Bevindingen DIL demo fase 1

Joost Diepenmaat, Remco van 't Veer

16 april 2024

Samenvatting

Tijdens de implementatie van de DIL demo fase 1 zijn we tekortkomingen, problemen en onduidelijkheden tegen gekomen. In dit document zetten we deze bevindingen op een rijtje.

Inhoudsopgave

1 iSHARE documentatie

We hebben gebruik gemaakt van de volgden bronnen:

- https://dev.ishareworks.org/
- https://ishare-3.gitbook.io/ishare-trust-framework-collection
- https://app.swaggerhub.com/apis/iSHARE/i-share_satellite/1.0

Samen geven ze een redelijk compleet beeld van de te gebruiken API's en de betekenis van de opties daarin. Helaas spreken ze elkaar echter op sommige punten tegen (zie commentaar verder op) en is het dan onduidelijk welke bron leidend is.

1.1 JWT documentatie incorrect

Op https://dev.ishareworks.org/reference/jwt.html#jwt-payload staat:

```
The iss and sub claims MUST contain the valid iSHARE
identifier (EORI) of the client.

The aud claim MUST contain only the valid iSHARE identifier of
the server. Including multiple audiences creates a risk of
impersonation and is therefore not allowed.
```

Bovenstaande lijkt geschreven voor JWT's (client assertions) die gebruikt worden voor het aanvragen van een access token bij een server, maar dat is niet vermeld. Voor andere iSHARE JTW's die door een server worden afgegeven geldt dit in ieder geval niet.

Bij delegation evidence is de `iss` de AR server, en `sub` de data consumer / client.

De `aud` claim bij delegation evidence is dan weer de service provider die de delegation mask aanvraag bij het Autorisatie register doet – dat is weer een andere server dan de `iss`, hierboven.

2 Associatie register (Satellite)

Bij het aansluiten bij de iSHARE satellite zijn certificaten voor ons gegenereerd. Bij een eerdere aansluiting waarbij certificaten werden geleverd viel al op dat iSHARE PKCS#12-bestanden levert in een verouderd formaat. Bij het uitpakken verschijnt de volgende foutmelding:

```
Error outputting keys and certificates
C0218E66F87F0000:error:0308010C:digital envelope routines:\
inner_evp_generic_fetch:unsupported:\
crypto/evp/evp_fetch.c:373:\
Global default library context, Algorithm (RC2-40-CBC : 0), Properties ()
```

Er wordt waarschijnlijk gebruik gemaakt van een verouderde versie van OpenSSL. De laatste versie waar dit algoritme standaard in gebruikt werd is 1.1.1, sinds december 2021 is versie 3.0 uit. Zie ook: $\text{OSSL}_{\text{PROVIDER}}$-legacy.

2.1 Token endpoint

`/connect/token` geeft onverwachte status `202 Accepted` en inhoud "invalid client id" als `client_id` geen EORI is. Dit zou waarschijnlijk een `400 Bad Request` of `422 Unprocessable Content` status moeten zijn. Echter, volgens de documentation zou dit een "OpenID Connect 1.0 client ID" moeten toestaan (bron: Access Token parameters) welke ook andere waarden dan een EORI toestaat.

2.2 Parties/Party endpoint

Ene keer wordt `under_score` dan weer `camelCase` gebruikt, dit is lelijk en foutgevoelig. Voorbeelden: `date_datime`, `certified_only`, `adherenceStartDate` en `registrarSatelliteID`.

Ook zit er in het `/party` endpoint resultaat een spelfout: `authregistery`. Dit zit niet in het `/parties` endpoint.

`/parties/{party_id}` met een niet bestaat ID geeft een `200 OK` met een `party_info` dat leeg is. Volgens de documentatie is dat niet valide (bron: Swagger definitie /parties/{party_{id}}. Gezien het ID op het pad staat, ligt het voor de hand dat hier een `404 Not Found` response terug wordt gegeven. In de dev.ishare documentatie wordt deze variant van het parties endpoint niet beschreven en het is daarmee onduidelijk welke documentatie leidend is.

3 Authorization Register

3.1 Modelering Delegation Evidence

De structuur van Delegation Evidence past niet bij deze use case. Hij is bedoelt om API (REST?) toegang te verlenen tussen entiteiten. In dit geval verlenen we geen API toegang maar fysieke toegang (pickup) en zijn niet alle partijen geregistreerde entiteiten (chauffeur en kenteken).

- `target.resource`

  Als resource id gebruiken we nu een "plat" opdrachtnummer, maar dit zou een URN moeten zijn (bron: Structure of delegation evidence). Het definiëren van een juist URN vergt nog wat creativiteit.

- `target.environment`

  Het verschil tussen target.environment in de policysets en policies is onduidelijk en de inhoud ervan verhoud zich slecht tot deze use cases.

- `target.environment.licenses`

  Het is onduidelijk wat het nut is van dit verplichte veld en wat er allemaal gebruikt kan worden. Na wat zoeken komen we tot de volgende lijst Licenses in iShare Wiki, deze zijn echter niet met "ISHARE." geprefixed zoals als in het voorbeeld op de dev site (bron: Decoded JWT Payload).

- `target.environment.serviceProviders`

  Volgens de dev documentatie dit optioneel (bron: Policies) maar op gitbook is het verplicht (bron: Structure of delegation evidence. De iSHARE implementatie staat een lege lijst toe (`[]`) maar bij de Poort8 implementatie hebben we `["Dummy"]` in moeten voeren.

  Het onduidelijk wat de bedoeling van dit veld is. Moeten systemen zelf checken of ze op de lijst staan?

- `target.accessSubject`

  Volgens de documentatie moet een iSHARE identifier (rechtspersonen) gebruikt worden maar dat is voor deze use-case echter te breed omdat we in deze use case met chauffeurs en kentekens te maken hebben.

3.2 Opvragen Delegation Evidence

Het antwoord bij het opvragen van Delegation Evidence (DE) is verwarrend. Bij een "match" wordt het originele DE terug geleverd maar bij een afwijzing

wordt de aangeleverde "Delegation Evidence Mask" mask verpakt als DE terug geleverd met daarin de rules aangepast met `effect` gezet op `Deny` ivm `Permit` om aan te geven dat het *niet* mag. Dit laatste heeft als probleem dat ingrediënten die missen in de mask erbij "verzonnen" moeten worden (bijvoorbeeld `notBefore`).

Het is niet duidelijk of er meerdere policies op een vraag terug kunnen komen en zoja welke conclusies daaruit genomen mogen worden. Het enige wat de documentatie hierover lost laat, is dat deze elkaar niet tegen mogen spreken (bron: Structure of delegation evidence).

Het antwoord bij het opvragen is verpakt als JWT. In theorie kan dit later gebruikt worden om de beschreven actie uit te voeren. De geldigheidsduur van de JWT is echter niet verbonden aan de (verplichte) `notBefore` en `notOnOrAfter` velden. De JWTs zijn meestal slechts 30 seconden geldig. Dat betekent dat als er toegang verleend wordt voor bijvoorbeeld "volgende week" het JWT al verlopen is als het gebruikt gaat worden. Voor de huidige implementatie is dat geen obstakel omdat er altijd gevraagd wordt voor toegang "nu". Als in de toekomst deze JWTs als langdurige authorisatie tokens bewaard gaan worden (de verifiable credentials achtige workflow) de verlooptijd van de JWTs ook verlengd moet worden.

3.3 iSHARE implementatie

`/delegation` geeft onverwacht status `404 Not Found` bij het insturen van een delegation mask met een onbekend "target.accessSubject". Dit is niet conform de documentatie (bron: Delegation Endpoint HTTP status codes).

`/policy` geeft status `200 OK` bij het succesvol opvoeren van een policy. Het ligt meer voor de hand dat hier een `201 Created` status teruggegeven wordt.

Policies worden overschreven op basis van een identieke `policyIssuer` en `accessSubject`. We hebben in deze demo gekozen voor de iSHARE AR als AR voor de verlader, een logische waarde voor `accessSubject` zou dan zijn de transporteur EORI zijn, dat betekent echter dat er maar één autorisatie per transporteur per verlader uitgegeven kan worden. Dat maakt dit AR onbruikbaar voor deze use-case. Als workaround plakken we nu het klantorder nummer achter het EORI als subject, dat mag niet volgens de specificatie maar werkt wel.

Onze verwachting is dat je eigenlijk altijd per resource X policyIssuer X accessSubject een unieke policy zou willen beheren en opvragen, en liefst meerdere (om bijvoorbeeld meerdere tijdvakken aan te geven waarbinnen toegang afgegeven is).

Het lijkt onmogelijk om een policy in te trekken. Het is wel mogelijk een policy te overschrijven (zie vorige opmerking). Dit hebben we als workaround

gebruikt; het inschieten van een regel dat iets *niet* mag (met `effect` waarde `Deny`). Echter het AR geeft een status `500 Internal Server Error` als deze weer opgevraagd wordt wat uiteraard niet wenselijk is.

3.4 Poort8 implementatie

Geeft bij verkeerd gebruik `500 Internal Server Error` terug ipv meer informatieve `400 Bad Request` met eventueel wat uitleg.

Het was niet mogelijk om `notBefore` en `notOnOrAfter` door te gegeven in nieuwe policy. Dit is meteen door poort8 opgepakt en kan nu wel, echter als parameter naam wordt `expiration` gebruikt bij inschieten voor veld `notOnOrAfter`. Zelfde veldnaam gebruiken is duidelijker en minder foutgevoelig.

`/delegation` geeft bij afwijzing `notBefore` en `notOnOrAfter` as `-1` terug. Dat is conform JSON schema (bron: iSHARE Scheme Specification) maar niet heel nuttig. De iSHARE specificatie schiet hier te kort en zou op z'n minst deze velden optioneel moeten maken in het geval van een delegation mask response.

4 OpenTripModel

We gebruiken nu de "owner" actor om het EORI van de verlader op te slaan. Dit is waarschijnlijk niet de juiste manier omdat de verlader niet meer dan de opdrachtgever van een transport opdracht is en niet de eigenaar van de goederen hoeft te zijn. Er is geen beter actor type voorhanden.