Edukoppeling

Discussiestuk OAuth/Secure API-profiel

Edustandaard

Datum: November 2022

edustandaard

Inhoudsopgave

1. Documenthistorie

Versie	Auteur	Datum	Opmerking
0.1	E. Reinhoud	12 september 2022	Initiële versie
0.2	E. Reinhoud	12 oktober 2022	Verwerking opmerkingen en besluiten 21 september 2022 en nieuwe voorstellen
0.3	E. Reinhoud	November 2022	Verwerking opmerkingen en besluiten 20 oktober 2022 en nieuwe voorstellen

2. Inleiding

2.1. Aanleiding

Edukoppeling is een belangrijke bouwsteen in de ketenreferentiearchitectuur ROSA[1]. Het functionele toepassingsgebied van de Edukoppeling betreft de M2M uitwisseling van vertrouwelijke gegevens via een beveiligde verbinding.

Voor hun koppelingen in de ECK-keten heeft Topicus een OAuth client credentials profiel in gebruik genomen. Dit profiel wordt nu toegepast bij uitwisselingen waarbij Topicus als bronhouder van persoonsgegevens die namens een onderwijsinstelling deze gegevens verstrekt aan een derde partij. In de ECK-keten zijn meerdere partijen die als bronhouder persoonsgegevens kunnen delen met derde partijen en dus is de verwachting dat meerdere partijen een dergelijk OAuth-profiel willen gaan gebruiken. Om die reden is de intentie reeds uitgesproken (en zijn er pilots ingericht) om op een eenduidige manier het OAuth client credentials profiel te implementeren in het SEM Ecosysteem.

Het risico hierbij is dat er bij die inrichting keuzes worden gemaakt die niet matchen met initiatieven die elders in het onderwijs plaatsvinden of plaats gaan vinden op dit gebied. Er is daarom door zowel SEM als door de Architectuurraad van Edustandaard aan de Edukoppeling werkgroep gevraagd om tot een gestandaardiseerd OAuth-profiel te komen die breed inzetbaar is binnen het onderwijs[2].

We pakken de ontwikkeling van het Edukoppeling OAuth-profiel op zonder vooraf al een duidelijke voorkeur te hebben over de nog te maken keuzes. Deze keuzes zijn in principe aan de werkgroep, maar we hebben wel te maken met de kaders van de ROSA. Daarnaast zijn er verschillende ontwikkelingen die het definiëren van een OAuth standaard complex maken. Met dit document hopen we de complexiteit te beperken door overzicht te bieden. In dit document willen we de aandachtspunten kaderen en de besluiten herleidbaar maken. Op basis van deze besluiten zal het uiteindelijke OAuth-profiel opgesteld worden.

2.2. Doel en status van dit document

Dit is versie 0.3 van dit document. Het zal geen onderdeel vormen van de uiteindelijke Edukoppeling afspraak, maar dient alleen voor transparantie van het proces door het inzichtelijk maken van relevante informatie en het traceerbaar maken van de verschillende te maken keuzes.

In deze versie zijn de besproken voorstellen vertaald naar besluiten in de vorm van uitgangspunten die voor het OAuth profiel gaan gelden. Voordat een profiel opgesteld kan worden moet er consensus zijn rond de uitgangspunten. Om welke uitgangspunten het uiteindelijk gaat is onderdeel van de discussie.

---

[1] https://www.wikixl.nl/wiki/rosa/index.php/Hoofdpagina
[2] Partijen buiten het onderwijs kunnen dit ook toepassen. De interoperabiliteit binnen het onderwijs heeft echter prioriteit.

3. Samenvatting bespreking oktober 2022

Op 20 oktober is de 0.2 versie van dit discussiestuk besproken. Hieronder wordt per laag hiervan verslag gegeven. Er zijn opmerkingen gekomen op de uitgangspunten van versie 0.2 en deze zijn in de nieuwe uitgangspunten verwerkt.

3.1. Algemeen

Begrippen
We hebben besloten een aantal begrippen te vervangen, we hebben het vanaf nu over Secure API profielen. We hebben het in de communicatie mogelijk nog wel over SaaS context[3]. We hebben in de huidige situatie de volgende Secure API-profielen:

- Secure API WUS (be, be-S en be-SE) profiel
- Secure API REST profiel
- Secure API OAuth profiel

Bij al deze profielen is in de uitwisseling de routering naar de eindorganisatie geregeld en worden in samenhang met het OSR gebruikt (procesafspraak Edukoppeling Secure API protocol).

We hebben besloten om het begrip mandatering niet door machtiging te vervangen. Digikoppeling heeft een handreiking opgesteld waarin een vergelijkbare procesafspraak (bevoegdheid intermediair/SAAS partij door 'machtigen'[4]) als een mandatering wordt beschreven, maar Digikoppeling gebruikt het begrip machtiging in plaats van een mandatering.

Opbouw Edukoppeling profielen binnen het Informatie-uitwisselingsmodel
De Secure API-profielen bevatten conform de huidige situatie voorschriften voor:

- registratie en verificatie van mandaat (procesafspraak in Secure API protocol);
- routeren o.b.v. een routeringskenmerk van een verwerker naar een bepaalde eindorganisatie (applicatielaag);
- beveiligd (P2P[5]) transport (infrastructuurlaag).

Secure API WUS profielen bieden extra functionaliteit. In de applicatielaag kan integriteit en vertrouwelijkheid op data niveau (WUS be-S en WUS be-SE) toegepast worden. Het nieuwe Secure API OAuth profiel heeft extra functionaliteit in de applicatielaag[6] rond autorisatie, zonder een geldig Access Token in het request is de uitwisseling niet mogelijk.

---

[3] De SaaS-context kan wel helpen als sprekend voorbeeld dat een SaaS-leverancier die als verwerker gegevens deelt in opdracht van een eindorganisatie.
[4] https://github.com/Logius-standaarden/Digikoppeling-Handreiking-Adressering-en-Routering/blob/main/documenten/Analyse_knelpunten_Routering_Intermediairs.md
[5] Het WUS profiel bevat ook voorschriften om berichten te ondertekenen en versleutelen.
[6] Binnen de architectuur is het wellicht wenselijk om met verschillende niveaus voor Secure API profielen te gaan werken.

Een overzicht van de verschillende Secure API profielen wordt bij de nieuwe voorstellen weergegeven in het Informatie-uitwisselingsmodel (zie Figuur 2).

Functioneel toepassingsgebied
In de Edukoppeling Architectuur moet duidelijk staan wat er met een Secure API profiel precies wordt bedoeld. Door het toepassen van een generiek interactiepatroon rond het gebruik van het OSR moet een partij die verwerker en eindorganisatie is toch een mandaat (voor zichzelf) registreren. Hiermee wordt het lastig om bijvoorbeeld nog een OAuth profiel te ontwikkelen die een partij binnen het onderwijs voor beveiliging van eigen API's (en eigen gegevens) kan gebruiken (dus buiten verwerker/eindorganisatie context). In Figuur 2 worden deze potentiële profielen weergegeven in het witte vlak. Om eenduidig te zijn is het wellicht toch beter om dit soort varianten (dus ook de open data variant) geen onderdeel te laten zijn van de Edukoppeling architectuur. Dit moet later nog besloten worden bij discussie rond nieuwe versie van de Architectuur. Wel is het goed om relatie eindorganisatie en verwerker in het functioneel toepassingsgebied te benoemen en dat we dit geborgd hebben met het verplicht gebruik van het OSR.

3.2. Proceslaag

P3: **Het juridisch kader vormt geen onderdeel van het Secure API protocol**
We maken geen onderscheid of het persoonsgegevens of andere vertrouwelijke gegevens betreft en expliciet benoemen van AVG en aansluiten op begrippen is zoals aangegeven niet noodzakelijk of wenselijk. Zodoende is het misschien beter om een ander begrip dan 'verwerker[7]' in Edukoppeling te gaan gebruiken en de definitie aan te scherpen. We hebben echter nog geen goed alternatief.

Binnen Edukoppeling gaat het om vertrouwelijke gegevens en het juridisch kader (AVG) wordt dus niet expliciet binnen het Secure API Protocol opgenomen. De Edukoppeling Secure API profielen voldoen wel aan algemeen geldende beveiligings- en privacy maatregelen die bij het verwerken van persoonsgegevens verwacht kunnen worden. Het Edustandaard Certificeringsschema[8] geeft aan wanneer deze toegepast dienen te worden.

P4: **Het Secure API protocol bevat de voorschriften rond het OSR.**
Het Secure API protocol bevat de procesafspraken rond het OSR (rollen en interacties rond registratie mandaat). Het beheer hiervan ligt in principe bij Kennisnet, maar ook de Edukoppeling werkgroep is hierbij een belangrijke stakeholder. Met het Secure API protocol wordt het OSR onderdeel van de normatieve Edukoppeling afspraken.

[7] Verwerker gedefinieerd in ROSA Begrippenmodel
[8] https://www.edustandaard.nl/standaard_afspraken/certificeringsschema-informatiebeveiliging-en-privacy-rosa/

4. Uitgangspunten

Op basis van de vorige versie en de samenvatting van het overleg van oktober zijn hieronder uitgangspunten geformuleerd. De uiteindelijke lijst met vastgestelde uitgangspunten vormen het kader voor het op te stellen Edukoppeling Secure API OAuth profiel.

1. SaaS-Context en SaaS profielen zijn begrippen die we in de communicatie kunnen blijven gebruiken, maar we veranderen de naam van de profielen in 'Secure API-profielen'. We onderkennen de volgende Secure API profielen[9]:
    a. Secure API WUS(be, be-S & be-SE)-profiel
    b. Secure API REST-profiel
    c. Secure API OAuth-profiel
2. In de Edukoppeling documentatie en die van OSR hebben we het over een mandatering. Digikoppeling heeft het in een vergelijkbare procesafspraak (bevoegdheid intermediair/SAAS partij door 'machtigen') over een machtiging. In Edukoppeling blijven we het begrip mandatering hanteren.
3. De procesafspraken rond het OSR[10] worden onderdeel van de normatieve voorschriften voor Edukoppeling. De procesafspraken worden beschreven in een Secure API Protocol (zie Figuur 1).
    a. Bij een Secure API profiel[11] mandateert een eindorganisatie een verwerker om als onderdeel van een bepaalde ketensamenwerking vertrouwelijke gegevens te laten verwerken[12]. De gemandateerde verwerkers controleren vooraf aan de uitwisseling hun eigen en elkaars mandaat. Bij alle Secure API profielen wordt altijd een mandaat toegepast.
    b. De interacties voor registratie en verificatie van het mandaat en bijbehorende kaders zijn onderdeel van het Edukoppeling Secure API Protocol.
    c. Er wordt een generiek interactiepatroon gehanteerd. Dit betekent dat ook een Agentschap of onderwijsinstelling die zowel de rol van eindorganisatie als verwerker heeft voor zichzelf een mandaat registreert om binnen een bepaalde ketensamenwerking vertrouwelijke gegevens uit te wisselen. Dit dus om partijen binnen de ketensamenwerking in staat te stellen een generiek interactiepatroon rond verificatie in te richten die vooraf aan de uitwisseling wordt uitgevoerd.

Met opmerkingen [BD1]: Check of de definitie in het ROSA Begrippenmodel overeenkomt met die uit de AVG: https://rosa.wikixl.nl/index.php/F9420848-2723-409d-9a76-c21f1543a450 (de bron is hiervoor NORA Begrippenkader)

---

[9] De WUS be-S & be-SE profielen bieden extra functionaliteit (integriteit en integriteit icm vertrouwelijkheid op data niveau) doordat de data in transport ondertekend of ondertekend en versleuteld kan worden. Het OAuth profiel biedt extra beveiliging doordat de autorisatie technisch geborgd is. Bij de uitwerking van de architectuur kunnen wellicht nog verschillende beveiligingsniveaus aan de profielen toekennen.

[10] Nader onderzoek moet nog uitwijzen of alle scenario's door OSR ondersteund kunnen (gaan) worden.

[11] Met Secure API duiden we profielen aan waarbij ook de machtiging aan een verwerker een rol speelt. Als we in de Architectuur meer profielen gaan ondersteunen, bijvoorbeeld API key, dan is het wellicht beter om per profiel met een beveiligingsniveau aanduiding te gaan werken.

[12] Verwerken is hier vergelijkbaar als in AVG gedefinieerd.

   d. Het OSR is naast een mandatenregister ook een service register. Vanuit die functie is het wenselijk (geen onderdeel van de normatieve afspraak) om ook AS metadata op te nemen in het OSR. Voor de Resource Server / Protected Resources ligt het al voor de hand dat dergelijk informatie in het OSR beschikbaar is[13]. Mede omdat OSR ook uitgaat van een ketensamenwerking (afspraak) gaan we er vanuit dat partijen ook via bilaterale kanalen over de benodigde informatie kunnen beschikken.
   e. Het beheer van het Secure API protocol ligt in principe bij Kennisnet, maar Edustandaard en specifiek de Edukoppeling werkgroep is hierbij een belangrijke stakeholder.
4. De Secure API profielen worden toegepast bij de uitwisseling van vertrouwelijke gegevens. Dit kunnen persoonsgegevens, maar ook bedrijfskritische gegevens zijn.
   a. Het gaat om vertrouwelijke gegevens en het juridisch kader (AVG) wordt dus niet expliciet binnen het Secure API Protocol opgenomen. De Edukoppeling Secure API Profielen voldoen wel aan algemeen geldende beveiligings- en privacy maatregelen die bij het verwerken van persoonsgegevens verwacht kunnen worden. Het Edustandaard Certificeringsschema geeft aan wanneer deze toegepast dienen te worden.
   b. We definiëren eigen Edukoppeling begrippen (eindorganisatie en verwerker) en sluiten dus niet direct aan op het juridisch (AVG) kader. Wel is het handig als we voor het begrip ‘verwerker’ nog een ander begrip te gaan gebruiken om verwarring te voorkomen. En moeten wellicht ook de definitie aanscherpen.
5. Het functionele toepassingsgebied van Secure API profielen blijft ongewijzigd.
6. We gebruiken (voorlopig[14]) de internationale open standaard OAuth (RFC6749[15] en RFC6750) als vertrekpunt voor de ontwikkeling van het OAuth profiel (dus niet het iGOV-NL OAuth). Onze use case past het beste bij het OAuth client credentials en dit OAuth profiel zal als basis dienen.
7. Het Secure API OAuth profiel gaat ervan uit dat er meerder lokale Authorization Servers en Resource Servers[16] zijn die door verschillende partijen worden beheerd.
8. In de context van het Secure API OAuth profiel wordt een meer specifieke rolaanduiding wenselijk. In de context van het OAuth profiel kunnen we namelijk spreken van een verwerker die een OAuth AS en RS beheert en een verwerker als client die de protected resource wil verwerken. De relatie tussen bestaande Edukoppeling rollen en nieuwe OAuth rollen wordt weergegeven in Figuur 7.

---

[13] Er vanuit gaande dat dit geen publieke info is, maar alleen toegankelijk voor partijen die als verwerker actief zijn binnen een bepaalde ketensamenwerking.
[14] Het Kennisplatform ontwikkelt nog OAuth profielen. Mochten we op een later moment kunnen aansluiten dan nemen we dat in overweging.
[15] https://datatracker.ietf.org/doc/html/rfc6749 (ook OAuth wordt doorontwikkeld: https://datatracker.ietf.org/doc/draft-ietf-oauth-v2-1/).
[16] We gaan bij deze versie uit van een lokale AS en RS. Er kan op termijn besloten dat er een centrale AS komt (wellicht OSR) voor de beveiliging van API’s bij meerdere Resource Servers van verschillende partijen. Dit zou dan wel meer een OSR implementatietraject zijn en een afsprakenstelsel zijn wat wellicht in het Secure API protocol beschreven kan worden. De interfaces van OSR beschrijven dan in zijn geheel het OAuth profiel.

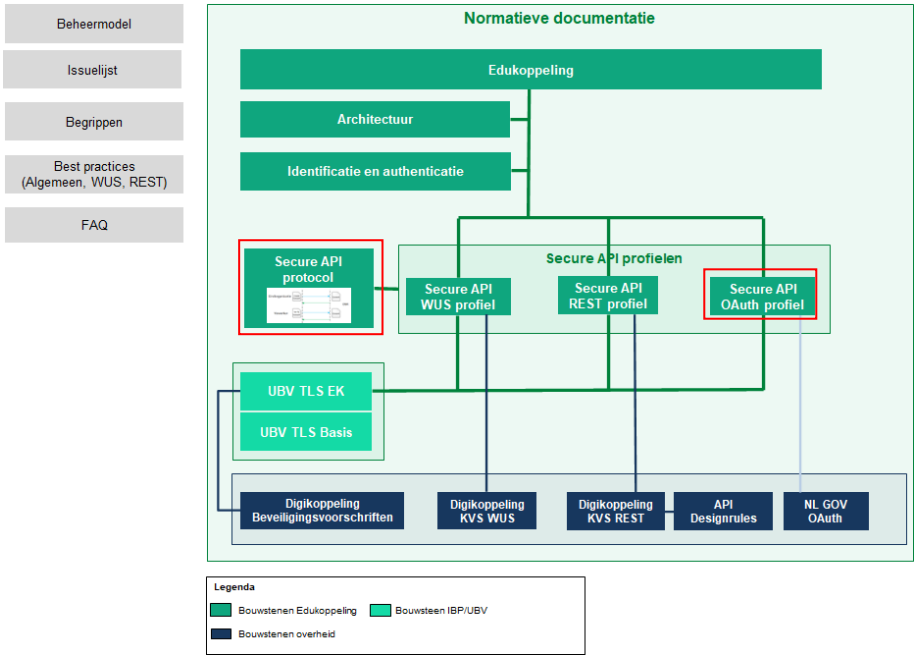
9. De lokale Authorization Servers moeten vooraf aan de uitgifte van een Access Token hebben geverifieerd of de client gemandateerd is.
    a. Hoe de AS de OSR-verificatie van client vertaalt[17] naar het wel of niet uitgeven van een access token valt buiten de Edukoppeling afspraak.
10. Het Secure API OAuth profiel (wijzigingen en/of aanvullingen op de internationale open standaard) wordt in het Engels geformuleerd.
11. Als wijzigingen overeenkomen met die van het iGOV-NL OAuth profiel dan wordt dit expliciet aangegeven met "<iGOV-NL>"[18]. We doen dit voor hele tekstblokken.
12. Het Secure API OAuth profiel gaat uit van het OAuth client credentials profiel[19]. Het gaat om confidential clients die een Access Token krijgen op basis van hun identiteit[20]. De client moet zich bij de AS registreren zodat identificatie mogelijk is om de verificatie van het mandaat bij het OSR uit te voeren en te kunnen bepalen of een Access Token geleverd mag worden.
13. Client identificatie op basis van OIN.
14. Verwerker met de AS/RS combi zijn zelf verantwoordelijk voor het registreren van een client. Hiervoor moet het OIN[21] gebruikt te worden.
15. Een client wordt geauthentiseerd op basis van het UBV TLS Edukoppeling profiel (mTLS). In OAuth (RFC6749) worden een aantal methoden voor client authenticatie[22] onderkend. Het Secure API OAuth vereist client authenticatie op basis van mTLS en PKIo certificaten.
16. Access Tokens worden beveiligd op basis van RFC8705[23]. Met het toepassen van mTLS kunnen we ook de Mutual-TLS Client Authentication and Certificate-Bound Access Tokens (RFC8705) standaard toepassen. We sluiten hiermee ook aan bij nieuwe ontwikkelingen rond OAuth[24]. Een Access Token is in principe een bearer

---

[17] Het OSR registreert een mandaat voor een bepaalde ketensamenwerking. De OAuth AS beveiligd een bepaalde resource. Er is geen 1-op-1 relatie te leggen vanuit de standaard.
[18] Gezien het iGOV-NL profiel gebaseerd is op het iGOV profiel (https://logius-standaarden.github.io/OAuth-NL-profiel/#bib-igov.oauth2) en hier ook teksten van overneemt kan het zijn dat binnen een <iGOV-NL> tag alleen iGOV teksten staan. Voor de leesbaarheid worden er niet geneste (iGOV/iGOV-NL) tags toegepast. Voor de iGOV-NL referenties wordt de volgende in ontwikkeling zijnde draftversie gebruikt: https://logius-standaarden.github.io/OAuth-NL-profiel/
[19] De clients binnen het Edukoppeling Secure API OAuth profiel betreffen alleen confidential clients in de vorm van een web application Zie toelichting bijlage A.
[20] Dus niet op basis van toestemming door een gebruiker (Resource Owner).
[21] Het OIN wordt nu binnen Edukoppeling en het OSR gebruikt. Het kan later blijken dat bij een AS een fijnmazige identificatie van een client nodig is. Zolang het een lokale (decentrale) AS betreft kunnen AS beheerders zou een client die onderdeel is van een bepaalde verwerkersorganisatie (OIN) aanvullend geïdentificeerd kunnen worden op basis van bijvoorbeeld een GUID.
[22] https://datatracker.ietf.org/doc/html/rfc6749#section-2.3
[23] https://datatracker.ietf.org/doc/rfc8705/
[24] Oauth 2.1 (https://datatracker.ietf.org/doc/draft-ietf-oauth-v2-1/)"*It is RECOMMENDED to use asymmetric (public-key based) methods for client authentication such as mTLS [RFC8705] or a JWT [RFC7523].*"

token, het kan worden gebruikt door iedereen die in het bezit is van het token. Met RFC 8705 wordt een bewijs van bezit (proof-of-possession) aan het token gebonden. Het bezit is de asymmetrische sleutel van het mTLS client certificaat. De koppeling van de sleutel aan het token wordt ook doorgegeven aan de beveiligde API (RS). De client kan het bezit aantonen doordat deze beschikt over de private sleutel en dit ook kan aantonen.

17. We adviseren het toepassen van beveiliging best practices, OAuth 2.0 threat model and security considerations [RFC6819]. Ook wordt aanbevolen om kennis te nemen van de OAuth 2.0 security best current practices [OAUTH-SBP] en JSON Web Tokens [JSONWT-BP].



Figuur 1 - Positionering van Secure API protocol en OAuth profiel binnen Edukoppeling

5. Nieuwe overwegingen

In dit hoofdstuk zijn nieuwe overwegingen opgenomen die we gebruiken om in een volgende versie tot uitgangspunten te komen. De overwegingen zijn ingedeeld naar het Edukoppeling informatie-uitwisselingsmodel.

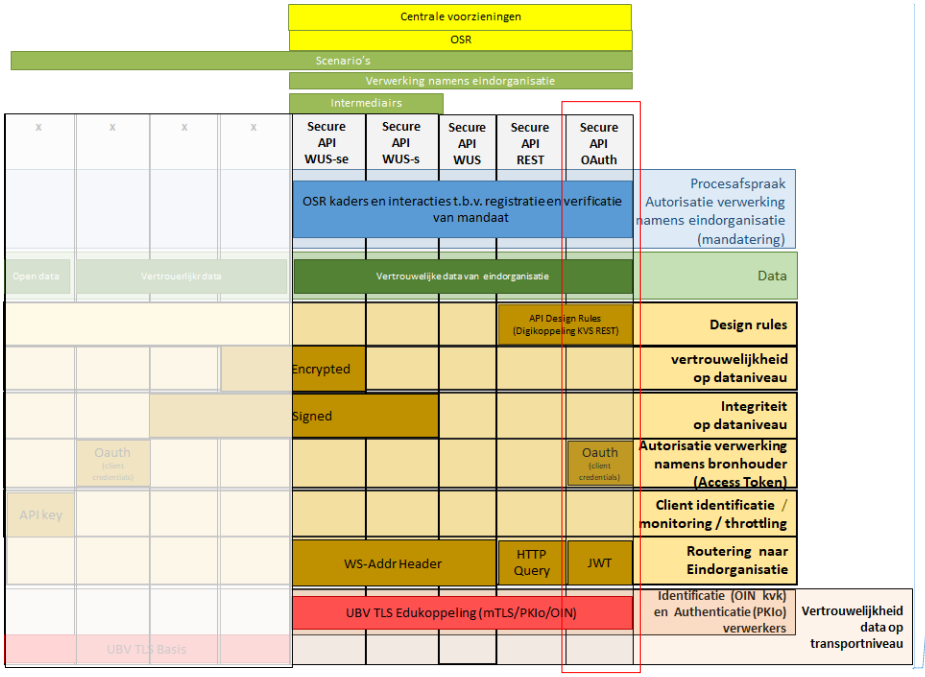
5.1. Algemeen

5.1.1. Edukoppeling Informatie-uitwisselingsmodel

In Figuur 2 wordt naast de opbouw van de bestaande profielen ook die van het Secure API OAuth profiel weergegeven. Hierbij geldt het volgende.

1. Autorisatie voor verwerking vertrouwelijke gegevens namens Eindorganisatie: We gaan uit van Verwerkers die namens Eindorganisaties vertrouwelijke gegevens verwerken. Hier wordt invulling aan gegeven door het Secure API protocol.
2. Design: De API (protected resource) conformeert aan de Design Rules. Hier wordt invulling aan gegeven door het Digikoppeling REST profiel[25].
3. Autorisatie verwerking vertrouwelijke gegevens namens bronhouder: Hier wordt invulling aan gegeven door het Secure API OAuth profiel. Het introduceert nieuwe rollen. De bestaande rol van Verwerker kan een OAuth client of een OAuth AS/RS combinatie zijn in beheer bij een bepaalde organisatie.
4. Routering naar Eindorganisatie: Hier wordt invulling aan gegeven door het Secure API OAuth profiel. Routering is op basis van een JWT token. Dit token is dus een JWT dat specifiek door Edukoppeling gespecificeerd wordt omdat een dergelijke functionaliteit (voor zover bekend) niet door een internationale open standaard ondersteund wordt.
5. Identificatie en authenticatie verwerkers: Invulling wordt gegeven door het UBV TLS Edukoppeling profiel
6. Vertrouwelijkheid data op transportniveau: Invulling wordt gegeven door het UBV TLS Edukoppeling profiel

[25] Het Edukoppeling Secure API REST profiel maakt ook gebruik van het Digikoppeling REST profiel, maar heeft de routering naar de eindorganisatie o.b.v. een query string ingericht.



Figuur 2 - Edukoppeling profielen binnen het Informatie-uitwisselingsmodel

Met opmerkingen [ER2]: We hebben eerder mbt de architectuur besloten dat we binnen EK ook profielen buiten de context van de eindorganisatie-verwerker relatie beschouwen, bijvoorbeeld een Open Data profiel. Het witte vlak geeft zo bijvoorbeeld een API key profiel aan dat bij open data gebruikt kan worden.

Bij de Secure API profielen hanteren we een generiek interactie rond verificatie van mandaat bij vertrouwelijke gegevens. Het is dus de vraag of een extra Oauth profiel zinvol is. Deze vallen eigenlijk al onder de situaties waarbij een eindorganisatie zichzelf al verwerker mandateert.

5.1.2. Aansluiting op de andere Edukoppeling Secure API profielen

De technische specs rond OAuth documenteren we in het Engels, maar we sluiten in de basis aan op de bestaande Nederlandse generieke voorschriften. In de bestaande Edukoppeling Secure API profielen (WUS/REST) hebben we een aantal generieke en specifieke voorschriften opgenomen. Het Secure API OAuth profiel sluit aan op de generieke en specifieke voorschriften van het Secure API REST profiel. De teksten hiervoor worden 1-op-1 overgenomen behalve met betrekking tot de routeringsfunctie. De routeringsfunctie wordt in het Secure API OAuth profiel ingevuld door een JWT. De overige voorschriften hebben o.a. betrekking op het volgende:

1. Het Edukoppeling Secure API OAuth profiel houdt expliciet rekening met gebruik van een openbaar netwerk (Internet).
2. Het Edukoppeling Secure API OAuth profiel stelt eisen aan transportbeveiliging;
3. Het Edukoppeling Secure API OAuth profiel stelt eisen aan identificatie van organisaties;
4. Het Edukoppeling Secure API OAuth profiel kan worden toegepast voor zowel bevragingen als meldingen;
5. Het Edukoppeling Secure API OAuth profiel MOET26 het Onderwijs Service Register gebruiken voor verificatie van de mandatering;

[26] In de huidige versies staat nog 'kan'

6. Het Edukoppeling Secure API OAuth -profiel stelt een aantal generieke eisen aan de foutafhandeling[27];
7. Het Edukoppeling Secure API OAuth profiel stelt aanvullende eisen rond het routeringskenmerk:
    a. MUST: Eindorganisatie routeringskenmerken zijn opgenomen als JWT
    b. MAY: Berichtbeveiligingsvoorschriften[28]
    c. SHOULD: Aansluiten op overheidsbrede afspraken rond REST
        i. Voor Secure API REST is dit een MUST omdat daar voor vrijwel het gehele profiel aangesloten kan worden op het Digikoppeling REST profiel.
        ii. Voor Secure API OAuth profiel is dit een SHOULD omdat we kunnen aansluiten op de api design rules, maar m.b.t. OAuth niet kunnen aansluiten omdat er (nog) geen iGOV-NL client credentials profiel is.

5.1.3. Gebalanceerde mate van specifiëren/voorschrijven

We hebben eerder besloten om niet aan te sluiten op Kennisplatform API's. We specificeren de wat de eisen zijn vanuit de specifieke Edukoppeling context. Met het ontwikkelen van een Edukoppeling Secure API OAuth profiel zijn we op zoek naar de juiste mate van specificeren waarbij we aandacht hebben voor zaken als implementeerbaarheid, standaardisatie, interoperabiliteit, beveiliging en privacy. De vraag is waar ligt de grens van de juiste mate van specificeren.

We zijn het profiel nu stapsgewijs aan het uitwerken. We nemen hierbij dus de internationale standaarden (RFC's) als basis. Eerst hebben we nu de contouren van het profiel binnen de context van Edukoppeling gedefinieerd (zie Figuur 2). De volgende stap is het definiëren van de relevante standaarden in de applicatielaag (RFC's zie Figuur 7). Hiermee kunnen al een groot deel van de doelen behaald worden. Een mogelijke toekomstige stap zou het definiëren van wijzigingen en/of aanvullingen binnen een RFC kunnen zijn (zie een eerste stap in bijlage A). Met name in deze stap moeten we wel aandacht hebben voor de implementeerbaarheid. We moeten besluiten of we deze stap willen maken voordat deze verder wordt uitgewerkt.

Verder zien we dat bij verschillende standaardisatietrajecten[29] een breed perspectief rond API's wordt gehanteerd. Wij richten ons heel specifiek op het OAuth client credentials profiel en de mandatering. We proberen vanuit onze eigen specificatie de relatie met anderen een plek te geven, maar zijn nog op zoek hoe dit het beste ingericht kan worden. Het is wenselijk dat het Edukoppeling Secure API OAuth profiel uiteindelijk duidelijk gepositioneerd moet

---

[27] Het Secure API OAuth profiel heeft ook een aantal specifieke eisen rond foutafhandeling.
[28] Voor REST zijn momenteel nog geen gestandaardiseerde Berichtbeveiligingsvoorschriften.
[29] Zoals iGOV-NL en ISA² IPS (https://ec.europa.eu/digital-building-blocks/wikis/download/attachments/323290750/%28ISA2%29.%28eDelivery%29.%28Piloting%20a%20REST%20API%20extension%20of%20CEF%20eDelivery%29.%28ISA%C2%B2%20IPS%20REST%20API%20Profile%29.%28v1.0%29.pdf?api=v2
) REST API Profile

kunnen worden in de andere initiatieven. Hoe meer we binnen detail zaken specificeren hoe lastiger het wordt om het Edukoppeling Secure API OAuth profiel aan andere initiatieven te relateren.

5.2. Applicatielaag

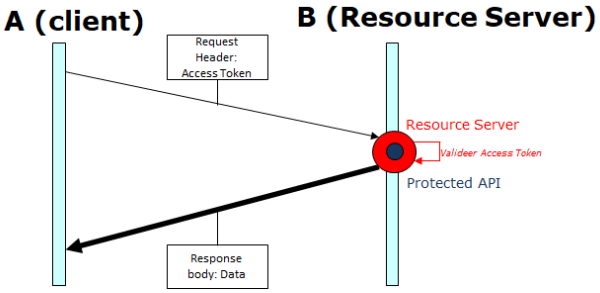
5.2.1. OAuth in de context van bedrijfstransactiepatronen

In de Edukoppeling architectuur onderkennen we een aantal bedrijfstransactiepatronen. Met name bij OAuth is het belangrijk om te duiden waar deze standaard relevant is. Oauth onderscheidt zich namelijk van de andere profielen doordat er ook sprake is van een autorisatie op basis van een Access Token. Voor de routeringsfunctie in de profielen achten we het (nog) niet relevant om de bedrijfstransactiepatronen apart te beschouwen omdat deze functie al sinds het ontstaan van Edukoppeling bestaat en hierover geen onduidelijkheid lijkt te bestaan.

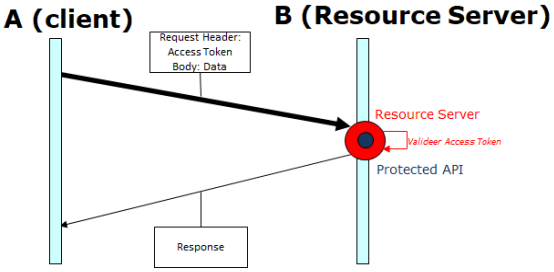
De hieronder geschetste interacties hebben dus betrekking op de autorisatie op basis van een Access Token. Er gelden de volgende uitgangspunten:

1. De client beschikt al over een valide Access Token (happy flow). Het ophalen van een Access Token bij het AS token endpoint is dus niet opgenomen in de transactiepatronen.
2. De Protected API (resource / collectie) is de bron[30] die door de Resource Server wordt beveiligd (controleert Access Token).
3. De client voert een bepaalde bewerking uit op de bron (HTTP verb/methode voor verwerking resource / collectie).
4. Een bewerking wordt beperkt door de rechten in het Access Token (scopes[31]).
5. Notificaties bevatten metadata en geen vertrouwelijke gegevens (een notificatie behoeft geen Access Token). Notificaties kunnen wel referenties bevatten naar vertrouwelijke gegevens, bijvoorbeeld een identifier. Het moet voorkomen worden dat deze identifier als privacy gevoelige informatie kan worden beschouwd.
6. Voor het ontwerp van de protected API gelden de Design Rules van het Digikoppeling REST profiel[32].

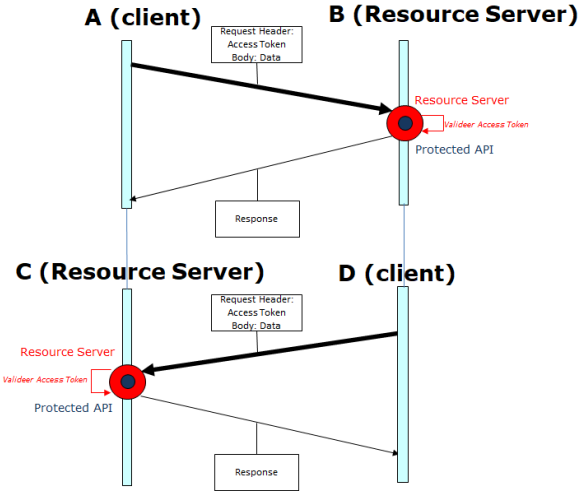
[30] We spreken van een bronhouder als deze binnen de ketensamenwerking verantwoordelijk is voor de levenscyclus van de bron (resource).
[31] Deze worden eventueel verder uitgewerkt in een volgende iteratie van het OAuth profiel.
[32] Zie ook Kennisplatform API's principes waar het Digikoppeling REST profiel naar verwijst.



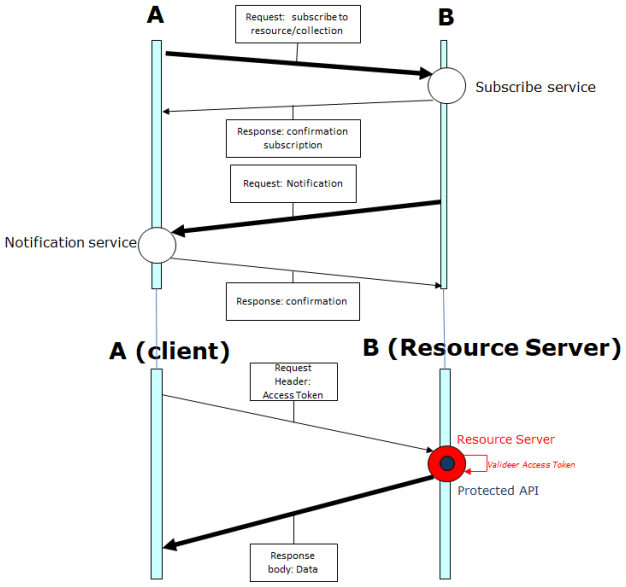
Figuur 3 - Patroon request-response (http GET)



Figuur 4 - Patroon melding- bevestiging (http POST/PUT/DELETE/…)



Figuur 5- Patroon Asynchrone uitwisseling



Figuur 6 - Patroon Abonneren op wijzigingen

5.2.2. Specificatie op het niveau van standaarden (RFC's)

Nu het Secure API OAuth profiel in de Edukoppeling context geduid is kunnen we de applicatielaag verder specificeren. De eerste stap is het duiden van de OAuth rollen in de context van Edukoppeling. Dit wordt in Figuur 7 schematisch weergegeven. Hiermee wordt ook de relatie tussen het Secure API OAuth profiel en het Secure API protocol verduidelijkt, het mandaat in het OSR heeft een relatie met de policy van de Autorization Server, maar is in de kern ontkoppelt. Daarnaast worden ook de relevante standaarden (RFC's) weer gegeven.

Er gelden de volgende uitgangspunten:

1. Een client die een push (bijv. POST, PUT of DELETE) naar de gehele bron (resource collectie) uitvoert kan als een anti-patroon beschouwd worden, want in dat geval zou de client als bron kunnen worden beschouwd (zie ook interactiepatronen). Vanuit Edukoppeling wordt hierin echter niet beperkt, mits een client een geldig Access Token hiervoor heeft. Verder moet de client hebben vastgesteld dat de protected API deze gegevens ook mag verwerken. Die autorisatie is op het niveau van de mandaat verificatie zoals in het Secure API protocol is gedefinieerd.
2. De verwerker die beheer voert over de Authorization Server moet vastgesteld hebben dat de client (verwerker) een valide mandaat heeft voordat de Authorization Server deze een Access Token levert (borging binnen AS policy).Hoe een Authoirzation Server exact tot het besluit komt om wel of geen Access Token te leveren is buiten scope van het Edukoppeling Secure API OAuth profiel.

a. Partijen richten dit op hun eigen manier in en hebben zelf de verantwoordelijkheid om de vertaling te maken van een mandaat voor een verwerker binnen een bepaalde ketensamenwerking en een client die voor een bepaalde protected resource een access token vraagt.
b. Een Access Token kent een bepaalde geldigheidsduur. Deze moet nog in het Secure API OAuth profiel worden gedefinieerd. Deze heeft een relatie met de geldigheid van een mandaat. De geldigheid van een mandaat is echter niet procedureel of technisch geborgd. De verificatie van het mandaat moet dus door beide verwerkers periodiek uitgevoerd worden (er is geen abonnement service). Zie voor verdere details rond het mandaat het Secure API protocol.

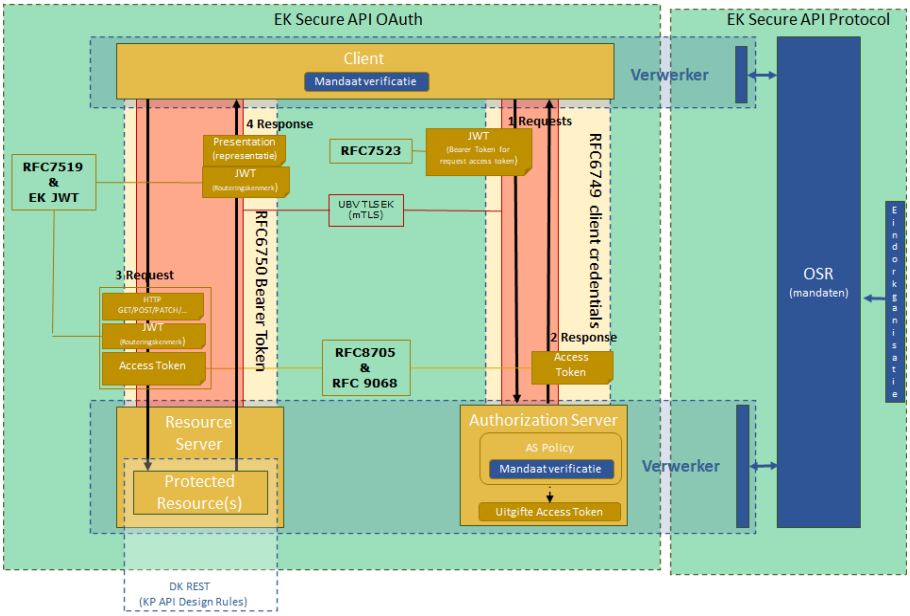
Interacties binnen het OAuth client credentials profiel:

I. Interactie client met Authorization Server token end point (RFC6749)
   1. Requests to the Token Endpoint
   2. Response from the Token Endpoint

II. Interactie client met protected resource (RFC6750)
   3. Request to protected resource (AT & EK Routeringskenmerk o.b.v. JWT)
   4. Response from protected resource

Deze interacties en kaders voor de rollen zijn verder uitgewerkt in het concept OAuth profiel (zie bijlage A).

Met opmerkingen [ER3]: Zoals eerder aangegeven is het de vraag of we verdere verdieping wenselijk achten. Het geeft binnen de RFC specifiek aan waar impact is. Het alternatief is om op het niveau van de RFC's te stoppen. Op dit niveau is ook de verificatie van een mandaat via het OSR al geborgd.



Figuur 7 – Overzicht van het Edukoppeling Secure API OAuth profiel

3. Bijlage A: Edukoppeling Secure API OAuth profiel (client credentials)

Deze bijlage vormt een voorlopige basis voor het Secure API OAuth profiel om bespreekbaar te maken of dit niveau van detaillering wenselijk en werkbaar is. Deze is zoals besloten geheel in het Engels. De indeling is overgenomen uit de RFC's. Over het algemeen zijn de originele teksten niet overgenomen en geven we enkel puntsgewijs een aantal Edukoppeling voorschriften aan. Een enkele keer is de originele tekst (cursief) overgenomen ter verduidelijking van de context, of doorgestreept om non-conformance te benadrukken. Daar waar hiervan sprake is wordt het begin aangegeven met "<EK>" en afgesloten met "</EK>".

Verder kijken we ook waar we nog kunnen aansluiten op iGOV-NL. Waar we dezelfde keuze maken als iGOV-NL wordt het begin van het tekstblok aangegeven met "<iGOV-NL>" en afgesloten met "</iGOV-NL>".

We houden het zo (hopelijk) overzichtelijk.

Requirements Notation and Conventions
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in **BCP 14 [RFC2119] [RFC8174**] when, and only when, they appear in all capitals, as shown here.

Terminology
The Oauth and JWT specifications use the terms "Access Token", "Authorization Code", "Authorization Endpoint", "Authorization Grant", "Authorization Server", "Client", "Client Authentication", "Client Identifier", "Protected Resource", "Resource Server", "Response Type", and "Token Endpoint" defined by [OAuth 2.0] [**RFC6749**] , the terms "Claim Name", "Claim Value", and "JSON Web Token (JWT)" defined by [JSON Web Token (JWT)] [**RFC7519**]. The Edukoppeling specification uses the terms "Verwerker", "Eindorganisatie", "Mandaat", "Routeringkenmerk", "OIN", "BRIN", "kvk-nummer".

Conformance
When an compliant component is interacting with other compliant components, in any valid combination, all components MUST fully conform to the features and requirements of this specification. All interaction with non compliant components is outside the scope of this specification.

A compliant OAuth 2.0 Authorization Server (AS) MUST support all features as described in this specification. A general-purpose authorization server MAY support additional features for use with non compliant clients and protected resources.

An compliant OAuth 2.0 client MUST use all functions as described in this specification. A general-purpose client library MAY support additional features for use with non compliant authorization servers and protected resources.

An compliant OAuth 2.0 Resource Server (RS) MUST use all functions as described in this specification. A general-purpose resource server MAY support additional features for use with non compliant authorization servers and clients.

3.1. Introduction

The baseline for this specification is defined by RFC6749[33] client credentials profile, RFC6750 and additional RFC's.

This specification is especially designed for the Edukoppeling use case where an Edukoppeling Verwerker as a client interacts (on behalf of an Edukoppeling Eindorganisatie) with a protected resources (under the control of another Edukoppeling Verwerker on behalf of an Edukoppeling Eindorganisatie).

3.1.1. Roles

<EK>

OAuth [RFC6749] defines four roles but for the client credentials profile the resource owner is not relevant. Hence the interactions with this role are not documented in this specification.

- ~~*resource owner*~~
  ~~*An entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end-user.*~~

Met opmerkingen [ER4]: Edukoppeling Oauth context: For the client credentials profile de resource owner is not in scope, there are only 3 roles relevant in the interactions.

</EK>

- *resource server*
  The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens.

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

- *authorization server*
  The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization.

3.1.2. Protocol Flow

This specification features additions for interaction between the following components (see Figuur 7).

---

[33] The international open standard RFC6749 is chosen as baseline since iGOV-NL (iGOV-NL is based on the iGOV.OAuth2 profile) is primarily focused on web-facing clients (code grand profile) and has yet to define a client credentials profile.

- Client to authorization server (1-2 RFC6749).
- Client to resource server (3-4 RFC6750).

3.1.3. Authorization Grant

- The grant type MUST be client credentials.
  a. Requests to the Token Endpoint (1-2): grant_type used by the client must be "client_credentials"

3.1.4. Client Credentials

<EK>
The client is Authorized (given an Acccess Token) by the AS if a valid 'mandaat' is present in het OSR that the Eindorganisatie registered for a particular Ketensamenwerking. See Secure API Protocol.

The client credentials (or other forms of client authentication) can be used as an authorization grant when the authorization scope is limited ~~to the protected resources under the control of the client, or~~ to protected resources previously arranged with the authorization server. Client credentials are used as an authorization grant ~~typically~~ when ~~the client is acting on its own behalf (the client is also the resource owner) or is~~ requesting access to protected resources based on an authorization previously arranged with the authorization server.
</EK>

3.1.5. Access Token

<EK>
~~Access token attributes and the methods used to access protected resources are beyond the scope of this specification and are defined by companion specifications such as [RFC6750].~~
</EK>

- Acces Tokens MUST comply with RFC9068.
  - The Access Token is not opaque. They carry all the information the RS needs to authorize without the need to check with the AS.
  - JWT access tokens MUST be signed. Authorization servers and resource servers conforming to this specification MUST include RS256 (as defined in [RFC7518]) among their supported signature algorithms.
  - This RFC is in part dependent on RFC6750 (resource server MUST handle errors as described in Section 3.1 of RFC6750).
- Acces Tokens scopes MUST comply with the scope notation convention of this specification (XXX).
- Acces Token "exp" claim MUST not exceed XXX from current NumericDate value.
  - Implementers MAY provide a few minutes leeway, to account for clock skew. However it is assumed that all systems in the ketensamenwerking follow the same clock synchronization scheme.

Met opmerkingen [ER5]: In deze bijlage wordt een voorzet gegeven voor een eigen EK Oauth profiel. Mochten we op deze manier verder specificeren (en niet enkel op niveau van RFC's of verwijzen naar externe specs) dan is een volgende stap oa verder specificeren signing specs. Dit raakt ook UBV specs
Rationale? Implies use of OIDC signing / or ISA IPS specs (For both payload-level and message-level signatures, the profile enforces the use of JWS detached signatures following the HttpHeaders Mechanism of the ETSI ESI JAdES specification [ETSI-JADES].)

3.1.6. Refresh Token

Refresh tokens are credentials used to obtain access tokens. Refresh tokens are issued to the client by the authorization server and are used to obtain a new access token when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope (access tokens may have a shorter lifetime and fewer permissions than authorized by the resource owner).

- The AS MUST NOT issue a refresh tokens.

3.1.7. TLS Version

Whenever Transport Layer Security (TLS) is used by this specification, the appropriate version (or versions) of TLS will vary over time, based on the widespread deployment and known security vulnerabilities. At the time of this writing, TLS version 1.2 [RFC5246] is the most recent version, but has a very limited deployment base and might not be readily available for implementation. TLS version 1.0 [RFC2246] is the most widely deployed version and will provide the broadest interoperability. Implementations MAY also support additional transport-layer security mechanisms that meet their security requirements.

- All conforming implementations MUST comply to UBV TLS Edukoppeling profile.
- All servers MUST conform to applicable recommendations found in the Security Considerations sections of [rfc6749] and those found in the "OAuth Threat Model Document" [rfc6819].

3.1.8. HTTP Redirections

This specification makes extensive use of HTTP redirections, in which the client or the authorization server directs the resource owner's user-agent to another destination. While the examples in this specification show the use of the HTTP 302 status code, any other method available via the user-agent to accomplish this redirection is allowed and is considered to be an implementation detail.

- All conforming implementations MUST NOT use HTTP Redirections.
    - This is not required by the client credentials profile

3.1.9. Interoperability

OAuth 2.0 provides a rich authorization framework with well-defined security properties. However, as a rich and highly extensible framework with many optional components, on its own, this specification is likely to produce a wide range of non-interoperable implementations.

In addition, this specification leaves a few required components partially or fully undefined (e.g., client registration, authorization server capabilities, endpoint discovery). Without these components, clients must be manually and specifically configured against a specific authorization server and resource server in order to interoperate.

This framework was designed with the clear expectation that future work will define prescriptive profiles and extensions necessary to achieve full web-scale interoperability.

3.2. Client Profile

3.2.1. Client Registration

<iGOV-NL>

- All clients MUST register with the authorization server.
- Client registration ~~MAY~~ MUST be completed by ~~either~~ static configuration (out-of-band, through an administrator, etc...) ~~or dynamically~~.
  - Registration is part of a certain Ketensamenwerking
  - Dynamic registration and discovery is not a required feature

</iGOV-NL>

3.2.2. Client Types

The OAuth specification has been designed around the following client types:

- *Confidential*
  Clients capable of maintaining the confidentiality of their credentials (e.g., client implemented on a secure server with restricted access to the client credentials), or capable of secure client authentication using other means.

- *Public*
  Clients incapable of maintaining the confidentiality of their credentials (e.g., clients executing on the device used by the resource owner, such as an installed native application or a web browser-based application), and incapable of secure client authentication via any other means.

The OAuth specification has been designed around the following client profiles:

- web application
  A web application is a confidential client running on a web server. Resource owners access the client via an HTML user interface rendered in a user-agent on the device used by the resource owner. The client credentials as well as any access token issued to the client are stored on the web server and are not exposed to or accessible by the resource owner.

- *user-agent-based application*
  A user-agent-based application is a public client in which the client code is downloaded from a web server and executes within a user-agent (e.g., web browser) on the device used by the resource owner. Protocol data and credentials are easily accessible (and often visible) to the resource owner. Since such applications reside within the user-agent, they can make seamless use of the user-agent capabilities when requesting authorization.

- *native application*
  A native application is a public client installed and executed on the device used by the resource owner. Protocol data and credentials are accessible to the resource owner. It is assumed

that any client authentication credentials included in the application can be extracted. On the other hand, dynamically issued credentials such as access tokens or refresh tokens can receive an acceptable level of protection. At a minimum, these credentials are protected from hostile servers with which the application may interact. On some platforms, these credentials might be protected from other applications residing on the same device.

- A client in the context of this Edukoppeling specification MUST be a web application, i.e. a confidential client running on a server.

3.2.3. Client Authentication

If the client type is confidential, the client and authorization server establish a client authentication method suitable for the security requirements of the authorization server. The authorization server MAY accept any form of client authentication meeting its security requirements.

Confidential clients are typically issued (or establish) a set of client credentials used for authenticating with the authorization server (e.g., password, public/private key pair).

The client MUST NOT use more than one authentication method in each request.

- *Clients MUST be authenticated through mTLS as specified by the UBV TLS Edukoppeling profile*

3.2.1. Connection to the Authorization Server (1-2)

OAuth Protocol Endpoints

The authorization process utilizes two authorization server endpoints (HTTP resources):

- ~~Authorization endpoint - used by the client to obtain authorization from the resource owner via user-agent redirection.~~
  - Not relevant for client credentials profile

- *Token endpoint - used by the client to exchange an authorization grant for an access token, typically with client authentication.*

- Client (confidential type) MUST NOT request a refresh token and the AS MUST NOT issue a refresh token.
- The issued Access Token MUST comply with RFC9068.
- The authorization server MUST support client_credentials grant.

3.2.2. Connection to the Resource Server (3-4)

- The contents and protocol of the Resource Request and Resource Response are out of scope of this profile.
- The Request MUST contain an Access Token as specified by RFC9068

- Clients MUST send a EK Routeringkenmerk JWT (spec XXX) with the OIN of the intended Eindorganisatie
- Resource Server MUST send a EK Routeringkenmerk JWT (spec XXX) with the OIN of the intended Eindorganisatie

Met opmerkingen [ER6]: We gebruiken een JWT ipv query string voor het routeringskenmerk. Op deze manier kan het routeringskenmerk in request en response opgenomen worden. Dit heeft dus wel een grotere impact op de implementatie.

3.3. Authorization Server Profile

- The AS MAY implement the OAuth 2.0 Token Introspection [RFC7662] interface. This is not required by this specification as the AT is not opaque to the RS.
    - If Token Introspection is used then the response token of the Token Introspection interface SHOULD conform to the JWT Response for OAuth Token Introspection [DRAFTIETF-OAUTH-JWT-INTROSPECTION].
- The authorization server MUST protect all communications to and from its OAuth endpoints using UBV TLS Edukoppeling profile.

3.4. Resource Server Profile

- The contents and protocol of the Resource Response is out of scope of this profile.
- The Resource Server MUST be able to process an Access Token as specified by RFC9068.

4. Bijlage B: Normatieve referenties en best practices

4.1. Normatief

[RFC9068]

JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens, V. Bertocci, 2021-10-21 , Proposed Standard URL https://datatracker.ietf.org/doc/rfc9068/
In-market use has shown that many commercial OAuth 2.0 implementations elected to issue access tokens using a format that can be parsed and validated by resource servers directly, without further authorization server involvement. The approach is particularly common in topologies where the authorization server and resource server are not co-located, are not run by the same entity, or are otherwise separated by some boundary. At the time of writing, many commercial implementations leverage the JSON Web Tokens (JWT) ***[RFC7519]*** *format.*

This specification aims to provide a standardized and interoperable profile as an alternative to the proprietary JWT access token layouts going forward. Besides defining a common set of mandatory and optional claims, the profile provides clear indications on how authorization request parameters determine the content of the issued JWT access token, how an authorization server can publish metadata relevant to the JWT access tokens it issues, and how a resource server should validate incoming JWT access tokens.

*Please note: although both this document and [**RFC7523**] use JSON Web Tokens in the context of the OAuth2 framework, the two specifications differ in both intent and mechanics. Whereas [**RFC7523**] defines how a JWT Bearer Token can be used to request an access token, this documents describes how to encode access tokens in JWT format.*

Rationale: We stellen voor deze draft op te nemen. De toepasbaarheid moet wel vooraf aan vaststelling bij implementaties getoetst worden.

Met opmerkingen [ER7]: Sinds het profiel is opgesteld, heeft IETF de RFC 9068 uitgebracht. RFC9068 beschrijft een standaard JWT formaat voor access tokens. Deze RFC meenemen in het profiel kan overwogen worden voor interoperabiliteit.
Concreet wordt met name sectie 3.2.1 van het profiel geraakt. Om dit in lijn met sectie 2.2 van RFC9068 te brengen, zullen de claims 'iat' en 'azp' als verplicht moeten worden toegevoegd. Verder moet de claim 'sub' ook verplicht worden gesteld, dit is lijn met issue #7.
Daarnaast is de claim 'client_id' toegevoegd, die zou overeenkomen met 'azp'. Hoe hiermee om te gaan zal nader uitgewerkt moeten worden.
Daarnaast is er ook een wijziging in the JWT header. De JWT header hoort 'typ' (content-type) 'at+jwt' te krijgen, conform RFC9068 sectie 2.1. Ook hiervoor zal nader uitgewerkt moeten worden hoe hiermee om te gaan.
Bovenstaande is geen volledige analyse, er kan dus nog verdere impact zijn.

<iGov-NL/>

[RFC7518]
JSON Web Algorithms (JWA), **RS256**

[RFC6749]
The OAuth 2.0 Authorization Framework. D. Hardt, Ed.. IETF. October 2012. Proposed Standard. URL: https://datatracker.ietf.org/doc/html/rfc6749

[RFC6750]
The OAuth 2.0 Authorization Framework: Bearer Token Usage. M. Jones; D. Hardt. IETF. October 2012. Proposed Standard. URL: *https://www.rfc-editor.org/rfc/rfc6750*

[RFC6819]
OAuth 2.0 Threat Model and Security Considerations. T. Lodderstedt, Ed.; M. McGloin; P. Hunt. IETF. January 2013. Informational.
URL: https://datatracker.ietf.org/doc/html/rfc6819

[RFC8705]
OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens Campbell, B., Bradley, J., Sakimura, N., and T. Lodderstedt, "
url: https://www.rfc-editor.org/rfc/rfc8705 Proposed Standard (February 2020). URL:
RFC 8705 - OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens (ietf.org)

[RFC2119]
Key words for use in RFCs to Indicate Requirement Levels. S. Bradner. IETF. March 1997. Best Current Practice. URL: https://www.rfc-editor.org/rfc/rfc2119

[RFC4122]
A Universally Unique IDentifier (UUID) URN Namespace. *P. Leach; M. Mealling; R. Salz. IETF. July 2005. Proposed Standard. URL:* https://www.rfc-editor.org/rfc/rfc4122

[RFC7009]
OAuth 2.0 Token Revocation. T. Lodderstedt, Ed.; S. Dronia; M. Scurtescu. IETF. August 2013. Proposed Standard. URL: https://www.rfc-editor.org/rfc/rfc7009

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

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

[RFC7662]
OAuth 2.0 Token Introspection. J. Richer, Ed.. IETF. October 2015. Proposed Standard. URL: https://www.rfc-editor.org/rfc/rfc7662

[RFC7800]
Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs). M. Jones; J. Bradley; H. Tschofenig. IETF. April 2016. Proposed Standard. URL: https://www.rfc-editor.org/rfc/rfc7800

[RFC8414]
OAuth 2.0 Authorization Server Metadata. M. Jones; N. Sakimura; J. Bradley. IETF. June 2018. Proposed Standard. URL: https://www.rfc-editor.org/rfc/rfc8414

Gewijzigde veldcode

[JWS.JWE.Algs] (UBV? Zelfde discussie als WUS-S en WUS-SE)

IANA JSON Web Signatures and Encryption Algorithms registry. Jim Schaad, Jeff Hodges, Joe Hildebrand, Sean Turner. IANA. URL: https://www.iana.org/assignments/jose/jose.xhtml#web-signature-encryption-algorithms

~~[BCP195]~~

Rationale: We gebruiken geen elementen uit BCP195 TLS. Alles t.a.v. TLS wordt door Edukoppeling UBV TLS gespecificeerd.

~~[HEART.OAuth2]~~

Unlike the core OAuth protocol, the HEART profile intends to allow compliant protected resources to connect to compliant authorization servers.
Rationale: We gebruiken geen elementen uit het Health Relationship Trust Profile for OAuth 2.0. Deze afspraken dienen als basis voor het iGOV profiel waar iGOV-NL gebruik van maakt. Er worden in iGOV-NL dus (deels) normatieve standaarden expliciet benoemd die al impliciet van toepassing zijn.

~~[iGOV.OAuth2]~~

Rationale: We gebruiken geen elementen uit iGov. De werkgroep heeft besloten om de interantionale open OAuth standaard als basis te nemen. We kijken wel naar iGOV-NL die het iGOV profiel wel als vertrekpunt gebruikt**.**

~~[OpenID.Core]~~

Rationale: We gebruiken geen elementen uit OpenID Connect Core 1.0.

~~[OpenID.Discovery]~~

Rationale: We gebruiken geen elementen uit OpenID Connect Discovery 1.0.

~~[rfc7515]~~

Rationale: We gebruiken geen elementen uit JSON Web Signature (JWS).

~~[rfc7516]~~

Rationale: We gebruiken geen elementen uit JSON Web Encryption (JWE).

~~[rfc7517]~~

Rationale: We gebruiken geen elementen uit JSON Web Key (JWK).

~~[rfc7518]~~

Rationale: We gebruiken geen elementen uit JSON Web Algorithms (JWA).

~~[rfc7591]~~

Rationale: We gebruiken geen elementen uit OAuth 2.0 Dynamic Client Registration Protocol. Dynamic Client Registration wordt bij client credentials profiel niet toegepast

~~[RFC7636]~~

Rationale: We gebruiken geen elementen uit Proof Key for Code Exchange by OAuth Public Clients. Public Clients worden bij client credentials profiel niet van toepassing

</iGov-NL/>

RFC 2616
Hypertext Transfer Protocol -- HTTP/1.1

[RFC5246]
The Transport Layer Security (TLS) Protocol

Met opmerkingen [ER8]: Betreft tls versie 1.2 en wordt vereist door RFC 6750. TLS 1.3 heeft voorkeur, onbekend of dit RFC6750 breekt, mogelijk ook relatie met RFC 9068 en RFC8705

4.2. Best Practices / Additions

[OAUTH-SBP]
OAuth 2.0 Security Best Current Practice (draft-ietf-oauth-security-topics-11). IETF. December 28, 2019. Best Current Practice. URL: https://datatracker.ietf.org/doc/draft-ietf-oauth-security-topics/

[JSONWT-BP]
JSON Web Token Best Current Practices (draft-ietf-oauth-jwt-bcp-04). IETF. November 08, 2019. Best Current Practice. URL: https://datatracker.ietf.org/doc/draft-ietf-oauth-jwt-bcp/

[RFC8174]
Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words. *B. Leiba. IETF. May 2017. Best Current Practice. URL:* https://www.rfc-editor.org/rfc/rfc8174

[RFC8725] JSON Web Token Best Current Practices
URL:https://datatracker.ietf.org/doc/html/rfc8725

[I-D.ietf-oauth-pop-architecture]
Rationale: We gebruiken OAuth 2.0 Proof-of-Possession (PoP) Security Architecture niet als onderdeel van de normatieve afsprakenset. Dit is nog een draft, mogelijk wordt wel gebruik gemaakt van het (niet normatieve) gedachtegoed (als best practice).

[DRAFTIETF-OAUTH-JWT-INTROSPECTION], T. Lodderstedt, Ed, 4 September 2021
URL: https://datatracker.ietf.org/doc/html/draft-ietf-oauth-jwt-introspection-response