IMPLEMENTATIEHANDLEIDING MITZ

Zorgaanbieder namens patiënt

Versie: 3.8.0
Status: Definitief
Datum: 28 juni 2021

WIJZIGINGENBEHEER

Versie	Hoofdstuk	Auteur	Opmerkingen
3.6.0.0		PvG	Initiële versie
3.6.0.0		FS	Ter goedkeuring versie 3.6.0.0 release na interne review
3.6.0.0		AV	Definitieve versie 3.6.0
3.6.0.1		DD	Correcties nav consultatie Remco Schaar
3.6.0.1		FS	Definitieve versie 3.6.0.1
3.7.0.0		DD	Doorontwikkeling versie 3.7.0
3.7.0.TV		FS	Doorontwikkeling versie 3.7.0 interne review
3.7.0.TG		FS	Doorontwikkeling versie 3.7.0 externe review
3.7.0		AV	Definitieve versie 3.7.0
3.8.0.TV		FS	Doorontwikkeling versie 3.8.0 interne review
3.8.0.TG		FS	Doorontwikkeling versie 3.8.0 externe review
3.8.0.D		DD	Transactietoken ondertekend met UZI-pas
3.8.0		AV	Definitieve versie 3.8.0

INHOUDSOPGAVE

WIJZIGINGENBEHEER **2**

1 INTRODUCTIE **4**

1.1 DOEL 4
1.2 TERMINOLOGIE 4

2 SECURE LINK **5**

2.1 OVERVIEW 5
2.2 OAUTH ROLLEN 5
2.3 TE IMPLEMENTEREN RFC'S 5
2.4 OAUTH SECURITY 6
2.5 TOKEN SPECIFICATIES 6
2.6 TOKEN VALIDATIE 8
2.7 TOKENS IN BERICHT 8
2.8 GEBOORTEDATUM 9
2.9 VOORBEELD 9
2.10 VERTROUWENSRELATIE TUSSEN XIS EN AUTORISATIESERVER 9

3 SMART LINK (NADER TE BEPALEN) **10**

4 TOESTEMMINGSKNOP **11**

BIJLAGE A OVERIGE REFERENTIES **12**

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 de benodigde koppelvlakken zodat een softwareleverancier kan aansluiten op de centrale voorzieningen die worden opgeleverd binnen het programma.

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.

Voor de definitie van de tokens wordt verwezen naar tokens die ook gebruikt kunnen worden voor het uitwisselen van gegevens (wat niet via Mitz, maar via een uitwisselingssysteem gebeurt). Bijvoorbeeld in de context van AORTA (LSP uitwisseling).

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'
- 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
7522		Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants	Client, authorization server
	7521	Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants	Client, authorization server
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 Best Current Practices	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
draft		JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens	Client, authorization server, resource owner

2.4 OAUTH SECURITY

Comply or explain to:

https://tools.ietf.org/html/draft-ietf-oauth-security-topics-15

2.5 TOKEN SPECIFICATIES

Tokens voor authenticatie

Het gaat om de volgende 3 specificaties [URL_tokens]:

- [IH_Mandaattoken]
- [IH_Inschrijftoken]
- [IH_Transactietoken]

Voor de Mitz toepassing wordt er niet gekeken naar de volgende attributen uit het AttributeStatement:

- MessageIdRoot
- MessageIdExt
- InteractionId
- ContextCodeSytem
- ContextCode
- ApplicationID
- Audience Restriction

Deze attributen zijn optioneel en mogen wel gevuld zijn maar hoeven niet gevuld te worden in de Mitz context.

In de Mitz context kan de datum van de BSN validatie bij het ondertekenen van het inschrijftoken ook de geplande datum zijn. Ondertekening in dat geval gebeurt door degene die het token ondertekent. Conform de eis uit PvE ZNP mag er alleen een geverifieerd BSN worden meegegeven op het Mitz koppelvlak, dus het inschrijftoken mag in dat geval pas worden vrijgegeven nadat de SBV-Z validatie succesvol is afgerond.

Het transactietoken mag slechts eenmalig gebruikt worden. Er wordt een replay detectie op het ID van het transactietoken uitgevoerd.

Access token

Het access token is een JSON Web Token (JWT). Een JSON Web Token (JWT) bestaat uit drie gedeelten:

1. de header
2. de payload
3. de signature.

Ieder van de gedeelten wordt weergegeven door een string, base64url encoded.
Het totale token zijn de drie strings gescheiden door een punt (h.p.s).

Header


```
{
  "alg": "HS256",
  "typ": "at+JWT"
}
```


“alg”: Hashing algoritme wat gebruikt is
“typ”: Type token, string hier is voor het access token

Payload


```
{
   "jti": "7422af7c-58c6-4779-b959-b9c44a96d42c",
   "iss": "https://autorisatieserver.ditIsSlechtsEenVoorbeeld.nl",
   "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"],
   "exp": 1311281870,
   "iat": 1311280970,
   "token_type": "Bearer",
   "scope": ["modify_consent"]
}
```


“jti”: JSON token ID, Een versie 4 UUID gegenereerd door de autorisatieserver.
"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 transactietoken).
"aud": audience, referentie naar Mitz
"exp": expiration, uitgedrukt in aantal seconden sinds 1970-01-01T0:0:0Z (in het voorbeeld 15 minuten na aanmaak iat)
"iat": issued at, aanmaaktijd uitgedrukt in aantal seconden sinds 1970-01-01T0:0:0Z
"token_type": altijd Bearer.
"scope": toegang tot creëren, modificeren en verwijderen van het toestemmingsprofiel van een patiënt.

Signature

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
   vour-256-bit-sec
)
```

Het secret heeft nu de waarde “your-256-bit-secret” maar wordt gegenereerd door de autorisatieserver.

Token op basis van bovenstaand voorbeeld:


```
eyJhbGciOiJIUzI1NiIsInR5cCI6ImF0K0pXVCJ9.eyJqdGkiOiI3NDIyYWY3Yy01OGM2LTQ3NzktYjk1OS1iOWM0NGE5
NmQ0MmMiLCJpc3MiOiJodHRwczovL2F1dG9yaXNhdGllc2VydmVyLkRpdElzRWVuVm9vcmJlZWxkLmNvbSIsInN1YiI6I
nVybjpobDdpaToyLjE2LjUyOC4xLjEwMDcuMy4zOjkzMzQ1MDA4IiwiYXVkIjpbInVybjpvaWQ6Mi4xNi44NDAuMS4xMT
M4ODMuMi40LjMuMTExLjIuMSJdLCJleHAiOjEzMTEyODE4NzAsImlhdCI6MTMxMTI4MDk3MCwidG9rZW5fdHlwZSI6IkJ
lYXJlciIsInNjb3BlIjpbIm1vZGlmeV9jb25zZW50Il19.yxcFgpPwYkKWXaUK-hy_KA7z_IxEhm8kNqejEsykrwg
```


Gecreëerd met https://jwt.io/

2.6 Token Validatie

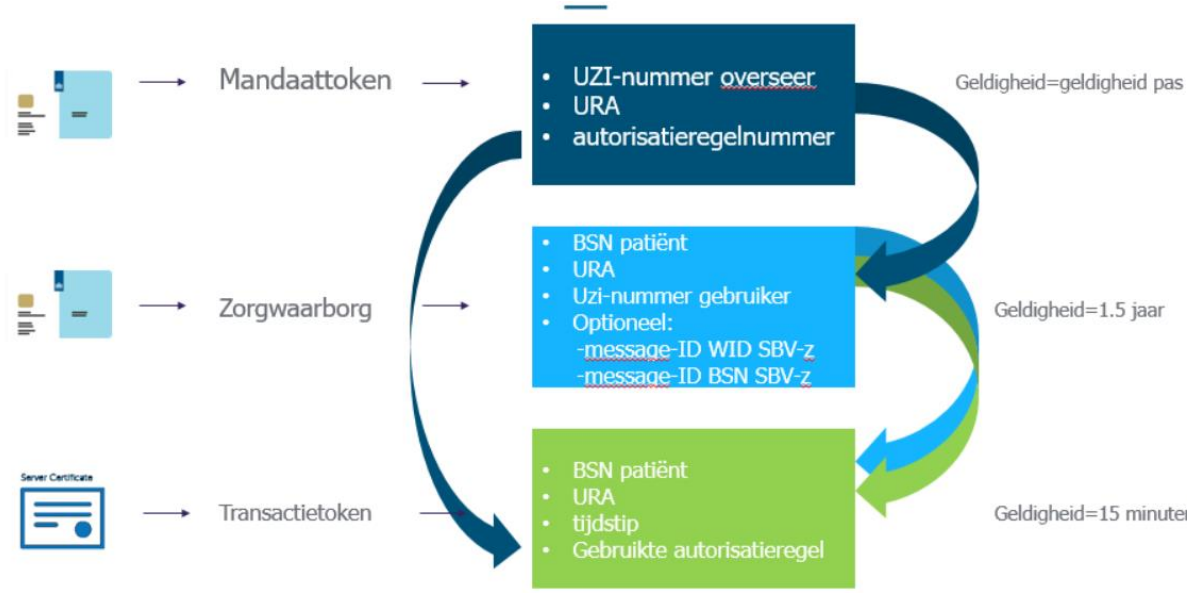
De tokens zijn onweerlegbaar (cryptografisch) met elkaar verbonden.

De tokens worden met elkaar verbonden om een stapeling van vertrouwen te creëren op het hoogste betrouwbaarheidsniveau. Er ontstaan op deze wijze gestapelde waarborgen (de waarborgen behoren onweerlegbaar bij elkaar), waarbij het mandaattoken op het hoogste betrouwbaarheidsniveau is gegenereerd, het inschrijftoken op niveau substantieel en het transactietoken op niveau laag. Hierom:

- Wordt de URA uit het mandaattoken gevalideerd tegen de URA uit het inschrijftoken
- Wordt de autorisatieregel URI uit de mandaattoken gevalideerd tegen de (daadwerkelijk gebruikte) autorisatieregel URI uit het transactietoken
- Wordt het BSN uit het inschrijftoken gevalideerd tegen het BSN uit het transactietoken
- Wordt de URA uit het inschrijftoken gevalideerd tegen de URA uit het transactietoken
- Indien het transactietoken ondertekend is met de UZI-pas van een zorgverlener, dan heeft dit token betrouwbaarheidsniveau hoog en zijn mandaattoken en inschrijftoken niet benodigd.

Voor overige validaties:

- Wordt de URA uit het transactietoken gevalideerd tegen de URA uit de TLS-tunnel
- De ondertekening van de tokens wordt gevalideerd
- De condities t.a.v. de geldigheid van de tokens worden gevalideerd (notBefore en notOnOrAfter)



2.7 Tokens in Bericht

De tokens worden los van elkaar meegestuurd in de HTTP body van het bericht respectievelijk in de elementen:

transactietoken=<assertion>…….</assertion>
mandaattoken=<assertion>…….</assertion>
inschrijftoken=<assertion>…….</assertion>

2.8 GEBOORTEDATUM

Het bericht 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.

birthdate=1957-02-17

2.9 VOORBEELD

```
POST /token.oauth2 HTTP/1.1
     Host: as.example.com
     Content-Type: application/x-www-form-urlencoded

     transactietoken%3D%3Cassertion%3E%u2026%u2026%3C/assertion%3E%26
     mandaattoken%3D%3Cassertion%3E%u2026%u2026%3C/assertion%3E%26
     inschrijftoken%3D%3Cassertion%3E%u2026%u2026%3C/assertion%3E%26
     birthdate%3D1957-02-17
```

2.10 VERTROUWENSRELATIE TUSSEN XIS EN AUTORISATIESERVER

Om de zorgaanbieders te ontlasten is configuratie van een vertrouwensrelatie vooraf niet nodig. De autorisatieserver zal de vertrouwensrelatie vaststellen op basis van:

- Het UZI-servercertificaat dat gebruikt is voor het opzetten van de TLS-verbinding
- De waarborgen uit de tokens

3 SMART LINK (NADER TE BEPALEN)

(niet in scope)

4 TOESTEMMINGSKNOP

Bij de toestemmingsknop dient het XIS, indien vereist, evenals bij de Secure Link (zie hoofdstuk 2) een access token aan te vragen bij de Autorisatieserver.

Verschil met de Secure Link is dat het aantal in te sturen tokens naar de Autorisatieserver afhankelijk is van de situatie (i.e. de situatiecode). De situatie en de vereiste tokens worden gedefinieerd in de Toestemmingscatalogus. Ook hier geldt dat als het transactietoken ondertekend is met de UZI-pas van een zorgverlener, dan voldoet alleen dit token (geen mandaattoken en inschrijftoken benodigd).

Daarnaast wordt naast de geboortedatum in het bericht voor de Secure Link bij de toestemmingsknop de situatiecode verwacht in de volgende vorm:

situatiecode=SIT002

De scope in het access token wordt vervangen door de situatiecode

```
"scope":                    ["SIT002"]
```

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