I.AM Connect
Client registration
Version 1.1

This document is provided to you free of charge by the

eHealth platform

Willebroekkaai 38

38, Quai de Willebroek

1000 BRUSSELS

Table of contents

1. Document management

1.1 Document history

Version	Date	Author	Description of changes / remarks
1.0	08/06/2017	eHealth platform	Initial version
1.1	12/08/2019	eHealth platform	Update

2. Introduction

2.1 Goal of the service

eHealth I.AM Connect allows client to access REST services for the eHealth domain. The eHealth platform divides clients and services into security realms. Each client or service needs to be registered in a realm. Clients and services that need to connect with one another, need to be defined in the same realm.

This document will serve as base to register the client and must contain all information required to add the partner to the federation.

Information in bold are required.

2.2 Goal of the document

This document contains all information, necessary to integrate with one of the eHealth environments.

Partners that want to use eHealth I.AM Connect must fill out this form, once for each client. Each client is linked to one realm. Realm registration form must be sent if the realm does not exist yet.

Available environments are:

- Integration
- Acceptation
- Production

The forms in this document must also be used to register updates (change of name, url, attributes, …).

For each registered update a new record should be added to the version table.

3. Support

3.1 Contact

eHealthDevSupport@ehealth.fgov.be

3.2 Support in general

For issues in production only

eHealth ContactCenter:

- Phone: 02/788 51 55
- 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)

FOR PARTNERS AND SOFTWARE DEVELOPERS ONLY

- For business issues please contact: ***info@ehealth.fgov.be***
- For technical issues in production please contact: ***support@ehealth.fgov.be*** ***or call 02/788 51 55***
- For technical issues in acceptance please contact: ***Integration-support@ehealth.fgov.be***

4. General information

A partner registered in the eHealth I.AM Federation is known as a Service Provider. He is registered with a unique ID (EntityID) you may choose yourself or ask eHealth Development team to give you one.

Realm	
Name	Medikit
Client	
ClientID	medikit-website
Name	Medikit
Description	Simple and efficient solution for your eHealth obligations
Consent required[1]	□

Depending on the client type, you will have to fill one of the subsection in section 5 Specific Information.

[1] If this is on, then user will get a consent page, which asks the user if he grants access to that application. It will also display the metadata that the client is interested in so that the user knows exactly what information the client is getting access to.

5. Specific Information

5.1 Access type: confidential

Confidential access type is for server-side clients who need to perform a browser login and require a client secret when they turn an access code into an access token, (see Access Token Request in the OAuth 2.0 spec for more details – Technical Specifications IAM Connect).

This type should be used for server-side applications.

Flows	
Standard Flow enabled[2]	□
Implicit Flow enabled[3]	□
Service Accounts enabled[4]	□

If Standard Flow and/or Implicit Flow are/is enabled, fill out this section:

URL	
Root URL	
Valid redirect URIs[5] **(separated by ;)**	
Base URL[6]	
Web Origins (separated by ;)	

[2] This enables Authorization Code Flow fot this client.

[3] This enables support of Implicit Flow for this client.

[4] This enables support of Client Credentials grant for this client.

[5] Valid URI pattern a browser can redirect to after a successful login or logout. Make your redirect URIs as specific as feasible.
Wildcards are allowed at the end of a URI.

[6] Default URL to use when the authorization server needs to redirect or link back to the client.

5.1.1 Credentials

Confidential access type requires a client secret when they turn an access code into an access token.

Signed JWT are used by default. We can import the public key (pem format) or we can configure JWKS URL[7].
When the key is about to change, the new key must be sent to us at least one month before the change. Key roll-over is not supported.
To know how to expose a key using JWK, you can refer to RFC7517[8].

Credentials	
JKWS URL	
Public key[9]	

5.2 Access type: public

Public access type is for client-side clients that need to perform a browser login. With a client-side application there is no way to keep a secret safe. Instead it is very important to restrict access by configuring correct redirect URIs for the client.

Public access type is not allowed for clients that handle medical data.

[7] When JWKS URL is provided, new keys will be always downloaded again when client generates new key pair.

[8] ***https://tools.ietf.org/html/rfc7517***

[9] Base64 encoded format of the certificate used to sign the JWT.

Flows	
Standard Flow enabled[10]	□
Implicit Flow enabled[11]	☒
Service Accounts enabled[12]	□

If Standard Flow and/or Implicit Flow is enabled, fill out this section:

URL	
Root URL	https://medikit.be
Valid redirect URIs[13] **(separated by ;)**	http://localhost:8080;https://localhost:8080;https://medikit.be
Base URL[14]	https://medikit.be
Web Origins (separated by ;)	http://localhost:8080;https://localhost:8080;https://medikit.be

5.3 Access type: bearer only

“Bearer-only access” type means that client will only verify bearer tokens and cannot obtain the tokens itself. If this is turned on, this application cannot participate in browser logins.

5.3.1 Credentials

Credentials are required if the client will use the validation or authorization flows.

Signed JWT are used by default. We can import the public key (pem format) or we can configure JWKS URL[15].

When the key is about to change, the new key must be sent to us at least one month before the change. Key roll-over is not supported.

To know how to expose a key using JWK, you can refer to RFC7517[16].

[10] This enables Authorization Code Flow fot this client.

[11] This enables support of Implicit Flow for this client.

[12] This enables support of Client Credentials grant for this client.

[13] Valid URI pattern a browser can redirect to after a successful login or logout. Make your redirect URIs as specific as feasible.
Wildcards are allowed at the end of a URI.

[14] Default URL to use when the authorization server needs to redirect or link back to the client.

[15] When JWKS URL is provided, new keys will be always downloaded again when client generates new key pair.

[16] ***https://tools.ietf.org/html/rfc7517***

Credentials	
JKWS URL	https://api.medikit.be/jwks
Public key[17]	

5.4 Mappers

In some cases, applications that receive ID Tokens or Access Tokens may need or want different user data than the basic one. Protocol mappers may be defined to map user data into protocol claims.

ID Tokens and Access Tokens contents are described in the IAM Connect technical specifications.

Don’t hesitate to contact ***ehealthdevsupport@ehealth.fgov.be*** to describe your specific needs.

[17] Base64 encoded format of the certificate used to sign the JWT.