EPJ API and Technical Specification

E-resept Forskrivningsmodul

Thula – Nordic Source Solutions

Date | 08/04/2013

Software version | FM 2.9.2
Document version | 6.0
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

© 201308/04/2013 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 ... 5
1.5 Reader comments ... 6
1.6 Glossary ... 6
2 Introduction ... 8
3 EPJ API specification ... 8
3.1 Methods in eResept.Forskrivningsmodul.1 ... 8
3.1.1 LesCave ... 9
3.1.2 LesVarslinger ... 9
3.1.3 LesTakst ... 10
3.1.4 LesVarerIBruk ... 10
3.1.5 SkrivBrukerInfo ... 11
3.1.6 SkrivCave ... 11
3.1.7 SkrivKorrespondanseInfo ... 11
3.1.8 StartPasient ... 12
3.1.9 StartInbox ... 13
3.1.10 LesAntallMeldingerForSignering ... 14
3.1.11 LesSisteVibCaveOppdatering ... 14
3.2 Methods in eResept.ForskrivningsmodulCallback.2 and 3 ... 14
3.2.1 StartPasientFerdig ... 15
3.2.2 StartInboxFerdig ... 17
3.2.3 StartCaveFerdig ... 19
3.3 Data model ... 19
3.3.1 XML types ... 19
3.3.2 XML elements ... 20
3.3.3 XSD ... 27
3.4 API technology ... 27
3.5 XML import ... 28
4 Technical specification ... 28
4.1 System structure ... 28
4.1.1 Primary components ... 29
4.1.2 Secondary components ... 29
4.2 Technology ... 30
5 Technical requirements ... 30
5.1 Workstation requirements ... 30
5.2 Application server requirements ... 31
5.3 Database server requirements ... 31

5.4 Other technical requirements ... 31
6 Appendix A: EPJ API XSD ... 31
7 Appendix B: EPJ API Test Application ... 39
7.1 Registering the application ... 39
7.2 Accessing the COM interface ... 39
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 ... 41
7.3.5 SkrivKorrespondanseInfo ... 41
7.3.6 LesCave ... 41
7.3.7 LesVarslinger ... 42
7.3.8 LesTakst ... 42
7.3.9 LesVarerIBruk ... 42
7.4 Testing the callback interface ... 42
7.5 Method parameter examples ... 42
7.5.1 StartPasient.xml ... 42
7.5.2 StartInbox.xml ... 43
7.5.3 SkrivBrukerInfo.xml ... 43
7.5.4 SkrivCave.xml ... 43
7.5.5 SkrivKorrespondanseInfo.xml ... 44
7.5.6 LesCave.xml ... 45
7.5.7 LesVarslinger.xml ... 45
7.5.8 LesTakst.xml ... 46
7.5.9 LesVarerIBruk.xml ... 46

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 2.9.1
2013-04-02	5.1	Atli Sturluson	Added new API method (Lukk)
			Release 2.9.0
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.
			Release 2.7.2
2012-10-11	3.1	Ægir Örn Leifsson	Added information about session ID in Start… methods and their callbacks.
			Release 2.7.0
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 (`eResept.ForskrivningsmodulCallback.2` and `eResept.ForskrivningsmodulCallback.3`).
2012-09-16	2.2	Ægir Örn Leifsson	Added documentation about new fields in the Resept class (see section 3.3.2.16), 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 3.3.2.19).
			Release 2.3.90
2012-04-12	1.35a	Atli Sturluson	Added the Read-only user role to the SkrivBrukerInfo type
			Release 2.3.80
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
			Release 2.3.60
2011-01-10	1.34a	Atli Sturluson	Updated the SkrivBrukerInfo type in section 3.3.2.17
2010-12-08	1.33a	Brynjar Bragason	Replaced screenshots in section 0. Added section 7.4 about the call back interface.
			Release 2.3.42
2010-11-24	1.32a	Ægir Örn Leifsson	Fixed an error in the XSD in Appendix A. The Varsling element had a cardinality of 1 in LesVarslingerSvar, this has been fixed and is now 0..unbounded. The text describing LesVarslingerSvar was correct (section 3.3.2.14).
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: • 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 PM 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 egir.leifsson@thula.is. If a review result in changes, all user 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.
PM	Forskrivningsmodul (Prescription Module), English abbreviation used throughout most of this document.
EPJ	Electronic patient journal. A computerized patient record/journal system that communicates with the PM through the PM 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 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 PM. The API provides the only way to initiate the prescription process in the PM. The API consists of two interfaces:

- eResept.Forskrivningsmodul.1 – this is the interface implemented by the PM 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 PM when the user completes his work in the PM 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 PM (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. 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 PM 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 PM. 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-PM integration.

Note that even though the EPJ API allows an EPJ to change the current patient in the PM (using StartPasient) the API does not guarantee that the EPJ and PM 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 PM. However, if the patient is already known to the PM, 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 PM. 6 – Patient identifier missing.

3.1.2 LesVarslinger

This method is used to determine if there is anything in the PM 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 PM 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 PM 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 PM Inbox.

Figure 1 - EPJ/PM user opens the PM 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 PM. However, if the patient is already known to the PM, 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 PM. 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 PM. However, if the patient is already known to the PM, 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 PM. 6 – Patient identifier missing.

3.1.5 SkrivBrukerInfo

This method is used to create and update user info in the PM. 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 PM.

NOTE: This is the only method that supports a LoginInfo element with an empty username and password. In this case the PM 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 PM 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 PM if the patient identifier provided cannot be found. If the patient is already known to the PM, 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 PM if a provided CAVE entry identifier cannot be found. If the CAVE entry is already known to the PM, 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 PM).

	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 PM user can manually choose to import the received information into the LIB/NIB/FIB.

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

This method will bring up the PM 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.

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

The EPJ can request a call-back from the PM when the doctor has finished his work in the PM. This mechanism is described in section 3.1.12.

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). 5 – Patient is not registered in PM.

Figure 3 (same as Figure 1) shows the context in which StartInbox can be used (steps 5 and 6). The EPJ periodically polls the PM 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 PM Inbox.

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

The EPJ can request a call-back from the PM when the doctor has finished his work in the PM. This mechanism is described in section 3.1.12.

3.1.10 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 where the assistant 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.

3.1.11 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 PM. 6 – Patient identifier missing.

3.1.12 Lukk

This method closes the PM 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.

3.2 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 PM when the user closes down the PM 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 PM when the doctor finishes his work in the PM.

Figure 4 shows the case where the EPJ has set the Callback parameter in StartPasient to request a call-back from the PM. When the doctor finishes his work in the PM, the PM 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 PM 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 PM. When the doctor finishes his work in the PM, the PM simply closes. How the user resumes his work in the EPJ cannot be influenced by the PM in this case.

3.2.2 StartInboxFerdig

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

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

Figure 6 shows the case where the EPJ has set the Callback parameter in StartInbox to request a call-back from the PM. When the doctor finishes his work in the PM, the PM 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 PM 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 PM. When the doctor finishes his work in the PM, the PM simply closes. How the user resumes his work in the EPJ cannot be influenced by the PM in this case.

3.2.3 StartCaveFerdig

This method may be called by the PM when the user closes down the PM 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/2013-03-12

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 PM.

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 PM.
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-PM 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 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.
RegisteringsDato	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.

3.3.2.2 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).

3.3.2.3 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.

3.3.2.4 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.

3.3.2.5 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.

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).

3.3.2.6 LoginInfo

This element is used to provide information on the EPJ user to the PM 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.

3.3.2.7 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.

3.3.2.8 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.

3.3.2.9 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.

3.3.2.10 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.

3.3.2.11 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.

3.3.2.12 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.

3.3.2.13 LesVarslinger

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

3.3.2.14 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.

3.3.2.15 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 identifier used to look up the corresponding patient in the PM (this is not done on the fødselsnummer). If the PM receives a Pasient element without an ID of type "XXX" it will return an error code 6 (patient identifier missing). If the PM receives a Pasient element where the "XXX" ID contains an unknown code, a new patient record will be created in the PM in some cases but in other cases an error will be returned (see section 3.1). If the PM 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.
Addresse	An address for the patient.
Telekommunikasjon	A list of information on phone numbers, etc.

3.3.2.16 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 PM 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.
SeponeringsInfomasjon	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: - LegemiddelMerkevare – LegmiddelMerkevare.Varenavn. - Legemiddelpakning – LegmiddelMerkevare.Varenavn of all referenced LegemiddelMerkevare instances are concatenated. - LegemiddelVirkestoff – Virkestoff.Navn of all references Virkestoff instances are concatenated. - Legemiddelblandning – Legemiddelblandning.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.

Note that the element will include ReseptDokLegemiddel or ReseptDokHandelsvare, never both.

3.3.2.17 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.
BrukerInfo/BrukerNavn	The username of the user.
BrukerInfo/Status	The status of the user: A = Active

	U = Inactive
BrukerInfo/Passord	The password (or a hash of one) of the user.
BrukerInfo/Role	One or more roles for the user. The defined roles are: 1 - Admin 2 - Doctor 3 - Assistant 4 - Read-only access NOTE: A user cannot have both the doctor and assistant role, or both doctor and read-onlyl role.
BrukerInfo/Etternavn	The family name of the user.
BrukerInfo/Mellomnavn	The middle name of the user.
BrukerInfo/Fornavn	The first (given) name of the user.
BrukerInfo/Fodseldato	The birthdate of the user.
BrukerInfo/Kjonn	The gender of the user (coding system 3101).
BrukerInfo/Nasjonalitet	The nationality of the user (coding system 9043).
BrukerInfo/Id	A list of identifiers for the user (coding system 8116).
BrukerInfo/Addresse	An address for the user.
BrukerInfo/Telekommunikasjon	A list of information on phone numbers, etc (coding system 9061).
BrukerInfo/Spesialitet	A list of specialities for a doctor (coding system 7426).

3.3.2.18 SkrivBrukerInfoSvar

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

3.3.2.19 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.

3.3.2.20 SkrivCaveSvar

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

3.3.2.21 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.

3.3.2.22 SkrivKorrespondanseInfoSvar

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

3.3.2.23 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 PM.
Innleggelse	Optional element containing information about the location reponsible 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 PM.
SessionId	An ID that the EPJ wants reported back in the StartPasientFerdig callback.

3.3.2.24 StartPasientSvar

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

3.3.2.25 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 PM.
CAVEId	The ID of a CAVE record to be focused when the FM opens (optional).
Innleggelse	Optional element containing information about the location reponsible for the patient
Callback	Indicates if the EPJ wants a call-back when the user closes down the PM.
SessionId	An ID that the EPJ wants reported back in the StartPasientFerdig callback.

3.3.2.26 StartCaveSvar

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

3.3.2.27 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.
Callback	Indicates if the EPJ wants a call-back when the user closes down the PM.
SessionId	An ID that the EPJ wants reported back in the StartInboxFerdig callback.

3.3.2.28 StartInboxSvar

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

3.3.2.29 Varsling

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

XML Element	Description
VarslingId	A 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).

3.3.2.30 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
AnsvarligEnhet	Element wrapping a single single Organisation object that stores the location that is responsible for the patient. This element is mandatory.
InleggelsesTidspunkt	The date and time of the patient's admission time, this parameter is optional

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.

3.3.2.32 LesAntallMeldingerForSignering

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

3.3.2.33 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.

3.3.2.34 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.

3.3.2.35 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.

3.3.2.36 Lukk

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

3.3.2.37 LukkSvar

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

3.3.2.38 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.

3.3.2.39 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.

3.3.2.40 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.

3.3.3 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 PM 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 PM 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 PM will create a reference to the COM object once, and hold on to it for the lifetime of the PM process.
- If the reference becomes invalid (happens if the EPJ is shut down), the PM will create a new reference and hold onto that instead.
- To get a reference to the COM object the PM 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 PM 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 PM 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 ImportPasientInfo 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 PM 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 PM has the following primary components:

- PM 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 PM.
    - 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.
- PM 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 PM 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 PM data, including system configuration.

4.1.2 Secondary components

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

- PM Client installer - An installation application that installs the PM Client application on the user's workstation.
- PM Application server installer - An installation application that installs the PM Application server on the server machine.
- PM Application server configuration wizard - An application that is installed along with the PM Application server and provides the following functionality:
    - Installs SQL Server 2008 R2 Express if required by the customer.
    - Installs the PM database schema on the database.
    - Configures PM 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 PM components. This client can be used by EPJ vendors when developing against the EPJ API.

4.2 Technology

The PM is implemented using the following technologies:

- Microsoft .NET Framework 4.0 - All of the PM 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:
  - PM Client application to PM Application server communication - WCF is used to expose a private set of services from the PM Application server that are used by the PM 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 PM Client application to access the PM Application server.
  - Web service communication - WCF is used for communicating with the RF and FEST.
- Windows Event Log - Used by the PM Application server for error reporting.
- Microsoft SQL Server 2008 R2 Express - This is a free database server that PM uses to persist data. PM 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 PM client application:

	Requirement
Operating system	The PM 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 PM client application requires Microsoft .NET Framework 4.0. NOTE: The installer will install the framework if it is not already installed.
CPU	The PM client application requires a modern CPU (Intel Core2Duo or similar).
Memory	The PM client application requires approximately 1GB of free memory.
Disk	The PM 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 PM 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 PM 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 PM application server:

	Requirement
Operating system	The PM application server requires one of the following operating system: Microsoft Windows Server 2003 SP2 Microsoft Windows Server 2003 R2 SP2 Microsoft Windows Server 2008 SP2 Microsoft Windows Server 2008 R2 - Recommended
Frameworks	The PM application server requires Microsoft .NET Framework 4.0. NOTE: The installer will install the framework if it is not already installed.
CPU	The PM application server requires a modern server class CPU (Intel Xeon or similar).
Memory	The PM application server requires approximately 1GB of free memory.
Disk	The PM 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 PM 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

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 PM 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 SQL Server 2008 R2 Express never uses more that 1GB memory, and can therefore easily be installed on the same server as the PM application server. Such an installation will require that 2GB of memory are free.

5.4 Other technical requirements

PM 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

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Definition of the EPJ API datatypes. -->
<schema xmlns:kith="http://www.kith.no/xmlstds"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema.xsd"
        xmlns="http://www.w3.org/2001/XMLSchema"
```

```
        xmlns:fk1="http://www.kith.no/xmlstds/felleskomponent1"
        xmlns:m1="http://www.kith.no/xmlstds/eresept/m1/2010-05-01"
        xmlns:epjapi="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-
03-12"
        xmlns:msgHead="http://www.kith.no/xmlstds/msghead/2006-05-24"

targetNamespace="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-03-12"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified">
  <import   namespace="http://www.kith.no/xmlstds"   schemaLocation="eResept-v2.4-2010-12-
08/felles/kith.xsd"/>
  <import namespace="http://www.kith.no/xmlstds/felleskomponent1" schemaLocation="eResept-
v2.4-2010-12-08/felles/felleskomponent1.xsd"/>
  <import                    namespace="http://www.kith.no/xmlstds/eresept/m1/2010-05-01"
schemaLocation="eResept-v2.4-2010-12-08/R0908-eResept-M1-M21-2010-05-01/ER-M1-2010-05-
01.xsd"/>
  <import                        namespace="http://www.kith.no/xmlstds/msghead/2006-05-24"
schemaLocation="eResept-v2.4-2010-12-08/felles/MsgHead-v1_2.xsd"/>
  <complexType name="typeForesporsel">
    <annotation>
      <documentation>
        Defines elements that are common to all EPJ API requests.
      </documentation>
    </annotation>
    <sequence>
      <element ref="epjapi:LoginInfo"/>
    </sequence>
  </complexType>
  <complexType name="typeSvar">
    <annotation>
      <documentation>
        Defines elements that are common to all EPJ API responses.
      </documentation>
    </annotation>
    <sequence>
      <element name="Returkode" type="int" minOccurs="0"/>
      <element name="Feilmelding" type="string" minOccurs="0"/>
      <element name="InternFeilmelding" type="string" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="typeCallback">
    <sequence>
      <element name="Brukernavn" type="string"/>
      <element name="SessionId" type="string" minOccurs="0"/>
    </sequence>
  </complexType>
  <element name="CAVE">
    <complexType>
      <sequence>
        <element name="CAVEId" type="string"/>
        <element name="HjelpestoffReaksjon" type="boolean"/>
        <element name="GrunnlagForCAVE" type="string"/>
        <element name="RegisteringsDato" type="date"/>
        <element name="Signatur" type="string"/>
        <element name="Innaktiv" type="boolean"/>
        <choice>
          <element name="LegemiddelMerkevare">
            <complexType>
              <sequence>
                <element name="LegemiddelId" type="string"/>
                <element name="LegemiddelNavn" type="string"/>
                <element name="ATCKode" type="kith:CV"/>
                <element ref="epjapi:Virkestoff" minOccurs="0" maxOccurs="unbounded"/>
              </sequence>
            </complexType>
          </element>
          <element ref="epjapi:Virkestoff"/>
          <element name="ATC">
            <complexType>
              <sequence>
                <element name="ATCKode" type="kith:CV"/>
              </sequence>
            </complexType>
          </element>
        </choice>
      </sequence>
    </complexType>
```

```
</element>
<element name="ImportPasientInfo">
  <complexType>
    <sequence>
      <element name="ImportPasient" minOccurs="0" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element ref="epjapi:Pasient"/>
            <element ref="epjapi:Resept" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="epjapi:ReseptHistorikk" minOccurs="0"/>
            <element ref="epjapi:CAVE" minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
<element name="LoginInfo">
  <complexType>
    <sequence>
      <element name="BrukerNavn" type="string"/>
      <element name="Passord" type="string"/>
    </sequence>
  </complexType>
</element>
<element name="LesCave">
  <complexType>
    <complexContent>
      <extension base="epjapi:typeForesporsel">
        <sequence>
          <element ref="epjapi:Pasient"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="LesSisteVibCaveOppdateringSvar">
  <complexType>
    <complexContent>
      <extension base="epjapi:typeSvar">
        <sequence>
          <element name="SisteVibOppdatering" type="dateTime" minOccurs="0"/>
          <element name="SisteCaveOppdatering" type="dateTime" minOccurs="0"/>
          <element name="SisteTakstOppdatering" type="dateTime" minOccurs="0"/>
          <element name="SisteOppdateringMax" type="dateTime" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="LesAntallMeldingerForSignering">
  <complexType>
    <complexContent>
      <extension base="epjapi:typeForesporsel"/>
    </complexContent>
  </complexType>
</element>
<element name="LesAntallMeldingerForSigneringSvar">
  <complexType>
    <complexContent>
      <extension base="epjapi:typeSvar">
        <sequence>
          <element name="AntallM1" type="int"/>
          <element name="AntallM5" type="int"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="LesSisteVibCaveOppdatering">
  <complexType>
    <complexContent>
      <extension base="epjapi:typeForesporsel">
        <sequence>
          <element ref="epjapi:Pasient"/>
        </sequence>
      </extension>
```

```
    </complexContent>
  </complexType>
</element>
<element name="LesCaveSvar">
  <complexType>
    <complexContent>
      <extension base="epjapi:typeSvar">
        <sequence>
          <element ref="epjapi:CAVE" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="LesVarslinger">
  <complexType>
    <complexContent>
      <extension base="epjapi:typeForesporsel"/>
    </complexContent>
  </complexType>
</element>
<element name="LesVarslingerSvar">
  <complexType>
    <complexContent>
      <extension base="epjapi:typeSvar">
        <sequence>
          <element ref="epjapi:Varsling" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="Varsling">
  <complexType>
    <sequence>
      <element ref="epjapi:VarslingId"/>
      <element ref="epjapi:Pasient" minOccurs="0"/>
      <element name="Melding" type="string"/>
      <element name="Avsender" type="string"/>
    </sequence>
  </complexType>
</element>
<element name="VarslingId" type="string"/>
<element name="LesTakst">
  <complexType>
    <complexContent>
      <extension base="epjapi:typeForesporsel">
        <sequence>
          <element ref="epjapi:Pasient"/>
          <element name="Dato" type="date"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="LesTakstSvar">
  <complexType>
    <complexContent>
      <extension base="epjapi:typeSvar">
        <sequence>
          <element name="Takst" minOccurs="0" maxOccurs="unbounded">
            <complexType>
              <sequence>
                <element name="TakstNavn" type="string"/>
                <element name="TakstId" type="string"/>
                <element name="Tidspunkt" type="dateTime"/>
              </sequence>
            </complexType>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element name="LesVarerIBruk">
  <complexType>
    <complexContent>
```

```
        <extension base="epjapi:typeForesporsel">
          <sequence>
            <element ref="epjapi:Pasient"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <element name="LesVarerIBrukSvar">
    <complexType>
      <complexContent>
        <extension base="epjapi:typeSvar">
          <sequence>
            <element ref="epjapi:Resept" minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <element name="Pasient">
    <complexType>
      <sequence>
        <element name="Etternavn" type="string" minOccurs="0"/>
        <element name="Mellomnavn" type="string" minOccurs="0"/>
        <element name="Fornavn" type="string" minOccurs="0"/>
        <element name="Fodseldato" type="date" minOccurs="0"/>
        <element name="Kjonn" type="kith:CS" minOccurs="0"/>
        <element name="Nasjonalitet" type="kith:CS" minOccurs="0"/>
        <element name="Id" type="fk1:Ident" minOccurs="0" maxOccurs="unbounded"/>
        <element name="Adresse" type="fk1:Address" minOccurs="0"/>
        <element      name="Telekommunikasjon"      type="fk1:TeleCom"      minOccurs="0"
maxOccurs="unbounded"/>
        <element name="Frikort" type="boolean" minOccurs="0"/>
        <element ref="m1:Utlending" minOccurs="0"/>
        <element name="FastlegeHprId" type="string" minOccurs="0"/>
      </sequence>
    </complexType>
  </element>
  <element name="Resept">
    <complexType>
      <sequence>
        <element name="Id" type="string"/>
        <element name="VibStatus" type="string" minOccurs="0"/>
        <choice>
          <element ref="epjapi:ReseptDokLegemiddel"/>
          <element ref="epjapi:ReseptDokHandelsvare"/>
        </choice>
        <element name="AnsvarligLege" type="string" minOccurs="0"/>
        <element name="InstituertAv" type="string" minOccurs="0"/>
        <element name="InstitueringsDato" type="date" minOccurs="0"/>
        <element name="ForrigeReseptId" type="string" minOccurs="0"/>
        <element name="SeponeringsInfomasjon" minOccurs="0">
          <complexType>
            <sequence>
              <element name="SeponertAv" type="string"/>
              <element name="Seponeringsdato" type="date"/>
              <element name="Seponeringsgrunn" type="string"/>
            </sequence>
          </complexType>
        </element>
        <element name="RegistreringsType" type="string" minOccurs="0"/>
        <element name="Kortdose" type="kith:CV" minOccurs="0"/>
        <element name="ForskrivningsDato" type="date" minOccurs="0"/>
        <element name="Varenavn" type="string" minOccurs="0"/>
        <element name="Styrke" type="string" minOccurs="0"/>
        <element name="Legemiddelform" type="string" minOccurs="0"/>
      </sequence>
    </complexType>
  </element>
  <element name="ReseptHistorikk">
    <complexType>
      <simpleContent>
        <extension base="base64Binary"/>
      </simpleContent>
    </complexType>
  </element>
  <element name="ReseptDokLegemiddel">
```

```
      <complexType>
        <simpleContent>
          <extension base="base64Binary"/>
        </simpleContent>
      </complexType>
    </element>
    <element name="Innleggelse">
      <complexType>
        <sequence>
          <element name="AnsvarligEnhet">
            <complexType>
              <sequence>
                <element ref="msgHead:Organisation" />
              </sequence>
            </complexType>
          </element>
          <element name="InnleggelsesTidspunkt" type="dateTime" minOccurs="0" />
        </sequence>
      </complexType>
    </element>
    <element name="ReseptDokHandelsvare">
      <complexType>
        <simpleContent>
          <extension base="base64Binary"/>
        </simpleContent>
      </complexType>
    </element>
    <element name="BrukerInfo">
      <complexType>
        <sequence>
          <element name="BrukerNavn" type="string"/>
          <element name="Status" type="kith:CS"/>
          <element name="Passord" type="string"/>
          <element name="Role" type="kith:CS" maxOccurs="unbounded"/>
          <element name="Etternavn" type="string" minOccurs="0"/>
          <element name="Mellomnavn" type="string" minOccurs="0"/>
          <element name="Fornavn" type="string" minOccurs="0"/>
          <element name="Fodseldato" type="date" minOccurs="0"/>
          <element name="Kjonn" type="kith:CS" minOccurs="0"/>
          <element name="Nasjonalitet" type="kith:CS" minOccurs="0"/>
          <element name="Id" type="fk1:Ident" minOccurs="0" maxOccurs="unbounded"/>
          <element name="Adresse" type="fk1:Address" minOccurs="0"/>
          <element       name="Telekommunikasjon"       type="fk1:TeleCom"       minOccurs="0"
maxOccurs="unbounded"/>
          <element name="Spesialitet" type="kith:CV" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </complexType>
    </element>
    <element name="SkrivBrukerInfo">
      <complexType>
        <complexContent>
          <extension base="epjapi:typeForesporsel">
            <sequence>
              <element ref="epjapi:BrukerInfo" minOccurs="0" maxOccurs="unbounded" />
            </sequence>
          </extension>
        </complexContent>
      </complexType>
    </element>
    <element name="SkrivBrukerInfoSvar">
      <complexType>
        <complexContent>
          <extension base="epjapi:typeSvar"/>
        </complexContent>
      </complexType>
    </element>
    <element name="SkrivCave">
      <complexType>
        <complexContent>
          <extension base="epjapi:typeForesporsel">
            <sequence>
              <element ref="epjapi:Pasient"/>
              <element ref="epjapi:CAVE" minOccurs="0" maxOccurs="unbounded"/>
              <element name="ErstatteAlle" type="boolean" minOccurs="0"/>
            </sequence>
          </extension>
        </complexContent>
```

```
    </complexType>
  </element>
  <element name="LesBrukerInfoSvar">
    <complexType>
      <complexContent>
        <extension base="epjapi:typeSvar">
          <sequence>
            <element ref="epjapi:BrukerInfo" />
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <element name="SkrivCaveSvar">
    <complexType>
      <complexContent>
        <extension base="epjapi:typeSvar"/>
      </complexContent>
    </complexType>
  </element>
  <element name="SkrivKorrespondanseInfo">
    <complexType>
      <complexContent>
        <extension base="epjapi:typeForesporsel">
          <sequence>
            <element ref="epjapi:Pasient"/>
            <element ref="epjapi:Resept" minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <element name="SkrivKorrespondanseInfoSvar">
    <complexType>
      <complexContent>
        <extension base="epjapi:typeSvar"/>
      </complexContent>
    </complexType>
  </element>
  <element name="StartPasient">
    <complexType>
      <complexContent>
        <extension base="epjapi:typeForesporsel">
          <sequence>
            <element ref="epjapi:Pasient"/>
            <element name="AnsvarligLege" type="string" minOccurs="0"/>
            <element ref="epjapi:Innleggelse" minOccurs="0" />
            <element name="Samstillingstekst" type="string" minOccurs="0"/>
            <element name="Callback" type="boolean"/>
            <element name="SessionId" type="string" minOccurs="0"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <element name="StartPasientSvar">
    <complexType>
      <complexContent>
        <extension base="epjapi:typeSvar"/>
      </complexContent>
    </complexType>
  </element>
  <element name="StartInbox">
    <complexType>
      <complexContent>
        <extension base="epjapi:typeForesporsel">
          <sequence>
            <element ref="epjapi:VarslingId"/>
            <element name="Callback" type="boolean"/>
            <element name="SessionId" type="string" minOccurs="0"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <element name="StartInboxSvar">
    <complexType>
```

```
        <complexContent>
          <extension base="epjapi:typeSvar"/>
        </complexContent>
      </complexType>
    </element>
    <element name="Lukk">
      <complexType>
        <complexContent>
           <extension base="epjapi:typeForesporsel"/>
        </complexContent>
      </complexType>
    </element>
    <element name="LukkSvar">
      <complexType>
        <complexContent>
          <extension base="epjapi:typeSvar"/>
        </complexContent>
      </complexType>
    </element>
    <element name="Virkestoff">
      <complexType>
        <sequence>
          <element name="VirkestoffId" type="string"/>
          <element name="VirkestoffNavn" type="string"/>
        </sequence>
      </complexType>
    </element>
    <element name="StartCave">
      <complexType>
        <complexContent>
          <extension base="epjapi:typeForesporsel">
            <sequence>
              <element ref="epjapi:Pasient"/>
              <element name="CAVEId" type="string" minOccurs="0"/>
              <element ref="epjapi:Innleggelse" minOccurs="0" />
              <element name="Callback" type="boolean"/>
              <element name="SessionId" type="string" minOccurs="0"/>
            </sequence>
          </extension>
        </complexContent>
      </complexType>
    </element>
    <element name="StartCaveSvar">
      <complexType>
        <complexContent>
          <extension base="epjapi:typeSvar"/>
        </complexContent>
      </complexType>
    </element>
    <element name="StartCaveFerdig">
      <complexType>
        <complexContent>
          <extension base="epjapi:typeCallback"/>
        </complexContent>
      </complexType>
    </element>
    <element name="StartPasientFerdig">
      <complexType>
        <complexContent>
          <extension base="epjapi:typeCallback"/>
        </complexContent>
      </complexType>
    </element>
    <element name="StartInboxFerdig">
      <complexType>
        <complexContent>
          <extension base="epjapi:typeCallback">
            <sequence>
              <element name="PasientXxxIds">
                <complexType>
                  <sequence>
                    <element       name="PasientXxxId"       type="string"       minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
                </complexType>
              </element>
            </sequence>
```

```
            </extension>
          </complexContent>
        </complexType>
      </element>
    </schema>
```

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 **Error! Reference source not found.**.

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:

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.

7.5 Method parameter examples

7.5.1 StartPasient.xml

```
<?xml version="1.0" encoding="utf-8"?>
<StartPasient xmlns:tns="http://www.kith.no/xmlstds"
xmlns:tnsa="http://www.kith.no/xmlstds/felleskomponent1"
xmlns="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-03-12"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <LoginInfo>
        <BrukerNavn>bjarni</BrukerNavn>
        <Passord>b</Passord>
    </LoginInfo>
    <Pasient>
        <Mellomnavn>string</Mellomnavn>
        <Fodseldato>1973-03-08</Fodseldato>
        <Id>
            <tnsa:Id>string</tnsa:Id>
            <tnsa:TypeId V="string" DN="string" />
        </Id>
        <Id>
            <tnsa:Id>12345</tnsa:Id>
            <tnsa:TypeId V="XXX" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Id>
            <tnsa:Id>44051900100</tnsa:Id>
            <tnsa:TypeId V="DNR" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Adresse>
            <tnsa:Type V="string" DN="string" />
            <tnsa:Postbox>string</tnsa:Postbox>
            <tnsa:County V="string" DN="string" />
            <tnsa:Country V="string" DN="string" />
        </Adresse>
        <Telekommunikasjon>
            <tnsa:TypeTelecom />
            <tnsa:TeleAddress V="http://www.liquid-technologies.com" />
        </Telekommunikasjon>
    </Pasient>
    <Callback>0</Callback>
</StartPasient>
```

7.5.2 StartInbox.xml

```
<?xml version="1.0" encoding="utf-8"?>
<StartInbox xmlns:tns="http://www.kith.no/xmlstds"
xmlns:tnsa="http://www.kith.no/xmlstds/felleskomponent1"
xmlns="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-03-12"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <LoginInfo>
        <BrukerNavn>bjarni</BrukerNavn>
        <Passord>b</Passord>
    </LoginInfo>
    <VarslingId>0FA3AB62-EC4B-4715-8173-93E29C4E1C42</VarslingId>
    <Callback>0</Callback>
</StartInbox>
```

7.5.3 SkrivBrukerInfo.xml

```
<?xml version="1.0" encoding="utf-8"?>
<SkrivBrukerInfo xmlns:tns="http://www.kith.no/xmlstds"
xmlns:tnsa="http://www.kith.no/xmlstds/felleskomponent1"
xmlns="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-03-12"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <LoginInfo>
        <BrukerNavn>bjarni</BrukerNavn>
        <Passord>b</Passord>
    </LoginInfo>
    <BrukerInfo>
        <BrukerNavn>bruker</BrukerNavn>
        <Status V="2"/>
        <Passord>passord</Passord>
        <Role V="2"/>
        <Etternavn>string</Etternavn>
        <Mellomnavn>string</Mellomnavn>
        <Fornavn>string</Fornavn>
        <Fodseldato>2019-02-10</Fodseldato>
        <Kjonn V="1" />
        <Nasjonalitet V="string />
        <Id>
            <tnsa:Id>1234567890</tnsa:Id>
            <tnsa:TypeId S="string" DN="string" OT="string" />
        </Id>
        <Id>
            <tnsa:Id>00112233</tnsa:Id>
            <tnsa:TypeId S="string" DN="string" OT="string" />
        </Id>
        <Adresse>
            <tnsa:Type V="string" />
            <tnsa:PostalCode>12345</tnsa:PostalCode>
            <tnsa:County V="ABC" />
            <tnsa:Country V="NO" />
        </Adresse>
        <Telekommunikasjon>
            <tnsa:TypeTelecom V="MC" />
            <tnsa:TeleAddress V="tel:+4773598600" />
        </Telekommunikasjon>
        <Telekommunikasjon>
            <tnsa:TypeTelecom DN="WD" />
            <tnsa:TeleAddress V="tel:+4773592233" />
        </Telekommunikasjon>
        <Spesialitet V="28" DN="Kjevekirurgi og munnhulesykd." />
        <Spesialitet V="195" DN="Hjertesykdommer" />
    </BrukerInfo>
</SkrivBrukerInfo>
```

7.5.4 SkrivCave.xml

```
<?xml version="1.0" encoding="utf-8"?>
<SkrivCave xmlns:tns="http://www.kith.no/xmlstds"
xmlns:tnsa="http://www.kith.no/xmlstds/felleskomponent1"
xmlns="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-03-12"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <LoginInfo>
        <BrukerNavn>bjarni</BrukerNavn>
        <Passord>b</Passord>
    </LoginInfo>
    <Pasient>
        <Mellomnavn>string</Mellomnavn>
```

```
        <Fodseldato>1973-03-08</Fodseldato>
        <Id>
            <tnsa:Id>string</tnsa:Id>
            <tnsa:TypeId V="string" DN="string" />
        </Id>
        <Id>
            <tnsa:Id>12345</tnsa:Id>
            <tnsa:TypeId V="XXX" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Id>
            <tnsa:Id>44051900100</tnsa:Id>
            <tnsa:TypeId V="DNR" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Adresse>
            <tnsa:Type V="string" DN="string" />
            <tnsa:Postbox>string</tnsa:Postbox>
            <tnsa:County V="string" DN="string" />
            <tnsa:Country V="string" DN="string" />
        </Adresse>
        <Telekommunikasjon>
            <tnsa:TypeTelecom />
            <tnsa:TeleAddress V="http://www.liquid-technologies.com" />
        </Telekommunikasjon>
    </Pasient>
    <CAVE>
        <CAVEId>351F1EC8-5797-4BDA-A112-7A5A91EAE406</CAVEId>
        <HjelpestoffReaksjon>false</HjelpestoffReaksjon>
        <GrunnlagForCAVE>Grunnlag for CAVE.</GrunnlagForCAVE>
        <RegisteringsDato>2010-11-19</RegisteringsDato>
        <Signatur>Ole Nordman</Signatur>
        <Innaktiv>false</Innaktiv>
        <LegemiddelMerkevare>
            <LegemiddelId>ID 77AAFA7B-32ED-46D6-9C5B-6056D6CA9091</LegemiddelId>
            <LegemiddelNavn>Sandimmun Neoral Kaps 100 mg</LegemiddelNavn>
            <ATCKode V="L04AD01"/>
        </LegemiddelMerkevare>
    </CAVE>
</SkrivCave>
```


7.5.5 SkrivKorrespondanseInfo.xml


```
<?xml version="1.0" encoding="utf-8"?>
<SkrivKorrespondanseInfo xmlns:tns="http://www.kith.no/xmlstds"
xmlns:tnsa="http://www.kith.no/xmlstds/felleskomponent1"
xmlns="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-0-27"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <LoginInfo>
        <BrukerNavn>bjarni</BrukerNavn>
        <Passord>b</Passord>
    </LoginInfo>
    <Pasient>
        <Mellomnavn>string</Mellomnavn>
        <Fodseldato>1973-03-08</Fodseldato>
        <Id>
            <tnsa:Id>string</tnsa:Id>
            <tnsa:TypeId V="string" DN="string" />
        </Id>
        <Id>
            <tnsa:Id>12345</tnsa:Id>
            <tnsa:TypeId V="XXX" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Id>
            <tnsa:Id>44051900100</tnsa:Id>
            <tnsa:TypeId V="DNR" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Adresse>
            <tnsa:Type V="string" DN="string" />
            <tnsa:Postbox>string</tnsa:Postbox>
            <tnsa:County V="string" DN="string" />
            <tnsa:Country V="string" DN="string" />
        </Adresse>
        <Telekommunikasjon>
            <tnsa:TypeTelecom />
            <tnsa:TeleAddress V="http://www.liquid-technologies.com" />
        </Telekommunikasjon>
    </Pasient>
```

```
    <Resept xmlns:fs="http://www.kith.no/xmlstds/eresept/forskrivning/2010-04-01"
xmlns:m1="http://www.kith.no/xmlstds/eresept/m1/2010-05-01">
        <Id></Id>
        <m1:ReseptDokLegemiddel>
            <m1:Varegruppekode></m1:Varegruppekode>
            <m1:Reiterasjon></m1:Reiterasjon>
            <fs:Forskrivning>
                <fs:LegemiddelMerkevare>
                    <fs:NavnFormStyrke></fs:NavnFormStyrke>
                    <fs:Reseptgruppe></fs:Reseptgruppe>
                    <fs:Varenavn></fs:Varenavn>
                </fs:LegemiddelMerkevare>
            </fs:Forskrivning>
        </m1:ReseptDokLegemiddel>
    </Resept>
    <Resept xmlns:fs="http://www.kith.no/xmlstds/eresept/forskrivning/2010-04-01"
xmlns:m1="http://www.kith.no/xmlstds/eresept/m1/2010-05-01">
        <Id></Id>
        <m1:ReseptDokLegemiddel>
            <m1:Varegruppekode></m1:Varegruppekode>
            <m1:Reiterasjon></m1:Reiterasjon>
            <fs:Forskrivning>
                <fs:LegemiddelMerkevare>
                    <fs:NavnFormStyrke></fs:NavnFormStyrke>
                    <fs:Reseptgruppe></fs:Reseptgruppe>
                    <fs:Varenavn></fs:Varenavn>
                </fs:LegemiddelMerkevare>
            </fs:Forskrivning>
        </m1:ReseptDokLegemiddel>
    </Resept>
</SkrivKorrespondanseInfo>
```


7.5.6 LesCave.xml


```
<?xml version="1.0" encoding="utf-8"?>
<LesCave xmlns:tns="http://www.kith.no/xmlstds"
xmlns:tnsa="http://www.kith.no/xmlstds/felleskomponent1"
xmlns="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-03-12"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <LoginInfo>
        <BrukerNavn>bjarni</BrukerNavn>
        <Passord>b</Passord>
    </LoginInfo>
    <Pasient>
        <Mellomnavn>string</Mellomnavn>
        <Fodseldato>1973-03-08</Fodseldato>
        <Id>
            <tnsa:Id>string</tnsa:Id>
            <tnsa:TypeId V="string" DN="string" />
        </Id>
        <Id>
            <tnsa:Id>12345</tnsa:Id>
            <tnsa:TypeId V="XXX" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Id>
            <tnsa:Id>44051900100</tnsa:Id>
            <tnsa:TypeId V="DNR" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Adresse>
            <tnsa:Type V="string" DN="string" />
            <tnsa:Postbox>string</tnsa:Postbox>
            <tnsa:County V="string" DN="string" />
            <tnsa:Country V="string" DN="string" />
        </Adresse>
        <Telekommunikasjon>
            <tnsa:TypeTelecom />
            <tnsa:TeleAddress V="http://www.liquid-technologies.com" />
        </Telekommunikasjon>
    </Pasient>
</LesCave>
```


7.5.7 LesVarslinger.xml


```
<?xml version="1.0" encoding="utf-8"?>
<LesVarslinger xmlns:tns="http://www.kith.no/xmlstds"
xmlns:tnsa="http://www.kith.no/xmlstds/felleskomponent1"
xmlns="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-03-12"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <LoginInfo>
        <BrukerNavn>bjarni</BrukerNavn>
        <Passord>b</Passord>
    </LoginInfo>
</LesVarslinger>
```

7.5.8 LesTakst.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LesTakst xmlns:tns="http://www.kith.no/xmlstds"
xmlns:tnsa="http://www.kith.no/xmlstds/felleskomponent1"
xmlns="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-03-12"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <LoginInfo>
        <BrukerNavn>bjarni</BrukerNavn>
        <Passord>b</Passord>
    </LoginInfo>
    <Pasient>
        <Mellomnavn>string</Mellomnavn>
        <Fodseldato>1973-03-08</Fodseldato>
        <Id>
            <tnsa:Id>string</tnsa:Id>
            <tnsa:TypeId V="string" DN="string" />
        </Id>
        <Id>
            <tnsa:Id>12345</tnsa:Id>
            <tnsa:TypeId V="XXX" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Id>
            <tnsa:Id>44051900100</tnsa:Id>
            <tnsa:TypeId V="DNR" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Adresse>
            <tnsa:Type V="string" DN="string" />
            <tnsa:Postbox>string</tnsa:Postbox>
            <tnsa:County V="string" DN="string" />
            <tnsa:Country V="string" DN="string" />
        </Adresse>
        <Telekommunikasjon>
            <tnsa:TypeTelecom />
            <tnsa:TeleAddress V="http://www.liquid-technologies.com" />
        </Telekommunikasjon>
    </Pasient>
    <Dato>2010-11-18</Dato>
</LesTakst>
```

7.5.9 LesVarerIBruk.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LesVarerIBruk xmlns:tns="http://www.kith.no/xmlstds"
xmlns:tnsa="http://www.kith.no/xmlstds/felleskomponent1"
xmlns="http://www.kith.no/xmlstds/eresept/forskrivningsmodul/epjapi/2013-03-12"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <LoginInfo>
        <BrukerNavn>bjarni</BrukerNavn>
        <Passord>b</Passord>
    </LoginInfo>
    <Pasient>
        <Mellomnavn>string</Mellomnavn>
        <Fodseldato>1973-03-08</Fodseldato>
        <Id>
            <tnsa:Id>string</tnsa:Id>
            <tnsa:TypeId V="string" DN="string" />
        </Id>
        <Id>
            <tnsa:Id>12345</tnsa:Id>
            <tnsa:TypeId V="XXX" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Id>
            <tnsa:Id>44051900100</tnsa:Id>
            <tnsa:TypeId V="DNR" DN="string" OT="string" S="2.16.578.1.12.4.1.1.8116" />
        </Id>
        <Adresse>
            <tnsa:Type V="string" DN="string" />
            <tnsa:Postbox>string</tnsa:Postbox>
```

```
            <tnsa:County V="string" DN="string" />
            <tnsa:Country V="string" DN="string" />
        </Adresse>
        <Telekommunikasjon>
            <tnsa:TypeTelecom />
            <tnsa:TeleAddress V="http://www.liquid-technologies.com" />
        </Telekommunikasjon>
    </Pasient>
</LesVarerIBruk>
```