IMPLEMENTATIEHANDLEIDING MITZ

Introspectie autorisatieserver

Versie: 3.8.0
Status: Definitief
Datum: 28 juni 2021

WIJZIGINGENBEHEER

Versie	Hoofdstuk	Auteur	Opmerkingen
3.6.0.1		DD	Initiële versie
3.7.0.C		DD	Doorontwikkeling versie 3.7.0
3.7.0.TV		FS	Uitwerking migratie v3.7.0 interne review
3.7.0.TG		FS	Uitwerking migratie v3.7.0 externe review
3.7.0		FS	Definitieve versie 3.7.0
3.8.0.TV		FS	Uitwerking toestemmingsknop v3.8.0 interne review
3.8.0.TG		FS	Uitwerking toestemmingsknop v3.8.0 externe review
3.8.0.D		DD	Transactietoken ondertekent met UZI-pas
3.8.0		FS	Definitieve versie 3.8.0

INHOUDSOPGAVE

1 INTRODUCTIE

Het programma “Online Toestemmingsvoorziening (OTV)” streeft naar het gebruik van open standaarden, omdat daardoor een open koppelvlak ontstaat waarop de leveranciers die dat wensen hun producten en diensten kunnen koppelen. Daarnaast zijn open standaarden van belang omdat componenten die al gebouwd en gebaseerd zijn op deze standaarden, hergebruikt kunnen worden en dat is efficiënter en goedkoper dan maatwerk ontwikkelen en bouwen. In die situatie zouden bijvoorbeeld bestaande internationale componenten ‘slechts’ geconfigureerd hoeven te worden voor de Nederlandse situatie, in plaats van volledig opnieuw geprogrammeerd.

Dit document beschrijft het koppelvlak tussen Mitz en autorisatieserver ten behoeve van de introspectie van een ontvangen access token.

1.1 DOEL

Om een veilig koppelvlak te maken voor systemen van zorgaanbieders, wordt aangesloten bij ‘state-of-the-art security’. Het gebruik van open standaarden borgt dat ieder systeem kan aansluiten, indien het een systeem is waar een zorgaanbieder op kan inloggen (conform NEN-normen) en waar een zorgverlener of een gemandateerde medewerker een geregistreerde patiënt kan selecteren.

Voor een beschrijving van de OAuth 2.0 flow zie PvE ZNP. Voor een beschrijving van de gebruikersflow zie PSA.

1.2 TERMINOLOGIE

Er is aansluiting gezocht bij de internationale open standaard van OAuth 2.0 en daarom is (een mapping op) die terminologie gemaakt.

Er is rekening gehouden met andere OAuth 2.0 flow implementaties in de Nederlandse zorg. Zo gebruikt MedMij ook een OAuth 2.0 flow, maar daarbij zijn de rollen net omgekeerd.

2 SECURE LINK

2.1 OVERVIEW

Het voorliggende document bevat de (verwijzingen naar de) technische specificaties van de koppeling met Mitz om als zorgaanbieder toestemmingskeuzes namens de patiënt vast te leggen of te wijzigen. Ze geven een nadere invulling van de functionele eisen zoals beschreven in het Programma van Eisen Zorgaanbieder namens patiënt (zie [PvE ZNP]).

2.2 OAUTH ROLLEN

De rollen in de OAuth client credentials flow worden voor Mitz als volgt ingevuld (conform RFC 6749):

- Zorgaanbieder is de 'resource owner'
- TAP van Mitz is de 'resource server' bij de Secure Link
- REG van Mitz is de 'resource server' bij de Toestemmingsknop
- XIS met client applicatie op het Werkstation van Zorgaanbieder is de 'client'
- VZVZ autorisatieserver is de 'authorization server'

2.3 TE IMPLEMENTEREN RFC'S

	Sub rfc:	**Request for Comments titel:**	**OAuth rol die moet implementeren**
6749		The OAuth 2.0 Authorization Framework [De client credentials flow wordt geïmplementeerd.]	Client, authorization server, resource owner
6750		The OAuth 2.0 Authorization Framework: Bearer Token Usage	Client, authorization server, resource owner
6819		OAuth 2.0 Threat Model and Security Considerations	Client, authorization server, resource owner
8414		OAuth 2.0 Authorization Server Metadata	Client, authorization server
7523		JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants	Authorization server (client en resource owner geven door)
7519		JSON Web Token	Authorization server (client en resource owner geven door)
7662		OAuth 2.0 Token Introspection	Authorization server/resource owner
7009		OAuth 2.0 Token Revocation	Authorization server

2.4 OAuth Security

Comply or explain to: https://tools.ietf.org/html/draft-ietf-oauth-security-topics-15

2.5 Introspection request

Een ontvangen access token via stap 1.3 (zie figuur [PvE ZNP]) dient voor introspectie voorgelegd te worden aan de autorisatieserver. Het hiernavolgende voorbeeld geeft een weergave van het Introspection Request (ontleent aan RFC 7662, sectie 2.1).

```
POST /introspect HTTP/1.1
Host: server.example.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded

token=mF_9.B5f-4.1JqM&token_type_hint=access_token
```

De string `mF_9.B5f-4.1JqM` representeert het feitelijke JWT access token (zie hoofdstuk 2.5, [IH ZNP]).

2.6 Introspection response

Indien het acces token valide is geeft de autorisatieserver de volgende respons:

Response valide token

```
HTTP/1.1 200 OK
Content-Type: application/json
```


```
{
  "active": true,
  "iss": "https://autorisatieserver.DitIsEenVoorbeeld.com",
  "sub": "urn:hl7ii:2.16.528.1.1007.3.3:93345008",
  "aud": ["urn:oid:2.16.840.1.113883.2.4.3.111.2.1"],
  "token_type": "Bearer",
  "scope": ["modify_consent"],
  "exp": 1311281870,
  "iat": 1311280970,
  "mitz_personID": {
    "extension": "987654321",
    "root": "2.16.528.1.1007.4.1"
  },
  "mitz_uzi": {
    "extension": "244003201",
    "root": "2.16.528.1.1007.3.1"
  },
  "mitz_overseer_uzi": {
    "extension": "042392027",
    "root": "2.16.528.1.1007.3.1"
  },
  "birthdate": "1957-02-17"
}
```

"active": boolean waarde die aangeeft of het access token valide is.
"iss": issuer, URL naar de autorisatieserver die het token heeft gegenereerd.
"sub": subject, bevat de URA van de organisatie die de toegang vraagt (autorisatieserver haalt deze uit het "sub" element van het access token). 8 posities eventueel met voorloopnullen.
"aud": audience, referentie naar Mitz
"token_type": altijd Bearer.
"scope": toegang tot creëren, modificeren en verwijderen van het toestemmingsprofiel van een patiënt. Bij gebruik van de toestemmingsknop staat hier de Situatiecode.
"exp": expiration, uitgedrukt in aantal seconden sinds 1970-01-01T0:0:0Z (autorisatieserver haalt deze uit het "exp" element van het access token)
"iat": issued at, uitgedrukt in aantal seconden sinds 1970-01-01T0:0:0Z (autorisatieserver haalt deze uit het "iat" element van het access token)
"mitz_personID": bevat het gevalideerde BSN van de patiënt, 9 posities eventueel met voorloopnullen. Alleen gevuld wanneer het inschrijftoken is gebruikt (gegeven komt uit het inschrijftoken) of wanneer het transactietoken ondertekend is met de UZI-pas van een zorgverlener (gegeven komt uit het transactietoken). Anders is dit veld leeg.
"mitz_uzi": bevat de UZI van de persoon die de aanvraag doet (uit transactietoken), 9 posities eventueel met voorloopnullen. Alleen gevuld wanneer het transactie token is gebruikt. Anders is dit veld leeg.
"mitz_overseer_uzi": bevat de UZI van de mandaatgever (uit mandaattoken), 9 posities eventueel met voorloopnullen. Alleen gevuld wanneer het mandaattoken is gebruikt (gegeven komt uit het mandaattoken) of wanneer het transactietoken ondertekend is met de UZI-pas van een zorgverlener (gegeven komt uit het transactietoken). Anders is dit veld leeg.
"birthdate": bevat de geboortedatum van de patiënt, formaat YYYY-MM-DD conform [ISO8601-2004]. Noot: Alhoewel [ISO8601-2004] toelaat dat alleen het jaar wordt weergegeven of juist weggelaten, wordt hier een volledige datum verwacht.
"situatiecode": bevat de situatiecode. Alleen bij gebruik van de toestemmingsknop.

Response niet valide token

Indien het access token correct geformuleerd is maar bijvoorbeeld verlopen of voorzien van een onjuist secret geeft de autorisatieserver de volgende respons:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "active": false
}
```

Error Response

Voor overige foutmeldingen wordt verwezen naar RFC 7662, sectie 2.3

2.7 REVOCATION REQUEST

Het access token mag slechts eenmalig gebruikt worden. Dit betekent dat op het moment dat de introspectie is uitgevoerd de autorisatieserver, bij wijze van acknowledgement, een signaal moet krijgen dat het access token en geassocieerde data verwijderd mogen worden. Dit wordt gedaan via een token revocation (zie rfc 7009)

```
POST /revoke HTTP/1.1
Host: server.example.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
token=mF_9.B5f-4.1JqM&token_type_hint=access_token
```

De string `mF_9.B5f-4.1JqM` representeert het feitelijke JWT access token (zie hoofdstuk 2.5, [IH ZNP]).

2.8 REVOCATION RESPONSE

Indien het token valide is wordt het token inclusief de geassocieerde data door de autorisatieserver verwijderd.

De autorisatieserver reageert met een HTTP 200, ook als het token niet bestaat of inmiddels verlopen is.

Voor overige foutmeldingen wordt verwezen naar (rfc 7009 sectie 2.2.1).

BIJLAGE A OVERIGE REFERENTIES

Referentie	Document	Versie
[PSA]	VZVZ_Mitz_PSA	3.8.0
[PvE TAP]	VZVZ_Mitz_PvE_TAP_Toestemmingsapplicatie	3.8.0
[PvE_ZNP]	VZVZ_Mitz_PvE_ZNP	3.8.0
[URL_tokens]	https://www.vzvz.nl/ict-dienstverleners/aorta-standaardisatie/aorta-documentatie/infrastructuur-aorta-v8100 Kijk hierbij onder het kopje authenticatie, gebruik niet de ZIP	
[IH_mandaattoken]	Implementatiehandleiding Mandaattoken AORTA	8.2.0.0
[IH_inschrijftoken]	Implementatiehandleiding Inschrijftoken AORTA	8.2.0.0
[IH_transactietoken]	Implementatiehandleiding Berichtauthenticatie_Transactietoken AORTA	8.2.0.0
[ISO8601-2004]	International Organization for Standardization, "ISO 8601:2004. Data elements and interchange formats - Information interchange - Representation of dates and times," 2004.	2004