

**Identity & Authorization Management (IAM)
eXchange
Technical specifications**

Version 1.4

This document is provided to you free of charge by the

eHealth platform

Willebroekkaai 38

38, Quai de Willebroek

1000 BRUSSELS

All are free to circulate this document with reference to the URL source.

Table of contents

Table of contents	2
1. Document management	4
1.1 Document history	4
2. Introduction	5
2.1 Goal of the service	5
2.2 Goal of the document	5
2.3 eHealth platform document references	5
2.4 External document references.....	6
3. Support	7
3.1 Helpdesk eHealth platform	7
3.1.1 Certificates	7
3.1.2 For issues in production	7
3.1.3 For issues in acceptance.....	7
3.1.4 For business issues	7
3.2 Status	7
4. Global overview	8
4.1 Process overview for Trusted Platforms	8
4.2 Process overview for technical clients	9
5. Step-by-step	10
5.1 Technical requirements	10
5.1.1 Tracing.....	10
5.2 Process overview for Trusted platform.....	11
5.2.1 eHealth platform authentication	11
5.2.2 GET /profiles.....	11
5.2.3 POST /protocol/oauth/tokenExchange	13
5.3 End user workflow	18
5.4 Process overview for technical clients	21
5.4.1 eHealth platform authentication	21
5.4.2 GET /profiles/{ssin}	21
5.4.3 GET /profiles/{ssin} with pseudonyms	22
5.5 Reference implementation	25
5.5.1 General description.....	25
6. Risks and security.....	26
6.1 Risks & safety	26
6.1.1 End user consent.....	26
6.1.2 Token validity period.....	26
7. Test and release procedure	27
7.1 Procedure.....	27
7.1.1 Initiation	27
7.1.2 Development and test procedure	27



7.1.3	Release procedure	27
7.1.4	Operational follow-up	27
7.2	Test cases	27
8.	Error and failure messages	28

To the attention of: “IT expert” willing to integrate this web service.



1. Document management

1.1 Document history

Version	Date	Author	Description of changes / remarks
1.0	09/09/2021	eHealth platform	Initial version
1.1	29/05/2024	eHealth platform	Deletion Par 9: Annex A – Security commitment from the Trusted Platform This document is available on the portal of the eHealth platform.
1.2	18/07/2025	eHealth platform	Introduction of exchange to SAML 2.0 assertion
1.3	24/03/2026	eHealth platform	Update Tracing
1.4	10/09/2026	eHealth platform	GET /profiles/{ssin} with pseudonyms

2. Introduction

2.1 Goal of the service

In today's clouded world, thin clients have become more and more popular at the expense of fat clients.

In addition, all major browsers (most widely used thin clients) have given up support for Java Applets making it possible to embed full Java applications into a browser.

The Service Oriented Architecture (SOA) of the eHealth platform and its partners have so far been designed around a few protocols and principles that work rather well from system to system between the eHealth platform and its partners or with full Java or .net software packages on the desktops of the customers. However, when using simple thin clients such as a browser, things get more difficult, especially if that thin client is running on a mobile device.

Our services currently use:

- SOAP Protocol as extra layer above the HTTP Protocol to transport messages between client and server
- WS-Security for authentication, confidentiality and integrity of the messages sent between client and server
- Trusted certificates, issued by recognized Certificate Authorities (CA) to verify identity tokens (X509, SAML assertion)
- Triple-wrapped CMS messages to encrypt data end to end from (identified) sender to both known and unknown receivers.

To facilitate integration with existing eHealth and/or partner services, IAM eXchange can be used.

IAM eXchange issues SAML Holder-of-Key (HOK) session tokens, which assert that a client has a valid eHealth profile.

The SAML token can be used to authenticate the client to most eHealth or partner services by signing the Body of SOAP messages with the Private Key that corresponds with the Public Key mentioned in the SAML token which proves that the client is the legitimate owner of the token.

2.2 Goal of the document

This document is not a development or programming guide for internal applications. Instead, it provides functional and technical information and allows an organization to integrate and use the eHealth platform service.

However, in order to interact in a smooth, homogeneous and risk controlled way with a maximum of partners, these partners must commit to comply with the requirements of specifications, data format and release processes of the eHealth platform as described in this document.

Technical and business requirements must be met in order to allow the integration and validation of the eHealth platform service in the client application.

It should be used in complement to the Swagger API, which describes the interface of the service and the structure of the request and responses.

2.3 eHealth platform document references

On the portal of the eHealth platform, you can find all the referenced documents.¹ These versions or any following versions can be used for the eHealth platform service.

¹ <https://ehealth.fgov.be/ehealthplatform>

ID	Title	Version	Date	Author
1	IAM Mobile integration – Technical specifications	1.13	10/04/2026	eHealth platform
2	SOA – Error guide	1.0	10/06/2021	eHealth platform
3	Request test case template	3.0	22/02/2018	eHealth platform
4	Swagger API IAM-Exchange	N.A.	N.A.	eHealth platform
5	Blinded pseudonymisation REST service - Cookbook	1.9.1	23/07/2026	eHealth platform

2.4 External document references

All documents can be found through the internet. They are available to the public, but not supported by the eHealth platform.

ID	Title	Source	Date	Author
1	OAuth 2.0 Token Exchange	https://datatracker.ietf.org/doc/html/rfc8693	01/2020	M.Jones (Microsoft) A.Nadalin (Microsoft) B. Campbell (Ping Identity) J.Bradley (Yubico) C. Mortimore (Visa)

3. Support

3.1 Helpdesk eHealth platform

3.1.1 Certificates

In order to access the secured eHealth platform environment you have to obtain an eHealth platform certificate, used to identify the initiator of the request. In case you do not have one, please consult the chapter about the eHealth Certificates on the portal of the eHealth platform

- <https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten>
- <https://www.ehealth.fgov.be/ehealthplatform/fr/certificats-ehealth>

For technical issues regarding eHealth platform certificates

- Acceptance: acceptance-certificates@ehealth.fgov.be
- Production: support@ehealth.fgov.be

3.1.2 For issues in production

eHealth platform contact centre:

- Phone: 02 788 51 55 (on working days from 8 am till 6 pm)
- Mail: support@ehealth.fgov.be
- Contact Form:
 - <https://www.ehealth.fgov.be/ehealthplatform/nl/contact> (Dutch)
 - <https://www.ehealth.fgov.be/ehealthplatform/fr/contact> (French)

3.1.3 For issues in acceptance

Integration-support@ehealth.fgov.be

3.1.4 For business issues

- regarding an existing project: the project manager in charge of the application or service
- regarding a new project or other business issues: info@ehealth.fgov.be

3.2 Status

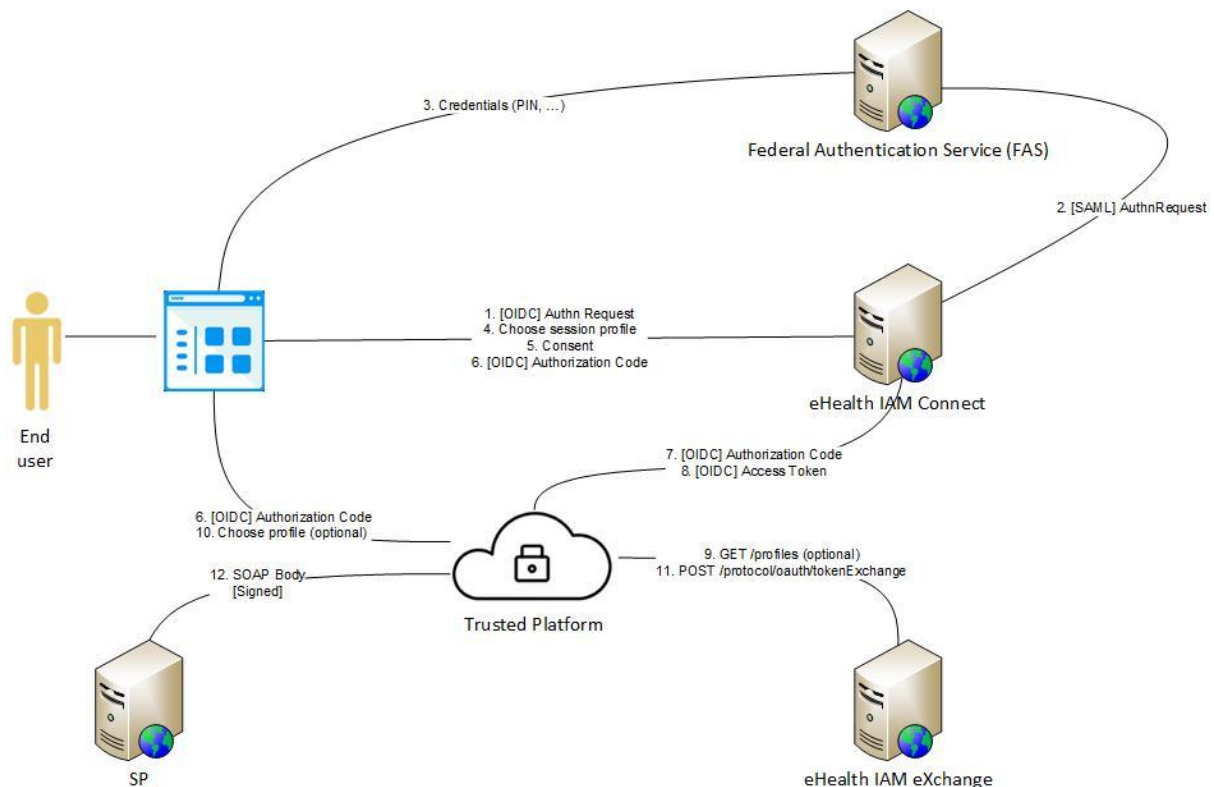
The website <https://status.ehealth.fgov.be> is the monitoring and information tool for the ICT functioning of the eHealth services that are partners of the Belgian eHealth system.

4. Global overview

In this section, we describe the 2 major ways to use IAM eXchange:

- IAM eXchange for Trusted Platforms
- IAM eXchange for technical clients

4.1 Process overview for Trusted Platforms



The end user uses his browser to contact (at least) one service provider (SP).

The client initiates the login (1) protocol with IAM Connect (Authorization Server).

IAM Connect relies on FAS service (2) for the authentication mechanism. End user is invited to provide his PIN (3) (or other credentials depending on the authentication method supported).

If the authentication succeeds, IAM Connect will propose a list of profiles² for the end user authenticated (4).

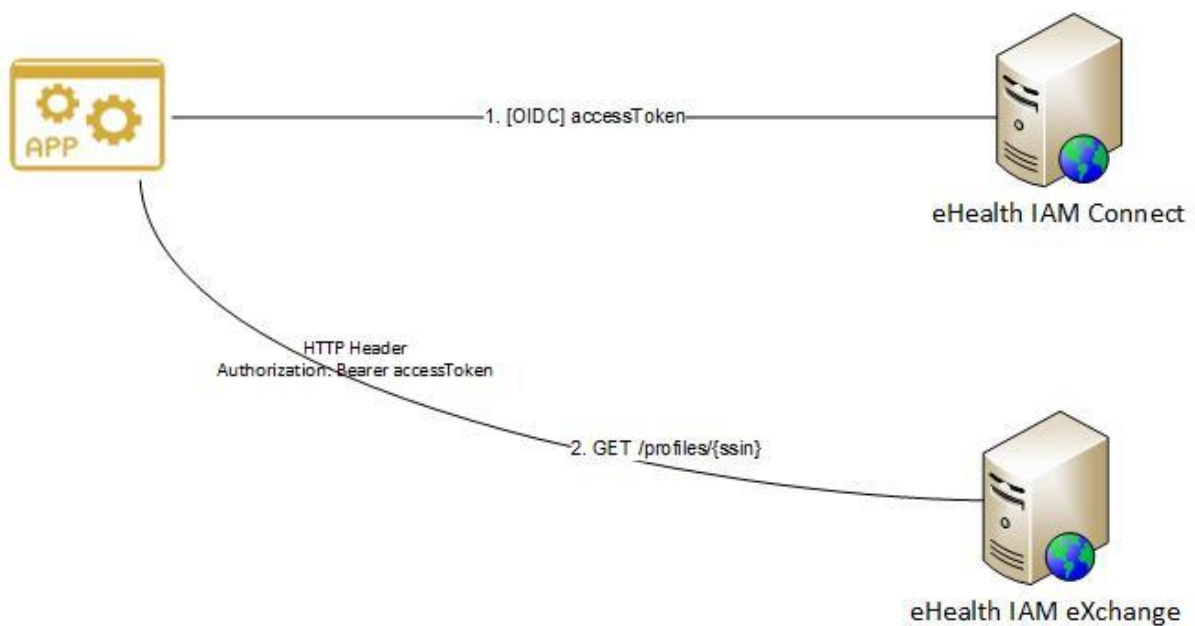
As the client will perform actions in the name of the end user, the latter must give his consent to the client in order to continue (5).

An AuthorizationCode is then sent from IAM Connect to the Trusted platform (6). With the AuthorizationCode, the Trusted Platform can obtain an Access Token (7, 8) which can be used to interact with IAM eXchange (9, 10, 11).

The SAML token obtained (11) can then be used by the Trusted Platform to contact the service provider in secured way (12).

² Supported profiles are managed by the eHealth platform. Depending on the profile selected, the SAML HOK assertion may contain different attributes.

4.2 Process overview for technical clients



The client uses client credentials flow to request an accessToken (1) with IAM Connect (Authorization Server). With this accessToken, the client can request (2) the list of profiles (for the SSIN provided in input) to IAM eXchange.

Technical clients do not have the possibility to perform any exchange with IAM eXchange. The exchange functionality is dedicated to trusted platforms.

5. Step-by-step

5.1 Technical requirements

5.1.1 Tracing

To use this service, the request SHOULD contain the following two http header values (see RFC <https://datatracker.ietf.org/doc/html/rfc7231#section-5.5.3>):

1. **User-Agent:** information identifying the software product and underlying technical stack/platform. It MUST include the minimal identification information of the software such that the emergency contact (see below) can uniquely identify the component.
 - a. Pattern: {minimal software information}/{version} {minimal connector information}/{connector-package-version}
 - b. Regular expression for each subset (separated by a space) of the pattern: `[[a-zA-Z0-9-\\]]*\\[[0-9azA-Z-_.]]*`
 - c. *Examples:*
User-Agent: myProduct/62.310.4 Technical/3.19.0
User-Agent: Topaz-XXXX/123.23.X freeconnector/XXXXX.XXX
2. **From:** email-address that can be used for emergency contact in case of an operational problem.

Examples:

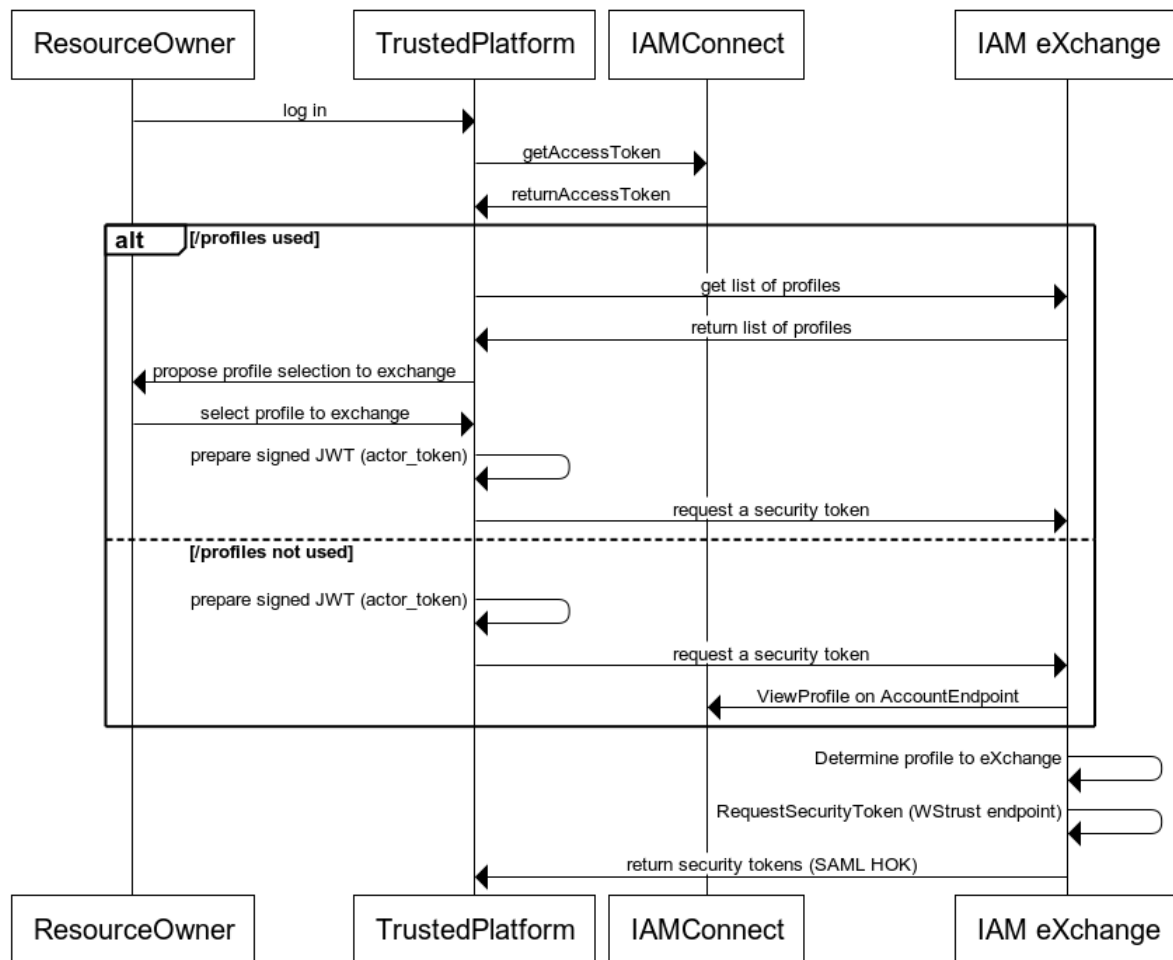
From: *info@mycompany.be*

Exception for Browser-Based Clients

Browsers control the User-Agent header natively and may block client-side modification for security reasons. When operating in a browser-based environment, supplying the custom User-Agent header is **not required**. That said, it **must** be included whenever the environment permits it.

The From header carries no such restriction and **must** be present in all requests, regardless of environment.

5.2 Process overview for Trusted platform



5.2.1 eHealth platform authentication

In order to use IAM eXchange service, the Trusted Platform must be able to obtain an accessToken. The Trusted Platform must use the Authorization Code flow to initiate the login protocol (see IAM Connect – Mobile integration for more information).

Two roles are available:

- profile: this role must be present in the accessToken in order to retrieve the list of profiles of the authenticated end user
- token-exchange: this role must be present in the accessToken in order to use token exchange

The Trusted Platform MUST request the client scope *iam:exchange:tokenexchange*.

If the Trusted Platform wants to use the */profiles* operation, the client scope *iam:exchange:profile* is also required.

5.2.2 GET /profiles

The operation must be used to retrieve the list of profiles of the authenticated end user.



The Trusted Platform may use this operation to propose to the end user which profile he/she wants to exchange with */protocol/oauth/tokenExchange* (see section 5.2.3.1).

If this operation is not used by the Trusted Platform, the exchange will only rely on the eHealth profile selected by the end user in eHealth IDP (see section 5.2.3.2).

In this section, we assume that the TrustedPlatform is already configured and recognised by the eHealth platform (see section 5.2.1).

5.2.2.1 Request

No specific input.

5.2.2.2 Response

If the operation succeeds, the result may contain a list of profiles (JSON format).

Element	Description
firstName	First name of the authenticated end user
lastName	Last name of the authenticated end user
ssin	SSIN of the authenticated end user
children	Child/children of the authenticated end user. Each child is represented with the following elements - ssin - firstName - lastName Child/children is/are not listed if the Trusted Platform is not concerned by this profile's subset.
mandators	Mandator(s) of the authenticated end user The mandate type(s) detected are specified in serviceNames for each mandator. Each service name listed corresponds to exactly one mandate type. For examples : <i>medicaldatamanagement (Gestion des données de santé/ Beheer van gezondheidsgegevens)</i> , <i>recipe (Mandat de prescription/ Voorschriftenvolmacht)</i> Mandators are not listed if the Trusted Platform is not concerned by this profile's subset.
organizations	Organizations related to the authenticated end user. Organizations are not listed if the Trusted Platform is not concerned by this profile's subset.

Example without profile found :

```
{
  "firstName": "John",
  "lastName": "Doe",
  "ssin": "12345678912"
}
```

Example with mandates and children:

```
{
  "firstName": "John",
  "lastName": "Doe",
  "ssin": "12345678912",
```



```

"children": [ {
  "lastName": "Doe",
  "firstName": "Junior1",
  "ssin": "23456789123"
},
{
  "lastName": "Doe",
  "firstName": "Junior2",
  "ssin": "34567891234"
}],
"mandators": [ {
  "firstName": "Grandfather",
  "lastName": "Doe",
  "ssin": "01234567891",
  "name": "Doe Grandfather",
  "serviceNames": ["medicaldatamanagement"]
}]
}

```

5.2.3 POST /protocol/oauth/tokenExchange

The operation must be used to exchange an access token into SAML HOK assertion.

This operation cannot be used by the Trusted Platform without a valid accessToken (obtained after a successful end user login). This accessToken is not sufficient. The Trusted Platform must also generate a signed JWT (see section 5.2.3.1).

In this section, we assume that the Trusted Platform is already configured and recognised by the eHealth platform (see section 5.2.1).

5.2.3.1 JWT token generation

The client application (the Trusted Platform) authenticates itself by signing a JWT (RFC7519) with its private key. The generated token will be used during exchange operation (it must be set in actor_token parameter – see section 5.2.3.2.1).

When the certificate used for signing the JWT approaches its expiration date, a request to renew the certificate must be sent to eHealth Platform via the procedure described here:

NL: [eHealth-certificaten | eHealth-platform](#), formulier “**Certificate Management: Renewal of public key**”

FR: [Certificats eHealth | Platform eHealth](#), formulaire “**Certificate Management: Renewal of public key**”

This request must be sent 8 weeks before the expiration for ACCEPTANCE certificate or 2 weeks before for PRODUCTION certificate.

Example:

Header

```

{
  "alg": "RS256"
}

```

Payload

```

{
  "iss": "frontendclient",
  "exp": 1516906514,
  "iat": 1513602283,
  "jti": "id123456"
}

```

Fields in the JWT payload are mandatory:

- iss: ‘Issuer’ identifies the principal that issued the JWT. It corresponds to the client id of the Trusted Platform.



- exp: "Expiration Time", identifies the expiration time on or after which the JWT must not be accepted.
- iat: "Issued At", identifies the time at which the JWT was issued
- jti: "JWT ID", provides a unique identifier for the JWT.

If the Trusted Platform uses */profiles*, the Trusted platform should add a claim (*sub*) representing the subject in the payload.

Example with sub claim :

```
Header
{
  "alg": "RS256"
}
Payload
{
  "iss": "frontendclient",
  "exp": 1516906514,
  "iat": 1513602283,
  "jti": "id123456",
  "sub": "90e9cedc5a771dce969c1388c4508783"
}
```

The value of this element MUST correspond to one of the sub claim presented in the access token (under claim *may_act*).

Example of *may_act* issued within an access token:

```
"may_act": [
{
  "sub": "90e9cedc5a771dce969c1388c4508783",
  "userProfile": {
    "children": [
      {
        "ssin": "23456789123"
      }
    ]
  }
},
{
  "sub": "cedc5a771dce969c1388c450878390e9",
  "userProfile": {
    "children": [
      {
        "ssin": "34567891234"
      }
    ]
  }
},
{
  "sub": "8a0f71a0d8a302166d4baa403954e511",
  "userProfile": {
    "mandators": [
      {
        "ssin": "01234567891"
      }
    ]
  }
}
]
```

5.2.3.2 Exchange access token

With the obtained access token and with the signed JWT, it is now possible to perform the exchange.



5.2.3.2.1 Request

In the Form Data, add the following parameters:

- *grant_type*: fixed value "urn:ietf:params:oauth:grant-type:token-exchange" indicates that a token exchange is being performed
- *requested_token_type*:
 - "urn:ietf:params:oauth:token-type:saml1" for SAML 1.1 assertion
 - "urn:ietf:params:oauth:token-type:saml2" for SAML 2.0 assertion
- *actor_token*: security token that represents the identity of the acting party. It corresponds to the JWT generated by the Trusted Platform (see section 5.2.3.1).
- *actor_token_type*: fixed value "urn:ietf:params:oauth:token-type:jwt"
- *subject_token*: security token that represents the identity of the party on behalf of whom the request is being made. Typically, the subject of this token will be the subject of the security token issued in response to this request.
- *subject_token_type*: fixed value "urn:ietf:params:oauth:token-type:access_token"
- *audience* must be empty
- *scope* must be empty
- *resource* must be empty

Example:

POST <https://api.ehealth.fgov.be/iam/v2/protocol/oauth/tokenExchange> HTTP/1.1

Host: as.example.com

Content-Type: application/x-www-form-urlencoded

```
grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-exchange
&requested_token= urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Asaml1
&actor_token= <signed token>
&actor_token_type= urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Ajwt
&subject_token=<access token>
&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Aaccess token
```

5.2.3.2.2 Response

If the operation succeeds, the result may contain an access token (in JSON format).

Example:

[illegible]

zNGeU1Y1R1amlQbW9GZ1lTaUpwWHFHbVF2allneDVXMzJSVmHbOVHVDdkM5R1dabGNKNWIIcNXTDV2SFZuL2VzTytwcHqYzNlxNFpR2F
0dnFPaUdsSlpVqitYZWorSWd1cXBTNJZMbkdya21TUzZNNkdjeS85bEVBSnFKUFRWU2oKUIRSTRBajNxVXY4dEh6SWdweVZGWUNRL3NSSEIV
eEpyS2lMcjhoM3NjdDdidXdzNnRoagNRZ2QzVmJkTgtFQ3pRUG1RMDlnbEdLbgp1bzfFFNmV5K1pvYnJOTkJKUFhobzVlVeINZQ0pPblArTdhFalZ
SSUowbFAzUfNMWnZRkZFPtXF3NjR1cWJkZzh3a241SW1UNFQ5bjFwCkRKNWNHbHBM220yMS96SE5VSHIROUx1b0MxeWpmNzNydVM3ei
s0M2pBZ01CQUFHAmdnRVVNSUICRURCRUJnZ3JCZOVGQlFJQkFRUTQKTURZd05BWUild1lCQlFVSE1BR0dL2gWZEHbNkx5OXZM053TFdWb
0xYQJbJR3RwTG10dmJtWnBaR1Z1Y3k1NlpYUmxeTVqYJlwdwplUUVIEVllwT0JCWUUVGSjKvRIltMEFjSGc5V3ExeVU4WkPzMHZ4UDdWTUf3R0E
xVWRfD0VCL3dRQ01BQXdld1IEVllwkJCZ3dGb0FVCKsR3ZWalBSM1RQbklSMUxJukdzbTdjS0R4a3dTd1IEVllwzKJfUXdRakJbB0QZ21BJWTZh
SFlwY0RvdkwyTnliQzFsYUMxd2RIQnlKYVM1amlyNW1hV1JsYm5NdWVtVjBaWE11WTISdEwxcERSVWhRVKZCTFNVTkJNREF4TG1OeWJEQU9
CZ05W5FE4QkFmOEVCCQU1DQmVBdwpIUUVIEVllwbEJCWXDGQVlUS3dZQkRVUUhBd0IHQ0NzR0FRVUZCd01FTUEwR0NTcUdTSWlZrFFFQkN3V
UFBNEIDQVFCVzNpMg1vQzEvClkwbDhJcXhmZThKcW1jbi80RmZnWlBwUkNhNlp4aG9IL2lxTVJ1YlI5R3BubUUhXV1FDRDVKZEhsWXNRYWkxM
kcydE9scWlNQ0R4YWEKVjNwUzdLN0NmK1h6MWx3ZEPU253dnE3VGZsRkgyRIJSTTlrcnYOVXNqYIEvOEErMmFBOWhsZDFVM3hBNUlpWLTli
cDhDOEx2VFBUQgPralY5QndMTUNqMHVjTnRISXpkUmh4THMweDhCZHN1c1JqNFhMdUgwdUY4MXo2VXh3QzBlZ2YcEZFBjJU3FKZm5Pal
NseGJpeXJ4CldSRWgrRVFfnQ3ZsNm0vSvVrMEEOtWlKvEhznHpaQ1BPR3FuazNibjg1VUEvNmEvVUpkcElrYlxcVTV4Q1pVWEVaRjFGJNBZ2RH
QkMKbGdHd0JMSHIMMVDlbbkZ5QUi4RE9aLzRxbDBRTXJaMW9iQm1rbG95d294ZjIbFJiNVVvZ2dEYVo0ay9acUZocEXlUulWtKrYayTTRgpDUGR
QVXlmWnZleXQdR2ZoRGE4NGRJBTFIMk5KQ2d1S0Nxbi95QJZ0YUFHKzJJUJNINStoL1VRWDJ5T2owNEc0S0RaUuZWYms0WnR1CnJfZnNLSTV
aY0pOYk9SFFfUitYaitVldqQURORmE3RDZlclFoNXyRaE5Gc1V4UC9pSDJEWm90cXJmbXcrTVZOMHJFVHJ4Wk5WskIKOflpd1kwcndPukfJTF
F3NmdmUnRHcHBDZVE3iU9NRWpyT2k2RFFKZGR5TFdlbVNVZWR4TW4xek1kL1VZROU3NC9oTmNIL2FjCjVGSAPfSkd4dWNYWEpqb0kRkM
0Y2cvQ1pJdVJOWkhHRGxsk2Q0kZrYnVrOHgzUGpJU1FWanNhMWdYTFBINGxdWNDWUE9PTwvZHM6WDUwOUnlcnRpZmljYXRlPgm0KCQk
JCQk8L2RzOlG1MDIEYXRhPg0KCQkQJCTwvZHM6S2V5SW5mbz4NCgkQJCTwvU3ViamVjdENvbmZpcm1hdGlvb34NCgkPC9TdWJqZWNOg0KCTw
vQXV0aGVudGJlYXRpb25TdGF0ZW1lbnQ+DQoJPEF0dHJpYnV0ZVN0YXRlbWVudD4NCgkJPfN1YmplY3Q+DQoJCQk8TmFtZUlkZW50aWZpZGI
gRm9ybWFOpSj1cm46b2FzaXN6bmFtZXN6dGME6U0FNTDoxLjE6bmFtZWklXVudC3BlY2lmaWVklBOYW1lUXVhbGlmaWVvPSJ1cm46YmU6Zl
mdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRlbnRpZmlj
YXRpb24tbmFtZXNwYWNlIj4NCgkQJCTxBdHRyaWJ1dGVWYXWx1ZT4xMjM0NTY3ODkxMjwvQXR0cmliXRlVmfSdWU+DQoJCTwvQXR0cmliXR
lPgm0KCQk8QXR0cmliXRlIEF0dHJpYnV0ZU5hbWU9InVyb3piZTpmZ292OmVoZWZsdGg6MS4wOmNlcnRpZmljYXRlAG9sZGVyOnBlcnNvb3pzc2l
ulBBDHRyaWJ1dGVOYW1l3c3BhY2U9InVyb3piZTpmZ292OmVklZW50aWZpY2F0aW9uLW5hbWVzcGFjZSI+DQoJCQk8QXR0cmliXRlVmfSdWU
+MTIzNDU2Nzg5MTI8L0F0dHJpYnV0ZVZhbHVlPgm0KCQk8L0F0dHJpYnV0ZT4NCgkJPfE0dHJpYnV0ZSBBdHRyaWJ1dGVOYW1lPSJ1cm46YmU6
ZmdvdjplaGVhbHROmJlhbTpleGNoNwY5NzSl+MTIzNDU2Nzg5MTI8L05hbWVlZGVudGlmawVvYpG0KCQk8L1N1YmplY3Q+DQoJCTxBdHRyaWJ1
dGUgQXR0cmliXRITmFtZT0idXJlOmJlOmZnb3Y6cGVyc29uOnNzaW4iEF0dHJpYnV0ZU5hbWVzcGFjZT0idXJlOmJlOmZnb3Y6aWRl

```

U+MTlZNEZha2VWYwX1ZTwwZHM6U2lnbmF0dXJlVmFsdWU+DQoJCTxkcZpLXlJbmZvPg0KCQkJPRzOlglMDIEYXRhPg0KCQkJCTxkcZpYNTA
5Q2VydGlmawNhdGU+TUIJROV6Q0NBL3VnQXdlJQkFnSVVMTk3amxtMGd6S1Y3NFduTnVFUmZ3aG9ldU13RFFZSkvWklodmNOQVFFTEJRQ
XdWakVMtUFRwPBMVVFQmhNQ1FrMHHHVEFYQmdOVkKBb01FRkYxYjFaaFpHbHhJRXhwYlksMFpXUXhMREFxQmdOVkBTU1JMUyXyYjFaa
FpHbHhJRIU5CmRYTjBJRUZ1WTJodmNpQkpkjM04xYVc1bkFTkzJWRWN5TUI0WERURTVNREI4TXpBM05ERXDOMW9YRFRJEU1ESXhNekEzTIRFd0
1Gb3ckZ2JNeEN6QUpcZ05WQkFZVEFrSkZNU3R1FZRFZRUUteQkpwHwldSbGNTnRnNRWR2ZG1WeWJtMWxibF4RHpBTkNtZlZCQXNNQmts
QgpUVWwXpVkrFEW1CVUdBMVVFQ3d3T1EwSkZQVEE0TURek9UUTBNamN4R1RBWEJnTIZCQXNNRUUVSVJVRk1WRWd0VUV4QlZFWIBVaz
B4CKlUQWZCZ05WQkFzTUdHVklaV0ZzZEdndGNHeGhkr1p2Y20wZ1FtVnNaMmwxYIRFZk1CMEdBMVVFQX3V1EwSkZQVEE0TURek9UUTA
KTWpjcl0IfbEJUVVwXpVkrDQ0FTSXdEUVIKS29aSWh2Y05BUUVCQlFBRGdnRVBRENDQVFvQ2dnRUJBS0RVb3FFQ1dzWUtKb0F0d0F4R4p1RV
hQQS9NZGdYSHNmUDVuRedUQUr1MEhXUnRna3dGZER1NmVXSnV5REdRLzhRUFVZSy9yWG1LnzFyOTBWWWEyZXdURWE4VUQ0NzRqCn
h3KzNGRGdhMWpnTndFbmo0d3BscmJTMmNxengyUjJQbTNN0ppVDNhOTZXZjQ3SGl1R1B6dzNTM2oxd0RMNFk4cmdPUeWrNWQ3dVgKY
2k1SIAxeWzhcVkd0dzbUxQTGTanJvbExOUEhRNHFUZE90L2IEUXovbkl0NWJvc2NsKzc4MwpmdlQwMG41T0ZEbVU1WTRlY1JaNaPVTArU
WN3ZwTxeXlVMzZBZ0ZSRStOU1I2WHZ2b016YndIT0VOOTV0WThZWGIVWUvRTTlIaW1wZDNmSVVYMEVCMktDWHlwU2VqQTR2CmJOUTQ
2Y2N1VXBhNFIsVnNCb01DQXdfQUFhT0NBWGt3Z2dGMU1COEdBMVVKsXdrWU1CYUFGS2pCbTjvbUNNQWNQmZ0eXdtOT0F5VG0KeUVO
Uk1IUUdDQ3NHQVFVRk13RUJCR2d3WmpBNEJnZ3JCZ0VGQlFjd0FvWVNhSFlwY0RvdKwzUnlkWE4wTG5GMWlzMmhaR2x6WjJ4dgpZbUzTG
1OdmJTOXhkbJlOwTJGbk1pNWpbjF3S2dZSut3WUJCUVVITUFHR0htaDBkSEE2THk5dlkzTndMbkyYjNaaFpHbHhPaMnh2CltRnNMbU52YIRC
UkNtIZIU0FFU2pCSU1FWUdEQ3NHQVFRQnZsZ0FCQUlBQVRBMk1EUUdDQ3NHQVFRk13SUJGaWhvZEHsd09pOHYKZDNkM0xurJFiM1poW
kdseloyeHZZbUzTG1OdmJTOXlaWEJ2YzJsMGlzSjVnQjBHQTFVZEUpRUvdNQlFHQ0NzR0FRVUZCd01DQmdncgpcZ0VGQlFjREJEQTdCZ05WSFI
4RU5EQXINRENnTHFbc2hpcG9kSFJ3T2k4dlkzSnNMbkYxYjNaaFpHbHhPaMnh2WW1Gc0xtTnZiUzI4CmRuUmhZMkZuTWk1amNtd3dlUUIEVllw
T0JCWUUGTEpOdUoPOTUFOc1lGeTJYMFV5ZTZtMzRPyZuTUeU0R0EXVWREd0VCL3dRRUF3SUUYKNERBTKJna3Foa2lHOXcwQkFRc0ZBQU9DQW
dFQVAVbUg2ZtBIOE1hKzJPSloxNk1aMUQwb0pnYkRCRVRsQ3JOUHEvSXA4WTcxVGxlegpnZFdHaXNZbytLcFdPYIZCUGxvV2krc1p0L3h5bHRjU0
JwRVd5QVhmemlIMFBjWdJlVGlJSGF4L2FtYlpXTDZVdStPMGd6bjdwNWdzCjhleEd0OEhiTXB6L08yVtdYckU2ZUkxVS8yUjJvc1hYeE1vZEJCR1R
YUTBiSE5NQ3E3ek5Qc1h3TnBoQzhYb0F6VWdqCHdzk1QT2gKNDRnQTJzNHJkRzFSajB3NFVVK2VKV0JncFMzZkEvZEcvK2R6MUZkemhLclZoa
zFYTmdXTVd1LzlpdElrZ21DQUJET0NoUnJHcTFhWQplbkU0R0Q0Sjd1M1hCS09WMmF5Sml5aTNqVm1aOVNDNFUvemtnazF3aHR6eExUcnJT
T3MrVmZsbllXNFVDSnJqcVB2OTVtd3NjTlMyCkNCZmlyZ29UMTh3VFdmeW5jNStuaHMxTlkva2paMWhqNWfYTWRRUXh1Y0pVaG1CZGdGb0
02UDhzc1h1cmQ1dkU4K0lmNkU4QVNjdTUKVVIltT2NzZTJMcW5PVE5Odk4vaU4rTnBnSmZycGlVeEt3TERkb0JpWTZ0bmpoeHVqTGV5azBDaV
prUVVVSXdkHKzdKOWhVNWpJUE1lgo1WTMrcmZQZm1t3p4UzJ4amxmQ0ljZGpkR2YxL1BhelfUaTUxbjJDNc9BcWtzZzh6bFJRTDRTZNVed0J
XNHZLzRybnJaYVvhil0dqCnV6aWU2eDjvVXISY1BHam5id3VUdWdCQnp3UW1OYTg3d2ZwdDhTQzNJTIpkNBH3NVcycWNUTFEyYkN2ZDJWcE
NqclRUSGxTU1hjMEYKYXNVYXdkWWJMRSsXOUhvZDBicjA4NDVKVlpjPTWvZHM6WDUwOUNlcnRpZmljYXRlPg0KCQkJPC9kcZpYNTA5RGFOYT
4NCgkJPC9kcZpLXlJbmZvPg0KCTWvZHM6U2lnbmF0dXJlPg0KPC9Bc3NlcnRpb24+",
"refresh_token": null,
"issued_token_type": "urn:ietf:params:oauth:token-type:saml1",
"scope": "",
"token_type": "N_A",
"expires_in": 43500
}

```



5.3 End user workflow

The end user needs to perform some actions in order to allow the client getting a SAML HOK token:

1. Authentication

The end user must select one of the authentication methods proposed.

The screenshot shows the CSAM login interface. At the top, there is a language selector with 'nl', 'fr', 'de', and 'en', and a '.be' domain indicator. Below this is the CSAM logo and the title 'Aanmelden bij de online overheid'. The main heading is 'Kies uw digitale sleutel om aan te melden', with a 'Hulp nodig?' link. The page is divided into two sections: 'Digitale sleutel(s) met eID of digitale identiteit' and 'Digitale sleutel(s) met beveiligingscode en gebruikersnaam + wachtwoord'. The first section contains two buttons: 'AANMELDEN met eID kaartlezer' and 'AANMELDEN via itsme', with a link 'Je itsme-account aanmaken' below the second button. The second section contains one button: 'AANMELDEN met beveiligingscode via mobiele app'. At the bottom, there is a 'Belangrijk om te weten!' section with a paragraph explaining that logging in with a digital key provides access to other government services.

nl fr de en .be

CSAM Aanmelden bij de online overheid

Kies uw digitale sleutel om aan te melden [Hulp nodig?](#)

Digitale sleutel(s) met **eID** of **digitale identiteit**

AANMELDEN met eID kaartlezer

AANMELDEN via itsme

[Je itsme-account aanmaken](#)

Digitale sleutel(s) met **beveiligingscode** en **gebruikersnaam + wachtwoord**

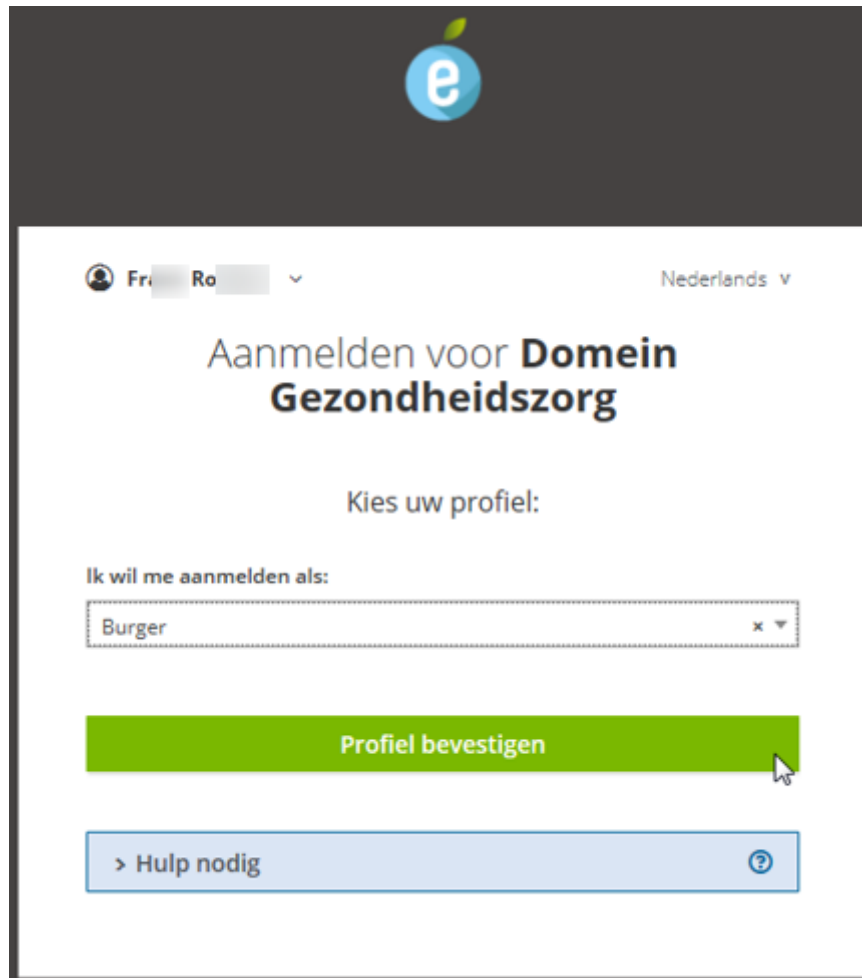
AANMELDEN met beveiligingscode via mobiele app

Belangrijk om te weten!

Eens u aangemeld bent met een digitale sleutel, hebt u via CSAM automatisch toegang tot andere onlinediensten van de overheid die met dezelfde sleutel beveiligd zijn. Dit geldt zolang uw browservenster actief is.

2. Profile selection

The end user has to select one of the available profiles in the dropdown list.



If the end user has no supported profile for the token eXchange, he will not be able to select any profile and the IDP will warn this end user.

3. Consent
The end user will have to give his/her consent



Nederlands v

IAM eXchange - default test client

heeft toegang nodig tot de volgende gegevens bij het domein van de gezondheidszorg

Hiermee kan deze toepassing het volgende doen:

- Uw rijksregisternummer raadplegen
- Uw profiel raadplegen (professional in de gezondheidszorg, patiënt, ouder van een kind, volmachthouder voor een andere patiënt, ...)
- Uw profiel bekijken als patiënt of beroepsbeoefenaar in de gezondheidszorg of als volmachthouder of als ouder
- Digitale sleutels aanmaken voor één profiel gekozen door de gebruiker waarmee gegevens kunnen worden opgehaald bij -of verzonden naar diensten van het eHealth platform
- Digitale sleutels aanmaken voor een lijst van profielen van een geauthentiseerde gebruiker waarmee gegevens kunnen worden opgehaald bij -of verzonden naar diensten van het eHealth platform

▼ Hulp nodig ?

Zorg ervoor dat u **IAM eXchange - default test client** vertrouwt.

U deelt mogelijk gevoelige gegevens met deze toepassing. U kan de toegang altijd bekijken of verwijderen in uw [account](#).

Meer info ? [Klik hier](#)

The end user can revoke his/her consent by using the account clients.

4. Choose another profile
If the Trusted Platform uses */profiles*, the end user may select within the application (Trusted Platform) some other profiles for the eXchange.

5.4 Process overview for technical clients

5.4.1 eHealth platform authentication

In order to use IAM eXchange service, the technical clients must be able to obtain an accessToken. The client must use the client credentials flow (see IAM Connect – Mobile integration for more information).

One role is available:

- profile-specific: this role must be present in the accessToken in order to retrieve the list of profiles for one individual person (identified by a SSIN)

The client MUST request the client scope *iam:exchange:profilesprofilespecific* during his onboarding.

5.4.2 GET /profiles/{ssin}

The operation must be used to retrieve the list of profiles for any individual identified by a SSIN.

In this section, we assume that the client is already configured and recognised by the eHealth platform (see section 5.4.1).

5.4.2.1 Request

Only one input must be provided : SSIN

Element	Description
ssin	SSIN of the individual person

Example (for ssin 12345678912):

GET <https://api.ehealth.fgov.be/iam/v2/profiles/12345678912>

5.4.2.2 Response

If the operation succeeds, the result may contain a list of profiles (JSON format).

Element	Description
ssin	SSIN of the authenticated end user
children	Child/children of the authenticated end user. Each child is represented with the following elements - SSIN - firstName - lastName Child/children is not listed if the Trusted Platform is not concerned by this profile's subset.
mandators	Mandator(s) of the authenticated end user The mandate type(s) detected are specified in serviceNames for each mandator. Each service name listed corresponds to exactly one mandate type. For examples : <i>medicaldatamanagement</i> (<i>Gestion des données de santé/ Beheer van gezondheidsgegevens</i>), <i>recipe</i> (<i>Mandat de prescription/ Voorschriftenvolmacht</i>) Mandators are not listed if the Trusted Platform is not concerned by this profile's subset.
organizations	Organizations related to the authenticated end user.



	Organizations are not listed if the Trusted Platform is not concerned by this profile's subset.
--	---

Example without profile found:

```
{
  "ssin": "12345678912"
}
```

Example with mandates and children:

```
{
  "ssin": "12345678912",
  "children": [ {
    "lastName": "Doe",
    "firstName": "Junior1",
    "ssin": "23456789123"
  },
  {
    "lastName": "Doe",
    "firstName": "Junior2",
    "ssin": "34567891234"
  }
],
  "mandators": [ {
    "firstName": "Grandfather",
    "lastName": "Doe",
    "ssin": "01234567891",
    "name": "Doe Grandfather",
    "serviceNames": ["medicaldatamanagement"]
  }
]
}
```

5.4.3 GET /profiles/{ssin} with pseudonyms

The operation must be used to retrieve the list of profiles for any individual identified by a pseudonymised SSIN.

In this section, we assume that the client is already configured and recognised by the eHealth platform (see section 5.4.1).

The pseudonym in transit (issued by domain ehealth_v1) must be converted to one-line argument.

The conversion result can then be provided in input.

Two conversion approaches exist (Base64 or SEC1).

The encoding used in the request will be used in the response.

For more information about this conversion, please consult the document Blinded pseudonymisation REST service - Cookbook (Annex Pseudonym in Transit: Conversion to one-line Argument)

5.4.3.1 Request

Only one input must be provided: SSIN

Element	Description
ssin	Pseudonymised SSIN (from domain ehealth_v1) of the individual person



5.4.3.2 Response

If the operation succeeds, the result may contain a list of profiles (JSON format).

Element	Description
ssin	Pseudonymised SSIN (from domain ehealth_v1)
children	Child/children of the authenticated end user. Each child is represented with the following elements - Pseudonymised SSIN (from domain ehealth_v1) - firstName - lastName Child/children is not listed if the Trusted Platform is not concerned by this profile's subset.
mandators	Mandator(s) of the authenticated end user The mandate type(s) detected are specified in serviceNames for each mandator. Each service name listed corresponds to exactly one mandate type. For examples: <i>medicaldatamanagement (Gestion des données de santé/ Beheer van gezondheidsgegevens), recipe (Mandat de prescription/ Voorschriftenvolmacht)</i> Mandators are not listed if the Trusted Platform is not concerned by this profile's subset. If the mandator is an individual person (citizen), the mandator will be represented with the following elements: - Pseudonymised SSIN (from domain ehealth_v1) - firstName - lastName
organizations	Organizations related to the authenticated end user. Organizations are not listed if the Trusted Platform is not concerned by this profile's subset.

Example without profile found:

```
{
  "ssin":
    "eyJpZC16IjdiMGUwMWZjLW12NzQtNDcyOC05YjhhLTJkY2RIMDE2ZDZhOCIsImRvbWVpbiI6ImVoZWZsdGhfdjEiLCJcnYiOiJQLTUyMSIsImhhdCI6MTc4OTA0OTc2MSwiZXhwIjoxNzg5MDUwMzYxLCJ4IjojQVZ3cVZ3dnJhRk00SkRQZErN2ljQlNaYXA1R3IzZ21PRHBNR3Z4Uktzci9Ob1Z2V2RqOWpyTis4ZG9UUUYwL1ImQ21vYVRpekFJY2ZCUFNtaFd5UCIsInkiOiJRWtNwUxSlcxakxYYWZDdUVOVU5uSFBWREFTRU1VMUNQSVpPRVQxQUhxNlUthcFoNC94TjRpZTc2a3hZcnp0cEM5dHhQUiRGNGkVYR3pqRWh3TU94MGc9liwidHJhbnNpdEluZm8iOiJleUpoZFdaU9pSm9kSFJ3Y3pvdkwYRndhUzFwYm5RdVpXaGxZV3gwYUM1bVoyOTJMbUpsTDNCelpYVmtieTkyTVM5a2lyMWVhVzV6TDJWb1pXRnNkR2hmZGpFaUxDSmxibU1pT2IKQk1qVTJSMESOSWI3aVpYaHdJam94TnpnNU1EVXdNell4TENKcFIYUWIPakUzT0Rrd05EazNOakVzSW1Gc1p5STZlbVJwY2Jlc0ltdHBaQ0k2SW1Zek5HVXIOak5sTFRSbE9EWXROREZsTmkwNU1qUTNMV1ZtWXPNeVl6ZzJNVFZsTUNKOS4uSzlPamJmMEILVDBBdkhvOC5UTnM0QzU5aENWa3NLZFNvX21GY2dKYkdSaGppUE1Nc0xydkp3cF9KMEdNVnNCRERSdWtBNWp5VWR5TVdWZ1JFOHFKbjRrYi1iRINKNWxLSGtsMVBVPV41S2NYTGJlb3dOWIVCa1VSeUJtMWg1cmdVNmMwbzJFa1VUazBhYXE0T0lkWG80WnJNekZMWUNpTWdnbk5LLVFoeEhSRnRzYkdLSFV3bG4zWW9yekx3TTRUeHJrMjUuYTI5Zm1WLVBaUlNjczVWc2VTRDJ5USJ9"
```

Example with mandates and children:

```
{
  "ssin":
    "eyJpZC16IjdiMGUwMWZjLW12NzQtNDcyOC05YjhhLTJkY2RIMDE2ZDZhOCIsImRvbWVpbiI6ImVoZWZsdGhfdjEiLCJcnYiOiJQLTUyMSIsImhhdCI6MTc4OTA0OTc2MSwiZXhwIjoxNzg5MDUwMzYxLCJ4IjojQVZ3cVZ3dnJhRk00SkRQZErN2ljQlNaYXA1R3IzZ21PRHBNR3Z4Uktzci9Ob1Z2V2RqOWpyTis4ZG9UUUYwL1ImQ21vYVRpekFJY2ZCUFNtaFd5UCIsInkiOiJRWtNwUxSlcxakxYYWZDdUVOVU5uSFBWREFTRU1VMUNQSVpPRVQxQUhxNlUthcFoNC94TjRpZTc2a3hZcnp0cEM5dHhQUiRGNGkVYR3pqRWh3TU94MGc9liwidHJhbnNpdEluZm8iOiJleUpoZFdaU9pSm9kSFJ3Y3pvdkwYRndhUzFwYm5RdVpXaGxZV3gwYUM1bVoyOTJMbUpsTDNCelpYVmtieTkyTVM5a2lyMWVhVzV6TDJWb1pXRnNkR2hmZGpFaUxDSmxibU1pT2IKQk1qVTJSMESOSWI3aVpYaHdJam94TnpnNU1EVXdNell4TENKcFIYUWIPakUzT0Rrd05EazNOakVzSW1Gc1p5STZlbVJwY2Jlc0ltdHBaQ0k2SW1Zek5HVXIOak5sTFRSbE9EWXROREZsTmkwNU1qUTNMV1ZtWXPNeVl6ZzJNVFZsTUNKOS4uSzlPamJmMEILVDBBdkhvOC5UTnM0QzU5aENWa3NLZFNvX21GY2dKYkdSaGppUE1Nc0xydkp3cF9KMEdNVnNCRERSdWtBNWp5VWR5TVdWZ1JFOHFKbjRrYi1iRINKNWxLSGtsMVBVPV41S2NYTGJlb3dOWIVCa1VSeUJtMWg1cmdVNmMwbzJFa1VUazBhYXE0T0lkWG80WnJNekZMWUNpTWdnbk5LLVFoeEhSRnRzYkdLSFV3bG4zWW9yekx3TTRUeHJrMjUuYTI5Zm1WLVBaUlNjczVWc2VTRDJ5USJ9"
```



```
"children": [ {
  "lastName": "Doe",
  "firstName": "Junior1",
  "ssin":
```

```
},
{
  "lastName": "Doe",
  "firstName": "Junior2",
  "ssin":
```

```
    },
    "mandators": [ {
      "firstName": "Grandfather",
      "lastName": "Doe",
      "ssin":
```

```
"name": "Doe Grandfather",
"serviceNames": ["medicaldatamanagement"]
}
```



5.5 Reference implementation

5.5.1 General description

The actual solution is based on RFC 8693 'OAuth 2.0 Token Exchange'

6. Risks and security

6.1 Risks & safety

6.1.1 End user consent

End user must give his consent to the client (the Trusted Platform) prior this client can use the end user credentials. The consent mechanism is present by default. No client will be able to act as the end user if the latter has not provided his consent once.

The end user can revoke his/her consent at any time.

If the user removes his consent for one client, this client cannot request a new access token and cannot exchange the token. But the client can still use a valid SAML token previously obtained.

6.1.2 Token validity period

When the end user gives his consent, the client (the trusted platform) can request a SAML HOK token during a given period.

The SAML HOK obtained has a limited validity period defined to 12 hours.

A more comprehensive set of security requirements is given in “IAM eXchange Annex A – Security commitment from the Trusted Platform”, available on the portal.

(See <https://www.ehealth.fgov.be/ehealthplatform/nl/service-i.am-identity-access-management>)

This document should be signed by a legal representative of the entity or by the information security consultant.

7. Test and release procedure

7.1 Procedure

This chapter explains the procedures for testing and releasing an application in acceptance or production.

7.1.1 Initiation

If you intend to use the eHealth platform service, please contact info@ehealth.fgov.be. The project department will provide you with the necessary information and mandatory documents.

7.1.2 Development and test procedure

You have to develop a client in order to connect to our WS. Most of the required info to integrate is published in the technical library on the portal of the eHealth platform.

Upon request, the eHealth platform provides you test cases (See Request testcase template) in order for you to test your client before releasing it in the acceptance environment.

7.1.3 Release procedure

When development tests are successful, you can request to access the acceptance environment of the eHealth platform. From this moment, you start the integration and acceptance tests. The eHealth platform suggests testing during minimum one month.

After successful acceptance tests, the partner sends his test and performance results with a sample of “eHealth request” and “eHealth answer” by email to his point of contact at the eHealth platform.

Then the eHealth platform and the partner agree on a release date. The eHealth platform prepares the connection to the production environment and provides the partner with the necessary information. During the release day, the partner provides the eHealth platform with feedback on the test and performance tests.

For further information and instructions, please contact: integration-support@ehealth.fgov.be.

7.1.4 Operational follow-up

Once in production, the partner using the eHealth platform service for one of his applications will always test first in the acceptance environment before releasing any adaptations of the application in production. In addition, he will inform the eHealth platform on the progress and test period.

7.2 Test cases

The eHealth platform recommends performing tests for all of the following cases:

For clients defined as a TrustedPlatorfm (with eXchange scope only)

- Request a SAML HOK token as a citizen (or as parent or as mandate holder or as healthcare professional)
- Use the SAML HOK token to call a protected service provider (example: WS SOAP KGSS)

For clients defined as a TrustedPlatorfm (with eXchange scope and profile scope)

- Consult with success a list of profiles (with one or more profiles)
- Request a SAML HOK token with one valid profile
- Use the SAML HOK token to call a protected service provider (example: WS SOAP KGSS)

For technical clients using client credentials flow:

- Consult with success a list of profiles for a valid SSIN (with one or more profiles)

8. Error and failure messages

Error codes originating from the eHealth platform for the IAM eXchange service are defined in the swagger file.

In the table below, you can find specific error messages for */protocol/oauth/tokenExchange*.

HTTP status code	Error code	Error message	Recommendation
401	unauthorized_client	SubjectToken Access Denied	Enduser may have revoked his/her consent or SubjectToken is expired. A new accessToken must be generated. Contact eHealth support for investigation if the problem persists.
400	invalid_client	Reason : error while invoking webaccess endpoint	Contact eHealth support for investigation.
400	invalid_request	ActorToken Access Denied: Authorized Party of subjectToken <code>{subjectToken.azp}</code> must be the same as issuer actorToken <code>{actorToken.iss}</code>	Contact eHealth support for investigation if it worked previously.
400	invalid_request	SubjectToken Access Denied: realm_access role token-exchange missing.	Contact eHealth support for investigation if it worked previously.
400	invalid_request	Invalid input for field actor_token_type	Review and adapt the input (as described in this document).
400	invalid_request	Invalid input for field actor_token	Review and adapt the input (as described in this document).
400	invalid_request	ActorToken Access Denied: client <code>{issuer}</code> not allowed (wrong signing algorithm)	Wrong signing algorithm (in JWT header). Review it and adapt the input (as described in this document).
400	invalid_client	ActorToken Access Denied: client <code>{issuer}</code> not allowed	ActorToken used is not known. Contact eHealth support for investigation if it worked previously.
400	invalid_request	ActorToken Access Denied: client <code>{issuer}</code> not allowed (wrong certificate)	Certificate used to generate the actorToken is not known at eHealth. Follow the certificate renewal procedure: NL: eHealth-certificaten eHealth-platform , formulier "Certificate Management: Renewal of public key" FR: Certificats eHealth Platform eHealth , formulaire "Certificate Management: Renewal of public key"
400	invalid_client	ActorToken expired	A new actorToken must be generated.
400	invalid_request	Invalid input for field audience	Review and adapt the input (as described in this document).

400	unsupported_grant_type	Invalid input for field grant_type	Review and adapt the input (as described in this document).
400	invalid_request	Invalid input for field requested_token_type	Review and adapt the input (as described in this document).
400	invalid_request	Invalid input for field resource	Review and adapt the input (as described in this document).
400	invalid_scope	Invalid input for field scope	Review and adapt the input (as described in this document).
400	invalid_request	Invalid input for field subject_token	Review and adapt the input (as described in this document).
400	invalid_request	Invalid input for field subject_token_type	Review and adapt the input (as described in this document).
400	invalid_request	SubjectToken Access Denied: untrusted issuer [\${subjectToken.iss}]	Your client is trying to use a token not suitable for this environment. Review your configuration.
400	Invalid_request	SubjectToken Access Denied: Authentication level not satisfied.	Contact eHealth support for investigation if it worked previously.
401	unauthorized_client	ActorToken Access Denied: failed to resolve attributes (Profile \${profile})	Contact eHealth support for investigation.
401	unauthorized_client	ActorToken Access Denied: failed to determine profile (Profile option type \${profileOptionType})	Contact eHealth support for investigation.
401	unauthorized_client	SubjectToken Access Denied: Invalid authentication level.	Contact eHealth support for investigation.
500	unknown	Reason: unable to resolve signing key	Contact eHealth support for investigation.
500	unknown	Reason: error while invoking account endpoint	Contact eHealth support for investigation.
500	unknown	Reason: \${failureStatusMessage}	Contact eHealth support for investigation.
500	unknown	Reason: unable to extract assertion from backend (empty response)	Contact eHealth support for investigation.
500	unknown	Reason: unable to extract assertion from backend (invalid response)	Contact eHealth support for investigation.
500	unknown	Reason: unable to encode assertion	Contact eHealth support for investigation.
500	unknown	Reason: wrong issued_token_type received from backend	Contact eHealth support for investigation.
500	unknown	Reason: wrong token_type received from backend	Contact eHealth support for investigation.
500	unknown	Reason: unable to determine assertionLifetime	Contact eHealth support for investigation.

Error example (http status code 400):

```
{
  "error" : "invalid_client",
  "error_description" : "ActorToken expired",
  "id" : "Id-1490f45e9e886f6fc635cd15"
}
```

In the table below, you can find specific error messages for */profiles/{ssin}*.

HTTP status code	Title	Detail	Recommendation
400	invalid_client	Invalid parameter: \${input} is not a valid SSIN.	Use a correct and valid input (SSIN)

Error example (http status code 400):

```
{
  "type": "https://www.gcloud.belgium.be/rest/problems/badRequest",
  "title": "Bad Request",
  "status": 400,
  "detail": "Invalid parameter: 'a' is not a valid SSIN.",
  "id": "Id-d5c7356182abed5af3d76ce2"
}
```