



EPJ API and Technical Specification

E-resept Forskrivningsmodul

Thula – Nordic Source Solutions

Date | 24/06/2015

Software version | FM 3.5.4

Document version | 15.0

Document status | Approved

Author | Thula

Contributor(s) | Ole Andreas Bjordal, Ole Martin Winnem

Distribution | Customer and EPJ vendors

File name | E-resept FM - EPJ API and Technical Specification.docx

© 2015 thula®

The document is only intended for Thula personnel and customers
Style and contents are confidential

Table of Contents

1	Document control	4
1.1	Revision tracking	4
1.2	Document source, storage and distribution	4
1.3	Revision history	4
1.4	Related documents	6
1.5	Reader comments	7
1.6	Glossary	7
2	Introduction	9
3	EPJ API specification	9
3.1	Methods in eResept.Forskrivningsmodul.1	9
3.1.1	LesCave	10
3.1.2	LesVarslinger	11
3.1.3	LesTakst	11
3.1.4	LesVarerIBruk	12
3.1.5	SkrivBrukerInfo	12
3.1.6	SkrivCave	12
3.1.7	SkrivKorrespondanseInfo	13
3.1.8	StartPasient	13
3.1.9	StartInbox	14
3.1.10	StartCave	15
3.1.11	LesSisteVibCaveOppdatering	15
3.1.12	LesAntallMeldingerForSignering	16
3.1.13	SlaSammenPasienter	16
3.1.14	Lukk	16
3.1.15	SkrivPasient	17
3.1.16	LesPasientStatus	17
3.1.17	SkrivUt	17
3.2	Methods in eResept.ForskrivningsmodulCallback.2 and 3	17
3.2.1	StartPasientFerdig	18
3.2.2	StartInboxFerdig	19
3.2.3	StartCaveFerdig	21
3.3	Data model	21
3.3.1	XML types	21
3.3.2	XML elements	22
3.3.3	XSD	35
3.4	API technology	35
3.5	XML import	35
4	Technical specification	36
4.1	System structure	36
4.1.1	Primary components	36

4.1.2	Secondary components	36
4.2	Technology.....	37
5	Technical requirements	38
5.1	Workstation requirements	38
5.2	Application server requirements	38
5.3	Database server requirements	38
5.4	Other technical requirements	39
6	Appendix A: EPJ API XSD	39
7	Appendix B: EPJ API Test Application.....	39
7.1	Registering the application.....	39
7.2	Accessing the COM interface	40
7.3	Calling methods on the COM interface	40
7.3.1	StartPasient	40
7.3.2	StartInbox	41
7.3.3	SkrivBrukerInfo	41
7.3.4	SkrivCave	42
7.3.5	SkrivKorrespondanseInfo.....	42
7.3.6	LesCave.....	42
7.3.7	LesVarslinger.....	42
7.3.8	LesTakst	42
7.3.9	LesVarerIBruk.....	42
7.4	Testing the callback interface	42

1 Document control

This section describes how to version, file, distribute and improve this document.

1.1 Revision tracking

This document is subject to revision control so that after each formal change a new version shall be created with a new data and revision number. At any given time the revision with the highest version number is considered the official and valid version of this document.

1.2 Document source, storage and distribution

The source of this document is maintained by Thula. It is stored in the Thula document repository in the following folder:

- E-resept \ System Documentation \ Paper Based

This document shall be distributed in PDF format only.

1.3 Revision history

Date	Version	Author/Approved by	Description
Release 3.5.4			
2015-06-24	15.0	Ægir Örn Leifsson	Added information about allowed values for Resept/Administrering. Reviewed changes from previous revision and approved document.
2015-06-24	14.2	Fernando Meira	Added section 3.3.2.38. Reviewed changes from previous revision.
2015-06-18	14.1	Ægir Örn Leifsson	Updated section 3.1.7 with information about supported user types for the SkrivKorrespondansInfo method.
Release 3.5.3			
2015.05.03	14.0	Ægir Örn Leifsson	Set document status to approved.
2015-04-15	14.0	Ægir Örn Leifsson	Added a reference to the "nurse" user type in sections 3.1.12, 3.3.2.2 and 3.3.2.50. Updated document version to 14.0.
Release 3.5.2			
2015-03-23	13.0	Ægir Örn Leifsson	Approved changes and updated document version to 13.0.
2015-02-05	12.1	Sveinn Ríkarður Jólsson	Updated document according to comments from OMW see RENO-9370
Release 3.5.0			
2014-10-24	12.0	Ægir Örn Leifsson	Approved changes and updated document version to 12.0.
2014-10-21	11.3	Atli Sturluson	Updated some outdated technical requirements for the database and FM server
2014-08-14	11.2	Atli Sturluson	Added documentation for new API methods and elements added in 3.5.
2014-06-19	11.1	Atli Sturluson	Updated TOC and EPJ API xsd
2014-06-19	11.0	Sveinn Ríkarður Jólsson	Added new API method (SkrivPasient)
Release 2.14.2			
2014-05-20	10.0	Ægir Örn Leifsson	Updated documentation to cover all API methods supported in namespace version 2014-01-14.

			Some reorganization of the document. Replaced all references to “PM” with references to “FM”.
<i>Release 2.14.1</i>			
2014-04-16	9.0	Ægir Örn Leifsson	Approved version 9.0.
2014-04-15	8.1	Ægir Örn Leifsson	Regenerated TOC.
<i>Release 2.13.0</i>			
2014-02-28	8.0	Ægir Örn Leifsson	Updated documentation to namespace version 2014-01-14. Schema changes are: <ul style="list-style-type: none"> • Organization and admission data in StartPasient and StartCave have been restructured. • RESH-Id of default avdeling can be specified in SkrivBrukerInfo. Removed method parameter examples (section 7.5).
<i>Release 2.12.1</i>			
2013-09-03	7.0	Fernando Meira	Version number updated for 2.12.1 release.
<i>Release 2.9.1</i>			
2013-04-02	5.1	Atli Sturluson	Added new API method (Lukk)
<i>Release 2.9.0</i>			
2012-03-26	5.0	Ægir Örn Leifsson	Approved changes for version 2.9.0.
2012-03-13	4.2	Ægir Örn Leifsson	Changed InVib to VibStatus in the Resept element.
2013-03-12	4.1	Ægir Örn Leifsson	Added InVib and InstituertAv fields to the Resept element. Updated XML namespace version.
2013-02-27	4.0	Ægir Örn Leifsson	Added information about Samstillingstekst element in StartPasient. Updated XML namespace version.
<i>Release 2.7.2</i>			
2012-10-11	3.1	Ægir Örn Leifsson	Added information about session ID in Start... methods and their callbacks.
<i>Release 2.7.0</i>			
2012-10-07	3.0	Ægir Örn Leifsson	Version number updated for 2.7.0 release.
2012-09-27	2.4	Ægir Örn Leifsson	Updated documentation for version 3 of the callback interface.
2012-09-21	2.3	Ægir Örn Leifsson	Added documentation for versions 2 and 3 of the callback interface (<code>eResept.ForskrivningsmodulCallback.2</code> and <code>eResept.ForskrivningsmodulCallback.3</code>).
2012-09-16	2.2	Ægir Örn Leifsson	Added documentation about new fields in the Resept class (see section Villa! Uppruni ilvísunar finnst ekki.), also added documentation for existing fields in the same class that lacked proper documentation. Updated the namespace version (to 2012-09-15). Added documentation about the new LesSisteVibCaveOppdatering and LesAntallMeldingerForSignering methods.

2012-09-12	2.1	Magnús Kristjánsson	Reformatted using new Thula template.
2012-09-11	2.0	Ægir Örn Leifsson	Added information about the ErstattAlle attribute in SkrivCave (see section Villa! Uppruni tilvísunar finnst ekki.).
<i>Release 2.3.90</i>			
2012-04-12	1.35a	Atli Sturluson	Added the Read-only user role to the SkrivBrukerInfo type
<i>Release 2.3.80</i>			
2012-02-21	1.34a	Viðar Júlíusson	Adding section 3.2.2.28 and updating 3.3.2.23 - admission information that can be added to StartCave and StartPasient
<i>Release 2.3.60</i>			
2011-01-10	1.34a	Atli Sturluson	Updated the SkrivBrukerInfo type in section Villa! Uppruni tilvísunar finnst ekki.
2010-12-08	1.33a	Brynjar Bragason	Replaced screenshots in section 7. Added section 7.4 about the call back interface.
<i>Release 2.3.42</i>			
2010-11-24	1.32a	Ægir Örn Leifsson	Fixed an error in the XSD in Appendix A. The Varslinger element had a cardinality of 1 in LesVarslingerSvar, this has been fixed and is now 0..unbounded. The text describing LesVarslingerSvar was correct (section Villa! ppruni tilvísunar finnst ekki.).
2010-11-22	1.31a	Ægir Örn Leifsson	Added Appendix B containing information about an EPJ API test application.
2010-10-26	1.3a	Ægir Örn Leifsson	Approved by Jon Tysdahl
2010-10-22	1.3d	Ægir Örn Leifsson	Added the following: <ul style="list-style-type: none"> • Use of unique patient identifiers in all patient-related methods. • StartForskrivning renamed to StartPasient. • LesInbox renamed to LesVarslinger. • The CAVE element (used in Les/SkrivCave) includes a unique record identifier to enable record-by-record updating. • LesInbox extended to include detailed information about unread notifications. • A new call-back interface has been defined that allows the FM to notify an EPJ when the processes started by StartPasient and StartInbox are done. This call-back is optional and opt-in for the EPJs. • Updated the glossary.
2010-10-06	1.2d	Ægir Örn Leifsson	Added sequence diagrams and more detailed descriptions to some parts of section 3.1. Updated the XSD in section 6. Applied some minor fixes in the text here and there.
2010-10-05	1.1d	Bjarni Gunnar Ívarsson	Added Role to SkrivBrukerInfo datamodel. Updated CAVE data model.
2010-10-04	1.0d	Magnús Kristjánsson	Created, by copying sections from the Detail Specification.

1.4 Related documents

The following background documents are relevant to the EPJ API:

Document	Description
Krav_til_forskrivningsmodul_ssa- u_bilag_1	The customer requirements specification.
DFS_versjon_1 9	Detailed functional specification for e-resept in general.
http://msdn.microsoft.com/en-us/library/ms143506.aspx	Hardware and software requirements for MS SQL Server.

1.5 Reader comments

If you have any comments on the contents of this document please send those by e-mail to thula@thula.is. If a review result in changes, all users of this document should be notified.

1.6 Glossary

This section lists some terms used in this document and specifies how they are to be interpreted within the scope of the document.

Abbreviation	Explanation or web reference
Forskrivningsmodul	Prescription module. The software being described in this document.
FM	Forskrivningsmodul.
EPJ	Electronic patient journal. A computerized patient record/journal system that communicates with the FM through the FM EPJ API and import/export of user and patient data (described in section 3).
API	An application programming interface (API) is an interface implemented by a software program that enables it to interact with other software. See http://en.wikipedia.org/wiki/API .
LIB	Medications in use (legemidler i bruk). A list of the medications in use by a single patient. This list may at any time be up-to-date or not up-to-date (the difference being the certainty the user can place in the accuracy of the information presented). LIB can contain medication or food supplement (kosttilskudd) prescriptions that have been sent to the RF or printed out on paper. But it can also contain prescriptions that are only stored within the LIB. Behind each prescription in the LIB there can be a chain of earlier prescriptions that are retained in medication history and possibly resept history.
NIB	Nutrition support in use (næringsmidler i bruk). A list of nutrition support currently prescribed to a single patient. This list may at any time be up-to-date or not up-to-date (the difference being the certainty the user can place in the accuracy of the information presented).
FIB	Medical disposables in use (medisinsk forbruksmateriell i bruk). A list of the medical disposables currently prescribed to a single patient. This list may at any time be up-to-date or not up-to-date (the difference being the certainty the user can place in the accuracy of the information presented).
VIB	LIB, NIB and FIB combined.
Prescription	Specification, created by a physician, of a medication treatment to be followed by a patient. Prescriptions can be registered with one of a number of status flags that specify if the prescription is simply a record in the LIB or if it also entails an order from a physician to a pharmacy to deliver certain medication in a certain amount to a given patient (resept).
Resept	An order from a physician to a pharmacy to deliver certain medication in a certain amount to a given patient.
RF	Reseptformidleren. Reseptformidleren er et sentralt elektronisk helseregister/database som de aller

flESTE meldinger i e-resept går gjennom. Her oppbevares den elektroniske resepten og her slettes den, 4 uker etter at den er blitt ugyldig, det vil si ferdig ekspedert eller utløpt på dato.

2 Introduction

This document is the result of the detail specification work performed as part of the e-resept Forskrivningsmodul project in co-operation between Helsedirektoratet and Thula. During several design workshops the two parties have defined the functionality and flow of the system, in addition to the EPJ API and technical requirements.

This document contains the EPJ API, technical specification and technical requirements, copied from the Detail Specification document.

3 EPJ API specification

The EPJ API is the interface used by EPJ systems to interact with FM. The API provides the only way to initiate the prescription process in the FM. The API consists of two interfaces:

- eResept.Forskrivningsmodul.1 – this is the interface implemented by the FM and exposed to EPJ systems (the only supported version is eResept.Forskrivningsmodul.1).
- eResept.ForskrivningsmodulCallback.2 – this interface can optionally be implemented by those EPJ systems that want to get notifications from the FM when the user completes his work in the FM and wishes to return to the EPJ (two versions are currently supported, eResept.ForskrivningsmodulCallback.2 and eResept.ForskrivningsmodulCallback.3).

Both of these interfaces define a set of methods that can be called by the EPJ (eResept.Forskrivningsmodul.1) or the FM (eResept.ForskrivningsmodulCallback.2 and 3).

The methods are defined in the following sections.

3.1 Methods in eResept.Forskrivningsmodul.1

All methods on eResept.Forskrivningsmodul.1 are XML document based, i.e. they take in one XML document as a parameter and return an XML document with the results.

The API supports two older namespaces:

- <http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-03-12>
- <http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2014-01-14>
- <http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2014-05-02> (current)

Each method accepts an XML document with a specific root element. All of the root elements include at least the following XML element (defined in typeForesporsel in the WSDL):

XML element	Description
LoginInfo	Includes a username and hashed password that is used by the FM to authenticate the EPJ user. The hash must match the hash that is specified when the user is created (using SkrivBrukerInfo).

Each method also returns an XML document with a specific root element. All of the root elements include at least the following XML elements (defined in typeSvar in the WSDL):

XML element	Description
Returkode	An integer specifying the return code for the method. The following values are defined: 0 – OK.

	1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 4 – The UI is blocked (e.g. unsaved data). 5 – Patient is not registered in FM. 6 – Patient identifier missing. 7 – Other content error (a missing or invalid field). 8 – UI was closed with possible loss of changes (only returned by Lukk). 9 – UI was closed when there are unsent RF messages (only returned by Lukk).
Feilmelding	Optional error message (in Norwegian) suitable to display to the user.
InternFeilmelding	Optional internal error message (in English) that is not suitable to display to the user. This should be used by the EPJ vendor to aid in debugging problems with EPJ-FM integration.

Note that even though the EPJ API allows an EPJ to change the current patient in the FM (using StartPasient) the API does not guarantee that the EPJ and FM patient contexts always stay synchronized.

The following sections define the methods exposed through the eResept.Forskrivningsmodul.1 interface. Each section has a short description of the semantics of the method, as well as a table like the following:

	Description
Parameter XML	This defines the XML element that is accepted as the root element of the parameter XML document. Note that all of the elements extend typeForesporsel and therefore include a LoginInfo element.
Return XML	This defines the XML element that is the root element of the returned XML document. Note that all of the return elements extend typeSvar and therefore include Returkode, Feilmelding and InternFeilmelding.
Return codes	This defines the possible return codes for the method.

3.1.1 LesCave

This method is used to read current CAVE information for the specified patient. No historical information (i.e. CAVE information that has been marked as no longer valid) is returned by this method.

NOTE: This method will return error code 5 if the referenced patient cannot be found; it will not create a new patient in the FM. However, if the patient is already known to the FM, the patient information is updated to reflect what was passed in from the EPJ.

	Description
Parameter XML	LesCave
Return XML	LesCaveSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 5 – Patient is not registered in FM. 6 – Patient identifier missing.

3.1.2 LesVarslinger

This method is used to determine if there is anything in the FM that a user needs to give attention. Each user has an Inbox that includes notifications of events the user should be aware of and may need to act on. This method returns unread notifications from the user's FM Inbox.

	Description
Parameter XML	LesVarslinger
Return XML	LesVarslingerSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation.

Figure 1 shows the context in which LesVarslinger can be used (steps 1 and 2). The EPJ periodically polls the FM for the user's inbox status and gives some sort of notification when the inbox is not empty. The user can then ask the EPJ to open his FM Inbox.

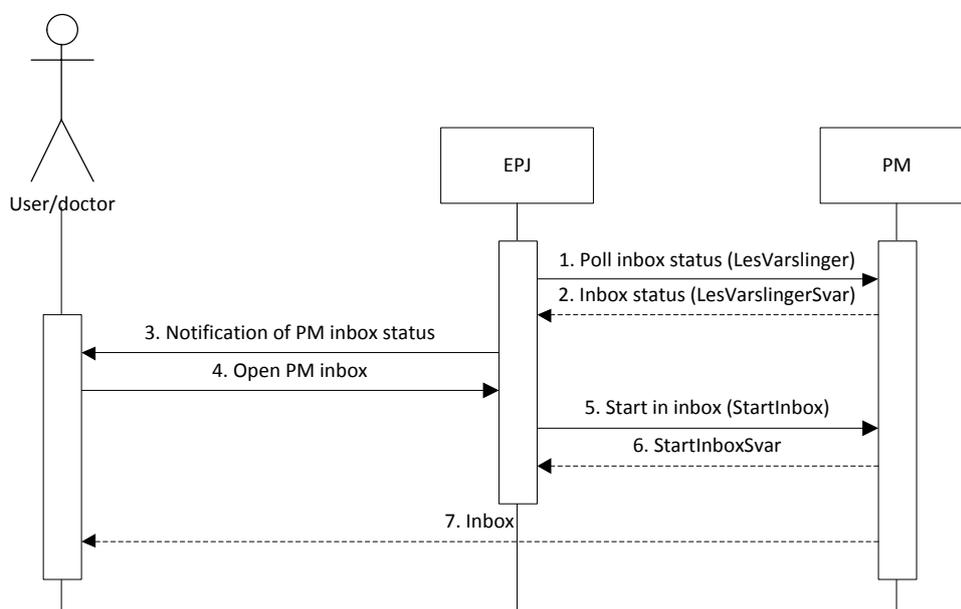


Figure 1 - EPJ/FM user opens the FM Inbox.

3.1.3 LesTakst

This method is used to get a list of rate (Norwegian: takst) records that have been registered for a specified patient at a specified date.

NOTE: This method will return error code 5 if the referenced patient cannot be found; it will not create a new patient in the FM. However, if the patient is already known to the FM, the patient information is updated to reflect what was passed in from the EPJ.

	Description
Parameter XML	LesTakst
Return XML	LesTakstSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 5 – Patient is not registered in FM.

6 – Patient identifier missing.

3.1.4 LesVarerIBruk

This method is used to get a list of medications (LIB), nutrition support (NIB) and medical disposables (FIB) for a specified patient, note that this method will not return any information about vaccines.

NOTE: This method will return error code 5 if the referenced patient cannot be found; it will not create a new patient in the FM. However, if the patient is already known to the FM, the patient information is updated to reflect what was passed in from the EPJ.

	Description
Parameter XML	LesVarerIBruk
Return XML	LesVarerIBrukSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 5 – Patient is not registered in FM. 6 – Patient identifier missing.

3.1.5 SkrivBrukerInfo

This method is used to create and update user info in the FM. It should be used both to initialize all user profiles after installation and to report changes to existing users and the introduction of new users after the initial installation and configuration of the FM.

NOTE: This is the only method that supports a LoginInfo element with an empty username and password. In this case the FM will ask the user to authenticate using a smart card (and pin code). The user that is allowed to authenticate in this way is defined during the FM installation process.

	Description
Parameter XML	SkrivBrukerInfo
Return XML	SkrivBrukerInfoSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation.

3.1.6 SkrivCave

This method is used to update or add to the current CAVE information for the specified patient.

NOTE: This method will create a new patient file in the FM if the patient identifier provided cannot be found. If the patient is already known to the FM, the patient information is updated to reflect what was passed in from the EPJ.

NOTE: This method will create a new CAVE entry in the FM if a provided CAVE entry identifier cannot be found. If the CAVE entry is already known to the FM, the CAVE entry is updated to reflect what was passed in from the EPJ (the previous version of the CAVE entry will be retained in CAVE history in the FM).

	Description
Parameter XML	SkrivCave
Return XML	SkrivCaveSvar

Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 6 – Patient identifier missing.
--------------	---

3.1.7 SkrivKorrespondanseInfo

This method is used by the EPJ to add to the list of medications (LIB), nutrition support (NIB) and medical disposables (FIB) for a specified patient. The information received will be shown at the bottom of the LIB/NIB/FIB screens, in the same area used for incoming M8 message and unknown prescriptions received in M9.6. From these, the FM user can manually choose to import the received information into the LIB/NIB/FIB.

SkrivKorrespondanseInfo should only be called by doctors (users with the user type “Doctor”). Use of this method by other user types is not supported.

NOTE: This method will create a new patient file in the FM if the patient identifier provided cannot be found. If the patient is already known to the FM, the patient information is updated to reflect what was passed in from the EPJ.

	Description
Parameter XML	SkrivKorrespondanseInfo
Return XML	SkrivKorrespondanseInfoSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 6 – Patient identifier missing.

3.1.8 StartPasient

This method is used to start the prescription process for the provided patient.

NOTE: This method will create a new patient file in the FM if the patient identifier provided cannot be found. If the patient is already known to the FM, the patient information is updated to reflect what was passed in from the EPJ.

This method will bring up the FM showing medications in use for the patient (the LIB).

	Description
Parameter XML	StartPasient
Return XML	StartPasientSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 4 – The UI is blocked (e.g. unsaved data). 6 – Patient identifier missing. 7 – Other content error (a missing or invalid field). 8 – UI was closed with possible loss of changes (only returned by Lukk). 9 – UI was closed when there are unsend RF messages (only returned by Lukk).

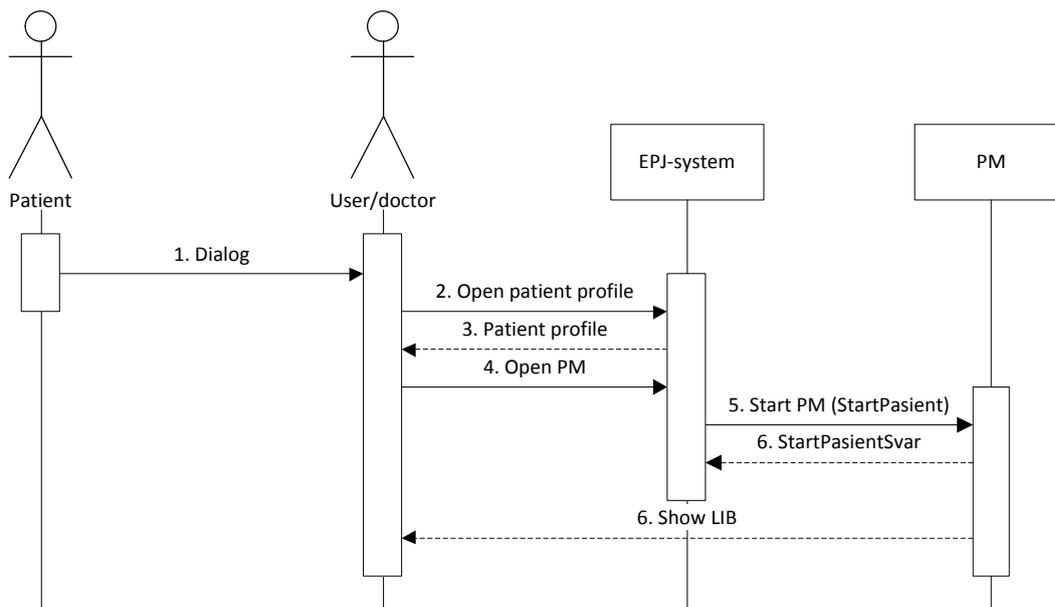


Figure 2 - EPJ/FM user opens a specific patient record.

The EPJ can request a call-back from the FM when the doctor has finished his work in the FM. This mechanism is described in section 3.2.1.

3.1.9 StartInbox

This method is used to open the user's Inbox.

	Description
Parameter XML	StartInbox
Return XML	StartInboxSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 4 – The UI is blocked (e.g. unsaved data). 7 – Other content error (a missing or invalid field). 8 – UI was closed with possible loss of changes (only returned by Lukk). 9 – UI was closed when there are unsend RF messages (only returned by Lukk).

Figure 3 (same as Figure 1) shows the context in which StartInbox can be used (steps 5 and 6). The EPJ periodically polls the FM for the user's inbox status and gives some sort of notification when the inbox is not empty. The user can then ask the EPJ to open his FM Inbox.

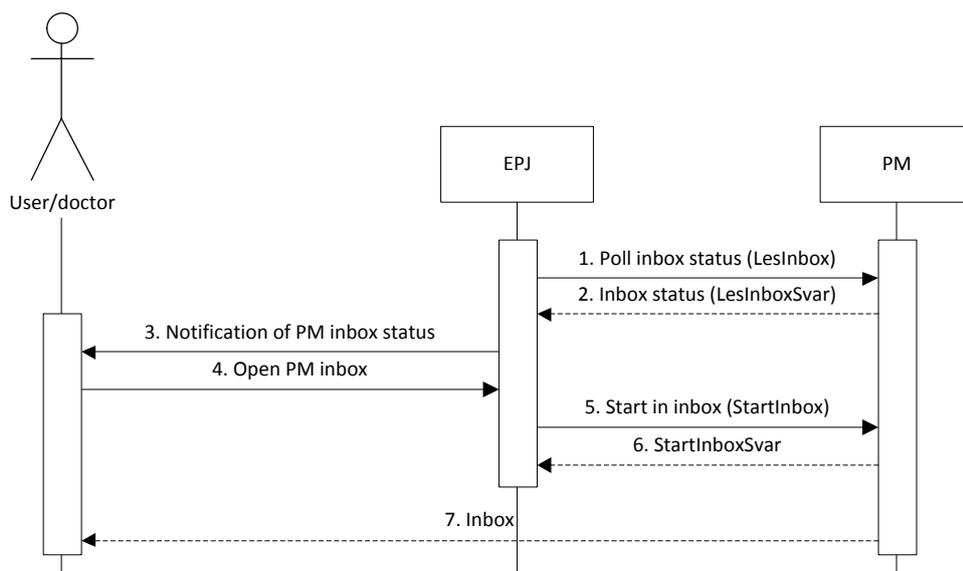


Figure 3 - EPJ/FM user opens the FM Inbox.

The EPJ can request a call-back from the FM when the doctor has finished his work in the FM. This mechanism is described in section 3.2.2.

3.1.10 StartCave

This method is used to start the FM at the CAVE registration screen for the provided patient.

NOTE: This method will create a new patient file in the FM if the patient identifier provided cannot be found. If the patient is already known to the FM, the patient information is updated to reflect what was passed in from the EPJ.

This method will bring up the FM showing CAVE for the patient.

	Description
Parameter XML	StartCave
Return XML	StartCaveSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 4 – The UI is blocked (e.g. unsaved data). 6 – Patient identifier missing. 7 – Other content error (a missing or invalid field). 8 – UI was closed with possible loss of changes (only returned by Lukk). 9 – UI was closed when there are unsend RF messages (only returned by Lukk).

The EPJ can request a call-back from the FM when the doctor has finished his work in the FM. This mechanism is described in section 3.2.3.

3.1.11 LesSisteVibCaveOppdatering

This method returns the date and time of the latest update to the VIB, CAVE and takst for a given patient.

	Description
Parameter XML	LesSisteVibCaveOppdatering

Return XML	LesSisteVibCaveOppdateringSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 5 – Patient is not registered in FM. 6 – Patient identifier missing.

3.1.12 LesAntallMeldingerForSignering

This method returns the number of M1 and M5 messages that are waiting to be signed and/or sent by a given user. Those are messages either created by the given user himself, or created by an assistant (or nurse) where the assistant/nurse has indicated the given user as the requesting or responsible doctor.

	Description
Parameter XML	LesAntallMeldingerForSignering
Return XML	LesAntallMeldingerForSigneringSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation.

3.1.13 SlaSammenPasienter

This method merges two patients from the FM database into a single patient. The patient file identified as secondary will be merged into the patient file identified as primary. After the merge the secondary patient file will no longer contain a VIB.

	Description
Parameter XML	SlaSammenPasienter
Return XML	SlaSammenPasienterSvar
Return codes	0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation.

3.1.14 Lukk

This method closes the FM user interface.

NOTE: The user will NOT be given a change to save potentially unsaved changes before closing the UI. If there are open windows that might have unsaved changes, the UI will be closed and return code 8 returned. If there are unsent messages on the signing queue, return code 9 will be returned.

	Description
Parameter XML	Lukk
Return XML	LukkSvar
Return codes	0 – OK. 3 – XML document does not pass validation. 8 – UI was closed with possible loss of changes. 9 – UI was closed when there are unsent RF messages.

3.1.15 SkrivPasient

This method creates or updates a patient. The parameter contains patient information and optional resept (0 – many), resepthistorikk (0 – 1) and CAVE (0 – many). If patient already exists the patient information is updated according to submitted info, prescriptions and CAVE with known Ids will be ignored but others will be added. ReseptHistorikk, if included, will overwrite existing. On success the reply will contain one of two return codes, Ok, the patient was created (0) or Existing patient updated (10).

	Description
Parameter XML	SkrivPasient
Return XML	SkrivPasientSvar
Return codes	0 – OK (success). 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 10 – Existing patient updated (success).

3.1.16 LesPasientStatus

This method returns status information for a group of patients. The input parameter contains the list of patients that status should be returned for.

	Description
Parameter XML	LesPasient
Return XML	LesPasientSvar
Return codes	0 – OK (success). 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation or contains no patient Ids.

3.1.17 SkrivUt

This method is used for printing out the same type of report for many patients. The input parameter contains information on which report to print, the patients that should be included and any parameters needed to create the report.

	Description
Parameter XML	SkrivUt
Return XML	SkrivUtSvar
Return codes	0 – OK (success). 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation or contains no patient Ids.

3.2 Methods in eResept.ForskrivningsmodulCallback.2 and 3

The methods in eResept.ForskrivningsmoduleCallback.2 and eResept.ForskrivningsmodulCallback.3 do return anything. Each method takes the following parameters:

- eResept.ForskrivningsmodulCallback.2 – all methods take a single string parameter that contains the username of the user who made the initial call.

- eResept.ForskrivningsmodulCallback.3 – all methods take a single string parameter that contains an XML document that has a root element with the same name as the callback method (see section 3.3.1.3 for a definition of the XML elements).

3.2.1 StartPasientFerdig

This method may be called by the FM when the user closes down the FM after it was started with the StartPasient method. The EPJ can choose to receive a call-back by setting the Callback parameter in the StartPasient element appropriately.

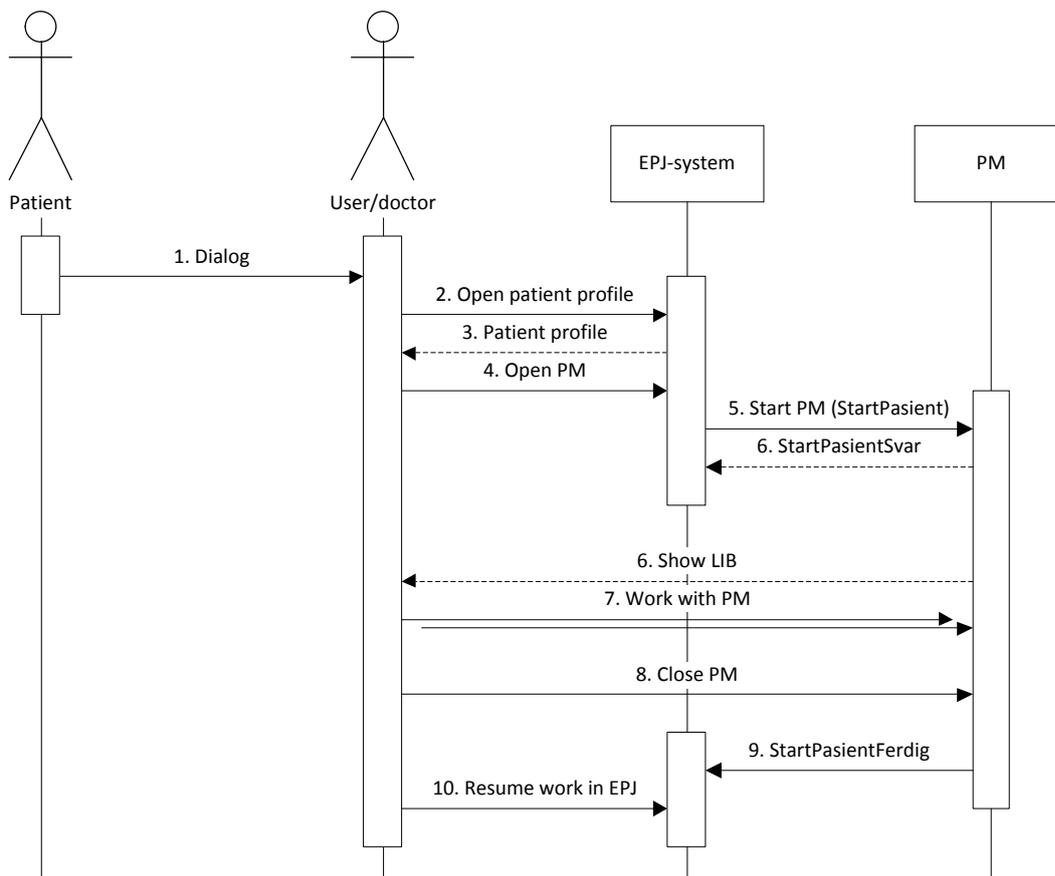


Figure 4 - EPJ receives call-back from FM when the doctor finishes his work in the FM.

Figure 4 shows the case where the EPJ has set the Callback parameter in StartPasient to request a call-back from the FM. When the doctor finishes his work in the FM, the FM makes a call-back using StartPasientFerdig to notify the EPJ that it is done and that the doctor is returning to the EPJ.

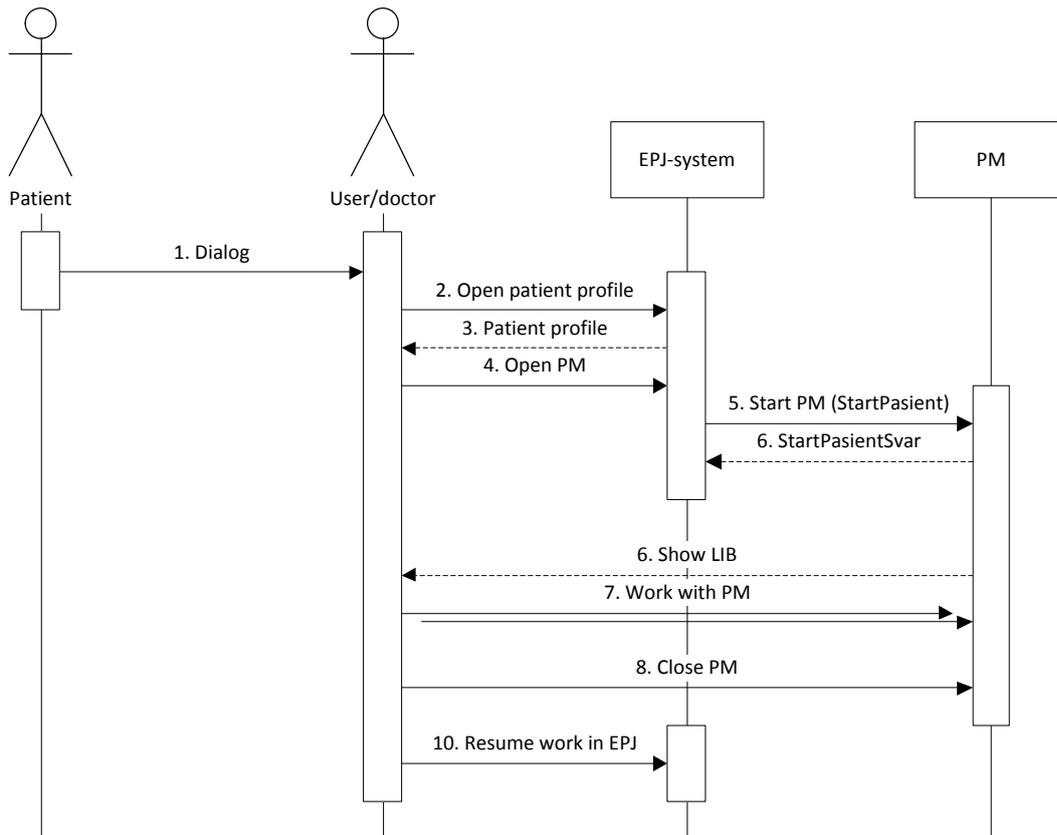


Figure 5 - The doctor finishes his work in the FM but the EPJ gets no call-back.

Figure 5 shows the case where the EPJ has set not the Callback parameter in StartPasient to request a call-back from the FM. When the doctor finishes his work in the FM, the FM simply closes. How the user resumes his work in the EPJ cannot be influenced by the FM in this case.

3.2.2 StartInboxFerdig

This method is called by the FM when the user closes down the FM after it was started with the StartInbox method.

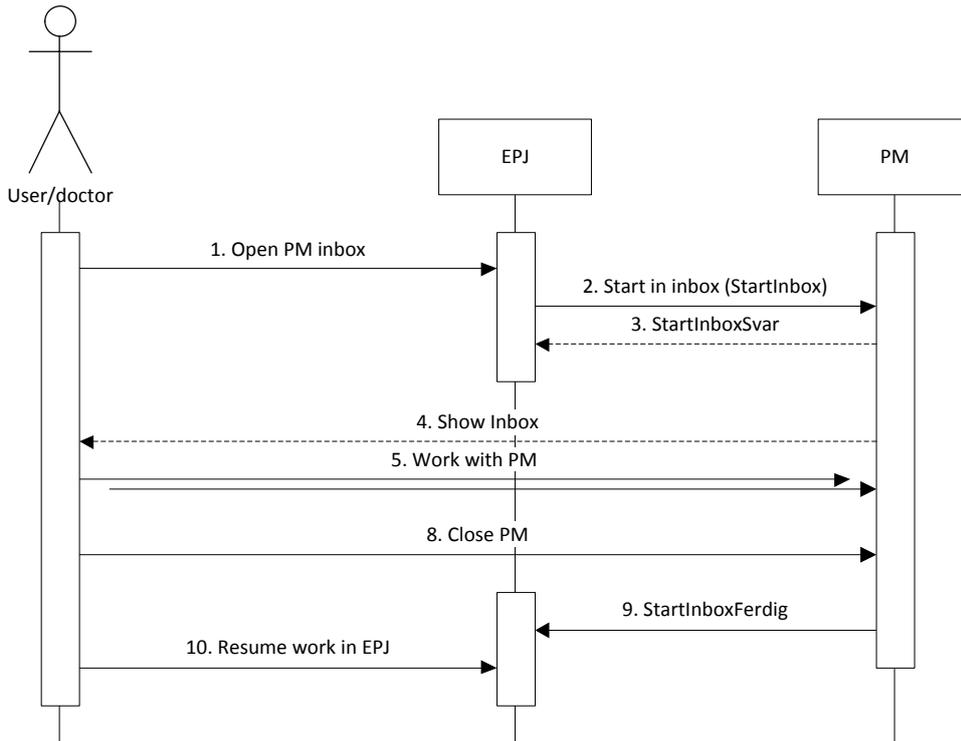


Figure 6 - EPJ receives call-back from FM when the doctor finishes his work in the FM.

Figure 6 shows the case where the EPJ has set the Callback parameter in StartInbox to request a call-back from the FM. When the doctor finishes his work in the FM, the FM makes a call-back using StartInboxFerdig to notify the EPJ that it is done and that the doctor is returning to the EPJ.

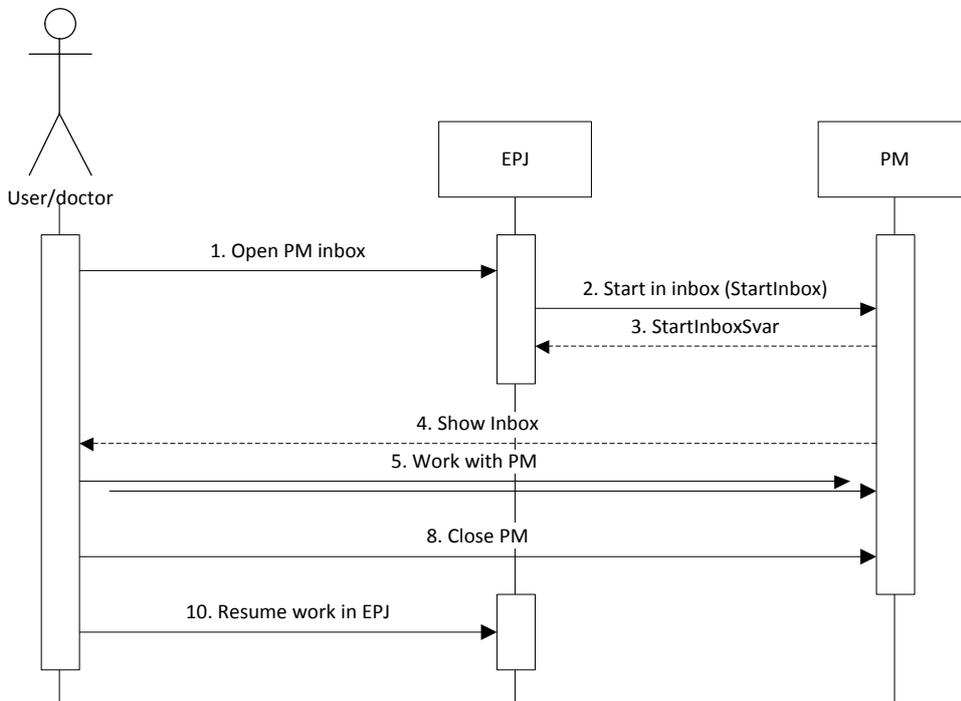


Figure 7 - The doctor finishes his work in the FM but the EPJ gets no call-back.

Figure 7 shows the case where the EPJ has set not the Callback parameter in StartInbox to request a call-back from the FM. When the doctor finishes his work in the FM, the FM simply closes. How the user resumes his work in the EPJ cannot be influenced by the FM in this case.

3.2.3 StartCaveFerdig

This method may be called by the FM when the user closes down the FM after it was started with the StartCave method. The EPJ can choose to receive a call-back by setting the Callback parameter in the StartCave element appropriately.

3.3 Data model

The following sections define the XML types and XML elements that are used by the EPJ API. The XML namespace for the data model is:

<http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2014-05-02>

3.3.1 XML types

3.3.1.1 typeForesporsel

Defines a base type that is extended by all request elements (i.e. elements that define the parameters accepted by the API methods).

XML Element	Description
LoginInfo	Contains information on the user who is logged into the EPJ. This information is used to perform user authorization in the FM.

3.3.1.2 typeSvar

Defines a base type that is extended by all *Svar elements. This includes information that is present in the return XML documents for all EPJ API methods.

XML Element	Description
Returkode	An integer specifying the return code for the method. The following values are defined: 0 – OK. 1 – User not authorized. 2 – System offline. 3 – XML document does not pass validation. 4 – The UI is blocked (e.g. unsaved data). 5 – Patient is not registered in FM. 6 – Patient identifier missing. 7 – Other content error (a missing or invalid field). 8 – UI was closed with possible loss of changes (only returned by Lukk). 9 – UI was closed when there are unsent RF messages (only returned by Lukk). 10 – Existing patient was updated (only returned by SkrivPasient).
Feilmelding	Optional error message (in Norwegian) suitable to display to the user.
InternFeilmelding	Optional internal error message (in English) that is not suitable to display to the user. This should be used by the EPJ vendor to aid in debugging problems with EPJ-FM integration.

3.3.1.3 typeCallback

Defines a base type that is extended by all *Ferdig elements.

XML Element	Description
Brukernavn	The user name that was used when the corresponding Start* method was called.
SessionId	The EPJ session ID that was passed into the corresponding Start* method.

3.3.2 XML elements

3.3.2.1 ATC

This element represents a single ATC entry. It is referenced from the CAVE element.

XML Element	Description
ATCKode	An ATC code on which CAVE has been registered.

3.3.2.2 BrukerInfo

Defines information about a single user of the FM.

XML Element	Description
BrukerNavn	The username of the user.
Status	The status of the user: A = Active U = Inactive
Passord	The password (or a hash of one) of the user.
Role	One or more roles for the user. The defined roles are: 1 - Admin 2 - Doctor 3 - Assistant 4 - Read-only access 5 - Nurse NOTE: A user who has the doctor role cannot have any of the roles assistant, nurse or read-only.
Etternavn	The family name of the user.
Mellomnavn	The middle name of the user.
Fornavn	The first (given) name of the user.
Fodseldato	The birthdate of the user.
Kjonn	The gender of the user (coding system 3101).
Nasjonalitet	The nationality of the user (coding system 9043).
Id	A list of identifiers for the user (coding system 8116).
Adresse	An address for the user.
Telekommunikasjon	A list of information on phone numbers, etc (coding system 9061).
Spesialitet	A list of specialities for a doctor (coding system 7426).
AvdelingResh	The RESH-Id of the default avdeling of the user.

3.3.2.3 CAVE

This element represents a single CAVE entry. It is referenced from the LesCaveSvar and SkrivCave elements.

NOTE: Only one of LegemiddelMerkevare, Virkestoff or ATC should be populated in each record.

XML Element	Description
CAVEId	A unique ID for the record. This should be a globally unique identifier (GUID).
HjelpestoffReaksjon	Indicates if the patient has had an adverse reaction to an inactive ingredient (true) or an active ingredient (false).
GrunnlagForCAVE	Text describing the reaction.
RegistreringsDato	Date of registration.
Signatur	The name of the user who registered the CAVE entry.

Innaktiv	Indicates if the CAVE entry is active (false) or inactive (true).
LegemiddelMerkevare	A medication.
Virkestoff	An active substance.
ATC	An ATC code.
Reaksjon	The type of allergic reaction (from coding system 7497)
Kilde	The source of the CAVE information (from coding system 7498)

3.3.2.4 Forskriver

This optional element is used/referenced in the Resept element and contains information on the responsible doctor. It consists of the doctor's local user name (BrukerNavn) and/or full name (FullNavn) along with his/hers hpr id (HprId). The schema file (epj-api.xsd) describes the BrukerNavn, FullNavn and HprId as optional but the FM depends on the Forskriver element being populated in the following way. If a user name is specified, the FM assumes the prescriber (Forskriver) is a local user and if found sets that user as the prescriber. If not specified, the FM requires the full name of prescriber along with the prescribers' hpr id. The FM will always attempt to match the prescriber with a local user if possible and processes the user name, before the hpr id, when looking up local users, so a locally known user name can override full name and hpr id in the case of mixed up Forskriver elements. The reverse is true when a locally known hpr id is present with an unknown or unspecified user name.

XML Element	Description
BrukerNavn	A conditionally optional string depicting user name of responsible doctor, this implies a local user. Required if HprId and FullNavn are not specified.
FullNavn	A conditionally optional string depicting full name of responsible doctor. Required if BrukerNavn is not specified or if the FM can't find a local user with the specified BrukerNavn or HprId.
HprId	A conditionally optional string depicting hpr id of responsible doctor. Required if BrukerNavn is not specified or if the FM can't find a local user with the specified BrukerNavn.

3.3.2.5 ImportPasientInfo

This element is not used by the EPJ API. It is used when importing patient information from XML files after the system is installed and also when exporting patient information from FM.

XML Element	Description
ImportPasient	One element per imported patient.
ImportPasient/Pasient	Patient information.
ImportPasient/Resept	0 or more Resept elements defining the list of medications (LIB), nutrition support (NIB) and medical disposables (FIB). This is stored when importing and populated when exporting.
ImportPasient/ReseptHistorikk	A Base64 encoded string, specifying freetext prescription history of the patient. This is stored when importing patient information and populated when exporting.
ImportPasient/CAVE	0 or more CAVE elements defining the patient's CAVE information. This is stored when importing and populated when exporting.
ImportPasient/StrukturertReseptHistorikk	0 or more elements defining the patient's prescription history of medications (LIB), nutrition support (NIB) and medical disposables (FIB). This is not used when importing, but is populated when exporting.

ImportPasient/HelfoSoknad	0 or more elements defining the HELFO applications. This is not used when importing, but is populated when exporting.
ImportPasient/SamtykkeM96	Boolean value that indicates if the patient has given permission for this installation to query RF for prescriptions from other doctors. This is not used when importing, but is populated when exporting.

3.3.2.6 Innleggelse

This element represents information about the patient's current admission. This is an optional element that can be used in StartCave and StartPasient EPJ API methods.

XML Element	Description
Id	A globally unique identifier for the admission.
InleggelsesTidspunkt	The date and time of the patient's admission time.

3.3.2.7 LegemiddelMerkevare

This element represents a single medication entry. It is referenced from the CAVE element.

XML Element	Description
LegemiddelId	The id of the medication (class LegemiddelMerkevare in FEST).
LegemiddelNavn	The name of the medication.
ATCKode	The ATC code of the medication.
Virkestoff	One entry per active substance in the medication (can be left empty when used in the SkrivCave method).

3.3.2.8 LesAntallMeldingerForSignering

Defines the parameter for the LesAntallMeldingerForSignering EPJ API method. This element extends typeForesporsel and adds no further elements.

3.3.2.9 LesAntallMeldingerForSigneringSvar

Defines the return value of the LesAntallMeldingerForSignering EPJ API method. This element extends typeSvar.

XML Element	Description
AntallM1	The number of M1 messages waiting to the signed and/or sent.
AntallM5	The number of M5 messages waiting to the signed and/or sent.

3.3.2.10 LesBrukerInfoSvar

This element is not used in the FM EPJ COM API, LesBrukerInfoSvar defines the return value of the LesBrukerInfo web service method on the FM back end. This element extends typeSvar.

XML Element	Description
BrukerInfo	Information about a single FM user.

3.3.2.11 LesCave

Defines the information needed as parameter to the LesCave EPJ API method. This element extends typeForesporsel.

XML Element	Description
Pasient	The patient for whom to read CAVE information.

3.3.2.12 LesCaveSvar

Defines the return value of the LesCave EPJ API method. This element extends typeSvar.

XML Element	Description
CAVE	A list of CAVE elements specifying the active CAVE information for the patient.

3.3.2.13 LesPasientStatus

Defines the information needed as parameter to the LesPasientStatus EPJ API method. This element extends typeForesporsel.

XML Element	Description
PasientXxxId	A list of XXX Ids for the patients that status should be fetched for..

3.3.2.14 LesPasientStatusSvar

Defines the return value of the LesPasientStatus EPJ API method. This element extends typeSvar.

XML Element	Description
PasientStatus	A list of elements specifying patient status information for each patient.
PasientStatus /PasientXxxId	The XXX Id of the patient.
PasientStatus /UkjentPasientXxxId	A patient with this XXX id was not found in FM.
PasientStatus /SisteOppdatering	The date/time of the last update, the same value as SisteOppdateringMax returned by LesSisteVibCaveOppdatering.
PasientStatus /KreverOppmerksomhet	A Boolean value indicating if the patient needs attention in the FM. This will be True if any of following conditions are fulfilled: <ul style="list-style-type: none"> ○ M25.1 received and has not been fully processed, the user will end up in samstemming when he opens the patient. ○ M25.3 has updated the delivery type of a prescription and that change has not been confirmed by a user. ○ Prescriptions in Andre forskrivninger ○ Patient has draft prescription(s) ○ Patient has draft seponeringer ○ Patient has messages in signing queue ○ A negative M12 message has been received since the patient was last opened ○ A negative M15 message has been received since the patient was last opened

3.3.2.15 LesSisteVibCaveOppdatering

Defines the parameter for the LesSisteVibCaveOppdatering EPJ API method. This element extends typeForesporsel.

XML Element	Description
Pasient	The patient for whom to read the date and time for the last VIB/CAVE updates.

3.3.2.16 LesSisteVibCaveOppdateringSvar

Defines the return value of the LesSisteVibCaveOppdatering EPJ API method. This element extends typeSvar.

XML Element	Description
SisteVibOppdatering	The point in time when the VIB was last updated for the patient.

SisteCaveOppdatering	The point in time when CAVE was last updated for the patient.
SisteTakstOppdatering	The registration date and time of the latest takst record of the patient.
SisteOppdateringMax	The maximum value of the three datetimes above.

3.3.2.17 LesTakst

Defines the information needed as parameter to the LesTakst EPJ API method. This element extends typeForesporsel.

XML Element	Description
Pasient	The patient for whom to read rate information.
Dato	The date for which to read rate information.

3.3.2.18 LesTakstSvar

Defines the return value of the LesTakst EPJ API method. This element extends typeSvar.

XML Element	Description
Takst	One element for each rate record being returned.
Takst/TakstNavn	The name of the rate record being returned.
Takst/TakstId	The identifier for the rate record being returned.
Takst/Tidspunkt	A timestamp indicating when the rate record was recorded.

3.3.2.19 LesVarerIBruk

Defines the information needed as parameter to the LesVarerIBruk EPJ API method. This element extends typeForesporsel.

XML Element	Description
Pasient	The patient for whom to read VIB.

3.3.2.20 LesVarerIBrukSvar

Defines the return value of the LesVarerIBruk EPJ API method. This element extends typeSvar.

XML Element	Description
Resept	One entry for each item being returned.
SisteM25Id	The Id of the last sent M25 message for the patient.

3.3.2.21 LesVarslinger

Defines the information needed as parameter to the LesVarslinger EPJ API method. This element extends typeForesporsel and adds no further elements.

3.3.2.22 LesVarslingerSvar

Defines the return value of the LesVarslinger EPJ API method. This element extends typeSvar.

XML Element	Description
Varsling	One entry for each item being returned.

3.3.2.23 LoginInfo

This element is used to provide information on the EPJ user to the FM so that the user can be authorized.

XML Element	Description
-------------	-------------

BrukerNavn	The username of the user.
Passord	The password (or the password hash) of the user. Note that this does not need to be an actual password. This simply needs to match the password provided to the SkrivBrukerInfo EPJ API method.

3.3.2.24 Lukk

Defines the information needed as parameter to the Lukk EPJ API method. This element extends typeForespørsel and adds no further elements.

3.3.2.25 LukkSvar

Defines the return value of the Lukk EPJ API method. This element extends typeSvar and adds no further elements.

3.3.2.26 Organisasjon

This element contains information about the current organizational context when the FM is started. This is an optional element that can be used in the StartCave, StartPasient and StartInbox EPJ API methods.

Note that this element is required in the Start... methods when more than one organization has been specified in the FM database.

XML Element	Description
Institusjon	This element contains a single Organisation object that specifies the current institution context. This element is mandatory.
Avdeling	This element contains a single Organisation object that specifies the current avdeling within the institution specified by the previous element. This element is optional.
Post	This element contains a single Organisation object that specifies the current post within the institution and avdeling specified by the previous elements. This element is optional.
Distrikt	This element contains a single Organisation object that specifies the Distrikt within the institution specified by the Institusjon element. This element is optional
Sone	String describing the zone associated with the Distrikt specified above. This element is optional.
Delsone	String describing the subsone associated with the Sone specified above. This element is optional.

3.3.2.27 Pasient

XML Element	Description
Etternavn	The family name of the patient.
Mellomnavn	The middle name of the patient.
Fornavn	The first (given) name of the patient.
Fodseldato	The birthdate of the patient.
Kjonn	The gender of the patient (coding system 3101).
Nasjonalitet	The nationality of the patient (coding system 9043).
Id	A list of identifiers for the patient (coding system 8116). The ID should always contain exactly one ID of the type "XXX" (Annet) (from 8116). This ID should contain a unique, unchanging identifier for the patient. An EPJ can e.g. use its own internal patient identifier to populate this identifier. This is the

	<p>identifier used to look up the corresponding patient in the FM (this is not done on the fødselsnummer).</p> <p>If the FM receives a Pasient element without an ID of type "XXX" it will return an error code 6 (patient identifier missing).</p> <p>If the FM receives a Pasient element where the "XXX" ID contains an unknown code, a new patient record will be created in the FM in some cases but in other cases an error will be returned (see section 3.1).</p> <p>If the FM receives a Pasient element where the "XXX" ID contains a previously known code, the patient information is updated to reflect what was passed in from the EPJ.</p>
Adresse	An address for the patient.
Telekommunikasjon	A list of information on phone numbers, etc.
Frikort	A boolean that indicates that the patient has a frikort.
Utlending	Contains information specific to foreign patients. Taken from the M1 KITH specification.
MedisinskInfo/Maling	A list of measurements relating to the patient (e.g. INR).
MedisinskInfo/Maling/Type	The type of measurement (currently only INR is supported).
MedisinskInfo/Maling/Verdi	The measurement value.
MedisinskInfo/Maling/Tidsstempel	Timestamp of the measurement.
AnsvarligLege	Information about either fastlege or LIB-ansvarlig lege.
AnsvarligLege/Hprld	The HPR-id of the fastlege or LIB-ansvarlig lege.
AnsvarligLege/Fornavn	The first name of the fastlege or LIB-ansvarlig lege.
AnsvarligLege/Etternavn	The last name of the fastlege or LIB-ansvarlig lege.
AnsvarligLege/Spesialitet	A medical speciality of the fastlege or LIB-ansvarlig lege (only one can be given).
AnsvarligLege/Rolle	Code from kodeverk 7490, only the values for Fastlege or Multidoseansvarlig (LIB-ansvarlig) should be used.

3.3.2.28 PasientListe

An unbounded sequence of pasient elements expressed in one or two elements

XML Element	Description
Pasient	Required element representing a single patient
Pasient/PasientXxxId	Required element expressed as a string representing a patient's xxx id.
Pasient/Organisasjon	Optional element describing organization patient is associated with.

3.3.2.29 Resept

Defines a single medication (LIB), nutrition support (NIB) or medical disposable (FIB) prescription. This element is used by LesVarerIBrukSvar and ImportPasientInfo

XML Element	Description
Id	A FM specific identifier for the entry being returned.
VibStatus	Indicates the current VIB status of the prescription i.e. if it is currently in the VIB, part of VIB history or has not yet been imported to VIB. This attribute can take the values "VIB", "Historikk" and "Import". This property is ignored in incoming data.
ReseptDokLegemiddel	A LIB record.

ReseptDokHandelsvare	A NIB or FIB record.
AnsvarligLege	The doctor responsible for the prescription. The FM populates this in outgoing data but ignores this field in incoming data.
InstituertAv	The person who started the treatment. This property is ignored in incoming data.
InstitueringsDato	The date when the treatment was started. Note that this is not the issue date of the prescription but rather the start of the treatment with the prescribed medication (many prescriptions may have been issues since).
ForrigeReseptId	The ID of the previous prescription when the current prescription was created by renewing an older prescription. The FM populates this in outgoing data but ignores this field in incoming data.
SeponeringsInformasjon	Information about the stop (seponering) of a prescription.
RegistreringsType	The type of prescription (eRp/uRp, eRp, uRp, fRp, tRp, iRp, Reg). The FM populates this in outgoing data but ignores this field in incoming data.
Kortdose	The kortdose that was used when the prescription was created. The FM populates this in outgoing data but ignores this field in incoming data.
ForskrivningsDato	The issue date of the prescription.
Varenavn	The varenavn of the prescribed medication. This is set as follows for specific types of medications: <ul style="list-style-type: none"> - LegemiddelMerkevare – LegemiddelMerkevare.Varenavn. - Legemiddelpakning – LegemiddelMerkevare.Varenavn of all referenced LegemiddelMerkevare instances are concatenated. - LegemiddelVirkestoff – Virkestoff.Navn of all references Virkestoff instances are concatenated. - Legemiddelblanding – Legemiddelblanding.Navn. The FM populates this in outgoing data but ignores this field in incoming data.
Styrke	The strength value(s) of the prescribed medication. The FM populates this in outgoing data but ignores this field in incoming data.
Legemiddelform	The galenic form(s) of the prescribed medication. The FM populates this in outgoing data but ignores this field in incoming data.
Kladd	True for draft prescriptions, not included for other prescriptions. The FM populates this in outgoing data but ignores this field in incoming data.
RegistrertAv	Username of the user who registered the prescription.
SignertAv	Username of the user who approved the prescription. This may be a doctor or a nurse in case of PLO.
Forskriver	Prescriber information, fullname along with user name and/or hpr id of responsible doctor.
InteraksjonsInformasjon	Information on the interaction of this medicine with other medicines in the list. FM populates this in outgoing data but ignores this field in incoming data.
InteraksjonsInformasjon /InteraksjonsNiva	The type (seriousness) of the interaction (from coding system 7483, with the addition of code V=0, DN="Interaksjon ikke vurdert").
InteraksjonsInformasjon	Textual description of the interaction.

/Kommentar	
VarselSlv	A list of warnings from SLV. FM populates this in outgoing data but ignores this field in incoming data.
Libld	The internal Id of the prescription. FM populates this in outgoing data but ignores this field in incoming data.
BegrunnelseReservasjon-GeneriskByte	If generic substitution should not be allowed for the prescription, this field will contain a comment on why it should not be allowed. FM populates this in outgoing data but ignores this field in incoming data.
Administrering	The delivery mode of the prescription. Allowed values are: <ul style="list-style-type: none"> • D – Dosett • M – Multidose • L – Neither dosett nor multidose, drugs managed by PLO • A – Neither dosett nor multidose, drugs managed by the patient

Note that the element will include ReseptDokLegemiddel or ReseptDokHandelsvare, never both. Both elements are simple containers for base64 encoded strings of ReseptDokLegemiddel and ReseptDokHandelsvare as defined in an m 1 (<http://www.kith.no/xmlstds/eressept/m1/2013-10-08>).

3.3.2.30 ReseptHistorikk

This element represents an unstructured “resept historikk” entry, a single simple content element comprised of text/html.

3.3.2.31 SeponeringsInfomasjon

This element represents information about the stoppage (seponering) of a prescription. This is an optional element in the Resept class.

XML Element	Description
SeponertAv	The name of the person who stopped the prescription.
Seponeringsdato	The date when the prescription stopped.
Seponeringsgrunn	The reason given for stopping the prescription.
Kladd	True when a draft prescription was stopped, not included for other prescriptions

3.3.2.32 SkrivBrukerInfo

Defines the information needed as parameter to the SkrivBrukerInfo EPJ API method. This element extends typeForesporsel.

XML Element	Description
BrukerInfo	One element for each user being stored.

3.3.2.33 SkrivBrukerInfoSvar

Defines the return value of the SkrivBrukerInfo EPJ API method. This element extends typeSvar and adds no further elements.

3.3.2.34 SkrivCave

Defines the information needed as parameter to the SkrivCave EPJ API method. This element extends typeForesporsel.

XML Element	Description
-------------	-------------

Pasient	Specifies the patient for whom to save CAVE information.
CAVE	A list of CAVE elements defining the active CAVE information for the patient.
ErstatteAlle	This is an optional element that, when set to True, causes the patient's CAVE record to be completely replaced by the contents of the CAVE parameter. After a call to SkrivCave the patient will be left with only the CAVE registrations in the CAVE parameter when ErstatteCave is True.

3.3.2.35 SkrivCaveSvar

Defines the return value of the SkrivCave EPJ API method. This element extends typeSvar and adds no further elements.

3.3.2.36 SkrivKorrespondanseInfo

Defines the information needed as parameter to the SkrivKorrespondanseInfo EPJ API method. This element extends typeForesporsel.

XML Element	Description
Pasient	The patient for whom to define current medications (LIB), nutrition support (NIB) and medical disposables (FIB).
Resept	A list of Resept elements defining the medications (LIB), nutrition support (NIB) and medical disposables (FIB) for the user.

3.3.2.37 SkrivKorrespondanseInfoSvar

Defines the return value of the SkrivKorrespondanseInfo EPJ API method. This element extends typeSvar and adds no further elements.

3.3.2.38 SkrivPasient

Defines the information needed as parameter to the SkrivPasient EPJ API method. This element extends typeForesporsel.

XML Element	Description
Pasient	The patient that is to be created by the FM.
Resept	0 or more Resept elements defining the list of medications (LIB), nutrition support (NIB) and medical disposables (FIB).
ReseptHistorikk	A Base64 encoded string, specifying freetext prescription history of the patient.
CAVE	0 or more CAVE elements defining the patient's CAVE information.

3.3.2.39 SkrivPasientSvar

Defines the return value of the SkrivPasient EPJ API method. This element extends typeSvar and adds no further elements.

3.3.2.40 SkrivUt

Defines the information needed as parameter to the SkrivUt EPJ API method. This element extends typeForesporsel.

XML Element	Description
Utskrift/Dosett	Parameters for printing the Dosett report.
Utskrift/Dosett/Pasienter	A list of XXX ids for the patients to print the report for.
Utskrift/Dosett/DosettType	The type of Dosett report to print.
Utskrift/Dosett/MedSeponert	If true, seponated prescriptions are included in the report.

Utskrift/Dosett/KunDosettLegemidler	If true, only prescriptions with dosett delivery are included.
Utskrift/Dosett/ForsteDoseringsDato	The start date of the report.
Utskrift/AkJournal	Parameters for printing the AkJournal report.
Utskrift/AkJournal/Pasienter	A list of XXX ids for the patients to print the report for.
Utskrift/AkJournal/ MedHistoriskeDoseringer	If true, the report will include the prescription history of the AK journal. Otherwise, only the current dosing will be printed.
Utskrift/AkJournalListe	Parameters for printing the AkJournalListe report.
Utskrift/AkJournalListe/Pasienter	A list of XXX ids for the patients to print the report for.
Utskrift/AkJournalListe/ListeType	The type of AK report: - IkkeSignert: the report will contain a list of all patients that have draft AK prescriptions. - NesteKontroll: the report will contain a list of all patients where the next INR control date is between the dates specified in NestelNrKontroll/Start and NestelNrKontroll/Slutt.
Utskrift/AkJournalListe/Sortering	Specifies the sort order of the report
Utskrift/AkJournalListe/NestelNrKontroll/Start Utskrift/AkJournalListe/NestelNrKontroll/Slutt	The start and end dates for next INR control date. Only used if ListeType is NesteKontroll
Utskrift/Vib	Parameters for printing the Vib report.
Utskrift/Vib/Pasienter	A list of XXX ids for the patients to print the report for.
Utskrift/Vib/TaMedNibFib	If true, NIB and FIB prescriptions will be included, otherwise only LIB prescriptions are printed.
Organisasjon	Optional element containing information about the organization (e.g. HF), avdeling and post context for the patient. Note that this field is semantically required if more than one organization has been configured in the FM database.
BrukStandardPrinter	A Boolean value indicating if the user's default printer should be used. If omitted or set to false or if the user does not have a defined standard printer, the user can select which printer to use.

3.3.2.41 SkrivUtSvar

Defines the return value of the SkrivUt EPJ API method. This element extends typeSvar and adds no further elements.

3.3.2.42 SlaSammenPasienter

Defines the information needed as parameter to the SlaSammenPasienter EPJ API method. This element extends typeForesporsel.

XML Element	Description
Primaer/Pasient	The patient who should continue to be used after the merge, i.e. the primary patient in the merge.. The VIB, CAVE, etc from the secondary patient will be moved over to the primary patient.
Sekundaer/Pasient	The patient who's file should be merged into the primary patient file, i.e. the secondary patient in the merge. This is the patient who should not continue to be used after the merge.
AnsvarligLegeHprld	A string representing the hpr id of responsible doctor

3.3.2.43 SlaSammenPasienterSvar

Defines the return value of the SlaSammenPasienter EPJ API method. This element extends typeSvar and adds no further elements.

3.3.2.44 StartCave

Defines the information needed as parameter to the StartCave EPJ API method. This element extends typeForesporsel.

XML Element	Description
Pasient	The patient that is to be selected by the FM.
CAVEId	The ID of a CAVE record to be focused when the FM opens (optional).
Organisasjon	Optional element containing information about the organization (e.g. HF), avdeling and post context for the patient. Note that this field is semantically required if more than one organization has been configured in the FM database.
Innleggelse	Optional element containing admission information for the patient
Callback	Indicates if the EPJ wants a call-back when the user closes down the FM.
SessionId	An ID that the EPJ wants reported back in the StartCaveFerdig callback.

3.3.2.45 StartCaveFerdig

Defines the parameter value used for the StartCaveFerdig method on the eResept.ForskrivningsmodulCallback.3 callback interface.

XML Element	Description
Brukernavn	The user name of the user who was authenticated for the StartCave call that lead to the callback.

3.3.2.46 StartCaveSvar

Defines the return value of the StartPasient EPJ API method. This element extends typeSvar and adds no further elements.

3.3.2.47 StartInbox

Defines the information needed as parameter to the StartInbox EPJ API method. This element extends typeForesporsel.

XML Element	Description
VarslingId	The ID of the notification that should be focused/selected when the Inbox opens. This is an optional parameter.
Organisasjon	Optional element containing information about the organization (e.g. HF), avdeling and post context. Note that this field is semantically required if more than one organization has been configured in the FM database.
Callback	Indicates if the EPJ wants a call-back when the user closes down the FM.
SessionId	An ID that the EPJ wants reported back in the StartInboxFerdig callback.

3.3.2.48 StartInboxFerdig

Defines the parameter value used for the StartInboxFerdig method on the eResept.ForskrivningsmodulCallback.3 callback interface.

XML Element	Description
Brukernavn	The user name of the user who was authenticated for the StartCave call that lead to the callback.
PasientXxxIds	A wrapper element for multiple PasientXxxId elements.

PasientXxxIds/PasientXxxId The XXX identifiers of the users who were opened from the FM Inbox.

3.3.2.49 StartInboxSvar

Defines the return value of the StartInbox EPJ API method. This element extends typeSvar and adds no further elements.

3.3.2.50 StartPasient

Defines the information needed as parameter to the StartPasient EPJ API method. This element extends typeForesporsel.

XML Element	Description
Pasient	The patient that is to be selected by the FM.
AnsvarligLege	The user name of the doctor who is responsible for the patient. This doctor will get an inbox notification of new prescriptions etc. when an assistant or nurse runs StartPasient. This element is ignored when a doctor runs StartPasient.
Organisasjon	Optional element containing information about the organization (e.g. HF), avdeling and post context for the patient. Note that this field is semantically required if more than one organization has been configured in the FM database.
Innleggelse	Optional element containing admission information for the patient
Samstillingstekst	Free text containing medication information that will be processed using the Vivit samstillingsmodul when the FM opens and matched with the current LIB.
Callback	Indicates if the EPJ wants a call-back when the user closes down the FM.
SessionId	An ID that the EPJ wants reported back in the StartPasientFerdig callback.

3.3.2.51 StartPasientFerdig

Defines the parameter value used for the StartPasientFerdig method on the eResept.ForskrivningsmodulCallback.3 callback interface.

XML Element	Description
Brukernavn	The user name of the user who was authenticated for the StartPasient call that lead to the callback.

3.3.2.52 StartPasientSvar

Defines the return value of the StartPasient EPJ API method. This element extends typeSvar and adds no further elements.

3.3.2.53 Varsling

This element represents a single notification (varsling) entry. It is referenced from the LesVarslingSvar element.

XML Element	Description
VarslingId	A string depicting unique ID for the notification.
Pasient	The patient to whom the notification relates (left out for non-patient related notifications).
Melding	A message or subject describing the nature of the notification.
Avsender	The sender of the message that triggered the notification (left out for non-message related notifications).

3.3.2.54 Virkestoff

This element represents a single active substance (virkestoff) entry. It is referenced from the CAVE element.

XML Element	Description
VirkestoffId	The ID of the active substance (class Virkestoff in FEST).
VirkestoffNavn	The name of the substance.

3.3.3 XSD

The EPJ API data model is defined in section 6.

3.4 API technology

Both `eResept.Forskrivningsmodul.1` and `eResept.ForskrivningsmodulCallback.2/3` are defined as COM interfaces. The FM is implemented as an out-of-process COM server exposing the `eResept.Forskrivningsmodul.1` interface. If an EPJ wants to get call-backs, it should implement an out-of-process COM server exposing the `eResept.Forskrivningsmodul.1` interface. If COM technology is not suitable for the EPJ vendor, the API can be wrapped and accessed through other means.

The following guidelines apply to the usage of the `eResept.Forskrivningsmodul.1` COM interface:

- The EPJ should create a reference to the COM object once, and hold on to it for the lifetime of the EPJ process.
- If the reference becomes invalid (happens if the FM is shut down), the EPJ should create a new reference and hold onto that instead.
- All method parameters and return values are strings containing XML documents.
- All methods have the same signature - `string MethodCall(string parameter)`.
- To get a reference to the COM object the following ProgId is used: `eResept.Forskrivningsmodul.1`

Experimenting with the COM object is relatively simple using e.g. Microsoft PowerShell.

The following guidelines apply when implementing the `eResept.ForskrivningsmodulCallback.2/3` COM interface:

- The FM will create a reference to the COM object once, and hold on to it for the lifetime of the FM process.
- If the reference becomes invalid (happens if the EPJ is shut down), the FM will create a new reference and hold onto that instead.
- To get a reference to the COM object the FM will use the following ProgId: `eResept.ForskrivningsmodulCallback.3`, if this ProgId is not available then the FM tries to get a reference using `eResept.ForskrivningsmodulCallback.2`. This means that the FM will always use `eResept.ForskrivningsmodulCallback.3` when available and fall back to `eResept.ForskrivningsmodulCallback.2` if `eResept.ForskrivningsmodulCallback.3` is not available.
- All of the methods on the `eResept.ForskrivningsmodulCallback.2/3` interface should return immediately.

3.5 XML import

The FM supports importing a list of patients and their list of medications (LIB), nutrition support (NIB) and medical disposables (FIB). This can be done via the FM system administration UI, and should be done before the system is put into use. The import is done by reading one or more XML files that have

ImportPatientInfo XML elements as the root element. Each patient should be represented in only one file.

After the initial file based XML import any updates to patient information should go through the EPJ API.

Note that importing of user information using XML files is not supported. User information should be imported through the EPJ API (SkrivBrukerInfo).

4 Technical specification

4.1 System structure

This section defines the structure of the FM components. The components are divided into two groups:

- **Primary components** - Components that are used during the running of the system.
- **Secondary components** - Components that are used during installation and initial configuration.

4.1.1 Primary components

The FM has the following primary components:

- FM Client application - This is the client application that is installed on the workstations and is used directly by the users of the system. The client application can run in two modes:
 - User UI - This is the UI used to manage prescriptions. This is always invoked from the EPJ through the EPJ-API and handles all the core functionality of FM.
 - Admin UI - This is the UI used to configure system parameters. This is invoked by the user from the start menu and requires a separate login (with administrator privileges).
 - The Admin UI is also used for importing initial patient data into the system after install.
 - The application can run in both modes at the same time, and they will not interfere with each other.
- FM Application server - This is a background service that runs on a server machine. The application server is responsible for the following activities:
 - Provide data access services to the FM Client application.
 - Manage synchronous messaging (web services).
 - Manage asynchronous messaging - This includes sending/receiving AppRec messages, signing messages with the organizational certificate, etc. Asynchronous messages are exchanged with the messaging brokers through two file system folders, one for incoming messages and one for outgoing messages.
- Database - This provides a persistent store for all FM data, including system configuration.

4.1.2 Secondary components

In addition to the primary components, the FM has the following secondary components:

- FM Client installer - An installation application that installs the FM Client application on the user's workstation.
- FM Application server installer - An installation application that installs the FM Application server on the server machine.
- FM Application server configuration wizard - An application that is installed along with the FM Application server and provides the following functionality:
 - Installs SQL Server 2008 R2 Express if required by the customer.
 - Installs the FM database schema on the database.
 - Configures FM Application server so that it can access the database.
 - Creates an initial "super user" that can log into the Admin UI. Further configuration is done in the Admin UI
- EPJ API Test client - A simple client that implements the EPJ API without needing access to the rest of the FM components. This client can be used by EPJ vendors when developing against the EPJ API.

4.2 Technology

The FM is implemented using the following technologies:

- Microsoft .NET Framework 4.0 - All of the FM components are implemented using this framework.
- Windows Presentation Foundation (WPF) - All UI components are built using WPF.
- Windows Communication Foundation (WCF) - This is used for the following:
 - FM Client application to FM Application server communication - WCF is used to expose a private set of services from the FM Application server that are used by the FM Client application. The services use a binary TCP transport with transport level security.
 - The services are exposed on a configurable IP port on the application server. The servers IP address and port is then the only configuration that is needed for the FM Client application to access the FM Application server.
 - Web service communication - WCF is used for communicating with the RF and FEST.
- Windows Event Log - Used by the FM Application server for error reporting.
- Microsoft SQL Server 2008 R2 Express - This is a free database server that FM uses to persist data. FM also supports using an existing SQL Server database if that is available. SQL Server 2008 R2 Express has the following limitations:
 - Uses only 1 CPU for processing.
 - Utilizes a maximum of 1GB memory.
 - Maximum database size is 10GB.

5 Technical requirements

Below are the initial technical requirements that will form the basis for the installation checklist.

5.1 Workstation requirements

The following requirements apply to the workstations intended to run the FM client application:

	Requirement
Operating system	The FM client application requires one of the following operating system: Microsoft Windows XP SP3 Microsoft Windows Vista SP2 + Platform Update Microsoft Windows 7 - Recommended
Frameworks	The FM client application requires Microsoft .NET Framework 4.0. NOTE: The installer will install the framework if it is not already installed.
CPU	The FM client application requires a modern CPU (Intel Core2Duo or similar).
Memory	The FM client application requires approximately 1GB of free memory.
Disk	The FM client application requires approximately 100MB of free disk space. NOTE: In addition the .NET Framework 4.0 requires 850 MB (32 bit) or 2 GB (64 bit) of free disk space.
Monitor	The FM client application requires that monitors have a resolution of at least 1024x768 pixels with a 16 bit color depth.
Other	Administration privileges are required to install the FM client application.

In addition to the listed requirements, the operating systems may have hardware requirements that are outside the scope of this document.

5.2 Application server requirements

The following requirements apply to the server used to run the FM application server:

	Requirement
Operating system	The FM application server requires one of the following operating systems: Microsoft Windows Server 2003 SP2 (not recommended for new installations) Microsoft Windows Server 2003 R2 SP2 (not recommended for new installations) Microsoft Windows Server 2008 SP2 (not recommended for new installations) Microsoft Windows Server 2008 R2 Microsoft Windows Server 2012 R2
Frameworks	The FM application server requires Microsoft .NET Framework 4.0. NOTE: The installer will install the framework if it is not already installed.
CPU	The FM application server requires a modern server class CPU (Intel Xeon or similar).
Memory	The FM application server requires approximately 1GB of free memory.
Disk	The FM application server requires approximately 100MB of free disk space. NOTE: In addition the .NET Framework 4.0 requires 850 MB (32 bit) or 2 GB (64 bit) of free disk space.
Other	Administration privileges are required to install the FM application server.

In addition to the listed requirements, the operating systems may have hardware requirements that are outside the scope of this document.

5.3 Database server requirements

FM currently supports the following versions of Microsoft SQL Server:

- SQL Server 2005 (not recommended for new installations)
- SQL Server 2008 R2
- SQL Server 2012

Note that SQL Server 2014 is currently not supported.

Microsoft defines hardware and software requirements for the database server in the following document: <http://msdn.microsoft.com/en-us/library/ms143506.aspx>. In addition to this FM will require disk space for the database. The initial size of the database is expected to be around 200MB and the database will gradually grow from there.

Note that the Express edition of SQL Server never uses more than 1GB memory, and can therefore easily be installed on the same server as the FM application server. Such an installation will require that 2GB of memory are free.

5.4 Other technical requirements

FM requires robust communication between the client application and the application server. A 100Mbps local area network is the minimum requirement for this communication.

6 Appendix A: EPJ API XSD

See the file epj-api.xsd, distributed along with this document.

7 Appendix B: EPJ API Test Application

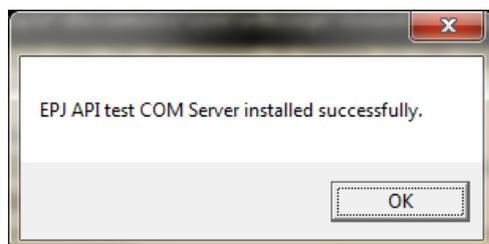
Thula has made available an EPJ API test application that can be used by EPJ vendors to test their technical implementation of the API. The EPJ API test application implements the same COM interface as the forskrivningsmodul but does not require any infrastructure (such as a database) in order to run.

7.1 Registering the application

Before the COM interface can be accessed, it needs to be registered. This is done by running the application executable with the parameter -Install:

```
FM.Client.EpjApiTestApp.exe -Install
```

This will register the COM component and should result in a confirmation dialog stating that the registration was successful:



Note that the registration needs to be run with administrator privileges.

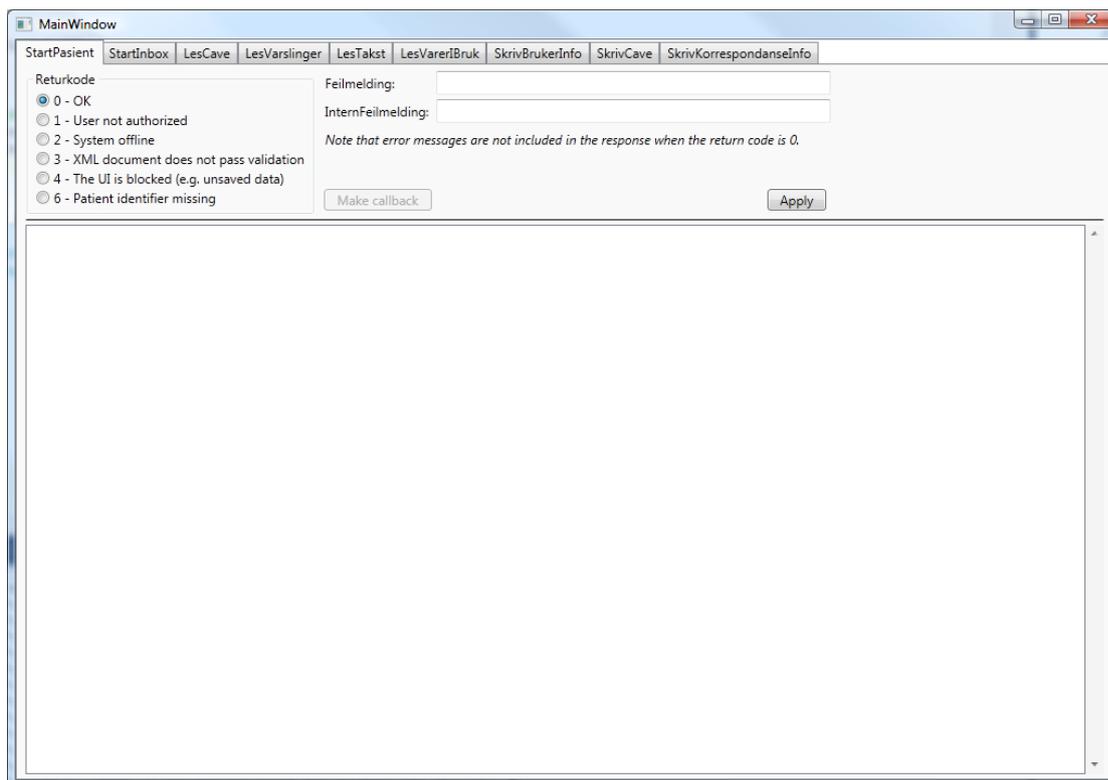
7.2 Accessing the COM interface

The COM interface exposed by the test application (eResept.Forskrivningsmodul.1) can be accessed from any COM compatible environment. What follows are examples in Windows PowerShell.

Once the COM component has been registered as described in section 7.1, start Windows PowerShell and execute the following to start an instance of FM.Client.EpjApiTestApp.exe and place a reference to it in the variable \$a:

```
$a = New-object -comObject eResept.Forskrivningsmodul.1 -strict
```

When the line above has been executed, the FM.Client.EpjApiTestApp.exe user interface should pop up on the screen:



The FM.Client.EpjApiTestApp.exe user interface contains one tab for each method call included in the eResept.Forskrivningsmodul.1 COM interface. On each tab the return code that the method should return can be specified and both an error message and an internal error message can be entered. Clicking the Apply button sets the return values of the method corresponding to the active tab.

7.3 Calling methods on the COM interface

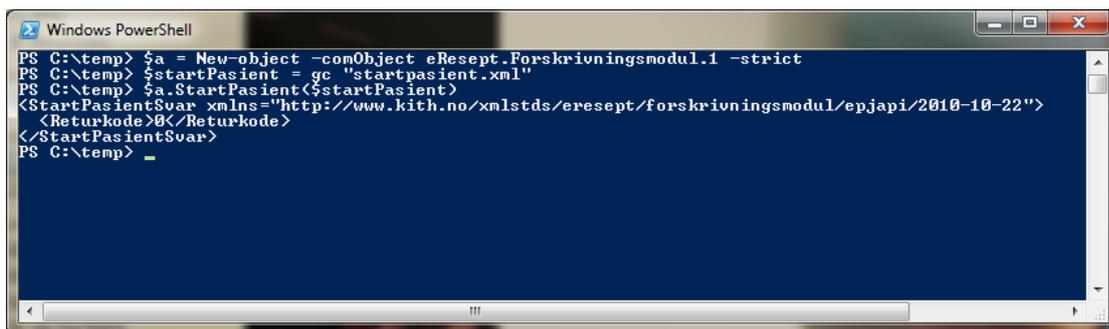
When FM.Client.EpjApiTestApp.exe has been registered and started as described in sections 7.1 and 7.2 the methods included in the eResept.Forskrivningsmodul.1 interface can be called. In the following Windows PowerShell examples this consists of loading an XML file containing method parameter values into a variable and then calling a method, passing in the variable as a parameter. Example parameter XML files are included in section **Villa! Uppruni tilvísunar finnst ekki..**

7.3.1 StartPasient

Load the parameter contained in startpasient.xml into the variable \$startPasient and call the StartPasient method:

```
$startPasient = gc "startpasient.xml"
$a.StartPasient($startPasient)
```

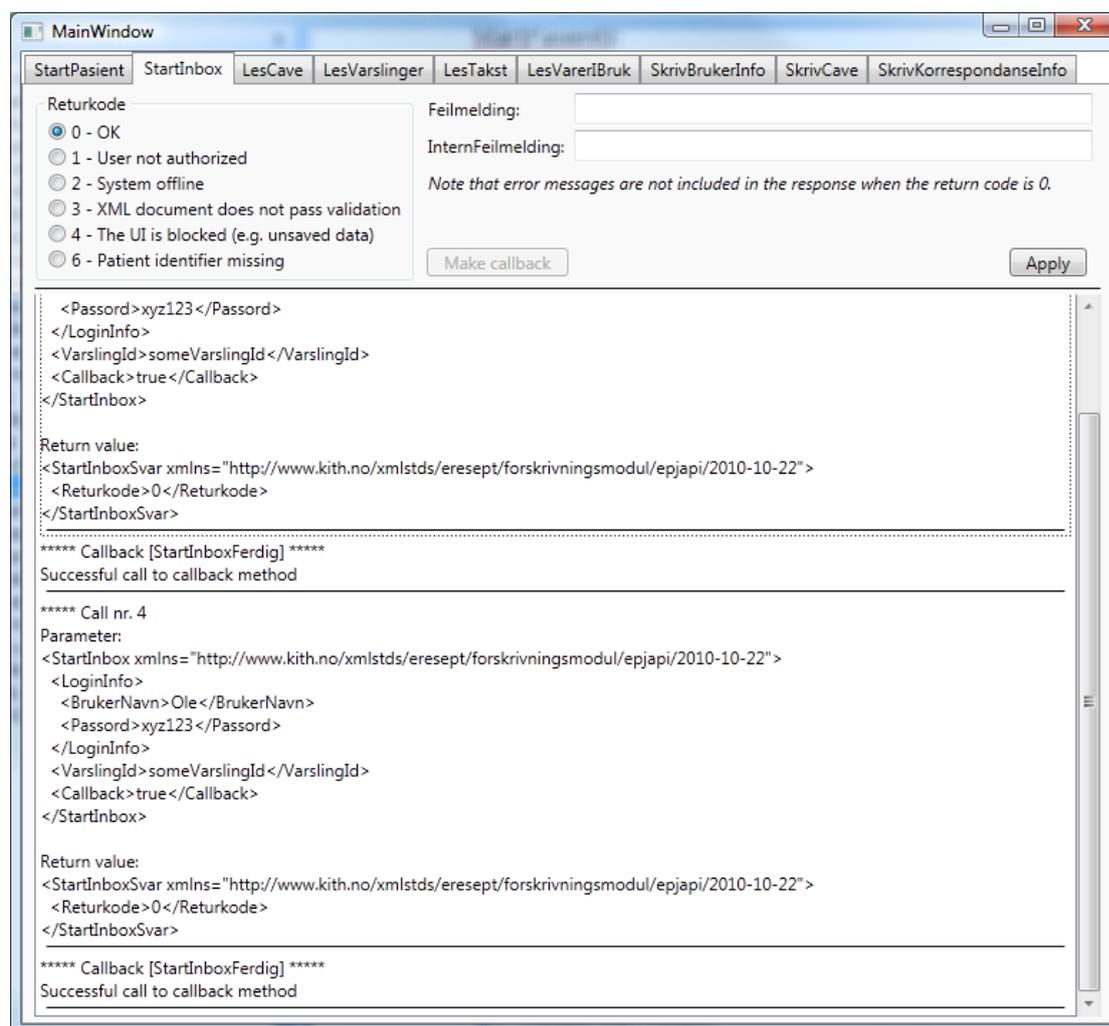
The results in PowerShell look as follows:



```

PS C:\temp> $a = New-object -comObject eResept.Forskrivningsmodul.1 -strict
PS C:\temp> $startPasient = gc "startpasient.xml"
PS C:\temp> $a.StartPasient($startPasient)
<StartPasientSvar xmlns="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2010-10-22">
  <Returkode>0</Returkode>
</StartPasientSvar>
PS C:\temp>
    
```

The FM.Client.EpjApiTestApp.exe UI is updated when calls are received as shown below (note that the call log is dedicated to each tab in the UI, so the call log for e.g. StartInbox is not affected by calls to StartPasient):



7.3.2 StartInbox

```

$startInbox = gc "startinbox.xml"
$a.StartInbox($startInbox)
    
```

7.3.3 SkrivBrukerInfo

```

$skrivBrukerInfo = gc "skrivbrukerinfo.xml"
$a.SkrivBrukerInfo($skrivBrukerInfo)
    
```

7.3.4 SkrivCave

```
$skrivCave = gc "skrivcave.xml"  
$a.SkrivCave($skrivCave)
```

7.3.5 SkrivKorrespondanseInfo

```
$skrivKorrespondanseInfo = gc "skrivkorrespondanseinfo.xml"  
$a.SkrivKorrespondanseInfo($skrivKorrespondanseInfo)
```

7.3.6 LesCave

```
$lesCave = gc "lescave.xml"  
$a.LesCave($lesCave)
```

7.3.7 LesVarslinger

```
$lesVarslinger = gc "lesvarslinger.xml"  
$a.LesVarslinger($lesVarslinger)
```

7.3.8 LesTakst

```
$lesTakst = gc "lesTakst.xml"  
$a.LesTakst($lesTakst)
```

7.3.9 LesVarerIBruk

```
$lesVarerIBruk = gc "lesvareribruk.xml"  
$a.LesVarerIBruk($lesVarerIBruk)
```

7.4 Testing the callback interface

The EPJ API Test Application is capable of testing the callback interface (currently only eResept.ForskrivningsmodulCallback.2) of an EPJ if it has been registered.

On the tabs for StartPasient and StartInbox there are buttons (Make callback) to make calls to the methods StartPasientFerdig and StartInboxFerdig on the callback interface.

The buttons will become enabled when a corresponding "Start call" (Pasient or Inbox) has been received and StartPasient/StartInbox parameter has the Callback value set to True. When pressed, the buttons will become disabled again until another start call is received. For each call made the result is written to the dedicated log.