien de JWKS bedoeld zijn voor het uitwisselen van gevoelige gegevens (minstens informatieklassse 3)¹, moet je zelf eigenaar zijn van de domeinnaam van de URL. Zulke JWKS mag je dus niet hosten op een URL zoals <https://www.dropbox.com/shared/dssdsdds> of <https://opslagruimte.eu-west-1.cloudprovider.com>.

Voorbeeld

De CodeSchool host voor elk van haar twee JWK sets een publieke webpagina. Vermits inschrijvingsgegevens gevoelige informatie bevatten, is de domeinnaam van de URL eigendom van de Codeschool. Ze kiest hierbij *niet* voor een voor de hand liggende naam als <https://inschrijvingen.codeschool.be>.



Figuur 5 JWKS sets gehost op een publieke URL

¹ [Informatieclassificatiemodel Vlaamse Overheid](#)

3.1.1.3 Sleutels in de JWKS vervangen

3.1.1.3.1 Frequentie

De sleutels in de JWKS moeten regelmatig vervangen worden. Hoe vaak je deze sleutels vervangt, hangt af van de informatieklasser² van de gegevens die uitgewisseld worden.

Informatieklasse	Omschrijving	Frequentie sleutels vervangen
	Open data lezen of wegschrijven in een databank van O&V. bv. De codeschool is een school met adres Bondgenotenlaan 1, 3000 Leuven.	Minstens: 1x per jaar
	Niet-gevoelige persoonsgegevens lezen of wegschrijven in een databank van O&V. bv. werkrelatie: Janine Janssens werkt in de Codeschool.	Minstens: 1x per jaar
	Gegevens met beperkte gevoeligheid lezen of wegschrijven in een databank van O&V. bv. privé-adres: Janine Janssens woont op Bondgenotenlaan 1, 3000 Leuven.	Minstens: 1x per jaar Aanbevolen: 1x per maand
	Persoonsgegevens met rijksregisternummer of met informatie over levensbeschouwing, handicap, ziekte, negatief studieresultaat, ... uitwisselen met O&V. bv. geaardheid: Janine Janssens neemt geboorteverlof (als meemoeder).	Minstens 1x per maand Aanbevolen: 1x per dag Maximum: 1x per 60 minuten

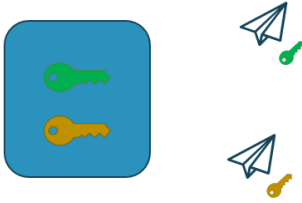
We raden sterk aan om de sleutelgeneratie te automatiseren, bv. met behulp van downloadbare versie van mkjwk. De webpagina met de publieke JWKS wordt idealiter ook automatisch geüpdatet.

3.1.1.3.2 Methode

In dit voorbeeld tonen we hoe je een sleutel vervangt in een JWKS met twee sleutels. Deze methode kan je uitbreiden naar een JWKS met meer sleutels.

1 Beginsituatie

De groene en de oranje private sleutels worden beide gebruikt om communicatie met O&V te handtekenen. De JWKS met groene en oranje sleutel wordt publiek gehost.



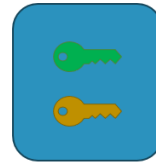
² [Informatieclassificatiemodel Vlaamse Overheid](#)

2 Oude sleutel niet meer gebruiken

De groene private sleutel wordt niet langer gebruikt om te handtekenen. De groene publieke sleutel blijft beschikbaar in de JWKS. Eerder verstuurde communicatie die met groen versleuteld was, kan zo nog steeds gevalideerd worden door O&V.

Minimum duur: 1 uur

Aangeraden duur: 4 uur



3 Nieuwe JWKS ter beschikking stellen

Een nieuwe JWKS wordt gehost. Deze bevat de oranje sleutel die nog steeds in gebruik is en een nieuwe rode sleutel.

Minimum duur: 1 uur



4 Nieuwe sleutel in gebruik nemen

Zodra de rode publieke sleutel enige tijd beschikbaar is, kan de rode private sleutel in gebruik worden genomen om communicatie te handtekenen.



////////////////////////////////////

3.1.2 Een client ID verkrijgen

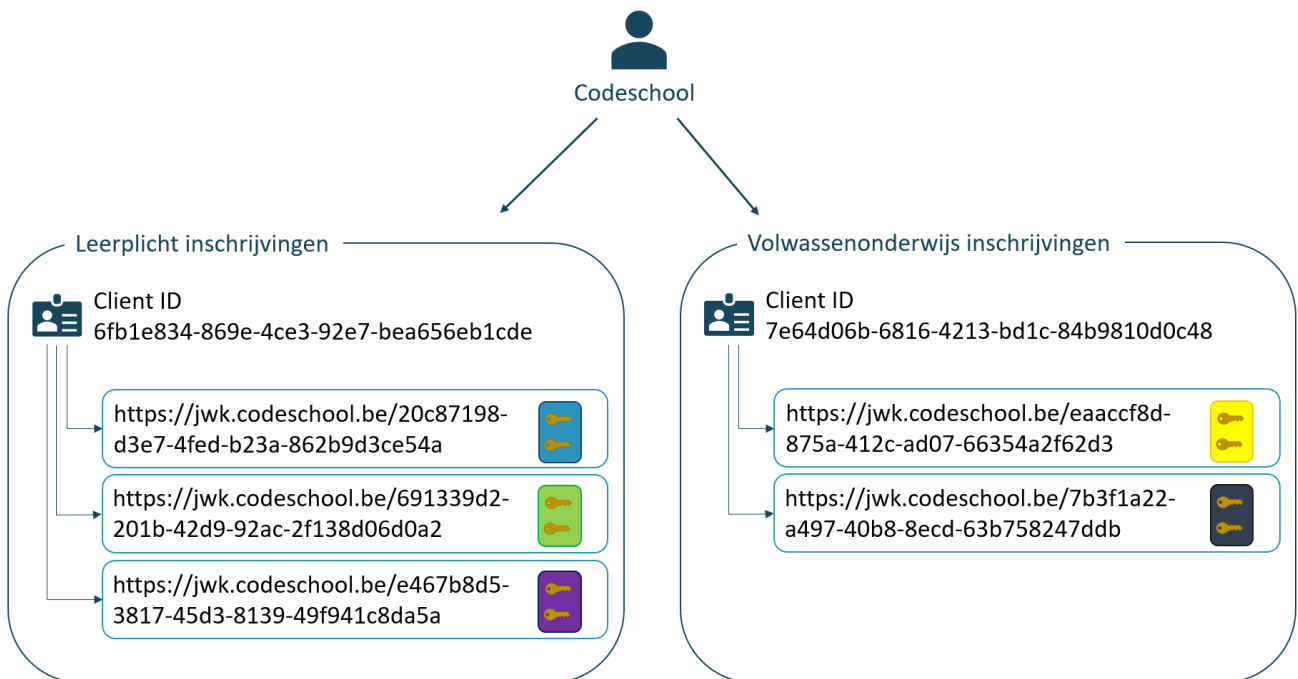
Heb je de URL('s) met publiek JWKS aan het Onderwijs API management bezorgd? Dan ontvang je een client ID. Dit client ID zal je samen met de private sleutels van je JWKS gebruiken om een access token aan te vragen.

Een client ID is geldig over alle omgevingen (acceptatie, stage en productie) heen. Dit in tegenstelling tot de gebruikte sleutels die per omgeving verschillen.

Gebruik je meerdere partner API's? Per partner API zal je dan verschillende JWK sets genereren en een nieuw client ID ontvangen.

Voorbeeld

De Codeschool wil naast de partner API Leerplicht inschrijvingen ook de partner API Volwassenonderwijs inschrijvingen gebruiken. Voor deze tweede partner API gebruikt zij een andere client ID en andere JWK sets.



Figuur 6 Afnemer van partner API met meerdere client ID's en JWK sets

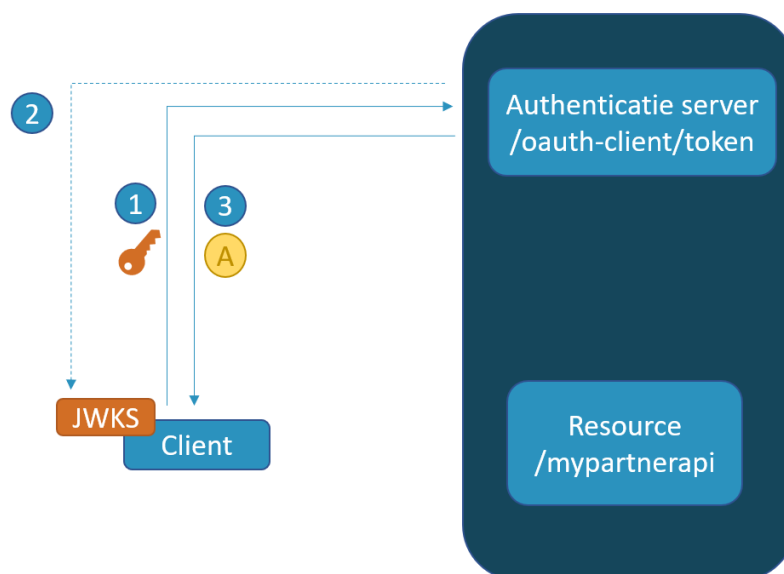
3.2 EEN ACCESS TOKEN AANVRAGEN

1. Een client dient zich te authenticeren bij de authenticatieserver om een access token te bekomen. Hiervoor genereert de client een JWT (JSON Web Token). Deze JWT wordt gehandtekend met de private sleutel van de JWKS. Dit gehandtekend JWT wordt aan de authenticatieserver aangeboden. De URL's voor het tokenverzoek zijn als volgt:

Omgeving	Token endpoint
stage	https://onderwijs-stage.api.vlaanderen.be/oauth-client/token
productie	https://onderwijs.api.vlaanderen.be/oauth-client/token

2. De authenticatieserver haalt de publieke sleutel op van de JWKS URL om het JWT van de client te valideren.
3. Het JWT van de client wordt gevalideerd. Bij een correct inhoud en handtekening wordt een access token geretourneerd. Dit access token blijft 60 minuten geldig.

3.2.1 Een JWT voor authenticatie



Figuur 7 Tokenverzoek bij de authenticatieserver

Een ondertekende JWT die de client binnen de autorisatieserver authenticaceert, zoals beschreven in RFC 7519. Het JWT voor de aanvraag van een Access token moet aan volgende vereisten voldoen.

3.2.1.1 Header

Parameter	Vereist	Omschrijving	Voorbeeld
kid	✓	key id: ID van de sleutel uit de JWKS waarmee de JWT gehandtekend is.	"XcCC1xuBs6iSaaIWpNvhB242m4f8YrNcq d8GH7UZ7E"
alg	✓	algorithm: het algoritme waarmee de JWT versleuteld werd.	"RS256"

3.2.1.2 Payload

Parameter	Vereist	Omschrijving	Voorbeeld
sub	✓	subject: identiteit van de aanroeper, het client_id van de toepassing.	"MyClientId123"
iss	✓	issuer: de client toepassing want deze "reikt" het token zelf uit	"MyClientId123"
aud	✓	audience: doelgroep van het JWT-token (URL is verschillend per omgeving)	"https://onderwijs.api.vlaanderen.be/oauth-client"
iat	✓	issued at: het tijdstip waarop het token is aangemaakt	1608287183
exp	✓	expiry: het tijdstip vanaf wanneer het token niet meer geldig zal zijn. De geldigheidsduur van een JWT-token is kort, typisch 5-tal minuten	1608287483
jti	✓	token id: een uniek nummer voor de aanvraag, om "replay attacks" te voorkomen	"c2da4cef-96ea-4d81-b686-c49007d00645"

3.2.1.3 Opmerkingen bij JWT

- **Geen hergebruik**
Een client wordt geacht om bij het ophalen van een nieuw access token, telkens een nieuw JWT te genereren en te handtekenen. Hierbij geef je dus telkens een nieuwe token id ("jti") en tijdstippen ("iat" en "exp") mee.
- **Beperkt in de tijd**
JWT tokens zijn maximaal een uur geldig. Een token dat vervalt vóór "iat" of dat langer dan 60 min na "iat" geldig blijft, wordt niet aanvaard.

////////////////////////////////////

- **Omgeving-gebonden**

Afhankelijk van de omgeving waarvoor je het access token wilt aanvragen, bevat aud(ience) een andere URL. Deze URL's vind je terug in de tabel op p. 13.

3.2.2 Verzoek bij de authenticatieserver

Een access token-verzoek wordt ingediend bij het token endpoint.

De request moet een HTTP POST request zijn met x-www-form-urlencoded parameters:

Parameter	Vereist	Omschrijving
grant_type	✓	vaste waarde: "client_credentials"
client_assertion_type	✓	vaste waarde: "urn:ietf:params:oauth:client-assertion-type:jwt-bearer"
client_assertion	✓	Een ondertekende JWT die de client authenticceert bij de autorisatieserver, zoals beschreven in rfc7519 . De inhoud van de JWT wordt beschreven in 3.2.1 Een JWT voor authenticatie.
scope		optioneel: lijst van scopes

De request moet bovendien de “X-API-Key” header bevatten met de waarde van de clientid.

3.2.2.1 Voorbeeld request

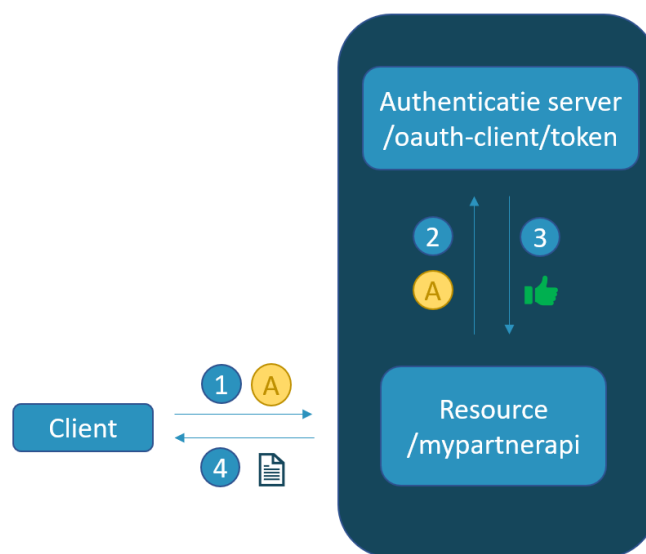
```
POST /oauth-client/token HTTP/1.1
Host: https://onderwijs.api.vlaanderen.be
X-API-Key: {{clientid}}
...
Content-Type: application/x-www-form-urlencoded
grant_type=client_credentials&client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer&client_assertion={{jwt token in base 64}}
```

3.2.2.2 Voorbeeld response

```
HTTP/1.1 200
...
Type: application/json; charset=UTF-8
{
    "access_token": "MyAccessToken",
    "token_type": "Bearer",
    "expires_in": 300
}
```

3.3 EEN PARTNER API AANROEPEN MET ACCESS TOKEN

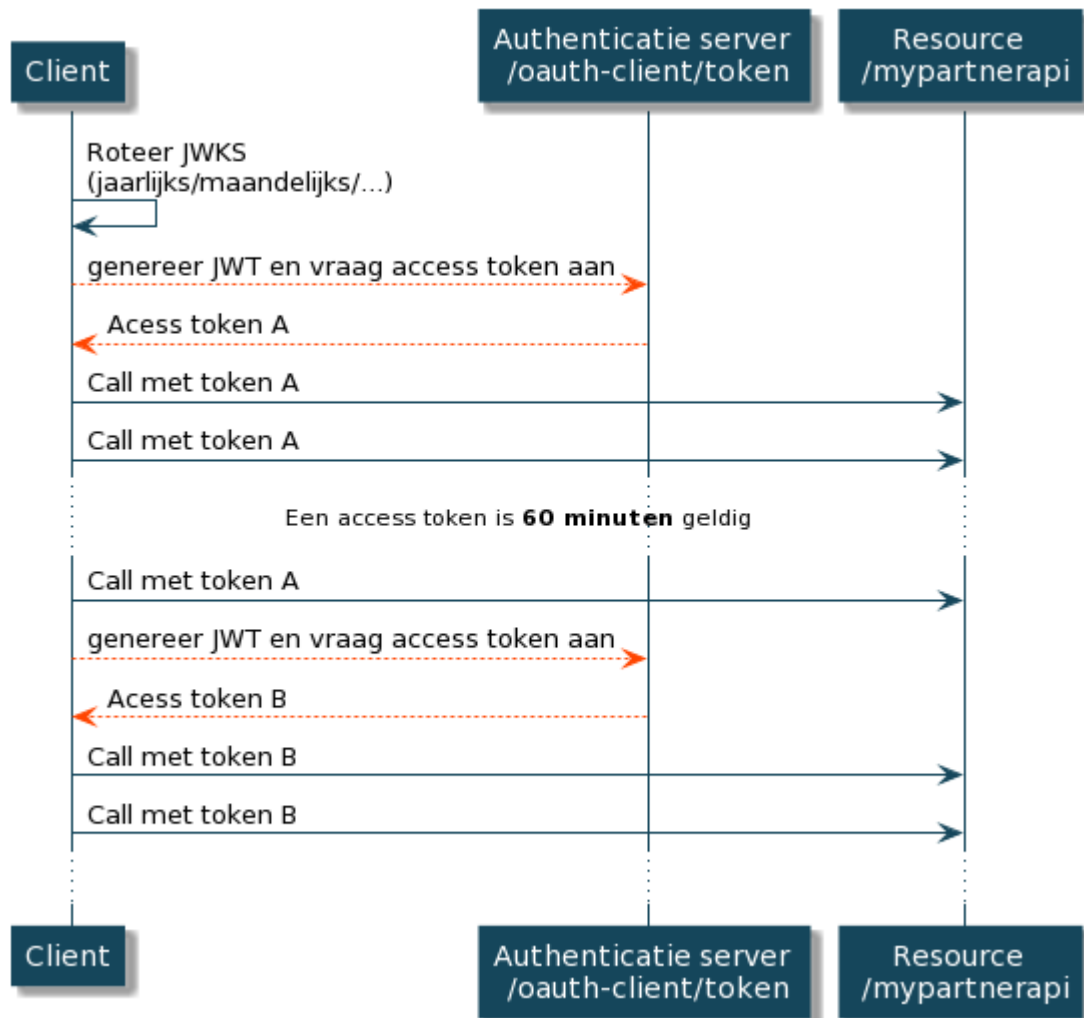
1. Het access token dat je bij de authenticatieserver hebt opgevraagd, gebruik je om de partner API aan te roepen. Dit access token vermeld je samen met het token type (Bearer) in de Authorization header. In de header "X-API-Key" vermeld je de clientid.
2. Het access_token wordt geverifieerd bij de authenticatieserver.
3. Bij een geldig token, krijg je toegang tot de resource. Als het token niet geldig is wordt een fout met de corresponderende foutmelding teruggegeven.



Figuur 8 Gebruik van een access token om een partner API aan te roepen

3.3.1.1 Voorbeeld request

```
POST /my-partner-api HTTP/1.1
Host: https://onderwijs.api.vlaanderen.be
Authorization: Bearer {{access token}}
X-API-Key: {{clientid}}
```



Figuur 9 De volledige Client Credentials stroom.

Als geïdentificeerde gebruiker moet je de sleutels in je JWKS regelmatig vernieuwen. De frequentie van deze sleutelrotatie is afhankelijk van de informatieklasse van gegevens die je uitwisselt (zie 3.1.1.3.1 Frequentie).

Met een actieve sleutel teken je het JWT waarmee je een acces token aanvraagt. Het access token dat je ontvangt blijft 60 minuten geldig. Je kan dus meerdere calls uitvoeren met dit acces token.

Wanneer de geldigheid van het acces token verstreken is, vraag je opnieuw een access token aan.

4 DOCUMENTHISTORIEK

Datum	Auteur	Opmerkingen
2020-12-21	Eline Meeus	Draft version
2020-12-23	Eline Meeus	Opmerkingen Sophie Kemper en Jan Castro verwerkt
2021-01-27	Jan Castro	Aanpassingen ivm audience parameter en endpoint url voor stage
2021-06-11	Eline Meeus	Geldigheid access token verduidelijkt. Overzicht client credentials stroom toegevoegd.
2021-09-07	Jan Castro	Toevoeging extra header "X-API-Key"
2023-10-24	Eline Meeus	Opsplitsing in interne en externe versie