UHMEP API

Referral prescription (RIZIV-INAMI)

JUNE 27, 2025

SMALS

Table of Contents

1 Document management

1.1 Document history

Version	Status	Date	Autor(s)	Modifications
0.1	Final	25/05/2023	Gaël Mavangui, Julien Beard, Cyprien Janssens	First version of the cookbook for the release 1.0 scope « MVP » of UHMEP
0.2	/		Julien Beard, Cyprien Janssens	Second version, changes are listed in the release note
0.3	Final	26/01/2024	Cyprien Janssens, Julien Beard, Gérard Bigaj	Alignment with the new version in acceptance
0.4	Final	03/07/2024	Cyprien Janssens, Julien Beard, Gérard Bigaj	Acceptance release of June 2024. API changes are listed in the release note.
0.5	Final	04/02/2025	Cyprien Janssens, Julien Beard, Lionel Cremer, Gérard Bigaj	Acceptance release of December 2024.
0.8	Final	25/06/2025	Cyprien Janssens, Julien Beard, Gérard Bigaj	Acceptance release of June 2025

1.2 Document reviews

Reviewers	Name(s)	Reviewed Version	Comments
SPOC Client	Maarten Cobbaert	0.1	
Project Manager	Maxime Daive	0.8	
Chain Service Manager	Nicolas Rogge	0.1	

1.3 Release note

This section highlights the differences between the current API release and the previous version. It provides a summary of changes such as new features and modifications to help quickly identify what has changed.

- The following sections have been added/adapted:
  - 6.2.5: Added section related to Attribute Authority checks on informed consent, therapeutic relationships and exclusions between a patient and a healthcare professional for all the operations available in UHMEP
  - 6.3.6 & 6.3.6.1: Added section about SSIN pseudonymization
  - 6.3.7: Added section on the impacts of pseudonymization on the free text fields
  - 8.5.2: Added explanations on the sorting in the list consultation
  - 8.1.2: Added the ***category*** on top of **code** and the ***orderDetail*** for template validation
- Annex 81 proposal: added decision (statusReason “approved” or “reject”) on the proposal
- List of prescriptions and proposals now sorted on the authoredOn date (most recent to oldest)
- Creation of a radiology prescription

Miscellaneous

- Replaced all occurrences of “BeReferralServiceRequestNursing” with “BeReferralServiceRequest”

2 Reference

2.1 eHealth reference

All referenced documents are available on the portal of the eHealth platform. These versions, or any following ones, can be used for the eHealth platform service.

ID	Title	Version	Date	Author
1	eHealth Services – Web Access	2.0	12/07/2018	eHealth platform
2	Identity & Authorization Management (I.AM) Mobile integration	1.7	31/03/2023	eHealth platform
3	Pseudonymization REST	1.0	04/04/2023	eHealth platform
4	Identity & Authorization Management (IAM) eXchange	1.1	28/06/2024	eHealth platform

2.2 FHIR references

FHIR documentation type	Link
Implementation guide	https://build.fhir.org/ig/hl7-be/referral/branches/earlyadopter/guidance.html
BeReferralServiceRequest	https://www.ehealth.fgov.be/standards/fhir/referral/StructureDefinition/be-referral-servicerequest-nursing
BeReferralTask	https://build.fhir.org/ig/hl7-be/referral/branches/earlyadopter/StructureDefinition-be-referral-task.html
BePerformerTask	https://build.fhir.org/ig/hl7-be/referral/branches/earlyadopter/StructureDefinition-be-performer-task.html
BeOrganizationTask	https://www.ehealth.fgov.be/standards/fhir/referral/StructureDefinition/be-organization-task
BePractitionerRole	https://build.fhir.org/ig/hl7-be/core//StructureDefinition-be-practitionerrole.html
BePractitioner	https://build.fhir.org/ig/hl7-be/core//StructureDefinition-be-practitioner.html
Examples	https://build.fhir.org/ig/hl7-be/referral/branches/earlyadopter/artifacts.html#example-example-instances

3 Document information

3.1 Glossary

Term	Definition
Reference ID	This is a unique number assigned by UHMEP to each referral prescription issued.
UHMEP	This is the name given to the API which allows to create and manage referral prescriptions and propositions. UHMEP stands for “Unaddressed Health Message Exchange Platform”.
Therapeutic link	It is a relation that a healthcare professional must establish with the patient to have access to his medical data.
Therapeutic exclusion	This relation can be created by the patient and allows him to prevent a healthcare professional to access to medical data coming from other professionals.
Prescriber	The person who can prescribe a referral prescription for a patient.
Caregiver	The caregiver is a general term for the person who provides care based on the information written on the referral prescription. Also known as "treatment provider" (prescriber as well as treatment provider are considered to be caregivers).
Patient	The individual which is the subject of the referral prescription.
Head nurse	A designated caregiver of an organization that will have access to prescriptions assigned to it. His/her role is to dispatch it to caregivers inside the organization.
Admin	A designated individual of an organization that will have access to prescriptions assigned to it. His/her role is to dispatch it to caregivers inside the organization.
Assignation	The assignation is a recommendation done by the prescriber or is done by the patient to help the caregiver to find the prescription easily.
Execution	The execution represents the status of the care that must be provided by a caregiver.

3.2 Formatting explanation

- When a word is in bold and italic: that is a field in the FHIR specification. Ex: ***status***.
- When a word is between quotations mark "": This is the value that the specific field can have. Ex: "ready".
- All FHIR resources mentioned in this document are described in the official eHealth FHIR implementation guide pages. Specific links are available in section 2.2.
- When a role is between parentheses in the begin of sentences, it means that the business rules explained next to it is applicable to this role. Ex: (Prescriber).

4 Goal of the document

The goal of this document is to provide all the information needed to integrate UHMEP.

This document will describe the project but also what is around the project like how an integrator can have some support and what is needed to call UHMEP.

UHMEP uses FHIR to exchange messages with the integrators. This document will explain some fundamental concepts to understand what must be given and what happens by doing the different operations.

The access management will be also explained to allow the integrators to know when an operation can be called.

After these explanations, this document will describe all the operations available respecting the following structure:

- A description of the operation.
- The endpoint to call.
- The roles that a user must have to call the operation.
- The rules implemented in UHMEP
- Some information about the request to send
- Some information about the response

5 Support

5.1 For issues in acceptance

Issues in acceptance can be reported by sending a mail to integration-support@ehealth.fgov.be.

5.2 Certificates

To access the secured eHealth platform environment each integrator must obtain an eHealth platform certificate, used to identify the initiator of the request. 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

6 Global overview

6.1 Context (Goal of the project)

UHMEP ("Unaddressed Health Message Exchange Platform") is an exchange platform that stores referral prescriptions and makes them available to healthcare professionals and patients.

The project wants to achieve 2 objectives:

- More efficient processing of the referral prescription
- Improve the delivery of care and its support activities

Referral prescriptions are the non-drug prescriptions that a patient receives when he goes to his doctor (the Prescriber) for a certain problem. A referral prescription is executed by the caregiver. For example, it can be a prescription for wound care, for an X-ray, etc.

A care proposal results from the reverse process, i.e. it is the caregiver who will create a care proposal to propose to prolongate an existing care or to communicate observations about a potentially new health problem.
This feature will come in a next release.

The digitization of referral prescriptions and care proposals will reduce the administrative load thanks to the reduction in the use of paper versions and thanks to the possibility of instantly retrieving a specific prescription. The prescription can also be consulted simultaneously by the different actors, which was not possible with the paper version.

Another advantage of this digitization is the centralization of all this information in one place, which is the UHMEP database.

In addition, UHMEP will use templates for each type of referral prescription as well as for care proposals. These models will be based on the international FHIR standard and adapted at the national level by the eHealth standardization team. FHIR is a standard that describes the format and the exchange of medical data between different computer systems.

The use of models will make it possible to avoid certain errors when writing and to avoid potential confusion when reading the prescription by the different actors.

The UHMEP database will therefore have all the information of referral prescriptions and medical proposals for all Belgian citizens. Referral prescriptions created within a hospital for internal use are not within the scope of UHMEP.

Around this database, an API will be set up to allow the various actors to access this information. The API will be used by several applications for healthcare professionals to enable them to carry out their work, but also by a web and mobile application for citizens so that they can consult the status of their prescriptions. This API will be deployed on the eHealth API Gateway.

6.2 Access management :

6.2.1 eHealth token

The token of eHealth allows UHMEP to identify who is connected. Each integrator must send the user token with the request if they want to use the UHMEP API.

To use UHMEP, the integrator needs to use the token exchange. Indeed, the user will connect himself to the integrator software and will receive rights for the pseudonymization service (identify + pseudonymize) that UHMEP **must not** receive. To avoid this transmission of rights by giving this access token, the integrator will exchange this token with another token without the pseudonymization rights but with the rights to call the back end. It is this exchanged token that the integrators must use to call UHMEP.

Technical information about token generation and token exchange is available in eHealth pages, see section 2.1.

The exchanged token has the following information that will be used for the access management :

- SSIN for the healthcare professional
- The type of professional which gives us the discipline
- The role of the user connected (Described in the section 6.2.2)

The SSIN of the patient will be pseudonymized and will be present in the *user info* token only.

There is other information present in the token but those are not used for the access management.

6.2.2 Roles

A connected user can have three roles which are used to make a first filter on which operation can be accessed or not.

The roles are:

- Prescriber
- Caregiver
- Patient
- Admin
- Head nurse

An extra role “Restricted” exists but has another purpose. If a connected healthcare professional has only this role, it means that he is suspended and he will have a limited access to UHMEP.
If this role is present with the role “Prescriber” or/and “Caregiver”, the healthcare professional is not suspended and will have a normal access to UHMEP.

The operations accessible by a patient are the following ones:

- Consultation of a prescription
- Consultation of a list of prescriptions
- Consultation of a task which describes the execution of a caregiver on the prescription
- Assignation of a caregiver/organization to a prescription

- Removal of a caregiver/organization from a prescription
- Cancellation of a prescription

There is no distinction between patients regarding access.

The operations accessible by a prescriber are the following ones:

- Creation of a prescription
- Consultation of a prescription
- Consultation of a list of prescriptions
- Consultation of a task which describes the execution of a caregiver on the prescription
- Cancellation of a prescription

The operations accessible by a caregiver are the following ones:

- Consultation of a prescription
- Consultation of a list of prescriptions
- Consultation of a task which describes the execution of a caregiver on the prescription
- Execution of a treatment (start, finish, cancel, interrupt)
- Assignation of a caregiver/organization to a prescription
- Removal of an assignation
- Transfer of an assignation: This specific end user operation is possible through two requests to UHMEP API: One assignation of caregiver (section 8.3.1) followed by a removal of an assignation (section 8.3.2) or an interruption of execution (section 8.4.3).

For the role Prescriber and Caregiver, after the first filter, the access matrix will be used to determine if the connected user can access to an operation depending on his discipline and the type of prescription he is handling.

The operations accessible by a head nurse or admin are the following ones:

- Consultation of a prescription
- Consultation of a list of prescriptions
- Consultation of a task which describes the execution of a caregiver on the prescription
- Assignation of a caregiver/organization to a prescription
- Removal of an assignation

6.2.3 Access matrix

In UHMEP, the access matrix checks the authorization for the healthcare professional when he tries to take an action on a prescription. It gives the access management rules and verifies if the user is allowed to make the request.

The access matrix is composed of :

- The methods available in UHMEP.

- The different types of prescription (templates) defined by INAMI
- The discipline of the healthcare professional (retrieved from the token).

Each time the connected user will try to make an action in UHMEP, the application will verify these three data and determine whether the user can act on the resource.

The access matrix is defined by the INAMI. The access matrix can be found in the business requirements release.

6.2.4 Suspension

If a healthcare professional is suspended, it means that he is not authorized to work during this period but can still introduce retroactively his work in UHMEP before his suspension. ***This notion must still be correctly defined by the different stakeholders working on the project UHMEP. The correct definition will come in a future release.***

When a professional is suspended, certain operations are not accessible anymore. For the ones to which he has still access, the operational behavior will be modified. This will be described in the section 8.

6.2.5 Informed Consent, Therapeutic Relationship & Exclusion

UHMEP verifies conditions related to informed consent, the therapeutic relationship, and any therapeutic exclusion between a patient and a healthcare professional. These checks determine if a user has the right to perform an operation. The following table details which checks are performed for each operation.

6.2.5.1 Terminology

- **Required**: The specified condition must be met (e.g., an active therapeutic relationship must exist) for the action to proceed.
- **Forbidden**: The specified condition must not be met (e.g., the patient must not have an exclusion for the professional) for the action to proceed.
- **N/A**: This check is not performed for the given operation.

6.2.5.2 Example Walkthrough

Looking at the row for an "Assignation" where the connected user is not the original requester:

- UHMEP checks the Exclusion between the connected user and the patient.
- The **Forbidden** state means the action will only succeed if there is **NO** exclusion in place.
- If an exclusion **is** found, UHMEP will block the action. In this case, it will return an “EXCLUSION_EXISTS” error. See section 9 for error management.
- No checks are performed for Informed Consent or Therapeutic Relationship.

6.2.5.3 *Summary Table*

Operation	Connected User	Informed consent	Exclusion	Therapeutic relationship
Create a prescription	All users	No therapeutic relation, exclusion and informed consent checks are applicable for this action.		
Cancellation of a prescription/proposals	All users	No therapeutic relation, exclusion and informed consent checks are applicable for this action.		
Assignation of a healthcare professional/of an organization *Additionally, the healthcare professional to be assigned should not be excluded by the patient to allow the assignation.*	• **Prescriptions**: Prescriber (also requester) • **Annex 81 proposals**: Nurse (also requester)	**N/A**	**N/A**	**N/A**
	• **Prescriptions:** Prescriber (not requester) • **Annex 81 proposals:** Nurse (not requester)	**N/A**	**Forbidden**	**N/A**
	• **Prescriptions**: Caregiver (assigned → _has:Task:focus:owner=himself) • **Annex 81 proposals**: Physician (assigned)	**N/A**	**Forbidden**	**N/A**
	• **Prescriptions:** Caregiver (not assigned) • **Annex 81 proposals:** Physician (not assigned)	**N/A**	**Forbidden**	**N/A**
	Patient	**N/A**	**N/A**	**N/A**
	Head Nurse	**N/A**	**Forbidden**	**N/A**
	Admin	**N/A**	**N/A**	**N/A**

Remove an assignation from a prescription/proposal (healthcare professional and organization)	All users	No therapeutic relation, exclusion and informed consent checks are applicable for this action.		
Start an execution	Caregiver (assigned → _has:Task:focus:owner= himself)	**N/A**	**N/A**	**N/A**
	Caregiver (not assigned)	**N/A**	**Forbidden**	**N/A**
Finish an execution	Caregiver connected is not yet assigned/has not yet started an execution (_has:Task:focus:owner != himself, only possible through a POST /Task with both start and end date of execution)	**N/A**	**Forbidden**	**N/A**
	Caregiver assigned	**N/A**	**N/A**	**N/A**
Cancellation of an execution	All users	No therapeutic relation, exclusion and informed consent checks are applicable for this action.		
Interruption of an execution	All users	No therapeutic relation, exclusion and informed consent checks are applicable for this action.		
Consultation of a list of prescriptions/proposal *Rules are described in section 8.5.2 . No error will be returned, but each item (prescription/proposal) that does not follow those rules, will not be returned.*	• **Prescriptions**: Prescriber (also requester) • **Annex 81 proposals**: Nurse (also requester)	**N/A**	**N/A**	**N/A**
	• **Prescriptions:** Prescriber (not requester) • **Annex 81 proposals:** Nurse (not requester)	**Required**	**Forbidden**	**Required**
	• **Prescriptions**: Caregiver (assigned → _has:Task:focus:owner= himself)	**N/A**	**Forbidden**	**N/A**

	• **Annex 81 proposals**: Physician (assigned)			
	• **Prescriptions:** Caregiver (not assigned) • **Annex 81 proposals:** Physician (not assigned)	**Required**	**Forbidden**	**Required**
	Patient	**N/A**	**N/A**	**N/A**
	Head Nurse (organization is assigned → _has:Task:focus:owner= his organization)	**N/A**	**Forbidden**	**N/A**
	Head Nurse (organization not assigned)	**Required**	**Forbidden**	**Required**
	Admin (organization is assigned → _has:Task:focus:owner= his organization)	**N/A**	**N/A**	**N/A**
	Admin (organization not assigned)	**Required**	**N/A**	**N/A**
Consultation of a specific prescription *This includes the consultation of Tasks, the Graph consultation and direct consultation of prescription/proposal. This is also valid for direct consultation via RefID and Short code + patient SSIN.*	• **Prescriptions**: Prescriber (also requester) • **Annex 81 proposals**: Nurse (also requester)	**N/A**	**N/A**	**N/A**
	• **Prescriptions:** Prescriber (not requester) • **Annex 81 proposals:** Nurse (not requester)	**N/A**	**Forbidden**	**N/A**
	• **Prescriptions**: Caregiver (assigned → _has:Task:focus:owner= himself) • **Annex 81 proposals**: Physician (assigned)	**N/A**	**Forbidden**	**N/A**
	• **Prescriptions:** Caregiver (not assigned)	**N/A**	**Forbidden**	**N/A**

	• **Annex 81 proposals:** Physician (not assigned)			
	Patient	**N/A**	**N/A**	**N/A**
	Head Nurse	**N/A**	**Forbidden**	**N/A**
	Admin	**N/A**	**N/A**	**N/A**
Create a proposal	All users	No therapeutic relation, exclusion and informed consent checks are applicable for this action.		
Approve/Reject an Annex81 proposal	Prescriber (assigned → _has:Task:focus:owner= himself)	**N/A**	**N/A**	**N/A**
	Prescriber (not assigned)	**N/A**	**Forbidden**	**N/A**

6.3 FHIR

FHIR (Fast Healthcare Interoperability Resource) is used by UHMEP as the transport layer. All communications with UHMEP should follow guidelines defined by HL7. The Belgian version of FHIR is managed by eHealth. All links in this document should refer to the Belgian FHIR resources when provided, or to international ones if no changes have been enforced in Belgium.

All resources used in UHMEP are defined and described in the implementation guide artifacts from HL7 Belgium. This document will not explain different fields of these resources but will refer to the FHIR profile definition used by UHMEP. In general, in each resource, UHMEP supports all fields marked as "Must Support" and all mandatory fields in the FHIR documentation. For instance, Figure 1Figure 1 shows a portion of the implementation guide of the BeReferralTask and it states that :

- ***status*** has a cardinality of "1..1" so it is mandatory and used by UHMEP
- ***statusReason*** and ***intent*** are marked as "Must Support" (the red "S"), so they are used by UHMEP
- ***businessStatus*** is optional and not marked as "Must Support", so it may be discarded by UHMEP ***(This will be confirmed in a future release)***.

status	?! Σ	1..1	code	draft | requested | received | accepted | + **Binding:** TaskStatus (required): The current status of the task.
statusReason	S Σ	0..1	CodeableConcept	Reason for current status **Binding:** BeVSTreatmentStatusReason (required)
businessStatus	Σ	0..1	CodeableConcept	E.g. "Specimen collected", "IV prepped" **Binding:** (unbound) (example): The domain-specific business-contextual sub-state of the task. For example: "Blood drawn", "IV inserted", "Awaiting physician signature", etc.
intent	S Σ	1..1	code	unknown | proposal | plan | order | original-order | reflex-order | filler-order | instance-order | option **Binding:** BeVsRequestIntent (required)
priority		0..1	code	routine | urgent | asap | stat **Binding:** RequestPriority (required): The task's priority.

Figure 1 : Part of the profile definition of the BeReferralTask

The implementation guide is a guide that should be used by both integrators and UHMEP just before sending the message and before processing it. This implementation guide can be used as a JSON validator on both end of the communication. All information about how to use it is available on the link :

https://build.fhir.org/ig/hl7-be/referral/branches/earlyadopter/index.html

6.3.1 Capability Statement

As a FHIR compliant server, UHMEP makes available a statement of the accessible features. It is described in the CapabilityStatement page from the FHIR International website.

This method is accessible to everyone, without any token from eHealth. The endpoint is the following :

`GET /metadata`

UHMEP will send back the resource CapabilityStatement filled with information about the API.

6.3.2 A prescription in FHIR resources

To create a Belgian prescription in UHMEP, the prescriber must create and send to UHMEP the resource BeReferralServiceRequest. Depending on its content, UHMEP will make available different other resources linked to this BeReferralServiceRequest. A BeReferralTask is always made available on creation of a prescription. It has the same ID as the one of the BeReferralServiceRequest and is linked to the BeReferralServiceRequest through its ***focus*** field and can be consulted with the endpoint :

GET /Task/{ID}

On prescription creation, BePractitionerRole and BePractitioner resources are also made available for consultation as described in section 6.3.4.

6.3.2.1 Prescription statuses

To follow the lifecycle of a prescription. Every business status has a mapping with FHIR available statuses through many resources (BeReferralServiceRequest + BeReferralTask).

The current mapping between business and FHIR statuses is :

Business Status	*BeReferralServiceRequest.status*	*BeReferralServiceRequest.statusReason*	*BeReferralTask.status*
Draft	“draft”	/	“draft”
Blacklisted	“entered-in-error”	/	“entered-in-error”
Pending	“active”	/	“draft”
Ready	“active”	/	“ready”
Canceled	“revoked”	“cancelingReason”	“cancelled”
Expired	“revoked”	“expired”	“cancelled”
In Progress	“active”	/	“in-progress”
Inactive	“active”	/	“on-hold”
Completed	“completed”	/	“in-progress”/”completed”

In the cookbook, business statuses will be used when talking about statuses to be more aligned with the requirements document and to lighten the explanation by avoiding expressing the value of each resource.

6.3.3 Execution of caregivers on prescriptions

The Belgian referral prescription is mainly stored in the BeReferralServiceRequest. Once created by the prescriber, clinical data in this resource are fixed and should not be changed afterward. This prescription will start its "prescription status lifecycle" like described in the business requirements release.

Another resource is used to handle caregiver interactions with the prescription, namely the BePerformerTask. This resource is linked to the BeReferralServiceRequest through a reference in its ***focus*** field and a reference to the BeReferralTask in its ***partOf*** reference.

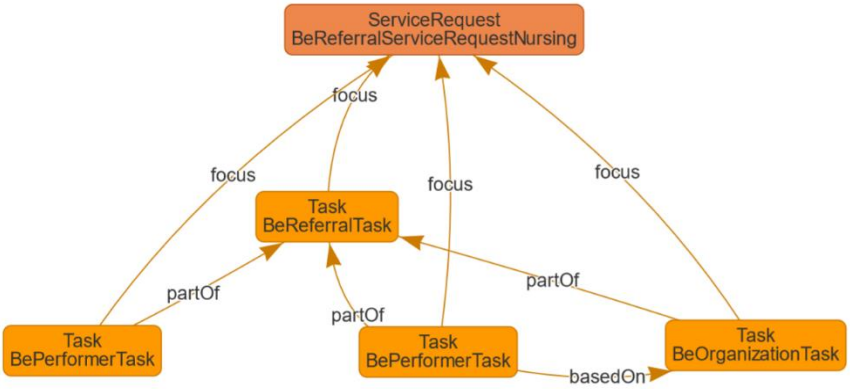

Figure 2 : Links between Task resources to the BeReferralServiceRequest

Similarly, the BeOrganizationTask resource is used to store an assignation of an organization on a prescription. If a caregiver is working for an organization, his BePerformerTask should be linked to the BeOrganizationTask through the ***focus*** fieldThe **Error! Reference source not found.** above shows links between BePerformerTasks, BeReferralTask, BeOrganizationTask and BeReferralServiceRequest when there are two executions on a prescription, when linked to an assignation to an organization.

All information about operations and consultation that can be done on this resource is available in section 8.

The BePerformerTask resource is created in two use cases :

- At the assignation of a caregiver on a prescription by a Prescriber or Patient
- When the caregiver starts his execution, and the resource wasn't already created (due to assignation)

After this, all operations are done using the PATCH operation of FHIR. The Figure 3 below shows the status flow of a BePerformerTask. Gray statuses mean that no further operation is allowed on this state. For instance, once the BePerformerTask ***status*** is “rejected”, “on-hold” or “completed”, the resource cannot be updated anymore.

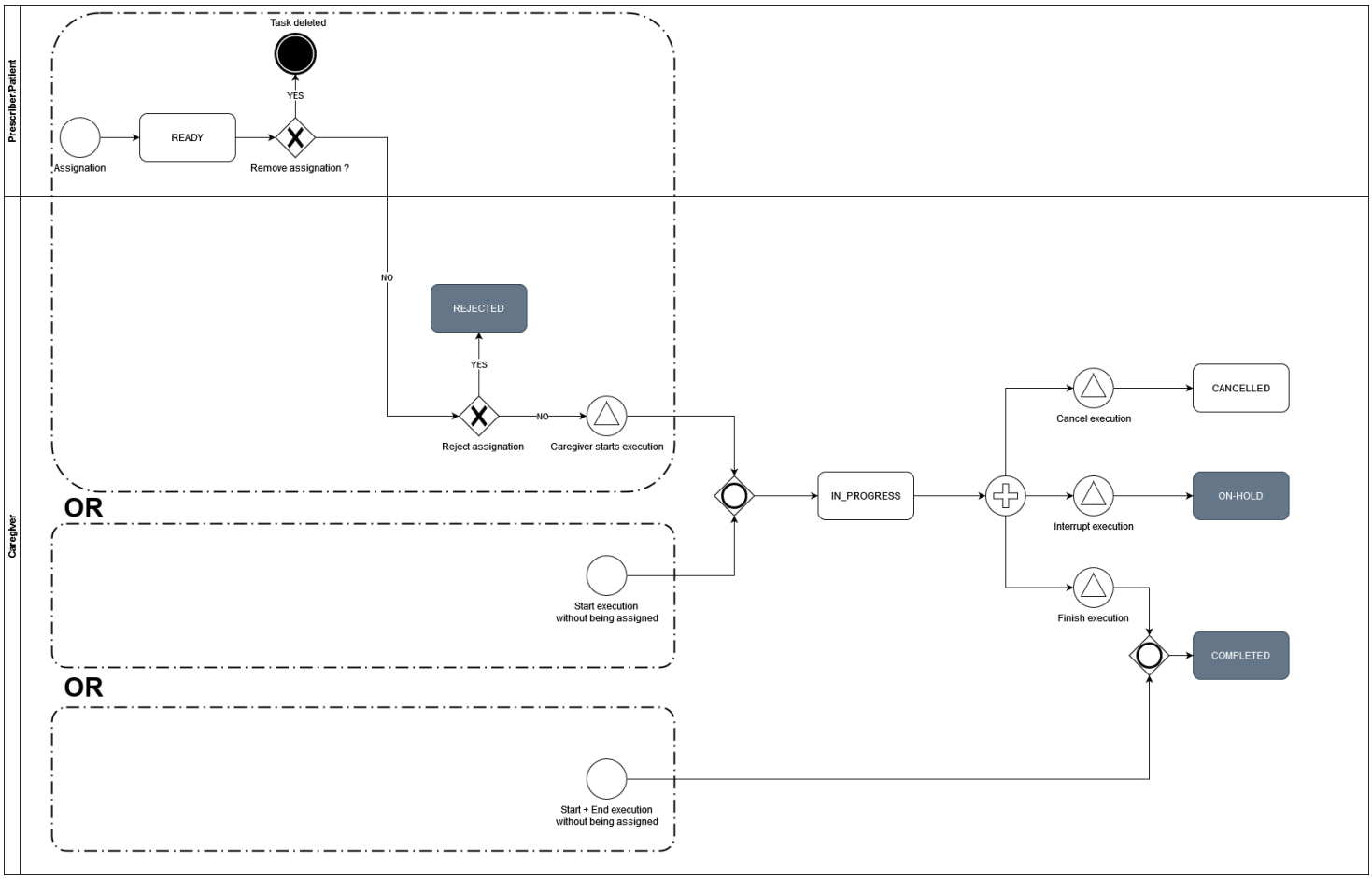

Figure 3 : Status diagram of BePerformerTask

6.3.4 References to healthcare professionals

As mentioned above, UHMEP uses FHIR as the transport layer for all communications. In resources like the BeReferralServiceRequest and BePerformerTask, references to healthcare professionals are done with references to BePractitionerRole FHIR resource.

IDs of BePractitionerRole in UHMEP are the concatenation between the SSIN and the discipline of the healthcare professional. For example, for a “NURSE” with “82042605839” as SSIN, the ID of the BePractitionerRole in UHMEP for this person would be : “82042605839-NURSE”.

Disciplines are defined by eHealth/INAMI and available through CoBRHA. Disciplines should be in UPPERCASE and all spaces replaced by underscores ("_").

For all resources, references can be put directly like "PractitionerRole/82042605839-NURSE". In the BePerformerTask for example, the reference of the ***owner*** field will have the format like below :

```
"owner": {
    "reference": "PractitionerRole/82042605839-NURSE"
}
```

Snippet 1 : Owner reference example in a BePerformerTask

The BePractitionerRole is created by UHMEP and can be consulted using the usual FHIR resource consultation endpoint format :

GET /PractitionerRole/82042605839-NURSE

This resource itself does a reference to a ***practitioner*** that is a reference to a BePractitioner FHIR resource. IDs of BePractitioner are only the SSIN of the healthcare professional.

An example is available below:

```
"practitioner": {
    "reference": "Practitioner/82042605839"
}
```

Snippet 2 : Practitioner reference example in a BePractitionerRole

Similarly, the BePractitioner resource is created by UHMEP and can be consulted using the usual FHIR resource consultation endpoint format:

GET /Practitioner/82042605839

6.3.5 References to organizations

BeOrganizationTask were introduced in section 6.3.3, and are used to store assignation of organizations on prescriptions. References to an organization should be a reference to a BeOrganization FHIR resource.

IDs of BeOrganization in UHMEP are the concatenation between the NIHII and the organization type of the organization. For example, for an organization with a type “940” and “94754895” as NIHII, the ID of the BeOrganization UHMEP for this organization would be : “NIHII-94754895-940”. “NIHII” is not case sensitive.

Organization types are defined by eHealth/INAMI and available through CoBRHA.

For all resources, references can be put directly like "Organization/NIHII-94754895-940". In the BeOrganizationTask for example, the reference of the ***owner*** field will have the format like below :

```
"owner": {
    "reference": " Organization/NIHII-94754895-940"
}
```

Snippet 3 : Owner reference example in a BePerformerTask

The BeOrganization is created by UHMEP and can be consulted using the usual FHIR resource consultation endpoint format :

GET /Organization/NIHII-94754895-940

6.3.6 Pseudonymization of SSIN

To ensure the protection of the patient data, UHMEP uses the pseudonymization for the patient identifier (SSIN).

All direct pseudonyms (representing an underlying identifier like an SSIN or an encryption key that has been processed by the pseudonymization service) **must** follow this structure:

urn:be:fgov:pseudo:v2:{SEC1}:{transitInfo}

- urn:be:fgov:pseudo:v2: Standard prefix indicating a version 2 direct pseudonym.
- {SEC1}: The first part of the pseudonym service's output.
- {transitInfo}: The second part, a JWE compact string. This JWE contains metadata critical for the pseudonymization service and the recipient

Please refer to the eHealth Pseudonymisation REST Cookbook, referenced in section 2.1.

6.3.6.1 FHIR example

Following eHealth FHIR recommendations for pseudonym handling, here is an example of how patient identifiers should be used in FHIR resources in UHMEP:

```
"subject": {
  "identifier": {
    "use": "official",
    "system":
"https://www.ehealth.fgov.be/standards/fhir/core/NamingSystem/ssin",
    "value": "urn:be:fgov:pseudo:v2:{SEC1}:{transitInfo}",
    "_value": {
      "extension": [
        {
          "url":
"https://www.ehealth.fgov.be/standards/fhir/infsec/StructureDefinition/be-ext-
pseudonymization",
          "extension": [
            {
              "url": "marker",
              "valueBoolean": true
            },
            {
              "url": "format",
              "valueCode": "direct"
            },
            {
              "url": "version",
              "valuePositiveInt": 2
            }
          ]
        }
      ]
    }
  }
}
```

6.3.7 Pseudonymization impacts on free text fields

In the free text fields, it is possible that the prescriber adds some sensitive data about the patient.

To avoid any exposure of sensitive data, all free text fields are encrypted and the key is pseudonymized. In FHIR, it means that each field which can contain a text must be attached with this FHIR extension "be-ext-pseudonymization". Fields concerned:

- BeCodedAnnotation.text
- CodeableConcept.text

Following eHealth Guidance, the key used for the encryption is provided in a pseudonymized way in the FHIR extension "be-ext-pseudonymization" which is contained in the FHIR extension "be-ext-key-pseudonymization" which is in the field "meta" of the ServiceRequest.

For each free text field, the identifier of the key (the "kid") must be the same as the one provided in the "meta" field. **All systems should use and reuse the key provided in the "meta" field of a FHIR resource**. Indeed, one key can encrypt and decrypt multiple free text in one FHIR resource. If no key has been already attached, or at the creation of a resource, it is up to the system to generate one following eHealth guidance (can be found in the Annex 9 of the Pseudonymisation REST Cookbook).

If multiple key are present, or no key linked to a free text, UHMEP will reject the request.

References to encrypted free text content **must** follow this structure:

urn:be:fgov:pseudo-encrypted:v1:{KID}:{JWE}

- urn:be:fgov:pseudo-encrypted:v1: Standard prefix indicating a version 1 encrypted payload reference.
- {KID}: The **Key Identifier** (a UUID) of the symmetric encryption key that was used to encrypt the actual free text. This KID allows systems to locate the pseudonymized encryption key.
- {JWE}: The compact JWE string representing the actual encrypted free text content.

6.3.7.1 FHIR Examples

6.3.7.1.1 Pseudonymized Encryption Key

```
{
  "resourceType": "ServiceRequest",
  "meta": {
    "extension": [
      {
        "id": "370f2865-03ff-4c7b-affd-897a4aae187e", // The KID for this key
        "url": "https://www.ehealth.fgov.be/standards/fhir/infsec/StructureDefinition/be-ext-key-pseudonymization",
        "extension": [
          {
            "url": "key",
            "valueString": "urn:be:fgov:pseudo:v2:{SEC1}:{transitInfo} ",
            "_valueString": {
              "extension": [
                {
                  "url": "https://www.ehealth.fgov.be/standards/fhir/infsec/StructureDefinition/be-ext-pseudonymization",
                  "extension": [
                    {
                      "url": "marker",
                      "valueBoolean": true
                    },
                    {
                      "url": "format",
                      "valueCode": "direct"
                    },
                    {
                      "url": "version",
                      "valuePositiveInt": 2
                    }
                  ]
                }
              ]
            }
          }
        ]
      }
    ]
  }
  // ... other resource fields, including encrypted text referencing this KID ...
}
```

6.3.7.1.2 Encrypted free text

```
"text": "urn:be:fgov:pseudo-encrypted:v1:370f2865-03ff-4c7b-affd-897a4aae187e:{JWE}", // The same KID for the key, alongside the encrypted free text in JWE format
"_text": {
  "extension": [
    {
      "url": "https://www.ehealth.fgov.be/standards/fhir/infsec/StructureDefinition/be-ext-pseudonymization",
      "extension": [
        {
          "url": "marker",
          "valueBoolean": true
        },
        {
          "url": "format",
          "valueCode": "encrypted"
        },
        {
          "url": "version",
          "valuePositiveInt": 1
        }
      ]
    }
  ]
}
```

7 Technical requirements

7.1 Tracing

To use this service, the request MUST 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.

    a. Pattern: {company}/{package-name}/{version} {platform-company}/{platform-package-name}/{platform-package-version}

    b. Regular expression for each subset (separated by a space) of the pattern: [[a-zA-Z0-9-\/]*\/[0-9a-zA-Z-_.]*

    c. Examples:
    User-Agent: MyCompany/myProduct/62.310.4 eHealth/Technical/3.19.0
    User-Agent: Topaz-XXXX/123.23.X Taktik/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

7.2 API documentation

The last version of REST interface described with a JSON / Swagger API is available on the eHealth API Portal :

Environment	Endpoint
Acceptance	https://portal-acpt.api.ehealth.fgov.be
Production (**not available at this moment**)	https://portal.api.ehealth.fgov.be

Endpoint to call UHMEP is described in the API documentation on the eHealth API Portal.

Different operations are available and textually described in section 8.

8 Description of the operations

8.1 Common validations

For all incoming requests, UHMEP applies a suite of common validations if the request is validated by the implementation guide described in section 6.3.

8.1.1 Validation of dates

In the payload, different types of date are used.

For dates, the format must be YYYY-MM-DD.
For the recorded date (***authoredOn*** field), UHMEP will return it with the timestamp format (dates and time format combined): YYYY-MM-DDThh:mm:ss+zz:zz (for example : `2023-05-12T13:37:42+02:00`). The timestamp format was integrator-defined.

Also, it is not allowed to create a resource with a timestamp in the future.

8.1.2 Validation of the template

UHMEP is using FHIR templates to exchange data between the applications. The templates follow INAMI's templates of referral prescriptions. The template must be known from UHMEP. If it is not the case, the user will receive an error.

Currently, for this version of the MVP, UHMEP supports the following templates:

8.1.2.1 For nursing care

- Preparation of drugs not refundable
- Diabetes education via convention center
- Diabetes education for patients with model of care/preliminary process "follow up of patients with diabetes type 2"
- Diabetic education for patients with care path
- Diabetic education for patients without a care path
- Assisting with personal hygiene
- Chronic peritoneal dialysis
- Bleeding
- Glycemic Test
- Parameters
- Sampling
- Annex 81
- Generic template
    - This template serves as a placeholder for the remaining templates that are yet to be implemented on UHMEP back-end's side

In the BeReferralServiceRequest, they are defined with a combination of ***code*** (sometimes) ***orderDetail*** and ***category.*** They are defined in INAMI document about templates. For example, "Diabetes education via convention center" prescription is a BeReferralServiceRequest with:

- ***code*** = "385805005"
- ***code system*** = "http://snomed.info/sct"
- ***orderDetail =*** "tmp-with-tion-7"
- ***orderDetail system*** = "https://www.ehealth.fgov.be/standards/fhir/referral/CodeSystem/be-cs-temp-requested-service-detail"
- ***Category:***
    - system: "http://snomed.info/sct",
    - code: ["9632001"]

The unique combination of the ***code (+system), orderDetail (+system***) as well as the **category** defines the template of the BeReferralServiceRequest resource.

8.1.2.2 For radiology

- Annex 82
    - The Annex 82, also called radiology prescription, is a Service Request with:
        - Code: ""tmp-img-ct","tmp-img-mri", "tmp-img-xray","tmp-img-us""
        - Code system: http://snomed.info/sct
        - Category:
            - System: "http://snomed.info/sct"
            - Code: ["363679005"]

8.1.3 Validation of the prescriber and caregiver information

The existence of the healthcare professionals is checked in the eHealth authentic source CoBRHA. UHMEP will compare the SSIN and the discipline given in the payload or retrieved from the token with CoBRHA. If the healthcare professional is not found, the user will receive an error. If he is found, UHMEP will make other validations linked to the operation called.

For the performance and to avoid useless calls, UHMEP will firstly make the check digits verification to verify that the given SSIN is a valid one respecting the structure definition.

8.1.4 Validation of the organization

The existence of the healthcare organization is checked in the eHealth authentic source CoBRHA. UHMEP will compare the NIHII number and organization type given in the payload or retrieved from the token with CoBRHA. If the referred organization does not follow INAMI's requirements, it will not be found, the user will receive an error.

8.1.5 Validation on the patient

The existence of the patient will also be checked by consulting the national register through the eHealth ConsultRN service (thanks to his pseudonymized SSIN). If the patient is not known by the national register, the prescriber will not be able to create a prescription for him. ***This validation will be implemented for the release 1.***

For the performance and to avoid useless calls, UHMEP will firstly verify the existence of the patient in the UHMEP database. If we don't find it, UHMEP will call consultRN.

In addition of this verification, every pseudonym in transit generated by UHMEP will be exclusively for this patient. This means that only this patient will be able to identify the pseudonym and it will not be possible for another citizen to identify this pseudonym.

This is done by adding the pseudonym in transit in the domain eHealth in the transit info attached to the generated UHMEP pseudonym in transit.

The healthcare professional will still be able to identify the pseudonyms.

8.1.6 Authentication

Each user using UHMEP must be authenticated to be allowed to use UHMEP. UHMEP will verify the validity of the access token. UHMEP will also use the other information contained in the token to accept or reject the request. This behavior has been described in the section 6.2 talking about access management.

8.1.7 Visible prescriptions

8.1.7.1 Depending on the template

When a prescriber/caregiver can access an operation because he has the correct role, only the prescriptions that he is allowed to see are returned. To know if he is allowed to see the prescription, a check is done by looking the consultation right in the access matrix like it is described in section 6.2.3.

8.1.7.2 Blacklisted prescriptions

When a prescriber/caregiver can access an operation because he has the correct role, some prescriptions in the status “Blacklisted” will not be returned.

A prescription can be blacklisted at the creation because the prescriber was suspended (see the suspension explanation in section 6.2.4) but it can also be blacklisted after its creation thanks to a retro-action script done by UHMEP. Only prescription created in the suspension period of the prescriber will be blacklisted.

This script is launched when the information of the prescriber suspension is received only some days after the creation.

Depending on the actual status of the prescription, the blacklisting script will have different behaviors :

- If the prescription was in one of the business statuses “Draft”, “Ready”, “Cancelled”, “Expired” :
  - The prescription is blacklisted.
  - The prescription will not be returned (in case of a specific consultation, an error will be returned).
- If the prescription was in the business status “In Progress”:
  - The prescription is blacklisted.
  - The status of all executions in “in-progress” is changed to “interrupted”.
  - The prescription will be returned.
- If the prescription was in one of the business statuses “Inactive”, “Completed” :
  - The prescription is blacklisted.
  - The prescription will be returned.

To summarize, when a prescription is in the status “Blacklisted”, it will be returned if there is one or more executions. If there is nothing or only assignations on it, the prescription will not be returned.

8.2 Operations on prescriptions

8.2.1 Creation of a prescription

This method allows the end user to create a referral prescription via the user interface in the UHMEP application.

8.2.1.1 Allowed roles

Prescriber, caregiver (only for radiology prescriptions)

8.2.1.2 Endpoint

POST /ServiceRequest

8.2.1.3 Implemented rules

There are different rules that this service must check to allow the end user to be able to create a prescription:

- The prescriber is not suspended.
  - If he is suspended, an error will be returned but the prescription will be created with the status “Blacklisted”.
- ***validity.start*** date can be maximum 5 days before the recorded date
  - 5 days is the current default value for all templates but it can be configured by INAMI for each template.
- ***validity.start*** =< ***validity.end*** =< ***latest***
- ***validity.end*** & ***latest*** > ***authoredOn***
- If an extend prescription is given (in the ***focus*** field), verification that the two prescriptions are for the same patient and the same template.
- When creating a request, the content of the request must match the content of the token. This means that a healthcare professional (prescriber/caregiver) is **always** combined by the **combination** of an **SSIN** and a **Discipline**.
  - The combination must exist in CoBRHA DB and to be exactly the same as in the token for the user to be able to create a prescription.
- The creation of a resource (prescription or proposal) with an intent other than “order” or “proposal” is forbidden
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

If the ***validity.start*** is empty, UHMEP will set this date with the value of the ***authoredOn*** field converted to the correct format YYYY-MM-DD.

For a **radiology** prescription, the caregiver is also able to proceed with a creation. In that case, the same checks should be carried out as for a nursing prescription created by a prescriber.

8.2.1.4 Request

The request must contain the information of the prescription in the body using the BeReferralServiceRequest resource.

Particularity coming from UHMEP :

Field	Particularity
intent	UHMEP will accept only the value “order”.
focus	UHMEP will accept only a reference to an existing ServiceRequest.

8.2.1.5 Response

On successful creation of the BeReferralServiceRequest resource, the server will respond with a HTTP 201. The ID of this newly created resource is accessible in the *Location* header. This ID can be used to further interactions with this resource.

8.2.2 Cancel a prescription

This method allows the end-user to cancel a referral prescription via his user interface in the UHMEP application. It means that the end-user would be able to continue to consult it but the prescription will no longer be executable.

8.2.2.1 Allowed roles

Prescriber and Patient.

8.2.2.2 Endpoint

PATCH /ServiceRequest/{prescriptionID}

8.2.2.3 Implemented rules

The cancellation of a specific prescription is possible only if all the following points are respected:

- The prescription is in one of the business statuses:
  - “Draft”
  - “Pending”
  - “Ready”
- (Prescriber) He is not suspended.
- (Prescriber) He is the prescriber of the prescription.
- (Patient) The prescription is for the patient making the request.
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.2.2.4 Request

The operation is done through a PATCH on the BeReferralServiceRequest resource. It should only replace the ***status*** of this resource to the one “revoked”. An example is available in Snippet 4 :

```
{
    "resourceType": "Parameters",
    "parameter": [
        {
            "name": "operation",
            "part": [
                {
                    "name": "type",
                    "valueCode": "replace"
                },
                {
                    "name": "path",
                    "valueString": "ServiceRequest.status"
                },
                {
                    "name": "value",
                    "valueString": "revoked"
                }
            ]
        }
    ]
}
```

Snippet 4 : PATCH example on a BeReferralServiceRequest to cancel a prescription

8.2.2.5 Response

On successful cancellation of the prescription, the server sends back an HTTP 200.

8.3 Assignation of a caregiver on a prescription

8.3.1 Add a caregiver

Once a prescription has been created, the system allows users to assign a specific caregiver on it. Giving them an ease of access to this prescription and some access rights.

8.3.1.1 Allowed roles

Patient, Caregiver, Head and Admin.

8.3.1.2 Endpoint

POST /Task

8.3.1.3 Implemented rules

The assignation of a caregiver on an existing prescription is possible only if all following rules are respected :

- (Patient/Caregiver) The prescription referenced in the ***focus*** field is in one of the business statuses :
  - "Draft"
  - "Pending"
  - "Ready"
  - "In progress"
  - "Inactive"
  - "Completed"
- (Patient/Caregiver) The ***status*** of the newly created BePerformerTask in the request is "ready".
- (Patient/Caregiver) No ***executionPeriod*** is provided.
- (Patient/Caregiver) No other BePerformerTask in ***status*** ["ready", "in-progress", "completed"] is already linked with ***focus*** to the same prescription for the same ***owner***.
- (Patient/Caregiver) The caregiver to assign is not currently suspended.
- (Patient/Caregiver) The caregiver to assign exists in CoBRHA database.
- (Patient/Caregiver) The discipline of the caregiver to assign is allowed to consult following the Access Matrix table for the template of the prescription.
- (Patient) The connected patient is the same as the one of the prescriptions.
- (Caregiver) The healthcare professional connected is not suspended.
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.3.1.4 Request

The assignation is done through the creation of a BePerformerTask resource.

For more information about how BePerformerTask are handled in UHMEP, see section 0.

Particularity coming from UHMEP :

Field	Particularity
focus	This is a reference to the prescription, the BeReferralServiceRequest. (ex: ServiceRequest/{referenceID})
partOf	This is a reference to a BeReferralTask, that currently has the same ID as the prescription. (ex: Task/{referenceID})
intent	UHMEP only allows “order”

8.3.1.5 Response

On successful creation of the BePerfomerTask resource, the server will respond with a HTTP 201. The ID of this newly created resource is accessible in the *Location* header. This ID can be used for further interactions with this resource.

8.3.2 Remove a caregiver

The inverse operation is also possible to remove an assigned caregiver from a prescription.

8.3.2.1 Allowed roles

Patient, Caregiver, Admin and Head

8.3.2.2 Endpoint

DELETE /Task/{executionID}

{executionID} being the ID of the BePerformerTask to be removed from a BeReferralServiceRequest.

8.3.2.3 Implemented rules

The removal of an assigned caregiver on an existing prescription is possible only if all following rules are respected :

- The prescription referenced in the ***focus*** field is in one of the business statuses:
  - “Draft”
  - “Pending”
  - “Ready”
  - “In Progress”
  - “Inactive”
- The execution of the caregiver (BePerformerTask) is in ***status*** “ready”.
- The user connected is not suspended.
- The patient connected is the same as the one of the prescriptions.
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.3.2.4 Request

In addition of the authorization and header described above, this request does not need a body.

8.3.2.5 Response

On successful deletion of the resource, the server sends back a HTTP 204. After this operation, the resource is definitively deleted and cannot be used anymore.

8.4 Execution of a prescription

This section will describe the different operations that a caregiver is able to do on a prescription. Technically, a caregiver cannot modify the BeReferralServiceRequest resource. He will update his execution through the BePerformerTask resource (see section 0).

The different operations available are :

- Start
- Finish
- Interrupt
- Cancel

8.4.1 Start an execution

The start operation allows the caregiver to indicate to UHMEP that he works or has started his work on the prescription.
His execution will be in the ***status*** "in-progress".

There are two ways in UHMEP to start an execution :

- By creating a BePerformerTask
- By updating a BePerformerTask thanks to a PATCH method

For more information about BePerformerTasks handling by UHMEP, see section 0.

The operation of "starting an execution" only adds a start date on the execution. This start date can be in the past, to allow caregivers to do their administrative work afterward.

8.4.1.1 Start an execution through a creation of resource

If the caregiver is not assigned on the prescription, this method must be used to create the execution directly in the ***status*** "in-progress".

8.4.1.1.1 Allowed Role

Caregiver.

8.4.1.1.2 Endpoint

POST /Task

8.4.1.1.3 Implemented rules

The creation of an execution task is possible only if all the following points are respected :

- The caregiver is/was not suspended on the date provided in the ***executionPeriod.start***.
- The prescription referenced in the ***focus*** field is in one of the business statuses:
  - "Pending"
  - "Ready"
  - "In progress"
  - "Inactive"
  - "Expired" (Only if the ***executionPeriod.start*** is before the ***extension.validity.end*** date of the BeReferralServiceRequest)
- No other execution task in "ready", "in-progress" or "completed" exists for this caregiver (execution with the same reference in the ***focus*** field).
- The task ***status*** is "in-progress".
- Only the ***executionPeriod.start*** date is provided for the execution dates.
- ***executionPeriod.start*** date is not in the future but can be in the past.
  If it is in the past:
  - [***BeReferralServiceRequest.authoredOn <= BeReferralServiceRequest.extension.validity.start***] ***executionPeriod.start*** cannot be before the ***authoredOn*** date of the BeReferralServiceRequest.
  - [***BeReferralServiceRequest.extension.validity.start < BeReferralServiceRequest.authoredOn***] ***executionPeriod.start*** cannot be before the ***extension.validity.start*** date of the BeReferralServiceRequest.
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.
- The information contained in the ***owner*** field matches with the information of the connected caregiver in the token.

8.4.1.1.4 Request

In addition of the authorization and header described above, this request needs a body containing the information of the task to create.

The execution task is represented in FHIR by the Belgian profile of the Task resource called BePerformerTask.

Particularity coming from UHMEP :

Field	Particularity
Intent	UHMEP will only accept "order"

8.4.1.1.5 Response

On successful creation of the BePerfomerTask resource, the server will respond with a HTTP 201. The ID of this newly created resource is accessible in the *Location* header. This ID can be used for further interactions with this resource.

8.4.1.2 Start an execution through a PATCH

If the caregiver is already assigned on the prescription, this method must be used to update the existing BePerformerTask (see section 0).

8.4.1.2.1 Allowed roles

Caregiver.

8.4.1.2.2 Endpoint

PATCH /Task/{executionID}

{executionID} being the ID of the BePerformerTask to be updated from a BeReferralServiceRequest.

8.4.1.2.3 Implemented rules

The update of an execution task is possible only if all the following points are respected :

- The caregiver is/was not suspended on the date provided in the ***executionPeriod.start***.
- The prescription referenced in the ***focus*** field is in one of the business statuses:
  - "Pending"
  - "Ready"
  - "In Progress"
  - "Inactive"
  - "Expired" (Only if the ***executionPeriod.start*** is before the ***extension.validity.end*** date)
- The Task must have previously the status "ready" or "cancelled".
- Only the ***executionPeriod.start*** is provided for the execution dates.
- ***executionPeriod.start*** date is not in the future but can be in the past.
  If it is in the past:
  - [***BeReferralServiceRequest.authoredOn <= BeReferralServiceRequest.extension.validity.start***] ***executionPeriod.start*** cannot be before the ***authoredOn*** date of the BeReferralServiceRequest.
  - [***BeReferralServiceRequest.extension.validity.start < BeReferralServiceRequest.authoredOn***] ***executionPeriod.start*** cannot be before the ***extension.validity.start*** date of the BeReferralServiceRequest.
- The information contained in the ***owner*** field must match with the information of the connected caregiver in the token.
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.4.1.2.4 Request

The operation is done through a PATCH on the BePerformerTask FHIR resource. It should only add the start execution date (YYYY-MM-DD) on this resource. An example is available below :

```
{
    "resourceType": "Parameters",
    "parameter": [
        {
            "name": "operation",
            "part": [
                {
                    "name": "type",
                    "valueCode": "add"
                },
                {
                    "name": "path",
                    "valueString": "Task.executionPeriod"
                },
                {
                    "name": "name",
                    "valueString": "start"
                },
                {
                    "name": "value",
                    "valueDateTime": "{{StartExecutionDate}}"
                }
            ]
        }
    ]
}
```

Snippet 5 : PATCH example on a BePerformerTask to start an execution

8.4.1.2.5 Response

On successful update of the execution, the server sends back a HTTP 200. The response does not contain a body.

8.4.2 Finish an execution

The finish operation allows the caregiver to indicate to UHMEP that he has finished his work on the prescription.
His execution will be in the ***status*** “completed”.

There are two ways in UHMEP to finish an execution :

- By creating a task
- By updating a task thanks to a PATCH method

8.4.2.1 Finish an execution through a creation of resource

If the caregiver is not assigned on the prescription, this method must be used to create the execution directly in the status “completed”.

8.4.2.1.1 Allowed roles

Caregiver.

8.4.2.1.2 Endpoint

POST /Task

8.4.2.1.3 Implemented rules

The creation of an execution task is possible only if all the following points are respected :

- The caregiver is not suspended.
- The prescription referenced in the ***focus*** field is in one of the business statuses:
  - “Ready”
  - “In Progress”
  - “Inactive”
  - “Expired”
- No other execution task in “ready”, "in-progress", “completed” and “cancelled” exists for this caregiver (execution with the same reference in the ***focus*** field).
- The BePerformerTask ***status*** is "completed" and the ***executionPeriod.start*** and ***executionPeriod.end*** dates are provided for the execution dates.
- Execution dates are not in the future.
- ***executionPeriod.start*** <= ***executionPeriod.end***.
- ***executionPeriod.end <= BeReferralServiceRequest.extension.latest.***
- ***executionPeriod.start*** date is not in the future but can be in the past.
  If it is in the past:
  - [***BeReferralServiceRequest.authoredOn <= BeReferralServiceRequest.extension.validity.start***] ***executionPeriod.start*** cannot be before the ***authoredOn*** date of the BeReferralServiceRequest.
  - [***BeReferralServiceRequest.extension.validity.start < BeReferralServiceRequest.authoredOn***] ***executionPeriod.start*** cannot be before the ***extension.validity.start*** date of the BeReferralServiceRequest.

- The information contained in the ***owner*** field matches with the information of the connected caregiver in the token.
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.4.2.1.4 Request

In addition of the authorization and header described above, this request needs a body containing the information of the task to create.

The execution task is represented in FHIR by the Belgian profile of the task resource called BePerformerTask.

Particularity coming from UHMEP :

Field	Particularity
intent	UHMEP will only accept the value “order"

8.4.2.1.5 Response

On successful creation of the BePerfomerTask resource, the server will respond with a HTTP 201. The ID of this newly created resource is accessible in the *Location* header. This ID can be used for further interactions with this resource.

8.4.2.2 Finish an execution through a PATCH

If the caregiver has already started an execution on the prescription, this method must be used to update the existing execution.

8.4.2.2.1 Allowed roles

Caregiver.

8.4.2.2.2 Endpoint

PATCH /Task/{executionID}

{executionID} being the ID of the BePerformerTask to be updated from a BeReferralServiceRequest.

8.4.2.2.3 Implemented rules

The update of an execution task is possible only if all the following points are respected :

- The caregiver is not suspended.
- The prescription referenced in the ***focus*** field is in one of the business statuses:
  - "In Progress"
- The BePerformerTask is in the ***status*** "in-progress".
- Only the ***executionPeriod.end*** date is provided in the PATCH request for the execution dates.
- ***executionPeriod.end*** date is not in the future.
- ***executionPeriod.start*** <= ***executionPeriod.end***.
- ***executionPeriod.end <= BeReferralServiceRequest.extension.latest***
- The information contained in the ***owner*** field must match with the information of the connected caregiver in the token.
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.4.2.2.4 Request

The operation is done through a PATCH on the BePerformerTask FHIR resource. It should only add the end execution date (YYYY-MM-DD) on this resource. An example is available below :

```
{
    "resourceType": "Parameters",
    "parameter": [
        {
            "name": "operation",
            "part": [
                {
                    "name": "type",
                    "valueCode": "add"
                },
                {
                    "name": "path",
                    "valueString": "Task.executionPeriod"
                },
                {
                    "name": "name",
                    "valueString": "end"
                },
                {
                    "name": "value",
                    "valueDateTime": "{{EndExecutionDate}}"
                }
            ]
        }
    ]
}
```

Snippet 6 : PATCH example on a BePerformerTask to finish an execution

8.4.2.2.5 Response

On successful update of the execution, the server sends back a HTTP 200. The response does not contain a body.

8.4.3 Interruption of an execution

A caregiver can interrupt his execution on a prescription.

8.4.3.1 Allowed roles

Caregiver

8.4.3.2 Endpoint

PATCH /Task/{executionID}

{executionID} being the ID of the BePerformerTask to be updated from a BeReferralServiceRequest.

8.4.3.3 Implemented rules

The interruption of an execution on an existing prescription is possible only if all following rules are respected :

- The caregiver is not suspended
- The BePerformerTask is in the ***status*** "in-progress".
- The information contained in the ***owner*** field matches with the information of the connected caregiver in the token.
- The business status of the prescription is "In Progress".
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.4.3.4 Request

The operation is done through a PATCH on the BePerformerTask FHIR resource. It should only replace the status of this resource to the one “on-hold”. An example is available below :

```
{
    "resourceType": "Parameters",
    "parameter": [
        {
            "name": "operation",
            "part": [
                {
                    "name": "type",
                    "valueCode": "replace"
                },
                {
                    "name": "path",
                    "valueString": "Task.status"
                },
                {
                    "name": "value",
                    "valueString": "on-hold"
                }
            ]
        }
    ]
}
```

Snippet 7 : PATCH example on a BePerformerTask to interrupt an execution

8.4.3.5 Response

On successful interruption of the execution, the server sends back a HTTP 200. This ID cannot be used anymore for further interactions with this resource.

8.4.4 Revoke an execution

A caregiver can revoke an execution on a prescription.

8.4.4.1 Allowed roles

Caregiver.

8.4.4.2 Endpoint

PATCH /Task/{executionID}

{executionID} being the ID of the BePerformerTask to be updated from a BeReferralServiceRequest.

8.4.4.3 Implemented rules

The revocation of an execution on an existing prescription is possible only if all following rules are respected :

- The caregiver is not suspended.
- The BePerformerTask is in the ***status*** "in-progress".
- The information contained in the ***owner*** field must match with the information of the connected caregiver in the token.
- The business status of the prescription is "In Progress".
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.4.4.4 Request

The operation is done through a PATCH on the BePerformerTask resource. It should only replace the status of this resource to the one “revoked”. An example is available below :

```
{
    "resourceType": "Parameters",
    "parameter": [
        {
            "name": "operation",
            "part": [
                {
                    "name": "type",
                    "valueCode": "replace"
                },
                {
                    "name": "path",
                    "valueString": "Task.status"
                },
                {
                    "name": "value",
                    "valueString": "cancelled"
                }
            ]
        }
    ]
}
```

Snippet 8 : PATCH example on a BePerformerTask to revoke an execution

8.4.4.5 Response

On successful revocation of the execution, the server sends back a HTTP 200. After this operation, the resource can be re-used by the caregiver.

8.5 Consultation of resources

8.5.1 Consultation of a BeReferralServiceRequest

This method allows the user to get information for one specific prescription by giving its reference ID.

It is also good to point out that after the creation of a prescription (BeReferralServiceRequest resource), in addition to the reference ID created for the prescription itself, a short code is also generated. The code is composed of 6 alphanumerical characters.

Therefore, the user can consult a BeReferralServiceRequest via the combination of the short code (=***identifier***) and patient SSIN (=***patient***).

If the user needs more resources around the BeReferralServiceRequest, like BePerformerTask, the recommended way is to use the $graph operation detailed in section 8.5.4. Multiple resources can be accessed with this $graph operation in one call to UHMEP API.

8.5.1.1 Allowed roles

Prescriber, Caregiver, Patient, Admin and Head

8.5.1.2 Endpoint

GET /ServiceRequest/{referenceId}

POST /ServiceRequest/_search (**for short code**)

8.5.1.3 Implemented rules

The consultation of a specific prescription is possible only if all the following points are respected :

- The referenceId given in the endpoint exists in UHMEP.
- The prescription is in another status than “Blacklisted” or in the same state with some executions.
- (Prescriber) He is not suspended if he is not the prescriber of the prescription.
- (Caregiver) He is not suspended if he does not have any BePerformerTask linked to this prescription.
- (Patient) The prescription is for the patient requesting the prescription.
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

In case of a suspension :

- A prescriber will only be able to see a prescription created by him.
- A caregiver will only be able to see a prescription on which he is assigned or he works.

8.5.1.4 Request

In addition of the authorization and header described above, this request does not need a body.

8.5.1.5 Response

On successful request, the response will contain the requested prescription.

This prescription is represented in FHIR by a Belgian profile of the ServiceRequest resource.

The Belgian profile used depends on the template of the prescription consulted :

Template	Profile
Nursing prescriptions	BeReferralServiceRequest

8.5.2 Consultation of a list of BeReferralServiceRequests

When searching for data through UHMEP API, the user can use filters to narrow down the results to specific subsets of prescription and proposals that match their needs. The filters are applied to the search query in a logical AND manner, meaning that only the combinations of filters that match all the conditions specified will be returned in the search results.

Here are the key principles to keep in mind when using filters:

- **Filters are applied in a logical AND manner**: The search results are filtered by each individual filter, and only the records that match all the filters are included in the final set of results.
- **If no filters are specified, all search results will be returned**.
- **Filters can be combined in complex ways**: The user can combine multiple filters in a single query to create complex search criteria. For example, the user can filter by a specific patient and then further narrow down the results by a specific status.
- **If there is no prescription matching all filters used by the user, UHMEP will send back an empty list of prescriptions.**
- **If an unknown filter is provided, it is ignored**: UHMEP will use only filters that it understands.
- **If an unknown value is provided for a filter that has a FHIR defined list of defined values**: Since filters are used in a logical AND manner, it should return an empty list of results.

8.5.2.1 Allowed roles

Prescriber, Caregiver, Patient, Admin and Head

8.5.2.2 Endpoint

`POST /ServiceRequest /_search`

8.5.2.3 Implemented rules

Each prescription or proposal that respect the requested search criteria will only be returned if it follows visibility rules of section 6.2 and 8.1.7.

8.5.2.4 Request

Here is a description of all filters available:

FHIR Filter	Value Required	Business Description
patient	Patient pseudonym in transit (domain UHMEP)	Filter by patient of the prescription
status	{status}	Filter by prescription status
statusReason	{statusReason}	Filter by additional info for prescription status
_has:Task:focus:status-_profile	{status}$Profile/be-referral-task	Filter by summary of executions status
requester	PractitionerRole/{Prescriber SSIN}-{Prescriber discipline}	Filter by prescriber of the prescription
_has:Task:focus:owner	PractitionerRole/{Caregiver SSIN}-{Caregiver discipline}	Filter by caregiver assigned to the prescription
_has:Task:focus:owner	Organization/{IdType}-{Id}-{OrgType}	Filter by organization assigned to the prescription
code	{code system}|{template code}	Filter on the template of the prescription
intent	{intent}	Filter between prescriptions and proposals
_count	Number	For pagination purpose, see description below
_offset	Number	For pagination purpose, see description below

Be aware that business statuses must be translated to FHIR statuses as referenced in section 6.3.2.1.

Those filters should be provided in the body of the POST request as *x-www-form-urlencoded* key-values.

Sorting

In UHMEP, ServiceRequests are sorted on their authoredOn fields when displaying a list. Upon searching for prescriptions/proposals using filters, the results are logically sorted from the newest on top, to the oldest at the bottom.

Pagination

While applying filters, the result may contain a large number of prescriptions. For performance reasons, UHMEP API uses a pagination.

_count

The _count filter is used to limit the number of search results returned by the API. It specifies the maximum number of results to return in a single page. For example, if the user sets _count=10, the API will return the first 10 matching records and omit the rest.

If this filter is not present, UHMEP will use the default and maximum allowed value that is currently set to 25. If the user requests more than this maximum, he will get the INVALID_COUNT_PARAMETER error.

_offset

The _offset filter is used to specify the starting point for the search results. It indicates the position of the first record to return in the result set. For example, if the user sets _offset=20, the API will start returning results from the 21st record onwards (i.e., skipping the first 20 records).

Together, _count and _offset can be used to implement pagination, allowing the user to navigate through large result sets in a controlled manner. For example:

- Set _count=10 and _offset=0 to retrieve the first 10 records.
- Set _count=10 and _offset=10 to retrieve the next 10 records, starting from the 11th record.
- And so on...

If this filter is not present, UHMEP will use the default value: 0

By using _count and _offset filters, the user can efficiently page through large result sets and retrieve only the data he needs.

Filter Access Control

FHIR Filter	Prescriber	Caregiver	Patient	Head Nurse	Admin
patient	Partially Mandatory	Partially Mandatory	Mandatory	Free	Free
status	Free	Free	Free	Free	Free
statusReason	Free	Free	Free	Free	Free
_has:Task:focus:status-_profile	Free	Free	Free	Free	Free
requester	Partially Mandatory + Partial Token Verification	Partially Mandatory + Partial Token Verification	Free	Free	Free
_has:Task:focus:owner	Partially Mandatory + Partial Token Verification	Partially Mandatory + Partial Token Verification	Free	Free	Free
_has:Task:focus:owner	Free	Free	Free	Mandatory	Mandatory
code	Free	Free	Free	Free	Free
intent	Free	Free	Free	Free	Free
_count	Free	Free	Free	Free	Free
_offset	Free	Free	Free	Free	Free

This table shows the specific rules depending on the role of the user accessing these search filters.

The following access control rules apply to each filter:

- **Free**: Users can choose to use or ignore this filter as needed.
- **Forbidden**: Users cannot use this filter. If they attempt to do so, they will receive an error message (SEARCH_CRITERIA_NOT_ALLOWED).
- **Mandatory**: Users must provide a value for this filter. If they do not, they will receive an error message (REQUIRED_SEARCH_CRITERIA_MISSING).
- **Partially Mandatory**: Users must use at least one of the *Partially Mandatory* filters. If they do not use any, they will receive an error message (REQUIRED_SEARCH_CRITERIA_MISSING).
- **Partially Mandatory + Partial Token Verification**: If the filter *patient* is **not** provided, then one of the Partial Token Verification should match with the token of the user (WRONG_REFERENCE_HEALTHCARE_PROFESSIONAL).
    - It implies that if the *patient* is provided, there should be **no** verification on the correspondence with the token content.

Example:

- If a Prescriber does a search without specifying a *patient,* he must:
    - Provide at least the *requester* or *_has:Task:focus:owner=PractitionerRole/....* filter. If not he will get (REQUIRED_SEARCH_CRITERIA_MISSING).
    - One of the *requesters* OR *_has:Task:focus:owner=PractitionerRole/....* filter should match his token. If not he will get (WRONG_REFERENCE_HEALTHCARE_PROFESSIONAL).

8.5.2.5 Response

On successful request, the response will contain a list of prescriptions corresponding to the filter(s) used.

The list will be sorted on the authoredOn date with the newest prescription/proposal on top and the oldest at the bottom. In addition, the list also contains a pagination to further improve the visibility.

The list is represented in FHIR by a Bundle resource.

Each prescription is represented in FHIR by a Belgian profile of the ServiceRequest resource.

8.5.3 Consultation of BeReferralTask and BePerformerTask

Alongside the direct consultation of FHIR resources using their specific IDs, it is also possible to do a search to receive Tasks linked on a BeReferralServiceRequest ID. Following the **Error! Reference source not found.**, it is possible to consult all BeReferralTask and BePerformerTask linked to a BeReferralServiceRequest but also do a filter to only get the BeReferralTask or only all BePerformerTask.

8.5.3.1 Consultation of all Tasks linked to a BeReferralServiceRequest

8.5.3.1.1 Allowed roles

Prescriber, Caregiver and Patient

8.5.3.1.2 Endpoint

GET /Task?focus={BeReferralServiceRequestID}

8.5.3.1.3 Implemented rules

The consultation of a specific prescription is possible only if all rules described in section 8.5.1 are followed. Since Tasks linked to a BeReferralServiceRequest are part of the referral prescription, all same rules should also be applied.
There is only one extra rule specific for BePerformerTasks. They will not be returned if the ***status*** of the execution task is “rejected”.

8.5.3.1.4 Request

In addition of the authorization and header described above, this request does not need a body.

8.5.3.1.5 Response

A Bundle of type “searchset” will be returned on successful request. Entries will contain each BeReferralTask and BePerformerTask linked to the BeReferralServiceRequest

8.5.3.2 Consultation filter of Tasks linked to a BeReferralServiceRequest

8.5.3.2.1 Allowed roles

Prescriber, Caregiver and Patient

8.5.3.2.2 Endpoint

```
GET /Task?focus={BeReferralServiceRequestID}&_profile={value}
```

Available “values” are described below :

value	Description
Profile/be-referral-task	Consult the BeReferralTask linked to the BeReferralServiceRequest
https://www.ehealth.fgov.be/standards/fhir/referral/StructureDefinition/be-referral-task	Consult the BeReferralTask linked to the BeReferralServiceRequest
Profile/be-performer-task	Consult all BePerformerTask linked to the BeReferralServiceRequest
https://www.ehealth.fgov.be/standards/fhir/referral/StructureDefinition/be-performer-task	Consult the BePerformerTask linked to the BeReferralServiceRequest

8.5.3.2.3 Implemented rules

The consultation of a specific prescription is possible only if all rules described in section 8.5.1 are followed. Since Tasks linked to a BeReferralServiceRequest are part of the referral prescriptions, all same rules should also be applied.
There is only one extra rule specific for BePerformerTasks. They will not be returned if the ***status*** of the execution task is “rejected”.

8.5.3.2.4 Request

In addition of the authorization and header described above, this request does not need a body.

8.5.3.2.5 Response

A Bundle of type “searchset” will be returned on successful request. Depending on the filter used, the Bundle will contain all BePerformerTask or the BeReferralTask linked to a BeReferralServiceRequest.

8.5.4 Consultation of a referral prescription (recommended)

All clinical information is stocked in the BeReferralServiceRequest, but other resources are often needed to have a global information. Every user that has the right to consult a prescription (see section 8.5) will also be able to ask for a Bundle that contains every resource that represents the global prescription. A visualization of resources that constitute a referral prescription is in Figure 4.

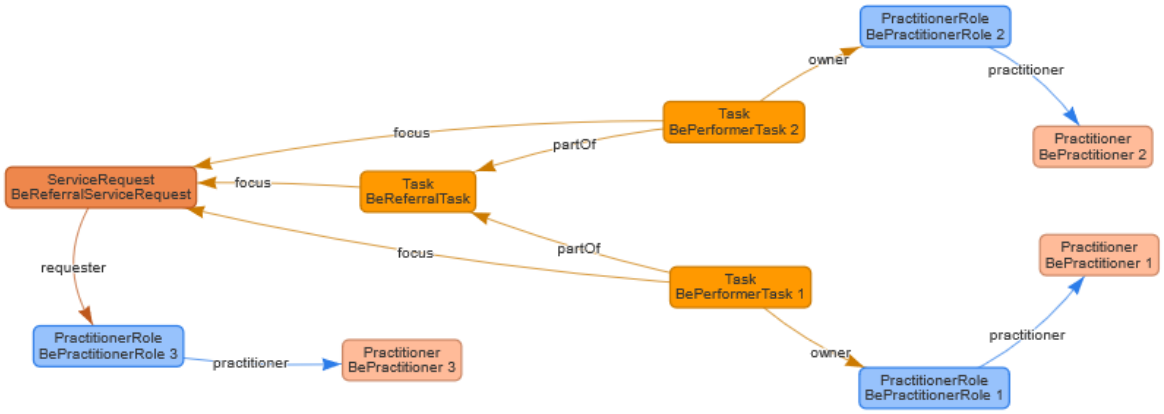

Figure 4 : Example of FHIR resources that compose a referral prescription with two different executions on it

Users will have the possibility to get all these resources, with as many BePerformerTask/BePractitionerRole/BePractitioner as needed, in one call.

8.5.4.1.1 Allowed roles

Prescriber, Caregiver and Patient

8.5.4.1.2 Endpoint

GET /ServiceRequest/{BeReferralServiceRequestID}/$graph?graph={GraphDefinitionURI}

GraphDefinitionURI available	Description
https://www.ehealth.fgov.be/standards/fhir/referral/GraphDefinition/full-referral-prescription	Entries will contain each BeReferralServiceRequest, BeReferralTask, BePerformerTask, BePractitionerRole and BePractitioner linked to the first BeReferralServiceRequest.

8.5.4.1.3 Implemented rules

The consultation of a specific prescription is possible only if all rules described in section 8.5.1 are followed. Since Tasks linked to a BeReferralServiceRequest are part of the referral prescription, all same rules should also be applied.
There is only one extra rule specific for BePerformerTasks. They will not be returned if the ***status*** of the execution task is “rejected”.

8.5.4.1.4 Request

In addition of the authorization and header described above, this request does not need a body.

8.5.4.1.5 Response

A Bundle of type “searchset” will be returned on successful request. Entries will contain each BeReferralServiceRequest, BeReferralTask, BePerformerTask, BePractitionerRole and BePractitioner linked to the first BeReferralServiceRequest.

For example, for the use case presented in Figure 4, the Bundle will contain 10 different entries.

8.6 Annex 81

8.6.1 Create an Annex 81 proposal

This method will allow the end user to create an Annex 81 proposal via UHMEP.

8.6.1.1 Allowed roles

The caregiver.

8.6.1.2 Endpoint

POST /ServiceRequest

8.6.1.3 Implemented rules

The following rules should be checked to create an Annex 81 proposal:

- The user connected is a caregiver
- The discipline of the caregiver has the right to “create a proposal”
- Use of the *intent* “proposal”
- Use of the code “tmp-prep-x081-2”

8.6.1.4 Request

The creation of the BeReferralServiceRequest resource must be done using the intent “proposal”. This is what makes it different from the creation of a prescription (which is a ServiceRequest with an intent “order”).

8.6.1.5 Response

On successful creation of the ServiceRequest resource, the server will respond with a HTTP 201. The ID of the proposal can be found in the location header.

8.6.2 Consult an Annex 81 proposal

This method allows the end user to consult an Annex 81 proposal by providing its RefID.

8.6.2.1 Allowed roles

Prescriber, Caregiver.

8.6.2.2 Endpoint

Same endpoint as for the consultation of a BeReferralServiceRequest (see 8.5.1)

8.6.2.3 Implemented rules

To consult an Annex 81 proposal, the following checks must be performed:

- The connected healthcare professional has the right to “consult a proposal”.
- The connected healthcare professional must not be suspended or he must be the creator or assigned on it.
- The connected patient must be the patient of the proposal that he wants to consult.
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.6.2.4 Request

The consultation of a proposal is possible by referencing its RefID.

8.6.2.5 Response

Upon a successful call, the server will respond with a 201 HTTP response and the user will be able to consult the BeReferralServiceRequest.

8.6.3 Assign an Annex 81 proposal

Allows a caregiver or a patient to assign a prescriber on an Annex 81 proposal for evaluation.

8.6.3.1 Allowed roles

Caregiver, patient.

8.6.3.2 Endpoint

POST /Task

8.6.3.3 Implemented rules

- Assignation possible only if the prescriber has the “evaluate proposal” right.
- The caregiver willing to assign has the “assign proposal” right.
- The patient must be the subject of the Annex 81 proposal as referred in the “focus” field of the performerTask.
- Connected caregiver must not be suspended.
- The prescriber must be able to practice.
- The prescriber must not already be assigned.
- Proposal should be in the “ready” status.
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.6.3.4 Request

Just like for the assignation of a caregiver, the assignation is done through the creation of a BePerformerTask resource (see 8.3.1).

8.6.3.5 Response

On successful creation of the BePerfomerTask resource, the server will respond with a HTTP 201. The ID of this newly created resource is accessible in the *Location* header. This ID can be used for further interactions with this resource.

8.6.4 Consult assignations on an Annex 81 proposal

This method allows a prescriber to consult the task created for the evaluation of an Annex 81 proposal.

8.6.4.1 Allowed roles

Prescriber

8.6.4.2 Endpoint

GET /Task?focus={{proposalID}}

8.6.4.3 Implemented rules

Same as for Consultation of BeReferralTask and BePerformerTask (see 8.5.3).

8.6.4.4 Request

Same as for Consultation of BeReferralTask and BePerformerTask (see 8.5.3).

8.6.4.5 Response

Same as for Consultation of BeReferralTask and BePerformerTask (see 8.5.3).

8.6.5 Remove assignation on an Annex 81 proposal

Assignation on an Annex 81 proposal can be removed by the end users.

8.6.5.1 Allowed roles

Prescriber, Caregiver, Patient.

8.6.5.2 Endpoint

`DELETE /Task/{executionID}`

8.6.5.3 Implemented rules

- (Prescriber/Caregiver/Patient) The proposal referenced in the “focus” field is in the “ready” business status.
- (Prescriber/Caregiver/Patient) The execution of the prescriber (BePerformerTask) is in status “ready”.
- (Prescriber/Caregiver) The prescriber/caregiver is not suspended.
- (Patient) The patient is the same as the one in the proposal.
- (Prescriber) The connected prescriber can only remove his/her own bePerformerTask (owner).
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.6.5.4 Request

This request does not need a body (see 8.3.2)

8.6.5.5 Response

On successful deletion of the resource, the server sends back a HTTP 204. After this operation, the resource is definitively deleted and cannot be used anymore.

8.6.6 Cancel an Annex 81 proposal

The end users are allowed to cancel an Annex 81 proposal when a medication preparation is not needed anymore.

8.6.6.1 Allowed roles

Caregiver, Patient.

8.6.6.2 Endpoint

PATCH /ServiceRequest/{{serviceRequestID}}

8.6.6.3 Implemented rules

- The proposal is in one of the combination of fields status and beReferralTask status:
  - "active" – "ready"
- Connected caregiver is not suspended.
- Connected caregiver is the caregiver of the proposal.
- Connected caregiver must have the right "cancel proposal".
- Connected patient is the patient of the proposal.
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

8.6.6.4 Request

Same as for Cancellation of a prescription (see 8.2.2.4).

8.6.6.5 Response

Same as for Cancellation of a prescription (see 8.2.2.5).

8.6.7 Consult a list of proposals

As per the list of prescriptions, end users can retrieve a list of Annex 81 proposals.

That list only contains ServiceRequests with the intent “proposal”. The list is also sorted and a pagination is present (when applicable).

8.6.7.1 Allowed roles

Prescriber, Caregiver, Patient

8.6.7.2 Endpoint

`POST /ServiceRequest/_search`

8.6.7.3 Implemented rules

Same as for the lists of prescriptions (BeReferralServiceRequest) (see 8.5.2). User should have the right to “consult_proposal”.

8.6.7.4 Request

Same as to retrieve a list of BeReferralServiceRequest with the exception that the body of the POST method may contain the intent “proposal” filter.

8.6.7.5 Response

Same as for 8.5.2.5: the response should contain a list of BeReferralServiceRequest with the intent “proposal”.

The same sorting logic applies for the proposal list consultation: the newest proposals are displayed on top of the page and the oldest at the bottom.

If applicable, a pagination is also put in place to avoid displaying too many results on the same page.

8.6.8 Evaluate (approve) an Annex 81 proposal

Allows an end user to approve the Annex 81 proposal to enable effective patient treatment.

8.6.8.1 Allowed roles

Prescriber.

8.6.8.2 Endpoint

`POST /ServiceRequest/{{serviceRequestID2}}/$approve`

8.6.8.3 Implemented rules

- Annex 81 proposal in status “ready”.
- Connected prescriber has the “evaluate_proposal” right.
- Optional note provided in the operation
- Informed Consent, Therapeutic Relations and Therapeutic Exclusions validations, please refer to section 6.2.5.3.

In addition, the approval of the Annex 81 proposal will automatically trigger:

- Proposal status changes to “completed”.
- Update/creation of the BePerformerTask to “completed”.
- Upon approval, automatic assignation of the caregiver who created the proposal.

8.6.8.4 Request

The approval is done through the “$approve” FHIR operation on the reference ID of the BeReferralServiceRequest with intent “proposal” (= Annex 81 proposal).

In addition, a particular body must be used in the FHIR request:

```
{
    "resourceType": "Parameter",
    "parameter": [{
        "name": "note"
        "valueMarkdown": "my note"
    }]
}
```

For additional details, please refer to this page: https://build.fhir.org/ig/hl7-be/referral/branches/earlyadopter/OperationDefinition-be-op-approve-annex81.html

8.6.8.5 Response

Upon successful approval, a 201 SUCCESS message should be sent. The ID created can then be used in the same way as the ID of a BeReferralServiceRequest resource with intent “order” (prescription).

8.6.9 Reject an Annex 81 proposal

Same as per the approval, end-users are also allowed to reject an Annex 81 proposal if it is deemed unnecessary/not required anymore.

8.6.9.1 Allowed roles

Prescriber.

8.6.9.2 Endpoint

POST /ServiceRequest/{{serviceRequestID2}}/$reject

8.6.9.3 Implemented rules

The same set of rules apply as for the Annex 81 approval (see 8.6.8).

8.6.9.4 Request

Same as for the Annex 81 approval (see 8.6.8).

8.6.9.5 Response

SUCCESS 200 HTTP response expected upon successful Annex 81 proposal rejection.

8.7 Organization assignations

8.7.1 Assignation to an organization

Once a prescription has been created, the system allows users to assign a specific organization on it. Giving them an ease of access to this prescription and some access rights.

8.7.1.1 Allowed roles

Head, Admin, Patient, Caregiver

8.7.1.2 Endpoint

This API request is the same as of an assignation of a caregiver. Please look on section 8.3.1.

8.7.1.3 Request

This operation is using a FHIR BeOrganizationTask resource, with the ***owner*** field as a reference to an organization, as described in section 6.3.5.

8.7.2 Remove assignation of an organization

This feature works as a DELETE on the BeOrganizationTask ID, it is the same feature as the “Remove a caregiver” in section 8.3.2.

8.7.3 Assignation of a caregiver working for an organization

A user being connected as working for an organization can assign a caregiver and specify that he/she is linked to the organization assignation.

8.7.3.1 Allowed roles

Head, Admin, Healthcare professional with organization information in their token

8.7.3.2 Endpoint

This API request is the same as of an assignation of a caregiver. Please look on section 8.3.1.

8.7.3.3 Request

The only field that changes from a simple assignation of a caregiver is the field ***focus*** that should be a reference to the existing BeOrganizationTask created in section 8.7.1. See Figure 2 for a visual representation of links between resources.

9 Error management

In the cycle of prescription, UHMEP API allows different type of requests and multiple different actors can interact with the API. If there is an error in the request, or if the action is forbidden by the system, UHMEP sends back an error. This section documents how errors will be represented by UHMEP. Alongside the correct HTTP code, UHMEP uses the FHIR resource OperationOutcome which is defined by FHIR to display the error.

An error can appear for different reasons. The errors returned by UHMEP are structured in a specific way to allow the end-user to quickly understand where the errors come from.

UHMEP uses issues with fields :

- ***severity*** : defines if this is an “error” or a “warning”. (https://hl7.org/fhir/r4b/valueset-issue-severity.html)
- ***code*** : FHIR error type. UHMEP uses (business-rule, value, security, forbidden, conflict, processing). (https://hl7.org/fhir/r4b/valueset-issue-type.html)
- ***details.coding.system*** : The system where the error is defined. Errors thrown and defined by UHMEP are in the system “urn:uhmep:errors”
- ***details.coding.code*** : The code that defines the error. In the document, it is referenced as the **UHMEP code**
- ***diagnostics*** : Contains the description of the UHMEP code to help understanding. It should always be the same message for one UHMEP code. This message ends with an error id that can used for support purpose since this error is logged and more information can be retrieved by the UHMEP team through the error tracking system of UHMEP.

All error messages returned by UHMEP are listed in the Excel document “UHMEP_API_ErrorCodes.xlsx” alongside this cookbook.