# TABLE OF CONTENTS

Table of contents ........................................................................................................... 2
External references ........................................................................................................... 4

1. Introduction to Recip-e ............................................................................................. 5
   1.1. Recip-e solution ................................................................................................. 5
   1.2. Description and purpose of the integration specifications ............................. 6
   1.3. Document scope ............................................................................................... 6

2. Responsibilities ......................................................................................................... 7
   2.1. Encoding ........................................................................................................... 7
   2.2. Identification of the patient ............................................................................ 7

3. Specification of the required SAML-token for hospitals (delivered by the Secure Token Service (STS) of eHealth-platform) .............................................. 15

4. Integration via the Recip-e webservices (as prescriber) ........................................... 16
   3.1. Overview .......................................................................................................... 16
       3.2. Implementation details .................................................................................. 18
           3.2.1. Authentication of the hospital ................................................................. 18
           3.2.2. Messaging and encryption ..................................................................... 19
               3.2.2.1. Overview of message creation ......................................................... 19
               3.2.2.2. Encryption for transport ................................................................. 20
           3.2.2.3. Non-addressed message encryption for storage (only for prescription) 21
           3.2.2.4. Addressed encryption for storage (only for feedbacks and notifications) 22
           3.2.2.5. Focus on compression ...................................................................... 22
           3.2.2.6. Focus on NA encryption .................................................................. 22
           3.2.2.7. Focus on end to end encryption ....................................................... 23
           3.2.3. Prescriber web services ........................................................................... 24
3.2.4. Error management ................................................................. 24
3.2.5. Use of eHealth-platform services ........................................... 25
3.3. Service inventory .................................................................. 25
  3.3.1. Administrative information .................................................. 25
  3.3.2. Party identification .............................................................. 25
  3.3.3. Prescriber service operations ............................................... 26
    3.3.3.1. Create prescription ......................................................... 26
    3.3.3.2. Send Notification .......................................................... 28
    3.3.3.3. Revoke Prescription ....................................................... 29
    3.3.3.4. Get prescription ............................................................ 31
    3.3.3.5. List Feedback ............................................................... 32
    3.3.3.6. List Open Prescription ................................................... 34
    3.3.3.7. Update Feedback Flag ................................................... 35
4. Appendix: Frequently Asked Questions ........................................ 38
## EXTERNAL REFERENCES

<table>
<thead>
<tr>
<th>Ref ID</th>
<th>Name</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.</td>
<td>Barcode print guideline</td>
<td>Specified in ISO/IEC 15416</td>
</tr>
</tbody>
</table>
1. INTRODUCTION TO RECIP-E

1.1. RECIP-E SOLUTION

The Recip-e solution concerns the generic (i.e. valid for all types of prescription: pharmaceutical, physiotherapist, nursing, ...) transfer of prescriptions from the prescriber to the care provider, for example from the general practitioner (GP) to the pharmacist or from a general hospital to the pharmacist, chosen freely by the patient or from the specialist to the physiotherapist.

With the Recip-e solution, data can be sent between various actors with a high level of security. This technological innovation also offers improvements for everyone involved in the project. Below is a list of the added value that the Recip-e solution offers:

- Ensure roles and responsibilities of everyone;
- Integration with medical platforms for the identification of the actors, the security of the data and the control of the insurability of patients to ensure (e.g. eHealth, MyCareNet)
- Improving the administrative process and reduce administrative burdens
- Reduce erroneous prescriptions (errors in prescriptions)
- Relationship between the electronic and the paper stream is guaranteed
- Traceability of the data between the different actors.
- Traceability of the data access (consult) for privacy logging

There are also immeasurable positive impacts foreseen thanks to the solution:
- Enhancing of the process (less fraud by avoiding patients to create fake prescriptions);
- ...

Technically speaking Recip-e is not only a system that manages non-addressed messages in the health sector. Recip-e is also a system that provides advanced functionality such as prescription state, prescription validation, unique document numbering...
1.2. DESCRIPTION AND PURPOSE OF THE INTEGRATION SPECIFICATIONS

This document establishes a set of requirements for the interface between Prescription Software and the RECIP-E system. It identifies agreed-upon design requirements and constraints that must be satisfied by the interfacing software. This document is intended for use by the developers of the applications identified, and by the test organizations responsible for the testing of these applications.

1.3. DOCUMENT SCOPE

This document outlines the interface requirements to support the following business events for Prescription Software (doctors, dentists, hospitals, ...)

- Create Prescription
- Cancel a prescription
- Read a feedback for a prescription
- Send a prescription notification
- List/Read Prescription

The document also details the requirements to support the following technical events:

- Encryption
- Authentication

Changes are required in Prescription software to integrate with RECIP-E. This document will detail the integration procedure expected to be implemented by the Software Providers of said software.
2. RESPONSIBILITIES

The software provider / implementer of the Recipe prescribing services will have to make sure that his application fulfills a set of requirements. These are documented as part of this chapter.

2.1. ENCODING

All Recip-e messages (prescriptions, feedback and notifications) should use the UTF-8 encoding.

2.2. IDENTIFICATION OF THE PATIENT

The Patient must be identified by his INSS number (also named NISZ, NISS, and SSIN). Therefore, the healthcare software has the responsibility to provide a verified/validated INSS number.

2.3. PRESCRIPTION FORMAT

A prescription is defined as being a specific XML KMEHR message (Kind Message for Electronic Healthcare Record) of type pharmaceutical prescription. This format is further described on eHealth website: https://www.ehealth.fgov.be/standards/kmehr/.

On the one hand, the prescription software has the responsibility to generate a valid KMEHR prescription. On the other hand, executor software must be able to load such valid prescription.

The validation of the prescription consists in a two-step validation process:

- XSD validation
- Additional validation

The validation must be performed, by the integrating software, before a prescription is submitted.

The XML schema defining the KMEHR standard (XSD file) is provided in the Recip-e-client package and can be found on eHealth website at this URL: https://www.ehealth.fgov.be/standards/kmehr/.

At this moment, the prescription must be compliant with the version “20100601-kmehr” of the XSD definition (XSD packaged in a zip can be downloaded at this URL: https://www.ehealth.fgov.be/standards/kmehr/sites/default/files/assets/kmehr/20100601/20100601-kmehr.zip).

2.3.1. ADDITIONAL VALIDATION

The kmehr standard defines many different type of message regarding the healthcare sector. In the first stage of Recip-e (pilot), only pharmaceutical prescription is accepted. However, in the next phases, the system will accept other kind of prescription (new version of this document will be available).

For pharmaceutical prescription, additional verifications must be performed before uploading the prescription in the Recip-e system.

The table below describes the different checks to be performed. For each check, a XML XPATH is defined. The result of the XPATH query must return the result defined in column "Expected Number of record".

<table>
<thead>
<tr>
<th>XPATH</th>
<th>Expected number of records</th>
</tr>
</thead>
<tbody>
<tr>
<td>/kmehrmessage/header/sender/hcparty/id[@S'ID-HCPARTY' and @SV'1.0']</td>
<td>1</td>
</tr>
<tr>
<td>/kmehrmessage/header/recipient/hcparty/id[@S'ID-HCPARTY' and @SV'1.0']</td>
<td>1</td>
</tr>
<tr>
<td>/kmehrmessage/folder/patient/id[@S'ID-PATIENT' and @SV'1.0']</td>
<td>1</td>
</tr>
</tbody>
</table>
2.3.2. SAMPLE PRESCRIPTION

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<kmehrmessage xmlns="http://www.ehealth.fgov.be/standards/kmehr/schema/v1">
  <header>
    <standard>
      <cd S="CD-STANDARD" SV="1.1">20100601</cd>
    </standard>
    <id S="ID-KMEHR" SV="1.0">14675011004.20090110090000000</id>
    <id S="LOCAL" SL="ID-RECIPE" SV="1.0">8e1c4ea4-3825-48e4-bcc2b8cadfa7a897</id>
    <date>2010-08-01</date>
    <time>09:00:00</time>
    <sender>
      <hcparty>
        <id S="ID-HCPARTY" SV="1.0">14675011004</id>
        <cd S="CD-HCPARTY" SV="1.0">persphysician</cd>
        <name>Dr. Duck Donald</name>
      </hcparty>
    </sender>
    <recipient>
      <hcparty>
        <id S="ID-HCPARTY" SV="1.0">RECIPE</id>
        <cd S="CD-HCPARTY" SV="1.0">orgpublichealth</cd>
        <name>Recip-e</name>
      </hcparty>
    </recipient>
  </header>
  <folder>
    <id S="ID-KMEHR" SV="1.0">1</id>
    <patient>
      <id S="ID-PATIENT" SV="1.0">87990949113</id>
      <firstname>Fred</firstname>
      <familyname>Flintstone</familyname>
      <birthdate>
        <date>1933-10-23</date>
      </birthdate>
      <sex>
        <cd S="CD-SEX" SV="1.0">male</cd>
      </sex>
    </patient>
    <transaction>
      <id S="ID-KMEHR" SV="1.0">1</id>
      <cd S="CD-TRANSACTION" SV="1.1">pharmaceuticalprescription</cd>
      <date>2010-08-01</date>
      <time>09:00:00</time>
      <author>
        <hcparty>
          <id S="ID-HCPARTY" SV="1.0">14675011004</id>
          <cd S="CD-HCPARTY" SV="1.0">persphysician</cd>
          <name>Dr. Duck Donald</name>
        </hcparty>
      </author>
      <iscomplete>true</iscomplete>
    </transaction>
  </folder>
</kmehrmessage>
```

For more information about XPATH queries, please refer to [http://www.w3.org/TR/xpath/](http://www.w3.org/TR/xpath/).
<isvalidated>true</isvalidated>
<expirationdate>2013-08-01</expirationdate>

<heading>
  <id S="ID-KMEHR" SV="1.0">1</id>
  <cd S="CD-HEADING" SV="1.1">prescription</cd>
</heading>

<item>
  <id S="ID-KMEHR" SV="1.0">1</id>
  <cd S="CD-ITEM" SV="1.1">medication</cd>
  <content>
    <medicinalproduct>
      <intendedcd S="CD-DRUG-CNK" SV="2.0">0318717</intendedcd>
      <intendedname>Adalat Oros 30 (c) 30mg</intendedname>
    </medicinalproduct>
  </content>
  <lifecycle>
    <cd S="CD-LIFECYCLE" SV="1.0">prescribed</cd>
  </lifecycle>
  <quantity>
    <decimal>1</decimal>
  </quantity>
  <frequency>
    <periodicity>
      <cd S="CD-PERIODICITY" SV="1.0">D</cd>
    </periodicity>
  </frequency>
  <dayperiod>
    <cd S="CD-DAYPERIOD" SV="1.0">evening</cd>
  </dayperiod>
  <posology>
    <text L="nl">1x</text>
  </posology>
</item>

<item>
  <id S="ID-KMEHR" SV="1.0">2</id>
  <cd S="CD-ITEM" SV="1.1">medication</cd>
  <content>
    <medicinalproduct>
      <intendedcd S="CD-DRUG-CNK" SV="2.0">1085885</intendedcd>
      <intendedname>Actrapid HM Penfill (c) 100IU/ml</intendedname>
    </medicinalproduct>
  </content>
  <lifecycle>
    <cd S="CD-LIFECYCLE" SV="1.0">prescribed</cd>
  </lifecycle>
  <quantity>
    <decimal>1</decimal>
  </quantity>
  <frequency>
    <periodicity>
      <cd S="CD-PERIODICITY" SV="1.0">D</cd>
    </periodicity>
  </frequency>
  <dayperiod>
    <cd S="CD-DAYPERIOD" SV="1.0">evening</cd>
  </dayperiod>
  <posology>
    <text L="nl">2x/d 12E voor de maaltijd SC</text>
  </posology>
</item>

<item>
  <id S="ID-KMEHR" SV="1.0">3</id>
  <cd S="CD-ITEM" SV="1.1">medication</cd>
  <content>
    <medicinalproduct>
      <intendedcd S="CD-DRUG-CNK" SV="2.0">1077718</intendedcd>
      <intendedname>Insulatard HM Penfill (c) 100IU/ml</intendedname>
    </medicinalproduct>
  </content>
  <lifecycle>
    <cd S="CD-LIFECYCLE" SV="1.0">prescribed</cd>
  </lifecycle>
  <quantity>
    <decimal>1</decimal>
  </quantity>
  <frequency>
    <periodicity>
      <cd S="CD-PERIODICITY" SV="1.0">D</cd>
    </periodicity>
  </frequency>
  <dayperiod>
    <cd S="CD-DAYPERIOD" SV="1.0">evening</cd>
  </dayperiod>
  <posology>
    <text L="nl">1x/d 7E voor het slapen</text>
  </posology>
</item>

<item>
  <id S="ID-KMEHR" SV="1.0">4</id>
  <cd S="CD-ITEM" SV="1.1">medication</cd>
  <content>
    <medicinalproduct>
      <intendedcd S="CD-DRUG-CNK" SV="2.0">1057959</intendedcd>
    </medicinalproduct>
  </content>
</item>
Example of a “product” medication item:

```xml
<item>
  <id SV="1.0" S="ID-KMEHR">1</id>
  <cd SV="1.2" S="CD-ITEM">medication</cd>
  <content>
    <medicinalproduct>
      <intendedname>Spironolactone E.G. (c) 100mg</intendedname>
      <lifecycle>
        <cd S="CD-LIFECYCLE" SV="1.0">prescribed</cd>
      </lifecycle>
      <quantity>
        <decimal>1</decimal>
      </quantity>
      <frequency>
        <periodicity>
          <cd S="CD-PERIODICITY" SV="1.0">D</cd>
        </periodicity>
      </frequency>
      <dayperiod>
        <cd S="CD-DAYPERIOD" SV="1.0">morning</cd>
      </dayperiod>
    </medicinalproduct>
    <posology>
      <text L="nl">1x/d</text>
    </posology>
  </content>
</item>
```

Example of a “substance” medication item:

```xml
<item>
  <id SV="1.0" S="ID-KMEHR">1</id>
  <cd SV="1.2" S="CD-ITEM">medication</cd>
  <content>
    <substanceproduct>
      <intendedname>paracetamol 1 g</intendedname>
    </substanceproduct>
  </content>
</item>
```
“Magistral prescription”: non-standardized (textual description):

```xml
<item>
  <id SV="1.0" S="ID-KMEHR">1</id>
  <cd SV="1.2" S="CD-ITEM">medication</cd>
  <content>
    <compoundprescription L="FR">Prescription magistrale</compoundprescription>
    ...
  </content>
</item>
```

2.3.3. CHECKING/SETTING THE AUTHOR OF THE PRESCRIPTION

The author of the prescription must correspond to the medical professional (doctor) prescribing the medication.

2.3.4. PRESCRIPTION TYPE

When the prescription software is creating a prescription, the attribute “prescription type” must be correctly defined.

During the pilot phase, only pharmaceutical prescriptions are in scope. Currently we have defined following 4 types of pharmaceutical prescriptions

- P0: Pharmaceutical prescription
- P1: Pharmaceutical prescription that necessitates information on the patient’s insurability
- P2: Pharmaceutical prescription that necessitates attestation information

The software providers are encouraged to already implement the distinction between P0, P1, and P2. This distinction can be made by looking at the BCFI database and the prescribed medication.

In general is expected from software provider to implement a solution open to any future updates of this prescription type.

Note: this prescription type is important because it is used by the Recip-e system to define access rights for executors. The executor will only be able to read (decrypt) messages of types to which he has access. (e.g. a pharmacist has access to pharmaceutical prescriptions).

2.4. NOTIFICATION FORMAT

When a prescriber creates a complex prescription, he has the possibility to send a notification message to a specific executor containing indication regarding the content of the prescription to be prepared or ordered (however, a notification message does not guarantee that the patient will come to this dedicated pharmacy).

The notification can also contain a copy of the prescription itself. The Notification message before encryption is a XML message that must be compliant with following description.

2.4.1. XSD VALIDATION

The File Notification.xsd describes the format of a notification.

Content of the file:

```xml
</xsd:schema>
```
2.4.2. SAMPLE FILE

```xml
<?xml version="1.0" encoding="UTF-8"?>
<p:notification xmlns:p="http://services.recipe.be"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://services.recipe.be notification.xsd">
    <text>this is a notification</text>
    <kmehrmessage>[the Kmehr prescription]</kmehrmessage>
</p:notification>
```

2.5. FEEDBACK FORMAT

When requested by the prescriber, a Feedback may be sent back by the prescription executor (pharmacist) once prescription is delivered. The Feedback before encryption is a XML message that must be compliant with following description.

2.5.1. XSD VALIDATION

The File Feedback.xsd describes the format of a feedback.

Content of the file:

```xml
<x:schema xmlns:x="http://www.w3.org/2001/XMLSchema"
    xmlns="http://services.recipe.be"
    targetNamespace="http://services.recipe.be">
    <x:element name="feedback">
        <x:complexType>
            <x:sequence>
                <x:element name="text" type="xs:string" maxOccurs="1" minOccurs="0"/>
            </x:sequence>
        </x:complexType>
    </x:element>
</x:schema>
```

2.5.2. SAMPLE FILE

```xml
<?xml version="1.0" encoding="UTF-8"?>
<p:feedback xmlns:p="http://services.recipe.be"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://services.recipe.be feedback.xsd">
    <text>this is a feedback</text>
</p:feedback>
```

2.6. BARCODE SPECIFICATION

On top of the prescription, a new barcode will be found that contains the Recip-e ID of the prescription. This barcode needs to be compatible with the current barcode readers found in pharmacies.
2.6.1. FORMAT RECIP-ID

The Recip-ID (or RID) has format BEPPXXXXXXXXX where

- BE is a 2-character alphanumerical id that stands for the country (BE = Belgium)
- PP is a 2-character alphanumerical id that stands for the type of prescription (allowed values are P0, P1 or P2). Refer to section “2.3.4 Prescription type” for more information on available prescription types.
- XXXXXXXX an 8-character alphanumerical sequence ID

The alphanumerical characters can be numbers or uppercase letters:

- Possible characters are 0123456789ABCDEFGHKLMNPRSTVWXYZ
- Due to possible ambiguity letters O, Q, I, J, U and V are not used

The Recip-ID will be provided by eHealth during the creation of a prescription.

2.6.2. FORMAT BARCODE

The 128A format is used for the barcode. The barcode symbology is specified in ISO/IEC 15417:2007

Example:

![Barcode Example]

2.7. PRINT PRESCRIPTION (FOR PRESCRIBERS)

2.7.1. BARCODE

The Prescription software must print the prescription ID as a barcode on the paper prescription. The prescription layout is defined on the INAMI/RIZIV website:


This barcode must be printed in the upper right part of the prescription.

The print quality of the barcode must be compliant with the ISO15416 specification (Bar Code Print Quality Guideline: this specification describes requirement in term of quiet zones, reflectance, contrast, ...).
Naam en voornaam van de voorschrijver
Dr. XYZ

DOOR DE VOORSCHRIJVER IN TE VULLEN:
Naam en voornaam van de rechthebbende: PATIENT X

Voorbehouden aan het verpakkingsvignet
R/Lyrica caps 14x 150mg

SPECIMEN

Stempel van de voorschrijver
Dr. XYZ
1-23456-78-999

Datum en handtekening van de voorschrijver
27/01/2016

Uitvoerbaar vanaf voornoemde datum of vanaf:

GENEESMIDDELENVOORSCHRIFT
2.7.2. MEDICATIONS

The list of medications must be printed on the prescription (middle right area). This “printable” area is limited in term of lines and characters per line.

The limit is currently set to 10 lines with each 80 characters. Each medication must be printed on a dedicated line. If a medication is printed on two lines, the prescription software must make sure that the overall prescription does not exceed 10 lines. If this is the case, a new prescription with a new RID must be created.

2.8. AUTHENTICATION AND AUTHORIZATION

The Recip-e services are exposed via eHealth-platform and the general principles and standards used for accessing eHealth-platform services are followed. This implies that the secure Recip-e services require a valid SAML token provided by the Secure Token Service (STS) of eHealth-platform.

In this document, what is named the “session” is in practice a SAML token generated by eHealth-platform STS. Once obtained by the care provider, this token allows calling each one of the Recip-e services (and other eHealth services). The validity of this token is limited in time, after expiration the SAML-token will not be accepted and a new token should be requested via STS.

More information on how to obtain a SAML-token via the Secure Token Service (STS) of eHealth-platform can be found on the eHealth-platform website:

- NL: https://www.ehealth.fgov.be/nl/support/sts-secure-token-service
- FR: https://www.ehealth.fgov.be/fr/support/sts-secure-token-service

2.8.1. SPECIFICATIONS OF THE REQUIRED SAML-TOKEN FOR HOSPITALS (DELIVERED BY THE SECURE TOKEN SERVICE (STS) OF EHEALTH-PLATFORM)

The SAML-token request is secured with the eHealth-platform certificate of the hospital. The certificate used by the Holder-Of-Key verification mechanism is the same eHealth certificate.

The required attributes are the following (Attribute namespace: "urn:be:fgov:identification-namespace"):

- The NIHII number of the hospital:
  - urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-number
  - urn:be:fgov:ehealth:1.0:hospital:nihii-number

Hospital must also specify which information must be asserted by eHealth-platform (designators):

- The NIHII number as identifier of the hospital (Attribute namespace: urn:be:fgov:identificationnspace):
  - urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-number
  - urn:be:fgov:ehealth:1.0:hospital:nihii-number

- To have access to the eHealth consent web service, the hospital must be a recognized hospital (Attribute Namespace: urn:be:fgov:certified-namespace:ehealth):
  - urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-number:recognisedhospital:Boolean
  - urn:be:fgov:ehealth:1.0:hospital:nihii-number:recognisedhospital:Boolean

Please note that the certificate used for signing the prescription (encryption for storage) must be identical to the certificate used as Holder of Key certificate in the SAML-token.
3. INTEGRATION VIA THE RECIP-E WEBSERVICES (AS PRESCRIBER)

3.1. OVERVIEW

The section below describes the technical scope of the development work, concerning functionalities to foresee by the Software Provider in the software of the Software Provider to integrate with the Recip-e system and eHealth-platform for prescribers.

The figure below shows all the actions to be performed by the Software Provider’s software and the way it is to be integrated with the Recip-e system.
The red rectangle represents the software of the Software Provider for Prescribers. The green/blue rectangle represents the Recip-e central system and eHealth-platform common services.

Two types of functionalities are considered:
- Communication functionality to be implemented by the Software Provider, based on this document: arrows between the green/blue and red rectangles
- Internal functionality within the Software Provider software: circle-arrows within the rectangle

Communication functionality

1. **Create Session (STS):** Send a SAML request to eHealth and receive a SAML token.
2. **Encryption Key Retrieval for addressed encryption (ETK Depot):** retrieve encryption keys in ETK depot of messages recipient.
3. **Encryption Key Generation (KGSS):** request encryption keys.
4. **Create Prescription:** Send a prescription to the Recip-e Integration module. The software receives in return the RID (Recip-e ID). The prescriber can decide if want to get a feedback from the executor.
5. **Send Notification:** The prescriber has the option to send a prescription directly to an executor. This could for instance contain a prescription to prepare or a personal message
6. **List Feedback:** Ask for feedback sent by executor. In return the software receives all feedbacks addressed to him.
7. **Revoke Prescription:** If the prescriber chooses so, he can revoke/cancel a prescription. The cancellation of a prescription by the prescriber should be communicated to the Recip-e system. The cancellation request is sent to the module with the Recip-e unique identification number (RID). The prescriber receives a confirmation that the prescription has been cancelled.
8. **List Open Prescriptions:** The Software Provider software requests an update from the Recip-e central system concerning the status of its prescriptions. It gets returned a list of prescription with their status.
9. **Get Prescription:** The Software Provider can retrieve prescription previously created.
10. **Update Feedback Flag:** the software provider change the feedback flag set previously (Action 1)

Internal functionality

A. Identify the patient through
   - the NISS/NISZ number found on the patients ID card
   - information on the patients SIS card
   - information on the patients mutuality sticker
   - directly into the system the executor (if the patient is already in the system)

B. Encrypt messages using addressed encryption framework provided by eHealth

C. Encrypt/decrypt prescription using non-addressed encryption framework provided by eHealth

D. Use MyCareNet services such as patient insurability (this is currently not in scope of the integration specifications, but is included here for clarity sake)

E. Create a prescription in the Software Provider software and attach it to the patient record in the Software Provider software

F. Add the unique identification number (Recip-e ID) of the electronic prescription (received from Recip-e) to the prescription, and print the prescription with RID on it in barcode format

G. The prescriber has the option to send a message directly to an executor. This could for instance contain a prescription to prepare or a personal message. The Service Supplier software needs to allow the prescriber to input such a message

H. Insert the received feedback into the patient record in the Software Provider software
I. Cancel the prescription from the patient record in the Software Provider software (only when requested by the prescriber)

J. For performance reasons, the prescriptions and their status (in process, delivered ...) need to be stored in the Service Supplier software. The prescription status (and the content of the prescription) can be updated in batch mode (e.g. once a day).

Remark: hospitals are encouraged to use the eHealth-platform timestamping service to officially timestamp the (electronic) prescriptions.

3.2. IMPLEMENTATION DETAILS

3.2.1. AUTHENTICATION OF THE HOSPITAL

Authentication of the care provider (hospital) is performed by eHealth-platform thanks to the Secure Token Service (STS). This service takes as an input the hospitals eHealth-platform certificate to generate a SAML token that can be used for Single Sign On.

Please refer to section “2.8.1 Specifications of the required SAML-token for hospitals (delivered by the Secure Token Service (STS) of eHealth-platform)” of this document for more information about the required contents of the SAML token.

Once the SAML assertion is retrieved from STS, it must be added to the SOAP header of each outgoing message addressed to Recip-e. Refer to the document http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0.pdf for more information about SAML and WS-Security.
3.2.2. MESSAGING AND ENCRYPTION

3.2.2.1. OVERVIEW OF MESSAGE CREATION

Health Care Software must communicate with Recip-e using message based communications. These messages must be highly secured with encryption. Three different types of encryption must be implemented:

- **Encryption for transport**: This kind of encryption concerns all type of messages. This encryption is based on the ETEE encryption framework provided by eHealth. It consists in encrypting the message so that only Recip-e can read it. This is based on Public key/Private key encryption.

- **Encryption for storage**: This encryption type is used to secure the storage of the message in Recip-e. Only allowed systems/recipient can decrypt these messages (Recip-e can’t decrypt them). There is two type of storage encryption:
  - **Non-addressed Encryption for Prescriptions**: Prescriptions are encrypted using the non-addressed encryption framework.
  - **Addressed Encryption for Feedbacks/Notifications**: these messages are encrypted, only the recipient can encrypt the message.

Please note that the certificate used for signing the prescription (encryption for storage) must be identical to the certificate used as Holder of Key certificate in the SAML-token.

The following section describes the different types of encryption needed to create messages in Recip-e. The steps to decrypt messages are identical (except the reverse order).

![Diagram showing inbound and outbound message encryption and decryption processes.](image)

**Inbound**

**HEADER**

- SAML ASSERTION HOK
  - SIGNED BY EH
    - DOCTOR_SSN
  - DOCTOR_NHII1
  - HOLDER_OF_KEY_CERT

- TIMESTAMP (MESSAGE- TTL)
- SIGN BODY, ASS, TIMESTAMP
  - [proof-of-possession HOK]

**BODY**

- ADMINISTRATIVE INFORMATION
  - KEY_ID
  - PRESCRIPTION_TYPE
  - PATIENT_SSN, DOCTOR_NHII

  - FEEDBACK_FLAG

  - KMEHR PRESCRIPTION
    - ETEE: SIGN
    - ETEE: SYM. ENCRYPT
    - ETEE: SIGN

  - ENCRYPT RECEIPE

**Outbound**

**HEADER**

- SAML ASSERTION SV
  - SIGNED BY EH
    - DOCTOR_SSN
  - DOCTOR_NHII1

- TIMESTAMP (MESSAGE- TTL)
- SIGN BODY, ASS, TIMESTAMP

**BODY**

- ADMINISTRATIVE INFORMATION
  - KEY_ID
  - PRESCRIPTION_TYPE
  - PATIENT_SSN, DOCTOR_NHII
  - DOCUMENT_ID

  - FEEDBACK_FLAG

  - KMEHR PRESCRIPTION
    - ETEE: SIGN
    - ETEE: SYM. ENCRYPT
    - ETEE: SIGN

  - ENCRYPT RECEIPE
3.2.2.2. ENCRYPTION FOR TRANSPORT

Following diagram illustrates the different steps of the transformation process to be implemented:

1. Message Content is encrypted using the public Key of Recipe (ETK: encryption token), this ETK is retrieved from eHealth service ETK depot.

2. Message is sent to Recip-e including a random SymmKey that only the sender knows. This SymmKey will be used by Recip-e to encrypt the response. This SymmKey should only be provided when a response is expected (optional for “void” API).
3.2.2.3. NON-ADDRESSED MESSAGE ENCRYPTION FOR STORAGE (ONLY FOR PRESCRIPTION)

1. Private Message Content (KMEHR prescription) is compressed using GZIP algorithm

2. Private Message is sealed using non-addressed encryption, the symmetric encryption key is retrieved from eHealth service KGSS.get[New]Key(). (The message will be stored as-is by Recip-e)

3. The message is enriched with non-confidential data (patient ID, Prescriber ID), “Encryption for Transport” described in previous section is then applied to secure the data transfer between the care provider workstation and the Recip-e central server. The overall is then included in a SOAP message secured by SAML authentication.
3.2.2.4. **ADDRESSED ENCRYPTION FOR STORAGE (ONLY FOR FEEDBACKS AND NOTIFICATIONS)**

1. Private Message Content (xml) is compressed using GZIP algorithm (more details in section).

2. Private Message is sealed using addressed encryption, the public encryption key of the recipient is retrieved from eHealth service ETKdepot.getETK(). (The message will be stored as-is by Recip-e)

3. The message is enriched with non-confidential data (patient ID, Prescriber ID), “Encryption for Transport” described in previous section is then applied to secure the data transfer between the care provider workstation and the Recip-e central server. The overall is then included in a SOAP message secured by SAML authentication.

3.2.2.5. **FOCUS ON COMPRESSION**

To decrease bandwidth consumption, xml messages are compressed using GZIP standard. (Only XML KMEHR prescription, XML notification & XML feedbacks are concerned).

Following Java code illustrates how the message must be compress:

```java
import java.util.zip.GZIPOutputStream;

public static byte[] compress(byte[] input) throws Exception {
    ByteArrayOutputStream outstream = new ByteArrayOutputStream();
    GZIPOutputStream out = new GZIPOutputStream(outstream);
    out.write(input);
    return outstream.toByteArray();
}
```

This compression is specified by standards RFC 1950; RFC 1951 and RFC 1952 (Refer to http://www.ietf.org/ for more information about these specifications).

3.2.2.6. **FOCUS ON NA ENCRYPTION**

See Ref 2 for generic information about NA Encryption.
As defined in previous mentioned document, the Process for NA Encryption is defined as follow:

1. The ETK (public Key) of KGSS system is retrieved from ETK depot (Key Id “CBE=0809394427”, Application ID : “KGSS”)
2. The New Key Request is created defining the Access Control List
3. The New Key Request is sealed using addressed encryption (public key of KGSS used)
4. The KGSS.GetNewKey() service is invoked

In the Recip-e specific case, the Access Control List (AllowedReader attribute of the GetNewKeyRequestContent message) must be defined as an Encrypted XML document. Only those three types of parties must be defined in the Access Control List:

- The Prescriber (prescribing doctor and/or hospital)
- The Patient
- All the Pharmacists (for “pharmaceutical” prescriptions).

(This list is used later on to allow a physical person reading the content of a prescription)

Sample Access Control List before encryption (to be replaced with correct values based on the actual prescription):

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:getNewKeyRequestContent xmlns:ns2="urn:be:fgov:ehealth:etee:kgss:1_0:protocol">
  <allowedReader>
    <name>urn:be:fgov:ehealth:doctor-nihii</name>
    <namespace>urn:be:fgov:ehealth:certified-namespace</namespace>
    <value>12659389004</value>
  </allowedReader>
  <allowedReader>
    <name>urn:be:fgov:ehealth:hospital-nihii</name>
    <namespace>urn:be:fgov:ehealth:certified-namespace</namespace>
    <value>12345678</value>
  </allowedReader>
  <allowedReader>
    <name>urn:be:fgov:ehealth:pharmacy-nihii</name>
    <namespace>urn:be:fgov:ehealth:certified-namespace</namespace>
    <value></value>
  </allowedReader>
  <allowedReader>
    <name>urn:be:fgov:identification-namespace</name>
    <namespace>urn:be:fgov:ehealth:certified-namespace</namespace>
    <value>84071230581</value>
  </allowedReader>
  <symmKey>[base64 symmKey]</symmKey>
</ns2:getNewKeyRequestContent>
```

### 3.2.2.7. FOCUS ON END TO END ENCRYPTION

See Ref 3 for generic information about and to end (ETE) encryption.

As defined in previous mentioned document, the process for ETE Encryption follows three steps:

1. The Public Key (ETK) of Recip-e is retrieved from ETK depot. (This step can pre-fetch, result can be cached)
2. The message is sealed using the Recip-e ETK
3. The Recip-e createPrescription() service is invoked

Recip-e ETK is retrieved from ETKDepot using KeyID “CBE=0823257311”.

Addressed encryption process has to be applied to all outgoing messages addressed to Recip-e. To allow Recip-e to unseal the message, the ETK (signed public key) of the care provider must be attached to the message (only if a response from Recip-e is expected).
That is why all messages addressed to recipe defined in the WSDL have the same structure:

- Request Message
  - SealedContent: the crypt request
  - ETK: the Encryption Token of the sender to be used by Recip-e to encipher the message

- Response Message
  - SealedContent: the crypt response (to be decrypted).

### 3.2.3. PRESCRIBER WEB SERVICES

The Recip-e Prescriber web service is described in a WSDL and XML schema. All required information, including the service URL is available via the eHealth-platform registry.

More information related to the eHealth-platform service registry is available on their website:

- NL: [https://www.ehealth.fgov.be/nl/support/registry](https://www.ehealth.fgov.be/nl/support/registry)
- FR: [https://www.ehealth.fgov.be/fr/support/registry](https://www.ehealth.fgov.be/fr/support/registry)

The following link can be used to access the registry in the acceptance and production environment directly:

- Acceptance: [https://services-acpt.ehealth.fgov.be/registry/uddi/bsc/web](https://services-acpt.ehealth.fgov.be/registry/uddi/bsc/web)
- Production: [https://services.ehealth.fgov.be/registry/uddi/bsc/web](https://services.ehealth.fgov.be/registry/uddi/bsc/web)

The Recip-e prescriber service is known as “Recip-ePrescriber v1” in the registry.

**Please note that the web service URL differs between the acceptance and production environment.**

### 3.2.4. ERROR MANAGEMENT

Each service may throw different type of error. The two types are:

- Business error for Functional Exception: indicates a functional error (such as missing information)
- SOAP Fault for Technical Exception: indicates a technical error due to the infrastructure (such as database unavailable, communication protocol issue).

For more information about SOAP Fault, please refer to [http://www.w3.org/TR/soap12-part1/#soapfault](http://www.w3.org/TR/soap12-part1/#soapfault)

**Sample Business error:**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<n1:AliveCheckResponse xmlns:n1="urn:be:fgov:ehealth:recipe:protocol:v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:be:fgov:ehealth:recipe:protocol:v1 ..\ehealth-recipeservices\XSD\recipeservices_protocol-1_0.xsd">
  <Status>
    <Code>500</Code>
    <Message Lang="FR">Erreur inattendue</Message>
    <Message Lang="EN">Unexpected error</Message>
  </Status>
</n1:AliveCheckResponse>
```

The message part contains the reference to the error message.

The label associated to the error message is also added in the detail of the fault in 4 languages (En, Fr, Nl, and Ge). The software provider can then choose the appropriate language for the error message.
3.2.5. USE OF EHEALTH-PLATFORM SERVICES

As described throughout this document, several eHealth-platform services must be integrated by the client software, e.g. Secure Token Service, ETK depot and KGSS. Detailed information related to these services is provided by eHealth-platform and might be available via the eHealth Technical Connector.

- **Secure Token Service (STS)**
  - NL: https://www.ehealth.fgov.be/nl/support/sts-secure-token-service
  - FR: https://www.ehealth.fgov.be/fr/support/sts-secure-token-service

- **ETK Depot and KGSS**
  - NL: https://www.ehealth.fgov.be/nl/support/basisdiensten/systeem-voor-end-end-vercijfering
  - FR: https://www.ehealth.fgov.be/fr/support/services-de-base/systeme-de-cryptage-end-to-end

3.3. SERVICE INVENTORY

This section lists all available services and operations as provided for prescribers. For each operation, the section “Implementation specification” details the different implementation steps.

Inputs/Outputs are also described for each service:
- **Input XML of the WS (Crypt Part of the message - Before Encryption process):** corresponds to the XML message to be generated before the encryption process.
- **Output XML of the WS (Encrypted Part of the message - After decryption process):** corresponds to the output that the Health Care Software will receive after decryption of the message.

3.3.1. ADMINISTRATIVE INFORMATION

Several requests should contain an AdministrativeInformation part next to the crypted content. This administrative part will be described for each operation (if applicable).

Example:

```xml
<xsd:complexType name="CreatePrescriptionRequestType">
  <xsd:complexContent>
    <xsd:extension base="protocol:RequestType">
      <xsd:sequence>
        <xsd:element name="SecuredCreatePrescriptionRequest" type="rc:SecuredContentType"/>
        <xsd:element name="AdministrativeInformation" type="rc:CreatePrescriptionAdministrativeInformationType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

3.3.2. PARTY IDENTIFICATION

Several outgoing messages addressed to Recip-e require an “IdentifierType” (i.e identification of a party) as part of the AdministrativeInformation, this should be defined as follows:

```xml
<xsd:complexType name="IdentifierType">
  <xsd:sequence>
    <xsd:element name="Id" type="xsd:string"/>
    <xsd:element name="Type" type="xsd:string"/>
    <xsd:element name="SubType" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```
Where:

- **Id** is the unique identification number of the party
- **Type** is the type of the identification number (NIHII, SSIN)

This part of the message is used by eHealth for different purposes:

- **Logging**: the information is logged by eHealth.
- **Orchestration**: eHealth can start complex processing based on those IDs (such as retrieving patient insurability).

Even if that information is not mandatory, Software vendors are encouraged to fill in those IDs when the information is known (ex: patientId and prescriberId should be filled in for the message “createPrescription”).

### 3.3.3. PRESCRIBER SERVICE OPERATIONS

#### 3.3.3.1. CREATE PRESCRIPTION

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>createPrescription</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSDL</td>
<td>Available in eHealth-platform registry.</td>
</tr>
<tr>
<td>Description</td>
<td>This action stores the prescription (Kmehr format) in Recip-e central system. This function takes care of the encryption (transport and storage). It returns the RID (Recipe ID) to be printed on the paper prescription (RID = “BEPPXXXXXXXX” where Be is the country, PP is the prescription type (pharmaceutical prescription) and XXX the ALPHANUM sequence ID)</td>
</tr>
<tr>
<td>When to be triggered</td>
<td>When the prescriber has generated a local prescription (XML Kmehr format), this function must be used to store it in Recip-e secured storage.</td>
</tr>
<tr>
<td>Errors</td>
<td>Technical Services Exception such as Connection failure, Encryption failure. Encryption Error : when Recip-e can’t decrypt the message</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input Parameter Description</th>
<th>Order</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>before encryption process</td>
<td>1</td>
<td>Boolean</td>
<td>feedbackRequested</td>
<td>Set to true if the prescriber wants to receive a feedback from the executor</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Long</td>
<td>patientId</td>
<td>INSS number of the patient</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>byte[]</td>
<td>prescription</td>
<td>Prescription content as a Validated XML Kmehr message. If the Prescription Type is set to PP, the prescription must contains a valid pharmaceutical prescription. (See XSD KMEHR Validation)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>String</td>
<td>prescriptionType</td>
<td>Type of prescription, will be used to generate the prescription ID and define what the allowed profiles for the delivery are. If prescription type is set to ‘PP’ (pharmaceutical prescription), only pharmacist will be allowed to retrieve the prescription. Other types of prescription will be</td>
</tr>
</tbody>
</table>
### Administrative Information

<table>
<thead>
<tr>
<th>No.</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IdentifierType</td>
<td>PatientIdentifier</td>
<td>INSS number of the real prescriber (can be different from the authenticated user)</td>
</tr>
<tr>
<td>2</td>
<td>IdentifierType</td>
<td>PrescriberIdentifier</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>String</td>
<td>PrescriptionType</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Base64Binary</td>
<td>KeyIdentifier</td>
<td></td>
</tr>
</tbody>
</table>

### Output Parameter Description

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>rid</td>
<td>RID of the prescription (size = 12 chars)</td>
</tr>
</tbody>
</table>

### Input XML of the WS (Encrypted Part of the message - Before Encryption process)

**XSD**

```xml
<xs:complexType name="createPrescriptionParam">
  <xs:sequence>
    <xs:element name="feedbackRequested" type="xs:boolean"/>
    <xs:element name="patientId" type="xs:long" minOccurs="0"/>
    <xs:element name="prescriberId" type="xs:long" minOccurs="0"/>
    <xs:element name="prescription" type="xs:base64Binary" minOccurs="0"/>
    <xs:element name="prescriptionType" type="xs:string" minOccurs="0"/>
    <xs:element name="keyID" type="xs:string" minOccurs="0"/>
    <xs:element name="ETK" type="xs:base64Binary" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

**Example**

```xml
<createPrescription xmlns="http://services.recipe.be">
  <CreatePrescriptionParam xmlns="">
    <feedbackRequested>true</feedbackRequested>
    <patientId>10000000000</patientId>
    <prescriberId>11111111111</prescriberId>
    <prescription>c3RyaW5n</prescription>
    <prescriptionType>PP</prescriptionType>
    <keyId>ABDEABDEABDE</keyId>
    <ETK>ABDEABDEABDE</ETK>
  </CreatePrescriptionParam>
</createPrescription>
```

### Output XML of the WS (Encrypted Part of the message - After decryption process)

**XSD**

```xml
<xs:complexType name="createPrescriptionResult">
  <xs:sequence>
    <xs:element name="rid" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

**Example**

```xml
<createPrescription xmlns="http://services.recipe.be">
  <CreatePrescriptionResult xmlns="">
    <rid>BEPPAAAAAAAA</rid>
  </CreatePrescriptionResult>
</createPrescription>
```
**Implementation Specification**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check the SAML token.</td>
</tr>
<tr>
<td>2</td>
<td>Request the Encryption Token (Signed Public Key) of the KGSS service to ETK Depot (eHealth service)</td>
</tr>
<tr>
<td>3</td>
<td>Generate a &quot;Symmetric Key request&quot; (including the key access control list)</td>
</tr>
<tr>
<td>4</td>
<td>The message should be ciphered using the Encryption Token of the recipient (seal()), the personal Encryption Token of the sender is attached to the request. (Recipient is KGSS)</td>
</tr>
<tr>
<td>5</td>
<td>The Sealed Response is received from Kgss, decrypt it. It contains the secured encryption key</td>
</tr>
<tr>
<td>6</td>
<td>The KMEHR Prescription is ciphered with the &quot;sealForUnknown&quot; function of the ETE framework using the symmetric key</td>
</tr>
<tr>
<td>7</td>
<td>The message is enriched with meta data (recipient id...)</td>
</tr>
<tr>
<td>8</td>
<td>A ciphered message is generated with the &quot;seal&quot; function of the ETE framework (Recipient : Recip-e), the message is sent to Recip-e</td>
</tr>
<tr>
<td>9</td>
<td>The message is decrypted. (Method &quot;unseal()&quot; of the ETEE framework is used). The response contains The RID of The prescription.</td>
</tr>
</tbody>
</table>

### 3.3.3.2. SEND NOTIFICATION

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>sendNotification</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSDL</td>
<td>Available in eHealth-platform registry.</td>
</tr>
<tr>
<td>Description</td>
<td>This function addresses sends a notification to a specific executor. The notification indicates that a client is likely to retrieve/execute his prescription with him.</td>
</tr>
</tbody>
</table>
| When to be triggered | • When a patient has habits in a specific pharmacy  
• When the product requires an ordering, |
| Errors         | Technical Services Exception such as Connection failure, Encryption failure  
ETK not present: the executor does not have a registered public Key at eHealth |
<p>| Input Parameter Description (before encryption process) | |</p>
<table>
<thead>
<tr>
<th>Order</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>byte[]</td>
<td>notificationText</td>
<td>Xml message containing the notification. Must be validated by the provided XSD</td>
</tr>
<tr>
<td>2</td>
<td>Long</td>
<td>executorId</td>
<td>INSS id of the recipient. Must be a valid INSS ID.</td>
</tr>
<tr>
<td>Administrative information</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Order</td>
<td>Type</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>IdentifierType</td>
<td>executorIdentifier</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>IdentifierType</td>
<td>PrescriberIdentifier</td>
<td></td>
</tr>
<tr>
<td>Output Parameter Description (after decryption)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Name</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>#N/A</td>
<td>#N/A</td>
<td>#N/A</td>
<td></td>
</tr>
</tbody>
</table>
Input XML of the WS (Encrypted Part of the message - Before Encryption process)

XSD

```xml
<xs:complexType name="addressPrescriptionParam">
  <xs:sequence>
    <xs:element name="content" type="xs:base64Binary" minOccurs="0"/>
    <xs:element name="executorId" type="xs:long" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Example

```xml
<addressPrescription xmlns="http://services.recipe.be">
  <AddressPrescriptionParam xmlns="">
    <content>c3RyaW5n</content>
    <executorId>01234567890</executorId>
  </AddressPrescriptionParam>
</addressPrescription>
```

Output XML of the WS (Encrypted Part of the message - After decryption process)

XSD

#N/A

Example

#N/A

### Implementation Specification

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check the SAML token.</td>
</tr>
<tr>
<td>2</td>
<td>request the Encryption Token (Signed Public Key) of recipient to ETK Depot (eHealth service)</td>
</tr>
<tr>
<td>3</td>
<td>Generate an XML request.</td>
</tr>
<tr>
<td>4</td>
<td>The message is ciphered using the Encryption Token of the recipient (seal()), the personal Encryption Token of the sender is attached to the request. (The recipient is the prescriber)</td>
</tr>
<tr>
<td>5</td>
<td>The message is enriched with meta data (recipient id...)</td>
</tr>
<tr>
<td>6</td>
<td>The message is ciphered using the Encryption Token of the recipient (seal()), the personal Encryption Token of the sender is attached to the request. (The recipient is Recip-e)</td>
</tr>
<tr>
<td>7</td>
<td>Send the message to Recip-e, nothing is returned if the action is processed successfully (otherwise, a SOAP fault is thrown)</td>
</tr>
</tbody>
</table>

#### 3.3.3.3. REVOKE PRESCRIPTION

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>revokePrescription</td>
<td>Available in eHealth-platform registry.</td>
</tr>
</tbody>
</table>
Description

Revoke the prescription (the prescription is removed from Recip-e system). It is then not possible to deliver it any more.

When to be triggered
When prescriber has done a mistake, he can use this function to revoke the bad prescription and then create a new one with the function "createPrescription".

An executor can also decide to revoke a prescription. For instance in the case where the patient request the executor to delete the prescription without delivering the prescribed items.

Errors
Technical Services Exception such as Connection failure, Encryption failure

<table>
<thead>
<tr>
<th>Input Parameter Description</th>
<th>Order</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>before encryption process)</td>
<td>1</td>
<td>String</td>
<td>rid</td>
<td>RID of the prescription (size = 12 chars)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>String</td>
<td>reason</td>
<td>Revoke reason to be used in the audit trail</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Administrative information</th>
<th>1</th>
<th>IdentifierType</th>
<th>PrescriberIdentifier / ExecutorIdentifier</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Output Parameter Description</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(after decryption process)</td>
<td>#N/A</td>
<td>#N/A</td>
<td>#N/A</td>
</tr>
</tbody>
</table>

Input XML of the WS (Encrypted Part of the message - Before Encryption process)

**XSD**

```
<xs:complexType name="revokePrescriptionParam">
  <xs:sequence>
    <xs:element name="reason" type="xs:string" minOccurs="0"/>
    <xs:element name="rid" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

**Example**

```
<revokePrescription xmlns="http://services.recipe.be">
  <RevokePrescriptionParam xmlns="">
    <reason>Mistake</reason>
    <rid>BPPAAAAAAAA</rid>
  </RevokePrescriptionParam>
</revokePrescription>
```

Output XML of the WS (Encrypted Part of the message - After decryption process)

**XSD**

#N/A

**Example**

#N/A

Implementation

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 3.3.3.4. GET PRESCRIPTION

<table>
<thead>
<tr>
<th>Specification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check the SAML token.</td>
</tr>
<tr>
<td>2</td>
<td>Request the Encryption Token (Signed Public Key) of recipient to ETK Depot (eHealth service)</td>
</tr>
<tr>
<td>3</td>
<td>Generate an XML request.</td>
</tr>
<tr>
<td>4</td>
<td>The message is ciphered using the Encryption Token of the recipient (seal()), the personal Encryption Token of the sender is attached to the request. (The recipient is the prescriber)</td>
</tr>
</tbody>
</table>

#### Operation Name
- **getPrescription (for prescriber)**

#### WSDL
- Available in eHealth-platform registry.

#### Description
- This action returns the content of the prescription (KMEHR Standard)

#### When to be triggered
- When the prescriber has lost a prescription from his local storage (ex: computer crash), he can use this function to retrieve the content (XML KMEHR).

#### Errors
- Technical Services Exception such as Connection failure, Encryption failure
- Inconsistent Status : when the prescription has been revoked, archived, is in Process

#### Input Parameter Description

<table>
<thead>
<tr>
<th>Order</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>String</td>
<td>rid</td>
<td>RID of the prescription (size = 12 chars)</td>
</tr>
</tbody>
</table>

#### Administrative information

<table>
<thead>
<tr>
<th>Order</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IdentifierType</td>
<td>PatientIdentifier</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>IdentifierType</td>
<td>PrescriberIdentifier</td>
<td></td>
</tr>
</tbody>
</table>

#### Output Parameter Description

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte[]</td>
<td>prescription</td>
<td>Prescription content as a Validated XML Kmehr message.</td>
</tr>
</tbody>
</table>

#### Input XML of the WS (Encrypted Part of the message - Before Encryption process)

**XSD**

```xml
<xsd:complexType name="getPrescriptionForPrescriberParam">
  <xsd:sequence>
    <xsd:element name="ETK" type="xsd:base64Binary" minOccurs="0"/>
    <xsd:element name="rid" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

**Example**

```xml
<getPrescriptionForPrescriber xmlns="http://services.recipe.be">
  <GetPrescriptionForPrescriberParam xmlns="">
    <ETK>c3RyaW5n</ETK>
    <rid>BEPFAAAAAAAA</rid>
  </GetPrescriptionForPrescriberParam>
</getPrescriptionForPrescriber>
```
**Output XML of the WS (Encrypted Part of the message - After decryption process)**

**XSD**

```xml
<xsd:complexType name="getPrescriptionForPrescriberResult">
  <xsd:sequence>
    <xsd:element name="creationDate" type="xsd:dateTime" minOccurs="0"/>
    <xsd:element name="encryptionKeyId" type="xsd:string" minOccurs="0"/>
    <xsd:element name="patientId" type="xsd:long" minOccurs="0"/>
    <xsd:element name="prescription" type="xsd:base64Binary" minOccurs="0"/>
    <xsd:element name="rid" type="xsd:string" minOccurs="0"/>
    <xsd:element name="timestampingId" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

**Example**

```xml
<getPrescriptionForPrescriber xmlns="http://services.recipe.be" ns="http://services.recipe.be">
  <GetPrescriptionForPrescriberResult xmlns="">
    <creationDate>2002-05-30T09:00:00</creationDate>
    <encryptionKeyId>ABCDEEABCDE</encryptionKeyId>
    <patientId>0123456789</patientId>
    <prescription>1234AERTZZZ</prescription>
    <timestampingId>7845233</timestampingId>
  </GetPrescriptionForPrescriberResult>
</getPrescriptionForPrescriber>
```

**Implementation Specification**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check the SAML token.</td>
</tr>
<tr>
<td>2</td>
<td>Generate an XML request.</td>
</tr>
<tr>
<td>3</td>
<td>The message is ciphered using the Encryption Token of the recipient (seal()), the personal Encryption Token of the sender is attached to the request. (The recipient is Recip-e)</td>
</tr>
<tr>
<td>4</td>
<td>Message is sent to Recip-e, Recip-e returns the result as a SOAP message</td>
</tr>
<tr>
<td>5</td>
<td>The message is decrypted. (Method “unseal()” of the ETEE framework is used).</td>
</tr>
<tr>
<td>6</td>
<td>Request the Encryption Token (Signed Public Key) of the KGSS service to ETK Depot (eHealth service)</td>
</tr>
<tr>
<td>7</td>
<td>Generate a “Symmetric Key request”</td>
</tr>
<tr>
<td>8</td>
<td>The message is ciphered using the Encryption Token of the recipient (seal()), the personal Encryption Token of the sender is attached to the request. (Recipient is KGSS)</td>
</tr>
<tr>
<td>9</td>
<td>The Sealed Response is received from Kgss, the module decrypt it. It contains the secured encryption key</td>
</tr>
<tr>
<td>10</td>
<td>The message is decrypted using unsealForUnknown()</td>
</tr>
</tbody>
</table>

**3.3.3.5. LIST FEEDBACK**

**Operation Name**  
listFeedback

**WSDL**  
Available in eHealth-platform registry.

**Description**  
This action list all the feedbacks sent by prescription executors. This action requires that you have your personal private key (SSIN=<niss>.p12) enabled in the module. There is different configuration possible:

- Single practice: system certificate = personal certificate : the private key is
already enabled. No additional step required!

- Fallback Session: the fallback session automatically enable the private key (using the personal password). No additional step required!
- Shared workstation: system certificate is a group practice certificate or the certificate of a legal responsible different of the care provider.

When to be triggered

There are different options:

- can be invoked on a regular basis (ex : 2x per day) and display incoming messages as popup to the prescriber.
- can be invoked on demand by the prescriber (ex : when the prescriber open a specific screen)

Errors

Technical Services Exception such as Connection failure, Encryption failure

Encryption Error : ex: if available private keys do not match the public key used for the encryption

Input Parameter Description

Administrative information

<table>
<thead>
<tr>
<th>Order</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IdentifierType</td>
<td>PrescriberIdentifier</td>
<td></td>
</tr>
</tbody>
</table>

Output Parameter Description

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List&lt;ListFeedbackItem&gt;</td>
<td>listFeedback</td>
<td>List of feedbacks (See XML Response for the description of this complex type)</td>
</tr>
</tbody>
</table>

Input XML of the WS (Encrypted Part of the message - Before Encryption process)

XSD

```xml
<xs:complexType name="listFeedbacksParam">
  <xs:sequence>
    <xs:element name="ETK" type="xs:base64Binary" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Example

```xml
<listFeedbacks xmlns="http://services.recipe.be">
  <ListFeedbacksParam xmlns="">
    <ETK>aaaaa</ETK>
  </ListFeedbacksParam>
</listFeedbacks>
```

Output XML of the WS (Encrypted Part of the message - After decryption process)

XSD

```xml
<xs:complexType name="ListFeedbackItem">
  <xs:sequence>
    <xs:element name="rid" type="xs:string" minOccurs="0"/>
    <xs:element name="sentBy" type="xs:long" minOccurs="0"/>
    <xs:element name="sentDate" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="content" type="xs:base64Binary" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```
Example

```xml
<listFeedbacks xmlns="http://services.recipe.be">
  <ListFeedResult>
    <ListFeedbackItem>
      <rid>BEPPAAAAAAA</rid>
      <sentBy>0123456789</sentBy>
      <sentDate>2010-01-01-00:00:00</sentDate>
      <content>ABCDEABCD</content>
    </ListFeedbackItem>
  </ListFeedResult>
</ListFeedbackItem>
```

### Implementation Specification

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check the SAML token.</td>
</tr>
<tr>
<td>2</td>
<td>Generate an XML request.</td>
</tr>
<tr>
<td>3</td>
<td>The message is ciphered using the Encryption Token of the recipient (seal()), the personal Encryption Token of the sender is attached to the request. (The recipient is Recip-e)</td>
</tr>
<tr>
<td>4</td>
<td>Send the Message to Recip-e, Recip-e returns the result as a SOAP message</td>
</tr>
<tr>
<td>5</td>
<td>The message should be decrypted. (Method &quot;unseal()&quot; of the ETEE framework is used).</td>
</tr>
<tr>
<td>6</td>
<td>Each single feedback should then be decrypted using unseal() method</td>
</tr>
</tbody>
</table>

### 3.3.3.6. LIST OPEN PRESCRIPTION

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>listOpenPrescription</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSDL</td>
<td>Available in eHealth-platform registry.</td>
</tr>
<tr>
<td>Description</td>
<td>This action Lists prescriptions created by the prescriber that are still in state &quot;NotDelivered&quot;. This function requires a PatientId as input argument. In case no PatientId is provided, an &quot;empty&quot; message will be returned.</td>
</tr>
<tr>
<td>When to be triggered</td>
<td>There are different options:</td>
</tr>
<tr>
<td></td>
<td>• can be invoked on a regular basis (ex : 2x per day) and display incoming messages as popup to the prescriber.</td>
</tr>
<tr>
<td></td>
<td>• can be invoked on demand by the prescriber (ex : when the prescriber open a specific screen)</td>
</tr>
<tr>
<td>Errors</td>
<td>Technical Services Exception such as Connection failure, Encryption failure</td>
</tr>
</tbody>
</table>

#### Input Parameter Description (before encryption process)

<table>
<thead>
<tr>
<th>Order</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IdentifierType</td>
<td>PrescriberIdentifier</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>IdentifierType</td>
<td>PatientIdentifier</td>
<td></td>
</tr>
</tbody>
</table>

#### Administrative information

<table>
<thead>
<tr>
<th>Output Parameter</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
**Description**

List<String> List of rid

**Input XML of the WS (Encrypted Part of the message - Before Encryption process)**

**XSD**

```xml
<xs:complexType name="listOpenPrescription">
  <xs:sequence>
    <xs:element name="prescriberId" type="xs:long" minOccurs="0"/>
    <xs:element name="patientId" type="xs:long" minOccurs="0"/>
    <xs:element name="ETK" type="xs:base64Binary" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

**Example**

```xml
<listOpenPrescription xmlns="http://services.recipe.be">
  <ListOpenPrescriptionParam xmlns="">
    <prescriberId>11111111111</prescriberId>
    <patientId>10000000000</patientId>
    <ETK>AAA</ETK>
  </ListOpenPrescriptionParam>
</listOpenPrescription>
```

**Output XML of the WS (Encrypted Part of the message - After decryption process)**

**XSD**

```xml
<xs:complexType name="GetListOpenPrescriptionResult">
  <xs:sequence>
    <xs:element name="prescriptions" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

**Example**

```xml
<listOpenPrescription xmlns="http://services.recipe.be">
  <ListOpenPrescriptionResult xmlns="">
    <rid>BEPP12345AAA</rid>
    <rid>BEPP1D345AAA</rid>
  </ListOpenPrescriptionResult>
</listOpenPrescription>
```

**Implementation Specification**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check the SAML token.</td>
</tr>
<tr>
<td>2</td>
<td>Generate an XML request.</td>
</tr>
<tr>
<td>3</td>
<td>The message is ciphered using the Encryption Token of the recipient (seal()), the personal Encryption Token of the sender is attached to the request. (The recipient is Recip-e)</td>
</tr>
<tr>
<td>4</td>
<td>Send the message to Recip-e, Recip-e returns the result as a SOAP message</td>
</tr>
<tr>
<td>5</td>
<td>Decrypt the message. (Method &quot;unseal()&quot; of the ETEE framework is used).</td>
</tr>
</tbody>
</table>

**3.3.3.7. UPDATE FEEDBACK FLAG**

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>updateFeedbackFlag</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSDL</td>
<td>Available in eHealth-platform registry.</td>
</tr>
</tbody>
</table>
### Description
Action to be used to change the “feedback flag”.

### When to be triggered
The feedback flag is already set during the "CreatePrescription" step. This specific operation should be used by the prescriber if he changes his mind.

### Errors
- Technical Services Exception such as Connection failure, Encryption failure
- Inconsistent Status: when the prescription has another status than “NotDelivered”

### Input Parameter Description (before encryption process)

<table>
<thead>
<tr>
<th>Order</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>String</td>
<td>rid</td>
<td>RID of the prescription (size = 12 chars)</td>
</tr>
<tr>
<td>2</td>
<td>boolean</td>
<td>feedbackRequested</td>
<td>Should be set to true, if the prescriber don’t want a feedback from the executor, this flag should be set to false</td>
</tr>
</tbody>
</table>

### Administrative information

<table>
<thead>
<tr>
<th>1</th>
<th>IdentifierType</th>
<th>PrescriberIdentifier</th>
</tr>
</thead>
</table>

### Output Parameter Description (after decryption process)

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#N/A</td>
<td>#N/A</td>
<td>#N/A</td>
</tr>
</tbody>
</table>

#### Input XML of the WS (Encrypted Part of the message - Before Encryption process)

**XSD**

```xml
<xs:complexType name="updateFeedbackFlagParam">
  <xs:sequence>
    <xs:element name="allowFeedback" type="xs:boolean"/>
    <xs:element name="rid" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

**Example**

```xml
<updateFeedbackFlag xmlns="http://services.recipe.be">
  <UpdateFeedbackFlagParam xmlns="">
    <allowFeedback>true</allowFeedback>
    <rid>BEPPAAAAAAAA</rid>
  </UpdateFeedbackFlagParam>
</updateFeedbackFlag>
```

#### Output XML of the WS (Encrypted Part of the message - After decryption process)

**XSD**

```
#N/A
```

**Example**

```
#N/A
```

### Implementation

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>Check the SAML token.</td>
</tr>
<tr>
<td>2</td>
<td>Generate an XML request.</td>
</tr>
<tr>
<td>3</td>
<td>Ciphered the message is using the Encryption Token of the recipient (seal()), the personal Encryption Token of the sender is attached to the request. (The recipient is Recip-e)</td>
</tr>
<tr>
<td>4</td>
<td>Send the message to Recip-e, nothing is returned if the action is processed successfully (otherwise, an exception is thrown)</td>
</tr>
</tbody>
</table>
## 4. APPENDIX: FREQUENTLY ASKED QUESTIONS

Following is a list of frequently asked questions concerning the implementation of the integration specifications and the use of the integration modules.

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doesn’t it make more sense to add a string in front of the Recipe ID to indicate it is a Recip-e ID (instead of immediately beginning with BEPP)</td>
<td>This is currently not foreseen. During the course of the pilot this will be evaluated.</td>
</tr>
<tr>
<td>Which doctor id should we use, the one in the Kmehr message, or the one in the header?</td>
<td>The INAMI/RIZIV number</td>
</tr>
<tr>
<td>Is it possible to store more than one prescription in the same Kmehr message?</td>
<td>No</td>
</tr>
<tr>
<td>Since there is no archiving currently foreseen, what has to happen with the MarkAsArchived function during the pilot? Should we always call it?</td>
<td>This should only be called when the prescription has been archived. For testing purposes, we will have test cases where this functionality needs to be called.</td>
</tr>
<tr>
<td>Can e-health provide a service to get public keys of doctors?</td>
<td>Use the ETK depot service.</td>
</tr>
<tr>
<td>Please provide more information on the ‘10 items’ limitation.</td>
<td>The printable area of the prescription is limited to 80 chars wide on 10 lines height. If the prescription requires more space, the content should be split on multiple prescription having each one its own RID</td>
</tr>
<tr>
<td>If a physical prescription contains more than 10 items, and we thus have to split it up over more than one electronic one, can we reuse eHealth keys for both prescriptions?</td>
<td>No, each prescription has its own encryption key.</td>
</tr>
<tr>
<td>In the notification, do we have to encapsulate the full Kmehr message in notification xml message?</td>
<td>The field is optional</td>
</tr>
<tr>
<td>The feedback is in text (without code CNK?). In our application, we need to flag delivered drugs. How can make something with text when receiving the feedback?</td>
<td>The feedback is linked to the overall prescription. No link is foreseen with delivery items.</td>
</tr>
<tr>
<td>Should the software provider maintain a list of executors (pharmacists) with correct IDs or are they available on e-Health?</td>
<td>Currently neither Recip-e nor e-Health provides a pharmacist inventory. So at this time, implement it in the software itself.</td>
</tr>
<tr>
<td>When do we need to signal the prescription as delivered? After one of the items has been delivered, or after all of the items have been delivered?</td>
<td>If 1 of the items is delivered, the complete prescription is considered as delivered, and the MarkAsDelivered functionality should be called.</td>
</tr>
<tr>
<td>Which encoding should be used?</td>
<td>All Recip-e messages (prescriptions, feedback and notifications) should use the UTF-8 encoding.</td>
</tr>
</tbody>
</table>