RIZIV-INAMI

eHealth

RECIP-e
FHIR Cookbook

Version 3.0

Smals
ICT for society

Recip-e FHIR cookbook

Contents

1. DOCUMENT MANAGEMENT 3

1.1. History 3

1.2. Purpose 4

1.3. Support 4

1.4. FHIR high risk of future change 4

2. MAJOR DIFFERENCES WITH KMEHR 6

2.1. Actor services 6

2.2. Operations 6

3. ARCHITECTURAL PRINCIPLES 8

3.1. Pseudonymisation 8

3.2. Authentication 9

3.3. Authorization 15

4. FHIR INTERACTIONS 22

4.1. CapabilityStatement 22

4.2. OpenAPI Definition 22

4.3. Supported resource types 23

4.4. Use of pseudonyms 23

4.5. Error codes 24

5. HEADERS SUMMARY 27

5.1. Request Headers 27

5.2. Response Headers 27

6. FLOWS 29

6.1. Status flow 29

6.2. General flow 29

7. BUSINESS INFORMATION 33

7.1. Business rules 33

7.2. Privacy flags 35

1. Document management

1.1. History

Version	Status	Date	Author(s)	Changes
0.1	Draft	10.10.2025	Dries Vanstappen	First version
0.2	Draft	01.04.2026	Dries Vanstappen	First amendaments
0.3	Draft	20.04.2026	Dries Vanstappen	Addition of business rules and status flow
1.0	Official version	04.06.2026	Dries Vanstappen, Hendrik Janssen	FHIR-A-THON feedback
3.0	Official version	12.11.2026	Dries Vanstappen, Hendrik Janssen	Feedback from various sources Addition of KMEHR comparison Updated authentication, client onboarding and access-token guidance; clarified FHIR interactions, headers and error handling; sanitized examples.

1.2. Purpose

This cookbook is for developers, analysts and software producers involved in the integration into their software solution of the services delivered by Recip-e, specifically for the FHIR data.

Fast Healthcare Interoperability Resources (FHIR) is a next generation standards framework created by HL7. Solutions are built from a set of modular components called “Resources”. These resources can easily be assembled into working systems that can solve clinical and administrative problems. More about FHIR can be read on the FHIR website: http://hl7.org/fhir/.

FHIR has defined general, international definitions of resources. Nevertheless, within the standard, it is possible to create a local variant based on those international resources. In a co-creation with the industry, the healthcare-sector, HL7 and the different governments involved eHealth has created a Belgian version of international resources. These Implementation Guides can be retrieved on the eHealth website: https://www.ehealth.fgov.be/standards/fhir/index.html.

Note: For now the build version of BeMedicationPrescription is to be used: BeMedicationPrescription - Medication v1.1.1
The official publication of the BeMedicationPrescription will be used in the future once it is available.

The goal of the cookbook is to document how a software can accomplish a technical integration with Recip-e in FHIR.

The business use cases and data models themselves are not part of this cookbook.

This document is composed of three sections:

- Section Principles explains the architectural principles as authentication, authorization and security.
- Section FHIR interactions provides an overview of the supported interactions on the server.
- Section Examples of CRUD operations presents the structure of the requests to perform CRUD operations on the server.

The latest version of this CookBook can always be found at https://github.com/smals-belgium/shared-digital-medication-prescription.

More information about eHealth registration for softwarevendors of healthcare software can be found at Technische informatie en registratie voor softwareleveranciers van software voor zorgverleners | RIZIV

1.3. Support

For issues in both acceptance and production, please use the form on technicalsupport.recip-e.be to contact us. In order to guarantee the best possible support, please provide as many details when reporting an issue: environment, timestamp, URL, method, request parameters/body, response code/body, … .

1.4. FHIR high risk of future change

We foresee that some things might change in the future because we are limited by the implementation and publication by eHealth of FHIR standards, resources and profiles and bound to implement them.
Below you’ll find a list of things of which we think that have a high risk of changing in the future:

- The way to indicate types and/or classifications of the prescribed product
- The unit system used for units of measure which now is http://unitsofmeasure.org
- The system used in the recip-e identifier
- The usage of the BREW workflow (hl7-be/workflow: BREW - Belgian Requests & Execution Workflows (FHIR IG))
- SAMv2 validation of magistral formulas and limitation on prescribing certain medicinal products as AMPP

- Link to a MedicationLine
- Usage of BeMedication

2. Major differences with KMEHR

There are numerous differences between the new way of working using FHIR and the old way of working using KMEHR.
The major differences between both ways of working are detailed below.
You'll notice that we have added a section "Difference with KMEHR" to some chapters in this CookBook to detail specific differences concerning the things mentioned in that chapter.

2.1. Actor services

In KMEHR all actors have their own services.
There is a service for patients, a service for executors and one for prescribers.
In FHIR this is not the case.
All actors use the same service and we have a system which determines what type of actor is logged in and which services and capabilities are then provided to them.

2.2. Operations

Most operations that exist in KMEHR have a relevant operation in FHIR.
However some operations have been removed.
Below you'll find a mapping of the existing KMEHR operations to their related FHIR operation.

Actor	KMEHR operation	FHIR operation(s)	Notes
Executor	ExecutorServices.createFeedback	To be decided	
Executor	ExecutorServices.getPrescriptionForExecutor	searchMedicationRequest	
Executor	ExecutorServices.getPrescriptionStatus	searchMedicationRequest	
Executor	ExecutorServices.listNotifications	To be decided	
Executor	ExecutorServices.listOpenPrescriptions	searchMedicationRequests	
Executor	ExecutorServices.listOpenPrescriptionsBTG	searchMedicationRequests	
Executor	ExecutorServices.listRelations	/	Use IAM ListProfiles directly
Executor	ExecutorServices.listReservations	To be decided	
Executor	ExecutorServices.listRidsHistory	searchMedicationRequests	
Executor	ExecutorServices.listRidsHistoryBTG	searchMedicationRequests	
Executor	ExecutorServices.listRidsInProcess	searchMedicationRequests	
Executor	ExecutorServices.markAsArchived	/	No longer used
Executor	ExecutorServices.markAsDelivered	Task: code = "fulfill"	
Executor	ExecutorServices.markAsUnDelivered	/	No longer used
Executor	ExecutorServices.putRidsInProcess	/	No longer used
Executor	ExecutorServices.revokePrescription	Task: code = "abort"	
Patient	PatientServices.createReservation	To be decided	
Patient	PatientServices.getPrescription	searchMedicationRequest	
Patient	PatientServices.getPrescriptionStatus	searchMedicationRequest	
Patient	PatientServices.getVision	/	Part of searchMedicationRequest
Patient	PatientServices.listOpenPrescriptions	searchMedicationRequests	
Patient	PatientServices.listOpenRids	searchMedicationRequests	
Patient	PatientServices.putVision(prescribers)	Task: code = "change"	
Patient	PatientServices.putVisionForPatient	Task: code = "change"	
Patient	PatientServices.revokePrescription	Task: code = "abort"	
Prescriber	PrescriberServices.createPrescription	CreateMedicationRequest createBundle	
Prescriber	PrescriberServices.getPrescriptionForPre	searchMedicationRequest	

	scriber		
Prescriber	PrescriberServices.getPrescriptionStatus	searchMedicationRequest	
Prescriber	PrescriberServices.listFeedbacks	To be decided	
Prescriber	PrescriberServices.listOpenRids	searchMedicationRequests	
Prescriber	PrescriberServices.listPrescriptions	searchMedicationRequests	
Prescriber	PrescriberServices.listRidsHistory	searchMedicationRequests	
Prescriber	PrescriberServices.putVision	Task: code = “change”	
Prescriber	PrescriberServices.revokePrescription	Task: code = “abort”	
Prescriber	PrescriberServices.sendNotification	To be decided	
Prescriber	PrescriberServices.updateFeedbackFlag	/	No longer used

3. Architectural principles

Recip-e has selected some specifications in order to guarantee the regulations on the data shared in the ecosystem.

3.1. Pseudonymisation

To ensure privacy regulation, Recip-e has chosen to use pseudonyms for identifiers of patients in order to guarantee security around the medical data of each patient that is stored in the database.

To accomplish pseudonymization, eHealth's pseudo service, as trusted party, will be used to pseudonymize, convert and identify data.

Recip-e will expect incoming data to contain pseudonyms of specific data elements. Please refer to Use of pseudonyms on how to accomplish this in FHIR. Software integrators will need to onboard with eHealth to be able to pseudonymize data using the pseudo service. Access needs to be granted for the software package so it can pseudonymize, identify and/or convert to the UHMEP domain.

More details can be found on the eHealth documentation:

- Pseudonymisering & Anonimisering (NL)
- Pseudonymisation & Anonymisation (FR)

Technical information about the UHMEP domain can be retrieved here (a valid access token is required):

- uhmep_v1 domain (ACC)
- uhmep_v1 domain (PRD)

Smals recommends usage of the Pseudonymisation Helper library:

- Java Library
- .NET Library
- Javascript Library

This is a code example on how to construct a SEC1 UHMEP pseudonym in transit using the Java library:

```
public final class UhmepPseudonymizer {

    private static final String UHMEP_DOMAIN = "uhmep_v1";
    private static final String V2_URN = "urn:be:fgov:ehealth:pseudo:v2:";

    private UhmepPseudonymizer() {}

    public static String pseudonymize(
            final String ssin,
            final PseudonymisationHelper helper) {

        final PseudonymInTransit pseudonym = helper.getDomain(UHMEP_DOMAIN)
                .get()
                .valueFactory()
                .from(ssin)
                .pseudonymize()
                .get();

        return V2_URN + pseudonym.asString();
    }
}
```

3.1.1. Difference with KMEHR

In KMEHR the whole prescription was encrypted, whereas in FHIR only certain fields will be pseudonymized, namely the fields which contain personal data from the patient or mandateholder.

The following fields are expected to be transferred pseudonymized:

- Citizen SSIN
    - Patient SSIN
    - Mandateholder SSIN

3.2. Authentication

3.2.1. eHealth IAM

Recip-e expects access tokens from the authentication service of eHealth: I.AM Connect.

The Identity Access Management (IAM) Connect service is an authentication service provided by the eHealth platform. Its goal is to authenticate users and client applications and provide access tokens that allow access to services such as Recip-e.

eHealth IAM uses the OpenID Connect layer on top of the OAuth2.0 protocol, where it provides an access token for the client who can use it to identify itself at various eHealth services. Send the access token intended for the destination service in the Authorization header as a Bearer credential. Recip-e requests use the exchanged healthcare access token or the separately requested M2M access token without pseudonymization capabilities.

Individual prescribers and citizens use the healthcare realm to obtain their access tokens. Pharmacists connect to Recip-e via their pharmacy organization, using the M2M (Machine-to-machine) realm with the pharmacy client. In hospital context, prescribers use the M2M realm with the hospital client. Hospital pharmacies use their own pharmacy M2M client; they cannot reuse the hospital's client.

Actor flows and access-token renewal:

Actor/context	**Realm**	**OICD Flow**	**Access token sent to Recip-e**	**Renewal**
Citizen	healthcare	authorization_code; human login	Exchanged access token; target audience nihdi-uhmep-fhir-patient	Refresh the original access token, then exchange again
Individual prescriber	healthcare	authorization_code; human login	Exchanged access token; target audience nihdi-uhmep-fhir-hcp	Refresh the original access token, then exchange again
Hospital prescriber	M2M	client_credentials; hospital client and signed client assertion	Separately requested access token without pseudonymization capabilities	Request a new access token
Public pharmacy	M2M	client_credentials; pharmacy client and signed client assertion	Separately requested access token without pseudonymization capabilities	Request a new access token
Hospital pharmacy	M2M	client_credentials; its own pharmacy client and signed client assertion	Separately requested access token without pseudonymization capabilities	Request a new access token

Healthcare access tokens obtained with the authorization-code flow can be renewed using the refresh-token flow when a refresh token has been issued. After renewal, exchange the renewed original access token again before calling Recip-e. Exchanged access tokens are not refreshed directly. M2M access tokens are requested again using client_credentials and a new signed client assertion; no refresh-token flow is used.

Scope categories:

Scope category	**Scope names**	**Applies to the illustrated access tokens**
Protocol and IAM authorization	openid iam:authz	All illustrated flows

Scope category	Scope names	Applies to the illustrated access tokens
Profile information	profile	Prescriber and organizational access tokens
Professional identity/profile	iam:profile:professional ssin	Initial prescriber access token
Citizen profile selection	iam:profile:citizen iam:profile:parent iam:profile:mandatee	Initial patient access token
Citizen profile handling	pseudo:profile (initial) iam:pseudo (exchanged)	Patient access-token examples
Access-token exchange	iam:connect:exchange	Initial healthcare access tokens
Pseudonymization API capabilities	pseudo:api:identify pseudo:api:pseudonymize	Access tokens intended for the pseudonymization service
UHMEP pseudonymization access	nihdi:uhmep:pseudo	Initial prescriber and pseudonymization-capable organizational access tokens; absent from the initial patient example
Recip-e prescriber access	nihdi:uhmep:hcp	Exchanged prescriber access token
Recip-e citizen access	nihdi:uhmep:patient	Exchanged patient access token
Recip-e organizational access	nihdi:uhmep:organization	Hospital and pharmacy access tokens

The categories describe the scopes in the access-token examples. Register the scopes needed for the selected actor and operations with eHealth.

More information about I.AM Connect can be found here:

- I.AM (Identity & Access Management) (NL)
- I.AM (Identity & Access Management) (FR)

3.2.2. eHealth IAM Client Onboarding

A client should register itself in the healthcare or M2M realm. The following client registration document should be filled out and sent to eHealth to get access:

- I.AM Connect - HealthCare Client Registration Form
- I.AM Connect - M2M Client Registration Form

Client registration and identity verification are completed during onboarding. At runtime, citizens and individual prescribers authenticate through the healthcare realm. Hospital and pharmacy clients authenticate through the M2M realm using their registered organizational credentials.

Please find the list of needed information that would need to be provided below (current version 2.1).

Note: Patient channels need to indicate to eHealth that they want to receive the information of mandateholders.

3.2.2.1. General Client Information

Information (fields with an * are mandatory)	Explanation and allowed values	Example
Request date (*)	Please state the date on which you are submitting this request form.	01/03/2025
Partner organization (*)	Please state the full name of the partner organization that is requesting the IAM onboarding.	Aqme Care
Contact person (*)	Please state the full name, email address and phone number of the person that may be contacted by eHealth for information on the onboarding request.	Name: John Doe Email address: john.doe@aqmecare.be Feel free to add additional email

	Note that this contact person may be contacted for questions related to businessas well as technical aspects, so the contact person is expected to coordinate with all relevant departments within your organization.	addresses or a group mail address in case more people need to be kept informed. Phone number: 0470/12.34.56
Brief description of the purpose of your application and of the requested client. (*)	Please describe briefly what the purpose of your application is.	Aqme care is currently building an application so patient can access to their prescriptions on the way…..
Public or confidential client? (*)	Please indicate which of the two available authentication flows your application uses (select only one option): **Public client:** Your application is either a distributed software, or a webapp that runs 100% on the client of the end user. (It is not possible in this case to generate a private key to authenticate the application.) The user authenticates directly with eHealth IDP and FAS. Authorization code flow: PKCE integration. **Confidential client:** Your application runs partly on the server(s) of a recognized partner organization (private keys are used to authenticate the application). The user authenticates directly with eHealth IDP and FAS. Authorization code flow: an access token is sent by the client-component to the server component of the recognized organization.	Public client
Client ID (*)	The IAM client ID is the unique identifier of the IAM Connect client configured by eHealth for the partner. In case the partner organization already has an existing IAM Connect Healthcare client, the partner can choose (or not) to request for the existing client to be reused and adapted. (In some cases, though this is not possible, and in that case, eHealth will have to configure a new client.)	software-name
Scopes	Scopes are boundaries that are defined to technically limit the use of the IAM client to the	Scope selection depends on the actor, the target service and the access-token flow. Use the

	purpose/application for which it is requested. In case your request for an IAM client is in the context of an onboarding procedure for a specific application, check the onboarding documentation of that application for guidelines on scopes to be filled out in this field. If the documentation mentions no scopes, leave the field empty.	actor/flow matrix, scope categories and access-token examples in this section.
Redirect URI (*)	To redirect the user after a successful authentication, a valid redirect URI is needed in the configuration. This URI is also used for redirecting the user after a logout.	https://app-acc.software-name.be/iam-connect-endpoint/

3.2.2.2. Additional Confidential Client Information

This is only to be filled out if you have selected the option “Confidential client” higher in this form. This information does not apply to public clients. It is related to the credentials of the eHealth certificate used by your organization.

Information (fields with an * are mandatory)	Explanation and allowed values	Example
Type (*)	Please specify the type of identifier mentioned in your eHealth certificate.	NIHII-HOSPITAL (hospital) NIHII-PHARMACY (pharmacy) CBE (institution)
Identifier (*)	Please state the value of your certificate identifier (string value).	Type of identifier (e.g. "NIHII institution")
Application ID	If your eHealth certificate contains an application ID, please state it here. If not, leave this field empty	Application ID (if available)

3.2.2.3. IAM Exchange Security Commitment

The signature of these documents is required:

- IAM eXchange Annex A Security commitment from the Trusted Platform
- IAM Connect Token Exchange Security Commitment in the context of pseudonymization

3.2.2.4. Patient Client Onboarding

Register the patient-facing application in the healthcare realm using the Healthcare Client Registration Form for each environment. Specify the public or confidential client type, authorization-code flow, redirect URIs and registered client authentication method. Public clients must use PKCE. Arrange the applicable token-exchange security commitments with eHealth (via integration-support@ehealth.fgov.be).

Scope registration and initial access token

For citizen access, pseudonymized profile handling and access-token exchange, enable openid, iam:authz, iam:profile:citizen, pseudo:profile and iam:connect:exchange. To support the mandate and parent flows, explicitly request iam:profile:mandatee and iam:profile:parent. When the application itself calls the pseudonymization service, additionally request pseudo:api:identify and/or pseudo:api:pseudonymize for the operations it performs, and arrange the required pseudonymization-domain access.

The complete initial access-token scope set for a patient channel supporting both representation profiles and both pseudonymization operations is:

```
openid iam:profile:mandatee iam:profile:parent pseudo:api:identify iam:connect:exchange
pseudo:profile pseudo:api:pseudonymize iam:profile:citizen iam:authz
```

Access-token exchange for Recip-e

Ask eHealth to authorize this healthcare client to exchange its original access token towards audience nihdi-uhmep-fhir-patient. The expected exchanged access-token scopes are:

```
openid iam:pseudo nihdi:uhmep:patient iam:authz
```

The exchanged access token must be authorized for nihdi-uhmep-api with the patient role. The example has audiences nihdi-uhmep-fhir-patient and nihdi-uhmep-api. It must not carry the identify/pseudonymize capabilities of the original access token. Send the exchanged access token to Recip-e; use the original access token only for its permitted pseudonymization calls and IAM exchange.

UserInfo configuration for mandates and parent representation

Explicitly ask eHealth to configure the patient client so that Recip-e can call UserInfo with the exchanged access token and receive the authenticated citizen's pseudonymized userProfile.ssin, userProfile.children entries with ssin, and userProfile.mandators entries with ssin and the applicable serviceNames. The backend recognizes the service names *recipe* for prescription mandates and *medicaldatamanagement* for health-data mandates, and uses the returned children for parent representation.

Request confirmation that the original-client configuration and exchange-target configuration make this information available in the UserInfo response seen by Recip-e. Registering/requesting the scopes alone is insufficient if the relationships are not returned through that exchanged access token.

3.2.2.5. Prescriber Client Onboarding

Personal prescribing context - healthcare realm

Register the application with the Healthcare Client Registration Form for each environment. Request the professional profile configuration, authorization-code flow and the required redirect URIs. Public clients use PKCE; confidential clients authenticate to the IAM token endpoint with their registered method. Arrange access-token exchange and the applicable security commitments.

Enable openid, iam:authz, iam:profile:professional, profile, ssin and iam:connect:exchange for the initial healthcare flow. If the application directly pseudonymizes or identifies identifiers, additionally enable pseudo:api:identify, pseudo:api:pseudonymize and nihdi:uhmep:pseudo as required, together with the approved domain access. The complete initial access-token scope is:

```
openid iam:profile:professional profile ssin iam:connect:exchange pseudo:api:identify
pseudo:api:pseudonymize nihdi:uhmep:pseudo iam:authz
```

Authorize exchange towards audience nihdi-uhmep-fhir-hcp. The expected exchanged access-token scopes are:

```
openid profile nihdi:uhmep:hcp iam:authz
```

The exchanged access token is authorized for nihdi-uhmep-api with the prescriber and restricted roles shown in the example. Its audiences are nihdi-uhmep-fhir-hcp and nihdi-uhmep-api. Request the professional claims needed by Recip-e: the prescriber SSIN, profession, nihii11 identifier and relevant recognition/entitlement attributes.

Use the original access token for pseudonymization calls and the exchanged access token for Recip-e. Verify that the exchanged access token retains the required professional identity without retaining pseudonymization capabilities. If the original healthcare access token is renewed using an issued refresh token, exchange the renewed access token again.

Hospital prescribing context - M2M realm

Register a hospital organization client with the M2M Client Registration Form for each environment. Specify the hospital organization identifier, the applicable NIHII-HOSPITAL certificate identity, and signed-client-assertion authentication. Request the client_credentials grant and a hospital organization profile containing hospital.nihii and hospital.name within the organizations structure illustrated in the hospital example.

For an access token sent to Recip-e, request:

```
openid iam:authz nihdi:uhmep:organization profile
```

The access token must include nihdi-uhmep-api as an audience and the organization role for that resource. In this context, Recip-e organizational access uses nihdi:uhmep:organization rather than the individual prescriber's nihdi:uhmep:hcp scope.

If the hospital application directly uses the pseudonymization service, separately request the applicable identify/pseudonymize capabilities and UHMEP domain access. The full pseudonymization-capable access-token scope set is:

```
openid pseudo:api:identify iam:authz pseudo:api:pseudonymize nihdi:uhmep:organization profile
nihdi:uhmep:pseudo
```

That access token carries the illustrated organization/pseudo roles for nihdi-uhmep-api and identify/pseudonymize roles for ehealth-pseudo-api. Use it for the permitted pseudonymization operations. Obtain a separate access token without those capabilities for Recip-e. The M2M flow does not allow access-token exchange. Obtain new M2M access tokens with client_credentials and a fresh client assertion.

3.2.2.6. Executor Client Onboarding

Public pharmacy context

Register the pharmacy organization client in the M2M realm with the M2M Client Registration Form for each environment. Specify the pharmacy organization identifier, the applicable NIHII-PHARMACY certificate identity, signed-client-assertion authentication and the client_credentials grant. Request an organization profile identifying pharmacy.nihii and pharmacy.name in the organizations structure illustrated in the Public pharmacy example.

Hospital pharmacy context

Register the hospital pharmacy with its own pharmacy M2M client and pharmacy credentials. Do not reuse the hospital client. Ask eHealth to ensure that the hospital pharmacy is exposed as the intended pharmacy organization in the profile recognized by Recip-e.

The public pharmacy and hospital pharmacy contexts use the same scope lists below. Their registered pharmacy organization identities distinguish the contexts.

Access token for Recip-e

Enable and request the following scope set:

```
openid iam:authz nihdi:uhmep:organization profile
```

The access token must include nihdi-uhmep-api as an audience and its organization role, with the pharmacy organization profile. Use this access token for Recip-e FHIR requests. Pharmacy access uses nihdi:uhmep:organization.

Access token for the pseudonymization service

Where the pharmacy application directly identifies or pseudonymizes citizen identifiers, request the corresponding API permissions and approved pseudonymization-domain access. The full scope set for both operations is:

```
openid pseudo:api:identify iam:authz pseudo:api:pseudonymize nihdi:uhmep:organization profile
nihdi:uhmep:pseudo
```

Confirm organization/pseudo roles for nihdi-uhmep-api and identify/pseudonymize roles for ehealth-pseudo-api as illustrated. Use that access token for the authorized pseudonymization operations, and obtain a separate access token without those capabilities for Recip-e. Request new M2M access tokens with client_credentials and a fresh client assertion; do not depend on refresh-token issuance.

3.3. Authorization

3.3.1. Access Tokens

The access token that the client provides for professional and organizational users should be self-contained: the information provided in the access token is enough to perform all eHealth IAM.Connect-related authorization checks.

In case of citizen access tokens, the UserInfo endpoint will always be called to retrieve pseudonymized personally identifiable information.

Recip-e supports only eHealth access tokens coming from the IAM Connect healthcare or M2M realm.

The following OICD configuration should be used by your application:

- issuer-uri (healthcare):
  - ACC: https://api-acpt.ehealth.fgov.be/auth/realms/healthcare
  - PROD: https://api.ehealth.fgov.be/auth/realms/healthcare
- issuer-uri (M2M):
  - ACC: https://api-acpt.ehealth.fgov.be/auth/realms/M2M
  - PROD: https://api.ehealth.fgov.be/auth/realms/M2M
- client-id: <your_client_id>
- scope: <actor-specific scopes; see the scope categories>
- redirect-uri (healthcare authorization-code flow): <your_redirect_URI>
- healthcare grant: authorization_code; renewal: refresh_token when issued
- M2M grant: client_credentials; renewal: request a new access token

The retrieved access token should be sent to Recip-e with the header:

```
Authorization: Bearer <access_token>
```

3.3.2. Access Token Exchange

Healthcare access tokens obtained after a citizen or individual prescriber authenticates are exchanged before calling Recip-e. Use the original access token for permitted eHealth pseudonymization calls and as subject_token at the IAM token endpoint. Do not forward that original access token to Recip-e.

Request an exchanged access token for the configured Recip-e audience. The exchanged access token has the Recip-e actor permissions and no identify/pseudonymize capabilities for the pseudonymization service. Send this exchanged access token in the Authorization header of each Recip-e FHIR request.

Access-token destinations are summarized below.

Access token	Destination and purpose
Original healthcare access token	eHealth pseudonymization service; also used as subject_token at IAM for access-token exchange
Exchanged healthcare access token	Recip-e FHIR API; reduced capabilities for the configured UHMEP audience
M2M access token with pseudonymization capabilities	eHealth pseudonymization service
M2M access token without pseudonymization capabilities	Recip-e FHIR API

Note: for healthcare providers, the plain text SSIN will still be part of the exchanged access token, but for patients, the SSIN is stripped. Recip-e will retrieve a pseudonymized patient SSIN from the UserInfo endpoint, given the access token.

Technically, in order to perform the token exchange, a POST must be sent to this endpoint:

- ACC: https://api-acpt.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/token
- PROD: https://api.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/token

For a public client, use the following form body. A confidential client must also authenticate to the IAM token endpoint using its registered authentication method.

```
requested_token_type=urn:ietf:params:oauth:token-type:access_token
grant_type=urn:ietf:params:oauth:grant-type:token-exchange
subject_token_type=urn:ietf:params:oauth:token-type:access_token
subject_token=<original_login_access_token>
client_id=<source_client_id>
audience=<target_audience>
```

Integrators of patient software should use “nihdi-uhmep-fhir-patient” as audience, integrators of software for professionals should use “nihdi-uhmep-fhir-hcp” as audience.

3.3.3. Access Token Claims

Recip-e reads these claims from the authenticated professional user:

- ssin
- userProfile

The “userProfile” claim identifies the professional or organization. Professional entries include physician, nurse, dentist and midwife, with nihii11 identifiers. Organizational entries include pharmacy and hospital, with nihii identifiers.

Pharmacists connect to Recip-e via their pharmacy organization using its M2M client. This also applies to hospital pharmacies, which use their own pharmacy M2M client.

The access-token examples below show only the claims relevant to this integration. Personal identifiers are placeholders.

3.3.3.1. Prescriber Access Token Example

Initial Access Token

```
{
  "iss": "https://api-acpt.ehealth.fgov.be/auth/realms/healthcare",
  "aud": [
    "<prescriber_client_id>",
    "nihdi-uhmep-api",
    "ehealth-pseudo-api"
  ],
  "azp": "<prescriber_client_id>",
  "resource_access": {
    "nihdi-uhmep-api": {
      "roles": [
        "pseudo"
      ]
    },
    "ehealth-pseudo-api": {
      "roles": [
        "identify",
        "pseudonymize"
      ]
    }
  },
  "scope": "openid iam:profile:professional profile ssin iam:connect:exchange
pseudo:api:identify pseudo:api:pseudonymize nihdi:uhmep:pseudo iam:authz",
  "ssin": "<prescriber_ssin>",
  "userProfile": {
    "ssin": "<prescriber_ssin>",
    "physician": {
```

```
      "recognised": true,
      "entitled": true,
      "nihii11": "<prescriber_nihii11>"
    }
  }
}
```

Exchanged Access Token

```
{
  "iss": "https://api-acpt.ehealth.fgov.be/auth/realms/healthcare",
  "aud": [
    "nihdi-uhmep-fhir-hcp",
    "nihdi-uhmep-api"
  ],
  "azp": "nihdi-uhmep-fhir-hcp",
  "resource_access": {
    "nihdi-uhmep-api": {
      "roles": [
        "prescriber",
        "restricted"
      ]
    }
  },
  "scope": "openid profile nihdi:uhmep:hcp iam:authz",
  "act": {
    "azp": "<prescriber_client_id>"
  },
  "userProfile": {
    "ssin": "<prescriber_ssin>",
    "physician": {
      "recognised": true,
      "entitled": true,
      "nihii11": "<prescriber_nihii11>"
    }
  }
}
```

3.3.3.2. Patient Access Token Example

Initial Access Token

Use this initial healthcare access token for permitted pseudonymization calls and as subject_token when requesting the exchanged access token.

```
{
  "iss": "https://api-acpt.ehealth.fgov.be/auth/realms/healthcare",
  "aud": [
    "<patient_client_id>",
    "ehealth-pseudo-api"
  ],
  "azp": "<patient_client_id>",
  "resource_access": {
    "ehealth-pseudo-api": {
      "roles": [
        "identify",
        "pseudonymize"
      ]
```

```
    }
  },
  "scope": "openid iam:profile:mandatee iam:profile:parent pseudo:api:identify
iam:connect:exchange pseudo:profile pseudo:api:pseudonymize iam:profile:citizen
iam:authz"
}
```

Exchanged Access Token

Use this exchanged healthcare access token in the Authorization header of Recip-e API requests.

```
{
  "iss": "https://api-acpt.ehealth.fgov.be/auth/realms/healthcare",
  "aud": [
    "nihdi-uhmep-fhir-patient",
    "nihdi-uhmep-api"
  ],
  "azp": "nihdi-uhmep-fhir-patient",
  "resource_access": {
    "nihdi-uhmep-api": {
      "roles": [
        "patient"
      ]
    }
  },
  "scope": "openid iam:pseudo nihdi:uhmep:patient iam:authz",
  "act": {
    "azp": "<patient_client_id>"
  },
  "userProfile": {}
}
```

Notes:

- Recip-e supports parent-child representation, prescription mandates and health-data mandates. Authorization depends on the mandate and the requested operation.
- For more information on the eHealth IAM.Connect claims, see the claim mappers document on the eHealth page. Recip-e only supports V1.

3.3.4. M2M Access Tokens

3.3.4.1. Requesting an M2M access token

In this M2M flow, the organization requests access tokens directly with the client_credentials grant and a signed client assertion. Request an access token with pseudonymization capabilities for the eHealth pseudonymization service, and a separate access token without those capabilities for Recip-e. Do not send the pseudonymization-capable access token to Recip-e. Request both kinds of access token through:

- ACC: **POST** https://api-acpt.ehealth.fgov.be/auth/realms/M2M/protocol/openid-connect/token
- PROD: **POST** https://api.ehealth.fgov.be/auth/realms/M2M/protocol/openid-connect/token

With body :

```
grant_type=[client_credentials]
client_assertion_type=[urn:ietf:params:oauth:client-assertion-type:jwt-bearer]
client_assertion=<client_assertion>
client_id=[<client_id>]
scope=[openid iam:authz nihdi:uhmep:organization]
```

When requesting pseudonymization rights, the scopes change to:

```
scope=[openid iam:authz nihdi:uhmep:organization pseudo:api:pseudonymize
pseudo:api:identify nihdi:uhmep:pseudo]
```

The client assertion is a JWT that contains:

```
{
  "iss": "<client_id>",
  "sub": "<client_id>",
  "aud": [
    "https://api(-acpt).ehealth.fgov.be/auth/realms/M2M"
  ],
  "jti": "<random_uuid>",
  "iat": <issuer_at_time_epoch_seconds>,
  "exp": <expiration_time_epoch_seconds>
}
```

The client assertion must be signed with the private key associated with the M2M client's registered eHealth authentication certificate.

3.3.4.2. Public pharmacy access-token examples

Public pharmacy access token for the pseudonymization service:

```
{
  "iss": "https://api-acpt.ehealth.fgov.be/auth/realms/M2M",
  "aud": [
    "nihdi-uhmep-api",
    "ehealth-pseudo-api"
  ],
  "azp": "<pharmacy_client_id>",
  "resource_access": {
    "nihdi-uhmep-api": {
      "roles": [
        "organization",
        "pseudo"
      ]
    },
    "ehealth-pseudo-api": {
      "roles": [
        "identify",
        "pseudonymize"
      ]
    }
  },
  "scope": "openid pseudo:api:identify iam:authz pseudo:api:pseudonymize
nihdi:uhmep:organization profile nihdi:uhmep:pseudo",
  "userProfile": {
    "organizations": [
      {
        "pharmacy": {
          "nihii": "<pharmacy_nihii>",
          "name": "<pharmacy_name>"
        }
      }
    ]
  }
}
```

Public pharmacy access token for Recip-e:

```
{
```

```
  "iss": "https://api-acpt.ehealth.fgov.be/auth/realms/M2M",
  "aud": "nihdi-uhmep-api",
  "azp": "<pharmacy_client_id>",
  "resource_access": {
    "nihdi-uhmep-api": {
      "roles": [
        "organization"
      ]
    }
  },
  "scope": "openid iam:authz nihdi:uhmep:organization profile",
  "userProfile": {
    "organizations": [
      {
        "pharmacy": {
          "nihii": "<pharmacy_nihii>",
          "name": "<pharmacy_name>"
        }
      }
    ]
  }
}
```

3.3.4.3. Hospital access-token examples

Hospital access token for the pseudonymization service:

```
{
  "iss": "https://api-acpt.ehealth.fgov.be/auth/realms/M2M",
  "aud": [
    "<hospital_client_id>",
    "nihdi-uhmep-api",
    "ehealth-pseudo-api"
  ],
  "azp": "<hospital_client_id>",
  "resource_access": {
    "nihdi-uhmep-api": {
      "roles": [
        "organization",
        "pseudo"
      ]
    },
    "ehealth-pseudo-api": {
      "roles": [
        "identify",
        "pseudonymize"
      ]
    }
  },
  "scope": "openid pseudo:api:identify iam:authz pseudo:api:pseudonymize
nihdi:uhmep:organization profile nihdi:uhmep:pseudo",
  "userProfile": {
    "organizations": [
      {
        "hospital": {
          "nihii": "<hospital_nihii>",
          "name": "<hospital_name>"
        }
      }
    ]
  }
```

```
}
```

Hospital access token for Recip-e:

```
{
  "iss": "https://api-acpt.ehealth.fgov.be/auth/realms/M2M",
  "aud": [
    "<hospital_client_id>",
    "nihdi-uhmep-api"
  ],
  "azp": "<hospital_client_id>",
  "resource_access": {
    "nihdi-uhmep-api": {
      "roles": [
        "organization"
      ]
    }
  },
  "scope": "openid iam:authz nihdi:uhmep:organization profile",
  "userProfile": {
    "organizations": [
      {
        "hospital": {
          "nihii": "<hospital_nihii>",
          "name": "<hospital_name>"
        }
      }
    ]
  }
}
```

3.3.5. Application Identifier

Your application identifier must be sent with each request via the User-Agent header. The value must conform to the following regular expression:

```
[a-zA-Z0-9-]*+\[[a-zA-Z0-9-]*+\/[0-9a-zA-Z-_.]*+\](?:-[a-zA-Z0-9]+)*+
```

Example value:

- YourCompanyName[YourSoftwarePackage/1.2.3]-YourConnector

3.3.6. AttributeAuthority

Recip-e integrates with AttributeAuthority to verify the patient's informed consent, therapeutic exclusions and therapeutic relations according to the regulations. More information can be found here: NL, FR.

Patients can use the MyHealth webapp to:

- Activate their informed consent
  - ACC
  - PROD
- Manage their therapeutic relations
  - ACC
  - PROD
- Manage their therapeutic exclusions
  - ACC
  - PROD

3.3.7. Break The Glass

Pharmacists can use Break The Glass on MedicationRequest search to bypass the Therapeutic Relation and/or Exclusion check. Note that other authorization and privacy rules still apply. Technically, it's done by adding these search parameters:

```
bypassTherapeuticLinkApproved=true
bypassTherapeuticExclusionApproved=true
bypassTherapeuticLinkReason=<justification>
```

Use bypassTherapeuticLinkApproved=true to approve bypassing the absence of a therapeutic relationship. Use bypassTherapeuticExclusionApproved=true to approve bypassing a therapeutic exclusion. If either flag is true, a reason is mandatory.

Example:

```
POST /MedicationRequest/_search
Content-Type: application/x-www-form-urlencoded

patient=urn:be:fgov:ehealth:pseudo:v1:...&
status=active&
bypassTherapeuticLinkApproved=true&
bypassTherapeuticExclusionApproved=true&
bypassTherapeuticLinkReason=Emergency situation
```

4. FHIR Interactions

As mentioned before we follow the FHIR standards. It provides default resource definitions, created by HL7. In Belgium, eHealth (together with all parties stated above) defines specific FHIR profiles according to the Belgian requirements. The implementation guide for these profiles can be found here: https://www.ehealth.fgov.be/standards/fhir/index.html.

Recip-e supports only FHIR R4 as this is also the version used in the implementation guide.

4.1. CapabilityStatement

The FHIR CapabilityStatement describes the shared server capabilities and supported profiles (authorization required). The package 3 Recip-e subset is listed below; additional platform capabilities may belong to eReferral. Consult OpenAPI for search parameters and operation-specific requirements:

- ACC
- PROD

4.2. OpenAPI Definition

The OpenAPI specification of UHMEP can be found here. Use it for HTTP methods and paths, search filters, paging and sorting parameters, request/response definitions and operation-specific status codes:

- ACC
- PROD

4.2.1. Used operations

The UHMEP API is shared by Recip-e and eReferral. The operations relevant for Recip-e are:

- getMetaData: Fetch the server FHIR CapabilityStatement
- createBundle: Create a prescription with visibility settings. This is not a general-purpose batch or transaction endpoint.
- getMedicationRequest: Retrieve prescription data.
- createMedicationRequest: Create a new MedicationRequest instance

- searchMedicationRequests: search for MedicationRequest instances
- createTask: Submit a prescription cancellation, completion or visibility-change command. This does not provide general-purpose Task management.

4.2.2. Request body

FHIR resource bodies use JSON with Content-Type: application/fhir+json. MedicationRequest search uses POST with application/x-www-form-urlencoded; encode search values correctly. Bodyless GET requests do not need Content-Type. See Headers Summary for more information.

4.3. Supported resource types

Recip-e supports the following FHIR resource types:

- MedicationRequest (profiled: BeMedicationPrescription - version 1.1.1)
  - Option to bundle it with Consent to manage prescription visibility
- Task
- Bundle

Note: The shared UHMEP platform supports additional resource types and operations outside of the Recip-e scope.

4.4. Use of pseudonyms

Represent the patient pseudonym in MedicationRequest.subject.identifier. Keep the SSIN naming system; the identifier value contains the in-transit pseudonym.

The following excerpt illustrates the patient identifier. Replace some-value with an actual pseudonym.

```
"subject": {
    "identifier": {
      "system":
"https://www.ehealth.fgov.be/standards/fhir/core/NamingSystem/ssin",
      "value": "urn:be:fgov:ehealth:pseudo:v1:some-value",
      "_value": {
        "extension": [{
          "url":
"https://www.ehealth.fgov.be/standards/fhir/infsec/StructureDefinition/be-ext-
pseudonymization",
          "extension": [
            { "url": "marker", "valueBoolean": true },
            { "url": "version", "valuePositiveInt": 1 },
            { "url": "format", "valueCode": "direct" }
          ]
        }]
      }
    }
  }
```

Rules:

- Marker must always be true
- The extension version must match the version in the identifier value prefix.
- Version 1: urn:be:fgov:ehealth:pseudo:v1: with a BASE64-encoded in-transit pseudonym.
- Version 2: urn:be:fgov:ehealth:pseudo:v2: with a SEC1-encoded in-transit pseudonym.
- Always provide an explicit version and matching prefix. Do not rely on automatic format detection. The extension format direct identifies the direct pseudonymization representation.

4.4.1. Pseudonymization Negotiation Headers

Use these headers to negotiate the representation of pseudonyms in responses, either in BASE64 or SEC1 encodings.

4.4.1.1. Accept-Be-Pseudo

Clients may send the `Accept-Be-Pseudo` request header on Recip-e API calls when they want the response to contain patient pseudonyms in a specific supported in-transit format.

Supported values:

- `Accept-Be-Pseudo: urn:be:fgov:ehealth:pseudo:v1` (default)
- `Accept-Be-Pseudo: urn:be:fgov:ehealth:pseudo:v2`

4.4.1.2. Content-Be-Pseudo

Content-Be-Pseudo identifies the pseudonym representation used in the response, using the full URN values listed above. Resource-level pseudonym prefixes and extensions take precedence. See Headers Summary for applicability and defaults.

4.4.2. Additional Documentation

Please find the FHIR documentation on pseudonymization here:

- Pseudonymization Extension
- Content Negotiation for pseudonymization in FHIR

4.5. Error codes

Recip-e reports application errors using OperationOutcome. Inspect all returned issues, including warnings in successful responses. Authentication, gateway or proxy failures may occur before FHIR processing and may return a different format; check HTTP status and Content-Type before parsing.

Medication business-error codes use the urn:be:uhmep:mp:errors coding system. Identify them by issue.details.coding.system and code; use translated messages for display, not program logic. Framework or infrastructure errors may not use this coding system.

A list of all error codes and their default translations can be found here:

- https://github.com/smals-belgium/shared-digital-medication-prescription

Example:

```
{
    "resourceType": "OperationOutcome",
    "issue": [
      {
        "severity": "error",
        "code": "business-rule",
        "details": {
          "coding": [
            {
              "system": "urn:be:uhmep:mp:errors",
              "code": "PRESCRIPTION_NOT_FOUND"
            }
          ]
        },
        "diagnostics": "Prescription not found."
      }
    ]
  }
```

Below is some guidance to interpret the error codes and messages correctly.

First of all Recip-e follows the FHIR and HTTP specs as best as possible:

- 2xx codes are positive statuses
- 4xx statuses mean something is wrong with the request
- 5xx statuses mean something is wrong on the server

4.5.1. External services

Recip-e depends on external services. Dependency failures may result in 500, 503 or 504 responses; the body may not identify the underlying cause. Retry only when the operation can safely be repeated. After a timeout during creation or a prescription action, verify the outcome before resubmitting; do not assume automatic duplicate prevention.

4.5.2. Validations

Requests are validated before processing. Typical application responses are listed below; consult OpenAPI and the returned error code for operation-specific behavior:

- 400 Bad Request: malformed input or request-validation failure
- 422 Unprocessable Entity: the requested action cannot be processed under the applicable business or resource-state rules
- 404 Not Found: when the requested resource does not exist
- 412 Precondition Failed: the resource was modified concurrently and the optimistic-lock version check failed. Retrieve the latest version and reassess the change before retrying.
- Other technical HTTP errors may include 405 Method Not Allowed (unsupported HTTP method), 406 Not Acceptable (unsupported response format requested through Accept), and 415 Unsupported Media Type (unsupported request Content-Type).

4.5.3. Authentication and authorization

Authentication failures normally return 401; authorization failures normally return 403. Recip-e checks client access, actor permissions and, where applicable, therapeutic relationships, exclusions and patient consent. Security messages may deliberately be generic. Check the access token and the relationships required for the operation; contact support if access appears incorrectly denied.
For support, provide the environment, timestamp, HTTP method/path, status and returned Correlation-Id when available, together with an appropriately redacted request and response through the approved support channel. Remove bearer credentials, client assertions and other secrets, and avoid unnecessary patient information. Correlation-Id is the preferred trace reference.

4.5.4. Internationalization

For each error code, Recip-e also returns a default translated error message. These can be used by integrators that do not wish to provide their own translations. The language in which these are returned is steered by the `Accept-Language` request header, which can have the values:

- `nl`
- `fr`
- `en` (default)

4.5.5. OperationOutcome in successful prescription searches

A successful prescription search returns HTTP 200 with a Bundle of type searchset. Warnings are returned as an OperationOutcome resource in an additional Bundle.entry; they do not replace the prescription results or change the successful HTTP status.

Each warning is an OperationOutcome.issue with severity warning and code incomplete. Its details.coding contains system urn:be:uhmep:mp:errors, the specific warning code, and a translated display message.

Inspect entries by resourceType and process all OperationOutcome issues. Do not assume every entry is a MedicationRequest or count supporting resources and outcomes as prescription matches. A warning can indicate incomplete results.

An OperationOutcome entry is included only when there are warning issues to report.

5. Headers Summary

5.1. Request Headers

Header	Values	Required	Applicability and behavior
Authorization	• Bearer <access token>	Yes	Use the access token intended for Recip-e: the exchanged patient/prescriber healthcare access token, or the separately requested M2M access token without pseudonymization capabilities. See section Access Tokens.
User-Agent	• Example: YourCompanyName[YourSoftwarePackage/1.2.3]-YourConnector	Yes	Use the application identifier format defined in section Application Identifier. Spaces and ordinary browser User-Agent values do not satisfy the enforced format; missing or malformed values fail validation.
Accept-Language	• nl • fr • en	No	Selects the language used for localized responses, including validation and error messages. English (en) is the default when omitted or when no supported language is selected.
Accept-Be-Pseudo	• urn:be:fgov:ehealth:pseudo:v1 • urn:be:fgov:ehealth:pseudo:v2	No	Selects the pseudonym format in returned data. If omitted, v1 (BASE64) is used. To request SEC1/v2 explicitly, send urn:be:fgov:ehealth:pseudo:v2. The supported values are the full URNs listed here.
Content-Type	• application/fhir+json • application/x-www-form-urlencoded	Conditional	For a FHIR resource body, including a MedicationRequest, Bundle or Task: application/fhir+json. For a search using POST with form-encoded search parameters: application/x-www-form-urlencoded. Not needed for a bodyless GET read.
Accept	• application/fhir+json	No	Recommended explicitly to request a FHIR JSON response.
Correlation-Id	• <request correlation identifier>	No	A nonblank identifier supplied by the caller is returned for tracing. If omitted or blank, the server generates a UUID. Include the returned identifier when reporting an issue.

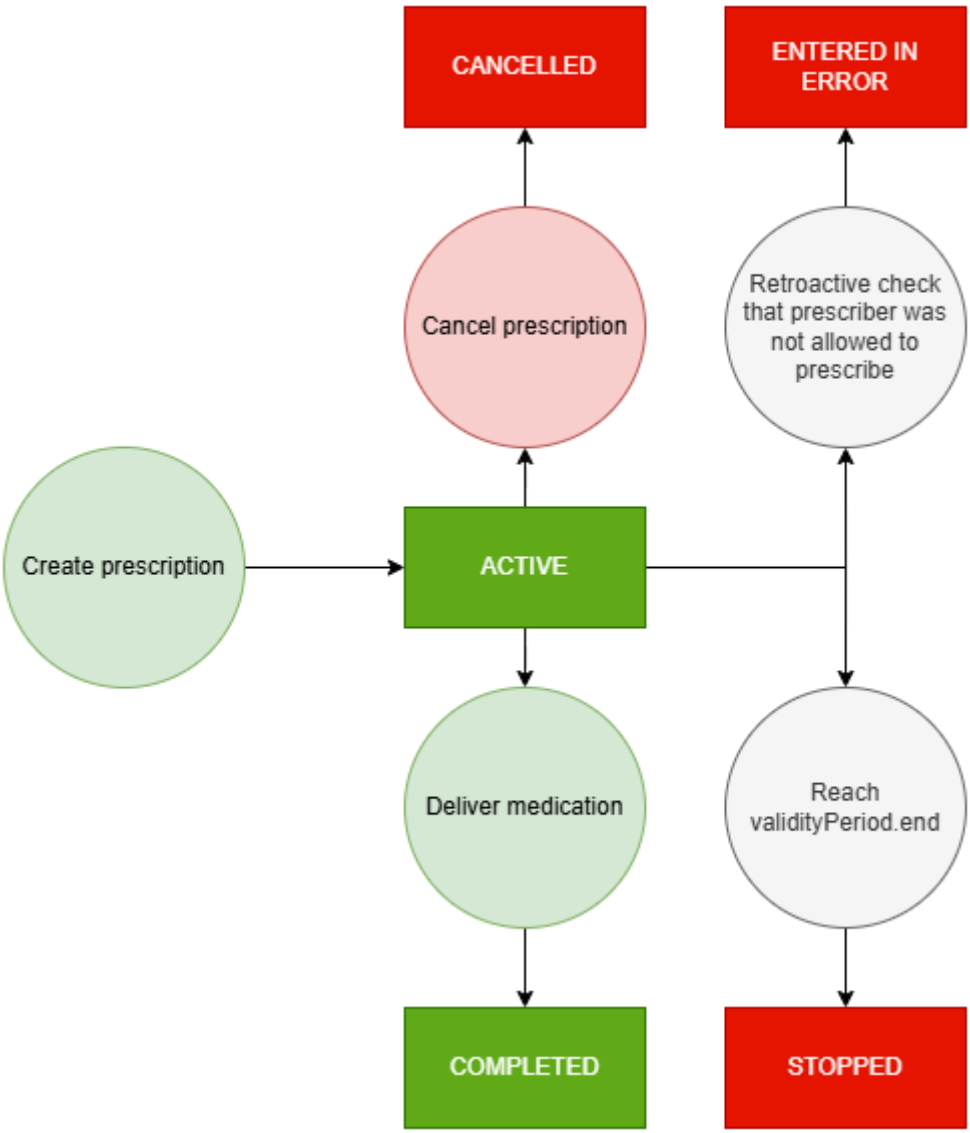
5.2. Response Headers

Header	Values	Required	Applicability and behavior
Content-Be-Pseudo	• urn:be:fgov:ehealth:pseudo:v1 • urn:be:fgov:ehealth:pseudo:v2	Conditional	Identifies the pseudonym format used in the returned data. The contract uses the full URNs

Header	Values	Required	Applicability and behavior
			shown here. Applicable to normal FHIR responses processed by the pseudonymization-aware services; it may be absent on early authentication, gateway or proxy failures.
Correlation-Id	<request correlation identifier>	Conditional	The supplied nonblank identifier, or a server-generated UUID if none was supplied. A caller-supplied identifier is not necessarily a UUID or unique. Failures generated before the service is reached may not carry this header.
Content-Type	• application/fhir+json	Conditional	Describes the response body. A bodyless response does not require this value. Early authentication, gateway or proxy errors may use a different response format.
ETag	• W/"<version-id>"	Conditional	On a MedicationRequest read, always present.
Last-Modified	• <HTTP-date>	Conditional	On a MedicationRequest read, always present.

6. Flows

6.1. Status flow



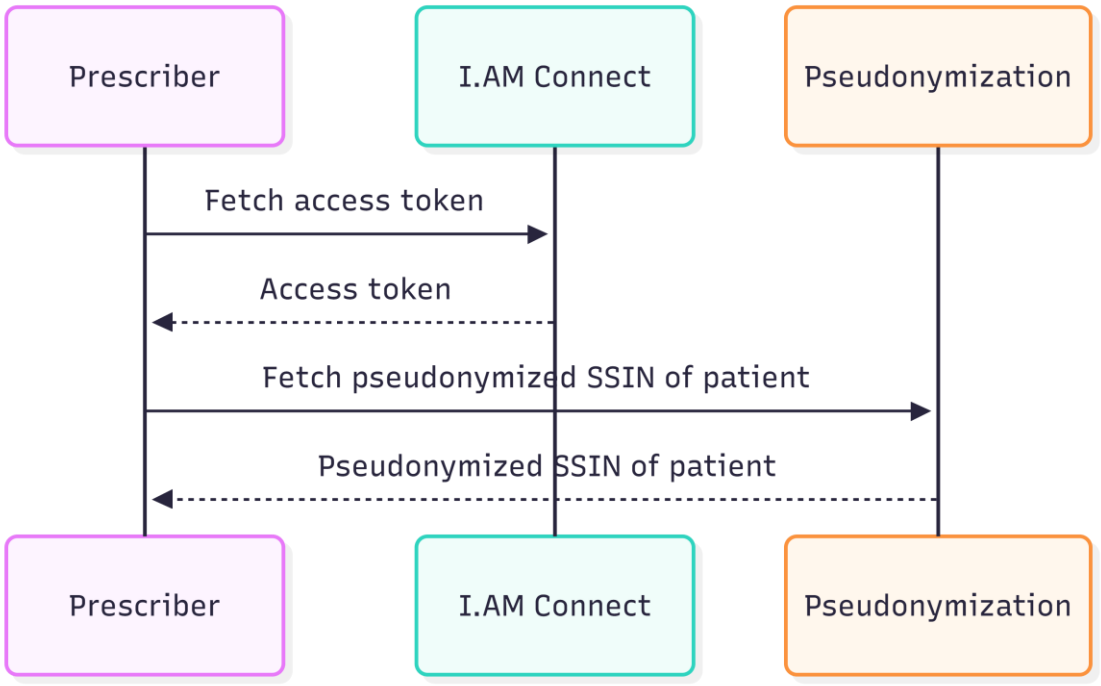
6.1.1. Explanation of the flow

- When a prescription is created then the status is set to ACTIVE.
- When an executor delivers the medication linked to the prescription then the status is set to COMPLETED.
- When a user cancels an ACTIVE prescription then the status is set to CANCELLED.
- Prescription rights can be revoke retroactively so at certain moments in the extended flow a check is done to ensure that the prescriber had the right to prescribe at the moment of prescription. When the prescribers right was revoked at the time of prescribing then the status is set to ENTERED IN ERROR.
- A validity end date is required. This date is automatically checked so that when that day is reached and passed the status of the prescription is automatically set to STOPPED.

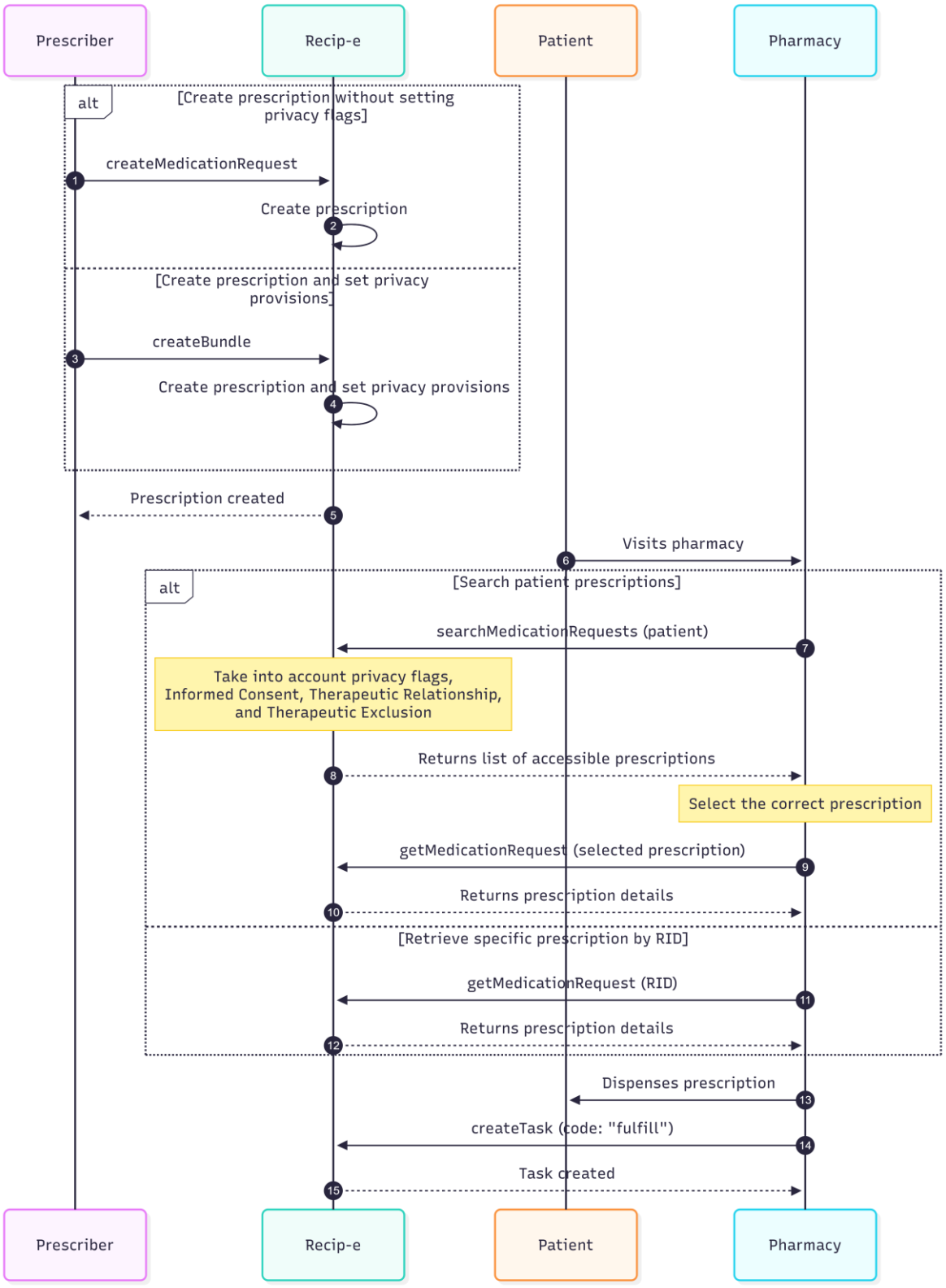
6.2. General flow

Below you'll find the happy flow of the creation of a prescription using the correct FHIR operations.
Note that before this flow a flow with eHealth will have to take place which is out-of-scope for Recip-e.

6.2.1. eHealth flow



6.2.2. Recip-e flow



6.2.3. Explanation of the flow

A prescriber creates a prescription. This can be done in two ways:

- (1) Via **createMedicationRequest** when the prescriber wants to create a prescription without setting the privacy flags.

- (3) Via **createBundle** when the prescriber wants to create a prescription and simultaneously set the privacy provisions.

(6) The patient visits a pharmacy.
The pharmacy can retrieve the correct prescription in different ways:

- (7) Via **searchMedicationRequests** to retrieve all prescriptions for the patient, taking into account the privacy flags, **Informed Consent**, **Therapeutic Relationship**, and **Therapeutic Exclusion**. From that list, the pharmacy can then retrieve the details of the correct prescription using **getMedicationRequest**.
- (11) By using the **RID**, the pharmacy can retrieve the details of a specific prescription using **getMedicationRequest**.

(14) When the pharmacy dispenses the prescription, it reports this through **createTask** with the code **"fulfill"**.

6.2.4. Alternatives

- A prescriber, citizen or pharmacy can cancel a prescription using a Task with code “abort”.
- A prescriber or citizen can change the privacy flags of a prescription using a Task with code “change” which includes a Consent.

7. Business information

7.1. Business rules

Most of the business rules defined in the Implementation Guide but not all rules can be defined as constraints on FHIR elements.
Those rules are written down and explained below:

- When prescribing using an Actual Medicinal Product Package (AMPP) then the requested dispense quantity should be exactly 1 package.
  If multiple packages are needed then multiple prescriptions have to be created.
- Optionally a display can be added to a AMPP, VMP, coded Magistral preparation or non-medicinal product. This display has to match the FAGG/AFMPS-name found in SAMv2.
- The expiration period of a prescription should be as short as possible and no longer than 1 year.
- The priority should always be set to “routine”
- The intent should always be set “order”
- Dates are used exactly as specified, and periods are inclusive.
  This means that, for example, if a user searches for prescriptions with a specific date, only results with that exact date will be returned. When searching using a period, the results will have a date that is greater than or equal to the start date of the period and less than or equal to the end date of the period.
  This means that only “eq”, “le” and “ge” are accepted as date search parameters, all other parameters are rejected.
- It is possible to add optional CTI-extended coding to Medication using the system “https://www.ehealth.fgov.be/standards/fhir/medication/NamingSystem/cti-extended-code”
  An example of a Medication element with both CNK and CTI-extended is shown below:

```
{
  "resourceType": "Medication",
  "id": "med1",
  "code": {
    "coding": [
      {
        "system": "urn:be:samv2:type",
        "code": "amppackage"
      }, {
        "system": "https://www.ehealth.fgov.be/standards/fhir/medication/NamingSystem/cnk-
codes",
        "code": "1591163",
        "display": "2LEV globuli verp. eenmalig gebr. 30",
        "userSelected": true
      }, {
        "system": "https://www.ehealth.fgov.be/standards/fhir/medication/NamingSystem/cti-
extended-code",
        "code": "51737701",
        "display": "2LEV globuli verp. eenmalig gebr. 30",
        "userSelected": true
      }
    ]
  }
}
```

- It is required to provide the name, first name and address of the prescriber who created the prescription.
  Optionally contactdetails such as phonenumber and emailaddress can also be provided.
  An example of a Practitioner with contactdetails is shown below:

```
{
  "resourceType": "Practitioner",
  "id": "5d79c025-c727-427c-8fb1-7eb0cba34ac2",
  "meta": {
    "profile": [
      "https://www.ehealth.fgov.be/standards/fhir/core/StructureDefinition/be-practitioner"
    ]
  },
  "identifier": [
    {
      "system": "https://www.ehealth.fgov.be/standards/fhir/core/NamingSystem/nihdi",
      "value": "<prescriber_nihii11>"
    }
  ],
  "name": [
    {
      "family": "Smals",
      "given": [
        "Rudy"
      ]
    }
  ],
  "telecom": [
    {
      "system": "phone",
      "value": "+32 16 555 30 00",
      "rank": 1
    },
    {
      "system": "pager",
      "value": "3000-101",
      "rank": 2
    },
    {
      "system": "email",
      "value": "rudy.smals@example.be",
      "rank": 3
    }
  ],
  "address": [
    {
      "line": [
        "Naamsestraat 101"
      ],
      "_line": [
        {
          "extension": [
            {
              "url": "http://hl7.org/fhir/StructureDefinition/iso21090-ADXP-streetName",
              "valueString": "Naamsestraat"
            },
            {
              "url": "http://hl7.org/fhir/StructureDefinition/iso21090-ADXP-houseNumber",
              "valueString": "101"
            }
          ]
        }
      ],
      "city": "Leuven",
      "postalCode": "3000",
      "country": "BE"
    }
  ]
}
```

7.2. Privacy flags

7.2.1. Explanation

Privacy flags determine, alongside informed consent, therapeutic relation, and therapeutic exclusion, the visibility of a prescription for prescribers and executors.
Two types of privacy flag exist:

1. The prescriber privacy flag, which limits the visibility of a prescription for prescribers
2. The pharmacy privacy flag, which limits the visibility of a prescription for executors

The prescriber privacy flag can be set to:

- All prescribers may see
- Only original prescriber
- Only original and GMR holder

An original prescriber will always be able to see a prescription. They cannot be blocked from viewing that prescription.

The pharmacy privacy flag can be set to:

- All pharmacies may see
- No pharmacy may see
- Only specific pharmacy may see

Note that when using “Only specific pharmacy may see” then you can only specify one pharmacy.

It is also possible to not set the privacy flags. When a privacy flag is not set, then it is considered to be open, so all related actors may see the prescription.

The privacy flags can be set by the prescriber when creating a prescription. They can set the prescriber privacy flags to any value they want. They can only set the pharmacy privacy flag to 'all pharmacies may see' or 'no pharmacy can see'.
The patient can also set the privacy flags. They have a full range of all the values of both privacy flags, and they can select a specific pharmacy to view their prescription.

Important to note is that when a patient sets the pharmacy privacy flag to 'only specific pharmacies may see' and selects a specific pharmacy, then the prescriber can see that the flag was set to 'only specific pharmacies may see,' but cannot have the info about which specific pharmacy is selected.

Circle Of Trust

A Circle-Of-Trust (COT) is a set of methodological and technical agreements between trusted parties that irrefutably secure information.
In this instance it's a set of agreements that is made between the hospital and eHealth. The COT determines everything about who can see which prescription so it decides itself about checking for informed consent, therapeutic relation, and therapeutic exclusion.

7.2.2. FHIR

Consent and Provision

The privacy flags are defined using the FHIR Consent element: Consent - FHIR v4.0.1
The Consent element has three fields which are required by FHIR and one which is required by our business rules. They are:

- Status: Indicates the state of the consent. Has to always be "active"
- Scope: Which of the four areas this Consent covers.
  - System: Has to always be "http://terminology.hl7.org/CodeSystem/consentscope"
  - Code: Has to always be "patient-privacy"
  - Display: Has to always be "Privacy Consent"
- Category: Classification of the Consent.
  - System: Has to always be "http://terminology.hl7.org/CodeSystem/v3-ActCode"

        - Code: Has to always be "INFA"
        - Display: Has to always be "Information access"
- Policy: Policies covered by this consent.
    - Authority: Has to always be "https://www.riziv.fgov.be"
    - Uri: Has to always be "https://www.ehealth.fgov.be/ehealthplatform/file/cc73d96153bbd5448a56f19d925d05b1379c7f21/9d861ec73faaa1b588a5ed0587fa680f33c35a99/10-085-n376-recip-e-en-webtoepassing-paris-gewijzigd-op-7-december-2021.pdf"

Inside a Consent element, a Provision element is defined. The Provision element details which specific denies and permits are used for the prescription. When more than one deny or permit is needed, then a nested Provision element can be added to the main Provision element.
This means that to use a permit for a certain role and a deny for the same or another role then a Provision element has to be added inside another Provision element.
An example of this can be found under Only original and GMD holder

A provision element has three required fields:

- Type: How a rule statement is applied, such as adding additional consent or removing consent. Has to be “deny” or “permit”.
- Actor: Who|what controlled by this rule (or group, by role). Pharmacy = “PHAR”, prescriber = “PROV”, GMD-holder = “AFFL”.
- Data: Data controlled by this rule. Contains a reference to the prescription using the fullUrl of the MedicationRequest.

Prescriber privacy flag

Empty

It is possible to not set the prescriber privacy flag. When this is done, then it is considered that all prescribers may see the prescription.

All prescribers may see

To allow all prescribers to see the prescription, you need to set a provision with type “permit”, actor.role.code “PROV” and actor.reference a practionerRole with code “persphysician” without a practitioner element.

```
"provision": {
  "type": "permit",
  "actor": [ {
    "role": {
      "coding": [ {
        "system": "urn:be:uhmep:mp:CodeSystem:privacy-provision-role",
        "code": "PROV",
        "display": "PROV"
      } ]
    },
    "reference": {
      "reference": "#pr1" // persphysician general
    }
  } ],
  "data": [ {
    "meaning": "instance",
    "reference": {
      "identifier": {
        "system": "http://ehealth.fgov.be/standards/fhir/medication/NamingSystem/be-ns-prescription",
        "value": "RID"
      }
    }
  } ]
}
```

Only original prescriber

To allow only the original prescriber to see the prescription, you need to set a provision with type “deny”, actor.role.code “PROV” and actor.reference a practionerRole with code “persphysician” without a practitioner element.

```
"provision": {
  "type": "deny",
  "actor": [ {
    "role": {
      "coding": [ {
        "system": "urn:be:uhmep:mp:CodeSystem:privacy-provision-role",
        "code": "PROV",
        "display": "PROV"
      } ]
    },
    "reference": {
      "reference": "#pr1" // persphysician general
    }
  } ],
  "data": [ {
    "meaning": "instance",
    "reference": {
      "identifier": {
        "system": "http://ehealth.fgov.be/standards/fhir/medication/NamingSystem/be-ns-
prescription",
        "value": "RID"
      }
    }
  } ]
}
```

Only original and GMD holder

To allow the original prescriber and the GMD holder to see this prescription, you need to set a provision with type “deny”, actor.role.code “PROV” and actor.reference a practionerRole with code “persphysician” without a practitioner element, and a provision with type “permit”, actor.role.code “AFFL” and actor.reference a practionerRole with code “persphysician” without a practitioner element.

```
"provision": {
 "type": "deny",
 "actor": [ {
  "role": {
   "coding": [ {
    "system": "urn:be:uhmep:mp:CodeSystem:privacy-provision-role",
    "code": "PROV",
    "display": "PROV"
   } ]
  },
  "reference": {
   "reference": "#pr1" // persphysician general
  }
 } ],
 "data": [ {
  "meaning": "instance",
  "reference": {
   "identifier": {
    "system": "http://ehealth.fgov.be/standards/fhir/medication/NamingSystem/be-ns-
prescription",
    "value": "RID"
   }
  }
 } ],
 "provision": [ {
  "type": "permit",
  "actor": [ {
   "role": {
    "coding": [ {
     "system": "urn:be:uhmep:mp:CodeSystem:privacy-provision-role",
     "code": " AFFL ",
     "display": "AFFL"
    } ]
   },
   "reference": {
    "reference": "#pr2" // persphysician general
   }
  } ],
  "data": [ {
   "meaning": "instance",
   "reference": {
    "identifier": {
     "system": "http://ehealth.fgov.be/standards/fhir/medication/NamingSystem/be-ns-
prescription",
     "value": "RID"
    }
   }
  } ]
 } ]
}
```

Pharmacy privacy flag

Empty

It is possible to not set the pharmacy privacy flag. When this is done, then it is considered that all pharmacies may see the prescription.

All pharmacies may see

To allow all pharmacies to see the prescription, you need to set a provision with type “permit”, actor.role.code “PHAR” and actor.reference a practionerRole with code “orgpharmacy” without a practitioner element.

```
"provision": {
  "type": "permit",
  "actor": [ {
    "role": {
      "coding": [ {
        "system": "urn:be:uhmep:mp:CodeSystem:privacy-provision-role",
        "code": "PHAR",
        "display": "PHAR"
      } ]
    },
    "reference": {
      "reference": "#pr1" // orgpharmacy general
    }
  } ],
  "data": [ {
    "meaning": "instance",
    "reference": {
      "identifier": {
        "system": "http://ehealth.fgov.be/standards/fhir/medication/NamingSystem/be-ns-
prescription",
        "value": "RID"
      }
    }
  } ]
}
```

No pharmacy may see

To deny all pharmacies from seeing the prescription, you need to submit a provision with the type ”deny”, actor.role.code “PHAR” and actor.reference a practionerRole with code “orgpharmacy” without a practitioner element.

```
"provision": {
  "type": "deny",
  "actor": [ {
    "role": {
      "coding": [ {
        "system": "urn:be:uhmep:mp:CodeSystem:privacy-provision-role",
        "code": "PHAR",
        "display": "PHAR"
      } ]
    },
    "reference": {
      "reference": "#pr1" // orgpharmacy general
    }
  } ],
  "data": [ {
    "meaning": "instance",
    "reference": {
      "identifier": {
        "system": "http://ehealth.fgov.be/standards/fhir/medication/NamingSystem/be-ns-
prescription",
        "value": "RID"
      }
    }
  } ]
}
```

Only specific pharmacy may see

To allow only a specific pharmacy to see the prescription, you set a provision with type “deny”, actor.role.code “PHAR” and actor.reference a practionerRole with code “orgpharmacy” without a practitioner element, and embed a provision with "deny", actor.role.code "PHAR", and actor.reference a practitioner role with code "orgpharmacy", with a practitioner element where the pharmacy organization is identified by its NIHII.

```
"provision": {
 "type": "deny",
 "actor": [ {
  "role": {
   "coding": [ {
    "system": "urn:be:uhmep:mp:CodeSystem:privacy-provision-role",
    "code": "PHAR",
    "display": "PHAR"
   } ]
  },
  "reference": {
   "reference": "#pr1" // orgpharmacy without NIHII
  }
 } ],
 "data": [ {
  "meaning": "instance",
  "reference": {
   "identifier": {
    "system": "http://ehealth.fgov.be/standards/fhir/medication/NamingSystem/be-ns-
prescription",
    "value": " RID "
   }
  }
 } ],
 "provision": [ {
  "type": "permit",
  "actor": [ {
   "role": {
    "coding": [ {
     "system": "urn:be:uhmep:mp:CodeSystem:privacy-provision-role",
     "code": "PHAR",
     "display": "PHAR"
    } ]
   },
   "reference": {
    "reference": "#pr2" // orgpharmacy with NIHII
   }
  } ],
  "data": [ {
   "meaning": "instance",
   "reference": {
    "identifier": {
     "system": "http://ehealth.fgov.be/standards/fhir/medication/NamingSystem/be-ns-
prescription",
     "value": "RID"
    }
   }
  } ]
 } ]
}
```