

# GOOD PRACTICES RECIP-E USE by software houses

## Content

Content.....	1
DOCUMENT HISTORY.....	3
I. Purpose of this document.....	4
II. Recip-e calls vs end-users business flows. ....	4
<b>A. Recip-e business case at the prescriber</b> .....	4
<b>B. Recip-e business case at the pharmacy</b> .....	6
III. Best practices .....	9
<b>A. Prescriber SWHs</b> .....	9
1. Automatic process (Batches) .....	9
2. Foresee guidance for prescriber to use GetPrescriptionStatus instead of GetPrescriptionForPrescribers .....	10
3. Expiration date in the header must be the same as in the Kmehr.....	10
4. Software package must check if the user present in the SAML Token is the same in the prescription and in the header (except for hospital ambulatory softwares).....	10
5. Hospital ambulatory softwares: Collective ListOpenPrescriptions not allowed .....	10
6. Handling of bandagist products .....	10
7. Conformity on error handling .....	11
8. Internet connection issue or Recip-e unavailability .....	11
9. Checking correctness of patient NISS .....	11
10. Keep created RID in software together with the prescription in the medical records of the patient.....	11
11. Special attention for three months validity period of prescriptions .....	11
12. Special attention for v2 prescriber softwares .....	11
13. Package identification.....	11
14. Correct mapping of CNK code and <Intended Name>.....	12
<b>B. Executors SWHs</b> .....	12
1. Automatic processes (Batches) to be avoided maximally! .....	12
2. MarkAsDelivered and MarkAsArchived functions are not reversible.....	13
3. Executor software must keep the status of the processed prescriptions .....	13
4. End of day process for In Process prescriptions .....	13

5.	It is mandatory to send notification of archivation towards Recip-e.....	13
6.	Implement a correct handling of ERR1000021 .....	13
7.	Package identification.....	13
<b>C.</b>	<b>Patient portal.....</b>	<b>14</b>
IV.	Rules for good use .....	14
1.	Prescribers .....	14
2.	Executors .....	15
3.	Patient .....	15
V.	Additional functions in V4 .....	15

## Document history

## DOCUMENT HISTORY

<b>Date</b>	<b>Version</b>	<b>Document Revision Description</b>	<b>Document Author</b>
21/02/2020	1.0	Initial version	Recip-e
11/05/2020	2.0	Additional BP	Recip-e
13/05/2020	2.1	Adding input of Marc Nyssens	Recip-e
19/05/2021	2,2	Update after review with Katrien's	Recip-e
03/06/2020	3.0	Multiple clean-ups (MN)	Recip-e
26/06/2020	4.0	Adding BP for having correct package identification	Recip-e
22/07/2020	5.0	Adding BP for alignment between used product intended name and CNK code.	Recipe
10/08/2020	6.0	Program identification must also be indicated in the connector-client.properties file	Recipe
10/08/2020	6.1	Adding special attention for V2 prescriber software	Recipe

## I. Purpose of this document

This document is targeted to all stakeholders in the process of creating, handling and delivering prescriptions. The purpose is to inform software designers on the correct call of recip-e services as well as the restrictions required to make sure the application works properly. It also describes the rules of good use of the application for prescribers, executors and patients.

Best practices and rules of good use must be followed correctly by end-users, as the patient is being cared for. Considering the large number of Recip-e users daily (prescribers, executors, patients), any misuse could lead to severe consequences on the application availability hence on the services provided to the patient.

## II. Recip-e calls vs end-users business flows.

Recip-e calls are performed at different moments of the patient journey. The system was designed to replace the traditional flow of paper-based prescriptions, following identical business steps, but making use of secure electronic messages and with a few additional functions.

### A. Recip-e business case at the prescriber

**Remark: Recip-e underlying calls are mentioned in: Green while status of prescription in Recip-e is in Blue**

1. The prescriber « opens » an e-health session with his e-ID or with the fall-back procedure, based on his eHealth- certificate, this session will potentially (unless the prescriber logs out) last for the working day. This is not a Recip-e function but a pre-requisite!
2. The prescriber opens the medical record of the patient he is going to meet, In case he wonders whether the patient has at least picked up the medication that was prescribed during previous consultation: Recip-e is consulted via “**listOpenPrescriptions PAT-ID**”.
3. To remember which medication was prescribed for this patient, the prescriber can refer to the medical record, that must be kept for the patient in the medical record, available to the prescriber. *Exceptionally*, when the medical record is not available and the prescriber wants to make sure which medication *was not picked up* by the patient, the specific prescription can be retrieved from Recip-e via the “**getPrescription RID**” call”
4. After or during the consultation, the prescriber wishes to prescribe medication, making use of the prescription module of his program; a pick-list hierarchy guides towards the medication of choice, to be completed with a posology (text and/or structured) and when applicable: validity period (by default: 3 months day by day including the current day).
5. The prescriber carefully verifies the content of the prescription as shown on the screen of the prescriber’s system.
6. When OK, the prescription is sent via “**createPrescription**” successfully to Recip-e and the prescriber can now print out the “proof of electronic prescription”, containing the RID identifier

and the medication that was prescribed. The Recip-e prescription uniquely identified by the RID is stored encrypted in the Recip-e database with status “**not delivered**”.

The prescriber is not allowed to add anything to the “proof of electronic prescription” by hand and should not sign!

If something goes wrong here, Recip-e does not return a RID and the prescriber’s program should generate and print a paper-based prescription, this document must be signed by the prescriber before being handed over to the patient.

7. If another prescription is required, we go back to point 4.
8. If the prescriber realizes that an error was made, for whatever reason, the “faulty” prescription can be “revoked” by the prescriber via the Recip-e call “**revokePrescription RID**”; the status of the electronic prescription is now “**revoked**” of course then the “proof of electronic prescription” should also be taken back from the patient, to avoid confusion!

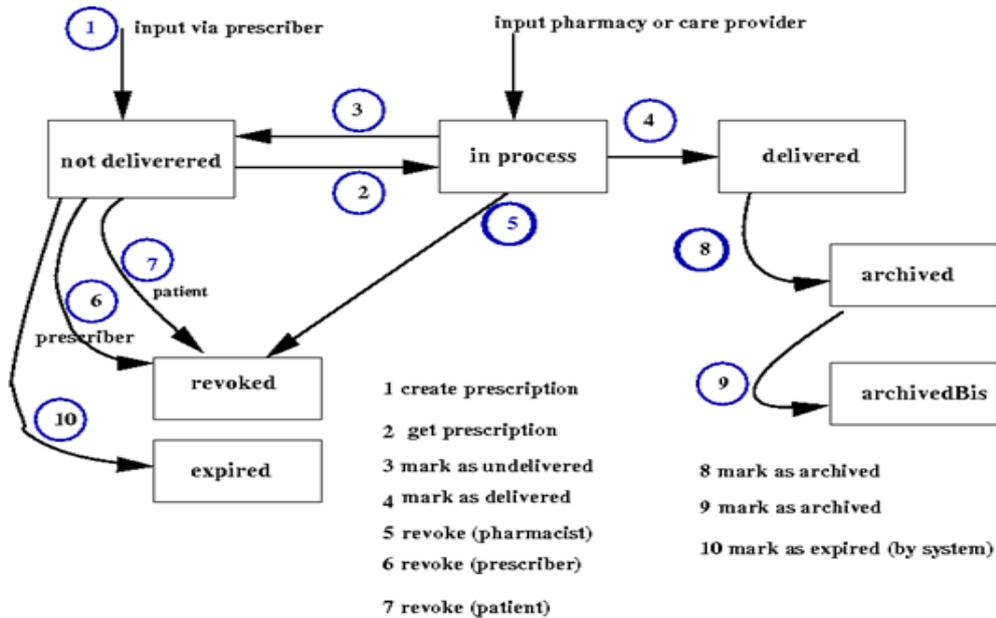
To resume the Recip-e part we see the following state diagram: a newly created electronic prescription obtains the state: “**not delivered**”, the prescriber and patient himself can request “**revokePrescription**” putting it in the state “**revoked**” and Recip-e will automatically set status “**expired**” when the expiration date is reached.

Prescriptions that are “revoked” or “expired” cannot be retrieved any more.

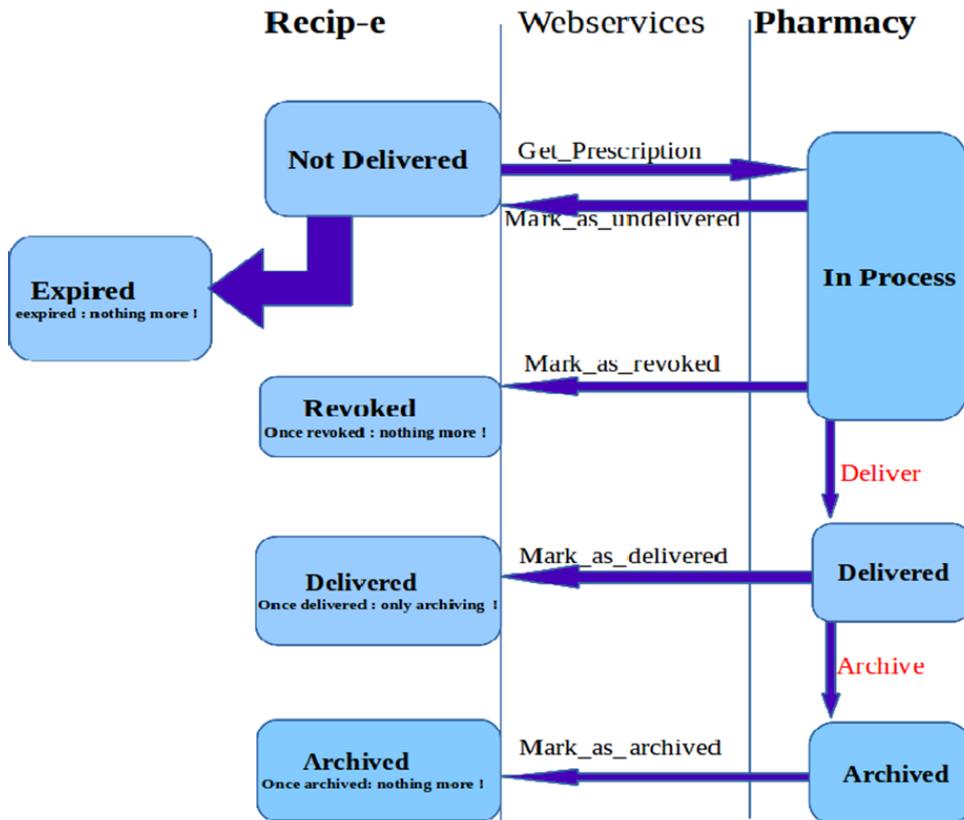
If another individual prescriber follows the first one, a new e-Health session SAML token must be obtained, corresponding to the identity of the new prescriber. This is not the case in the hospitals, where the hospital will generate a SAML token for all ambulatory prescriptions made from that institution.

The prescriber software should verify that the identity of the individual prescriber present in the SAML token is the same as the identity in the header of the Recip-e prescription and in the KMEHR-XML message of the prescription, if this is not the case: a new session must be initiated!

In hospital settings, the prescribing software must assure that the identity of the prescriber is correctly mentioned in the KMEHR-XML message of the prescription.



### B. Recip-e business case at the pharmacy



### A. with “proof of electronic prescription”

1. The pharmacist « opens » an e-health session with his e-ID or with the fall-back procedure, based on his eHealth- certificate, this session will potentially (unless the prescriber logs out) last for the working day. This is not a Recip-e function but a pre-requisite!
2. A patient enters the pharmacy and asks delivery of prescribed medication
  1. the patient hands over to the pharmacist a “proof of electronic prescription” of which the RID is scanned and the related prescription is obtained via Recip-e (**getPrescription RID**), contents are displayed on the screen; Recip-e status becomes “**in process**”.
  2. **A.** if the patient decides not to take at least one item of the prescription, the pharmacist can either give the proof of prescription back to the patient and announce to Recip-e “**markAsUndelivered RID**”. The prescription re-obtains the status “**not delivered**”, so that the patient can go to another pharmacy to obtain the required medication.
 

**B.** if the patient agrees, with the pharmacist that the medication on this prescription is no longer required, the pharmacist can revoke the prescription “**markAsRevoked RID**”, of course in this case, the prescription obtains the status “**revoked**”, it can never be delivered again and the “proof of electronic prescription” should also be thrown away, to avoid all confusion.
3. All delivered Recip-e prescriptions should be archived, for example by the RAOTD service of Farmaflux; after successful archiving, Recip-e should obtain a “**markAsArchived RID**” call, setting the status to “**archived**”, indicating that the prescription is safely archived and can be deleted from Recip-e.

#### **B. at de-materialisation, based on the patient’s identity**

1. The pharmacist « opens » an e-health session with his e-ID or with the fall-back procedure, based on his eHealth- certificate, this session will potentially (unless the prescriber logs out) last for the working day. This is not a Recip-e function but a pre-requisite!
2. A patient enters the pharmacy and asks delivery of prescribed medication, based on his/her identity
  1. if the pharmacist has a “therapeutic link” with the patient, he can proceed
  2. if the pharmacist does not have a “therapeutic link” with the patient, the pharmacist must initiate a therapeutic link with the patient, based on the e-ID of the patient
3. Once there is a therapeutic link between the patient and the pharmacist, the pharmacist can request Recip-e which are the “open” prescriptions (Recip-e status “**not delivered**”) for this patient via the Recip-e call “**ExecutorServices.listOpenPrescriptions Patient-ID**” (the status on Recip-e remains “**not delivered**”)
4. The pharmacist now sees on his screen the prescriptions, available via Recip-e and their content, asks the patient which ones should be delivered, the pharmacy program allows the pharmacist to “tick” the ones selected with the patient, when this selection is finalized, the program issues a Recip-e call: “**ExecutorServices.PutRidsInProgress RID1 RID2 ...**”
5. The pharmacist now selects which of the obtained prescriptions (that are now all in the status “**in process**” he will either deliver to the patient, either “undeliver” or “revoke”, this should result in a series of Recip-e calls: “**markAsDelivered**,” **markAsUndelivered**” and

“**markAsRevoked**”, corresponding to the actions taken for each prescription (and corresponding RID) in the pharmacy.

The pharmacy software should verify that for each RID that was put “**in process**”, a follow-up action was successfully taken.

6.

At the end of the day, every RID, that was put “**in process**”, should have been processed.

This can be verified by issuing the call: “**listRidsInProgress**”, which should result in an empty list.

### III. Best practices

This “best practices” section describes the rules SWHs must take into account in their development and the integration of Recip-e functionalities.

#### A. Prescriber SWHs

##### 1. Automatic process (Batches)

The general principle is that execution of automatic procedures interacting with the Recip-e system is forbidden except in some very specific circumstances. The table below summarizes for prescriber software which services require only atomic calls and the ones for which batch calls are exceptionally allowed.

Two “mode of operation allowed “exist;

- “Atomic” means that no batch call is allowed for the corresponding function
- “Batch” means that a batch call to the corresponding function is allowed, most of the time in some specific conditions

User	List of functions	Mode of operation allowed	Comment
<b>ambulatory GP + SP + Dentists + Midwives</b>	createPrescription	Atomic	The principle is no batch to create prescription. However, in some case (e.g. rest houses) this can be possible. TBC. If confirmed, we need to ask to put 1 sec between creation
<b>ambulatory GP + SP</b>	revokePrescription	Atomic	
<b>ambulatory GP + SP</b>	getPrescription	Atomic	
<b>ambulatory GP + SP</b>	sendNotification	Atomic	
<b>ambulatory GP + SP</b>	updateFeedbackFlag	Atomic	
<b>ambulatory GP + SP</b>	listFeedbacks	Batch	Allowed only 1x/day; prescriber software can provide a functionality for a prescriber to retrieve all the feedbacks from pharmacists for its patients. However, manual call is allowed only when 1 feedback is expected for patient for a specific prescription. Therefore, running a batch on listfeedback once a day is the maximum allowed.
<b>ambulatory GP + SP</b>	getPrescriptionStatus	Atomic	
<b>ambulatory GP + SP</b>	getValidationProperties	Atomic	
<b>ambulatory GP + SP</b>	listOpenRids	Atomic	
<b>ambulatory GP + SP</b>	listRidsHistory	Atomic	
<b>ambulatory GP + SP</b>	putVisionForPrescriber	Atomic	
<b>ambulatory HOSPITAL</b>	createPrescription	Atomic	
<b>ambulatory HOSPITAL</b>	revokePrescription	Atomic	
<b>ambulatory HOSPITAL</b>	getPrescription	Atomic	
<b>ambulatory HOSPITAL</b>	sendNotification	Atomic	

<b>ambulatory HOSPITAL</b>	updateFeedbackFlag	Atomic	
<b>ambulatory HOSPITAL</b>	listFeedbacks	Atomic	
<b>ambulatory HOSPITAL</b>	getPrescriptionStatus	Atomic	
<b>ambulatory HOSPITAL</b>	getValidationProperties	Atomic	
<b>ambulatory HOSPITAL</b>	listOpenRids	Atomic	
<b>ambulatory HOSPITAL</b>	listRidsHistory	Atomic	
<b>ambulatory HOSPITAL</b>	putVisionForPrescriber	Atomic	

2. Foresee guidance for prescriber to use GetPrescriptionStatus instead of GetPrescriptionForPrescribers

The software must foresee guidance of the prescriber, for him to use GetPrescriptionStatus instead of GetPrescriptionForPrescribers.

The call to GetPrescriptionForPrescriber is a heavy process and it brings too much unnecessary load on the Recip-e system.; it should be an exceptional call (e.g. when the medical record software has crashed). Normally, the prescriber should check what he has prescribed in the past, in his medical dossier.

The prescriber is guided through its software to call GetPrescriptionStatus that is less challenging for the Recip-e system.

3. Expiration date in the header must be the same as in the Kmehr

It is mandatory for SWH to implement a check to ensure that expiration date in the prescription is strictly identical to the expiration date in the header; the expiration date should be less than 1 year after the date of the creation of the prescription

4. Software package must check if the user present in the SAML Token is the same in the prescription and in the header (except for hospital ambulatory softwares)

It is mandatory for SWH to check that the user (prescriber) in the SAML token is strictly identical to the one in the Kmehr and in the header.

5. Hospital ambulatory softwares: Collective ListOpenPrescriptions not allowed

Retrieving open prescription for a list of patients is forbidden. Softwares must not allow for instance to retrieve prescriptions for all of hospital patients. A call of the service is allowed for specific patient but not for group of patients.

6. Handling of bandagist products

Bandagists products cannot be prescribed electronically.

#### 7. Conformity on error handling

All error messages produced by Recip-e must be transferred in a correct, functional and positive way to the prescriber, showing him the way forward.

#### 8. Internet connection issue or Recip-e unavailability

In case the internet connection is interrupted or issue on Recip-e while the prescribers is creating a prescription, the software must foresee to switch automatically for a paper prescription

#### 9. Checking correctness of patient NISS

Software must ensure that the NISS introduced is a valid number (correct structure).

#### 10. Keep created RID in software together with the prescription in the medical records of the patient.

It's required so that it is obvious to which prescription in the Electronic Medical Record of the patient, corresponds which Recip-e prescription

#### 11. Special attention for three months validity period of prescriptions

As from 01 November 2019 and as long as Recip-e V4 is not used, the validity period of a newly created prescription is setup to 3 months – 1 day by Recip-e server/

#### 12. Special attention for v2 prescriber softwares

As from 01 November 2019 and as long as Recip-e V4 is not used, prescriber softwares must guide prescribers to set correctly start of validity date. If Recip-e V4 ifnot used yet, the prescriber can only set a start of validity date within the 3 months. Prescribers will be able to setup a start validity date within 1 year as soon as they start using Recip-e V4 version of their software.

It is important to mention here that Recip-e server will not consider the start validity date when determining the default expiration date to 3 months – 1 days. The starting date for determining the expiration date is the creation date no matter if start validity date is indicated or not.

#### 13. Package identification

- It is mandatory for V4, to correctly identify prescriber software when calling Recipe services. The identification is through program name and version. The format for prescriber software vendor is as follow: **phyxxx[package/version]**
  - **phy** for physician
  - **xxx**=name or abbreviation of the software company
  - **package**= name of the package
  - **version**=version of the package

The package name must be requested to Recipe.

- Prescriber SDK

It is also mandatory to indicate software vendor in connector-client.properties file of the prescriber SDK. The default value of *programIdentification* parameter (DemoPrescriberClient.NET) must be replaced by the package identification as described above.

#### 14. Correct mapping of CNK code and <Intended Name>

The product CNK code retrieved from SAM V2 database must be mapped with the intended name used in the prescription. It means that, in the XML, the CNK code of the product <intended name> must correspond to the CNK code used in <intendedcd>. A check of those 2 CNKs must be performed before XML generation.

## B. Executors SWHs

### 1. Automatic processes (Batches) to be avoided maximally!

User	List of functions	Mode of operation allowed	Comment
executor	createFeedback	Atomic	
executor	getPrescriptionForExecutor	Atomic	
executor	markAsArchived	Atomic preferably Batch allowed only after 22 pm to 7 am	Only allowed during the night (after 22 pm to 7 am and with a limitation of 5 calls per second per pharmacy)
executor	markAsDelivered	Atomic, immediately corresponding to the handling of the patient	Must be performed before the patient leaves the pharmacy and proof of prescription taken away from the patient
executor	markAsUnDelivered	Atomic, immediately corresponding to the handling of the patient	Must be performed before the patient leaves the pharmacy and proof of prescription <b>given back</b> to the patient
executor	revokePrescription	Atomic	Must be performed before the patient leaves the pharmacy and proof of prescription thrown away
executor	listNotifications	Atomic	
executor	listOpenPrescriptions	Atomic	
executor	listReservations	Atomic	
executor	listRidsInProgress	Atomic	
executor	putRidsInProgress	Atomic	Designed to process 10 Rids max in 1 request
executor	getPrescriptionStatus	Atomic	
executor	listRidsHistory	Atomic	
executor	listRelations	Atomic	

## 2. MarkAsDelivered and MarkAsArchived functions are not reversible

Once a prescription is delivered or archived, it can't be put back to its previous status. The software must not allow the attempt to putting back the status to "undelivered", "in process", "revoked" or "Expired".

## 3. Executor software must keep the status of the processed prescriptions

The goal is to avoid unnecessary calls to Recip-e when the status can be checked locally. Mandatory to complete E2E delivery process

When a delivery process is started it must continue till the end. The software must not allow to close or interrupt the flow if the status is not delivered.

In case the delivery process cannot be conducted till the end for a valid reason (e.g. in case the patient has to come back few days later), the software must guide the pharmacist on how to correctly set the status, so that normal process can be continued afterwards.

## 4. End of day process for In Process prescriptions

At the end of the day, a process can verify if there are still "In Process" prescriptions in that pharmacy.

## 5. It is mandatory to send notification of archivation towards Recip-e

The software is responsible to create such messages and send it to Recip-e. All softwares must comply to that, independently how the archival is performed: via the RAOTD service, via other archival services for inside the hospital.

## 6. Implement a correct handling of ERR1000021

The software must avoid sending MarkAsUndelivered calls for RIDS that were already delivered or Undelivered. Should this happen anyway, then correctly interpret error message ERR1000021 (already delivered) to avoid unnecessary subsequent calls to MarkAsUndelivered. In fact, MarkAsUndelivered must be only called for "In Process" prescriptions.

## 7. Package identification

- It is mandatory for V4, to correctly provide executor software in the Recipe connector. The format for prescriber software vendor is as follow: **phaxxx[package/version]**
  - **pha** for physician
  - **xxx**=name or abbreviation of the software company
  - **package**= name of the package
  - **version**=version of the package

The package name must be requested to Recipe.

- It is also mandatory to indicate software vendor in connector-client.properties file of the executor SDK. The default value of *programIdentification* parameter (DemoExecuorClient.NET) must be replaced by the package identification as described above.

Many vendors still specify the package identifier as : "DemoExecuorClient.NET" !. All vendors must ensure that a correct identification of their package is done.

### C. Patient portal

User	List of functions	Mode of operation allowed	Comment
Patient	revokePrescription	Atomic	
Patient	ListOpenRids (previous ListOpenPrescriptions)	Atomic	
Patient	getPrescriptionForPatient	Atomic	
Patient	updateFeedbackFlag	Atomic	
Patient	PutvisionForPatient	Atomic	
Patient	GetVision	Atomic	
Patient	CreateReservation	Atomic	
Patient	ListRidsHistory	Atomic	
Patient	getPrescriptionStatus	Atomic	
Patient	CreateRelation	Atomic	
Patient	ListRelations	Atomic	
Patient	RevokeRelation	Atomic	

### IV. Rules for good use

On the other hand the rules of good use are targeted to end users (prescribers, executors, patients) to avoid misusing the system by performing unnecessary calls to Recip-e. (users should be guided by the end-user programs, that should be aware at all times of the statuses of the prescriptions that they are currently handling)

#### 1. Prescribers

- **Use GetPrescriptionStatus instead of GetPrescriptionForExecutor**

Users of prescriber software must get guidance to use GetPrescriptionStatus instead of GetPrescriptionFor executor whenever they want to verify if a prescription has been withdrawn at the pharmacy by the patient.

- **ListOpenPrescription □ ListOpenRids**

In case the prescriber wants to check the status of all prescriptions for the patient, he is encouraged to call ListOpenRids instead of ListOpenPrescriptions.

- **Special attention to deleted prescriptions**

When a prescriber deletes a prescription, he must ensure that no proof of prescription has been given to the patient.

In case a prescriber changes his mind after the patient has left, he must contact the patient to get back the proof of prescription for him to destroy the PeP. It's the prescriber's responsibility to get back the proof of electronic prescription even if the patient has already left.

- **Special attention for the start validity date**

As from November 2019, if prescribers want to use the start validity date to postpone the delivery of prescription, they must set it within the 3 months validity period. Recip-e is setting

anyway the expiration to 3 months after creation. The start validity date is not considered when setting the default expiration. Therefore, prescribers will have more flexibility after V4 roll-out since the start of validity date can be set within one year validity period.

## 2. Executors

- **Do not keep prescription status “in process” without setting correct status (delivered, undelivered, revoked) before the concerned patient leaves the pharmacy.**

Before serving another patient, the executor must ensure that current transaction is correctly closed. It means that for a prescription that is already delivered, the status must be set to “Delivered”. In case it is not delivered for different kind of reason, the status must be reset to “undelivered”. This is very important to avoid keeping a huge number of prescriptions in status “In Process”.

- **ListOpenPrescription to be use carefully together with PutRidsInProgress.**

The use of PutRidsInProgress must be followed by whether delivering, undelivering or deleting each of the selected RIDS.

- **Function markAsUndelivered must be only used for “In Process” prescriptions**

MarkAsUndelivered must never be used for “Expired”, “Revoked”, “Delivered” or “Archived” prescriptions

## 3. Patient

Use GetPrescriptionStatus instead of GetPrescriptionForPatients.

## V. Additional functions in V4

All the Recip-e V4 functions are already described in the general specifications dematerialisation document. Refer to that document for a complete description of those functions.