ELEKTRONISCHE FALLAKTE

Elektronische Fallakte e.V.

c/o Universitätsklinikum Aachen

Pauwelsstraße 30
52074 Aachen

http://www.Fallakte.de

Spezifikation Offline-Token

Addendum zur EFA v2.0 Spezifikation

Trial Implementation Rev. 1.0 25-10-2018

Tim Wilking, Salima Houta
Fraunhofer-Institut für Software- und Systemtechnik

1. Einleitung

1.1. Problemstellung

Die technische Spezifikation der Elektronischen Fallakte (EFA) unterstützt in der aktuellen Version 2.0 die Vergabe von Berechtigungen auf eine EFA über ein strukturiertes Berechtigungsmanagement. Ein bereits berechtigter EFA-Teilnehmer erteilt im Auftrag des Patienten nach schriftlicher Einwilligung des Patienten bei der Anlage der EFA oder bei der Anpassung der Berechtigungen einer EFA weiteren Einrichtungen oder Personen Zugriff auf die EFA. Um Berechtigungen auf eine Akte zu vergeben, muss der Arzt bereits zugriffsberechtigt auf die EFA sein. Diese Rahmenbedingungen sind nicht immer gegeben. Konsultiert ein Patient einen nicht berechtigten Arzt und möchte diesem ad hoc Zugriff auf eine seiner Fallakten erteilen, ist dies nicht möglich.

Außerdem ist es für verschiedene Überleitungsszenarien relevant, schnell neue Leistungserbringer für eine EFA zu berechtigten. Ein Kommunikationsaufbau zu berechtigten Leistungserbringern für die Editierung der Zugriffsberechtigungen ist zeitintensiv, organisatorisch komplex und nicht realistisch, da eine Zugriffsrechteerweiterung ohne Patient oder Vormund nicht möglich ist.

Um einen schnellen Zugriff für neue Beteiligte am Behandlungsszenario zu ermöglichen, soll das Offline-Token zum Einsatz kommen. Dies ist ein Mechanismus, der es dem Patienten ermöglicht, unabhängig relevante Leistungserbringer für den Zugriff auf die EFA zu berechtigen. Das Offline-Token gilt immer für eine bestimmte EFA und ermöglicht EFA-Teilnehmern nach Überreichung des Tokens durch den Patienten, sich selbst ein Zugriffsrecht auf diese EFA zu erteilen. Um dem Datenschutz und der Datensicherheit Rechnung zu tragen, kann das Offline-Token nur von Personen eingelöst werden, die im EFA-Netzwerk registriert sind. Die Offline-Token Transaktionen werden dabei umfassend protokolliert. Außerdem ist die schriftliche Einwilligung des Patienten oder eines Vormunds wesentlich.

Das EFA 2.0 Addendum Offline-Token umfasst die technische Beschreibung der Umsetzung des Offline-Tokens. Die Spezifikation setzt auf bestehende Mechanismen der EFA auf, um den Entwicklungsaufwand möglichst gering zu halten und auf bereits etablierte Verfahren aufzusetzen.

1.2. Aufbau des Dokuments

Das Dokument beschreibt zunächst ein Anwendungsszenario für das Offline-Token. Aus diesem werden anschließend Anwendungsfälle abgeleitet. Die Anwendungsfälle bilden die Grundlage für die Erweiterung der technischen Spezifikation der EFA. Die bestehenden EFA-Akteure werden um Offline-Token Funktionen erweitert. Das Informationsmodell der EFA wird um weitere Informationsobjekte ergänzt, die für die Umsetzung des Offline-Token Konzepts notwendig sind. Anschließend wird das technische Binding der Funktionen und Objekte spezifiziert und mit Beispielen untermauert.

2. Anwendungsszenario für das Offline-Token

Das hier aufgeführte Anwendungsszenario basiert auf Arbeiten des vom Land NRW und der EU geförderten Projektes I/E-Health NRW. Das dort gemeinsam mit Ärzten erarbeitete Szenario stellt einen konkreten Bedarf für das Offline-Token aus Behandlungssicht in der Modellregion Düren / Jülich dar. Für die Offline-Token Spezifikation wurde das Szenario leicht gekürzt / angepasst.

Kurzbeschreibung

Pflegebedürftige Menschen werden in Notfällen kurzfristig ins Krankenhaus aufgenommen. Nicht immer kann vorher die Hausärztin oder der Hausarzt, die/der den kranken Menschen kennt, konsultiert werden. Speziell an Wochenenden und Feiertagen führt dies dazu, dass das Krankenhaus auf Basis unvollständiger Informationen handeln muss. Gerade bei multimorbiden Menschen kann dies zu unnötigen Risiken bei der Behandlung führen.

Die beteiligten Ärztenetze möchten die genannten Risiken mittels einer EFA verringern. Die betreuenden Ärztinnen und Ärzte stellen dazu für Risikopatienten alle notwendigen Daten und Informationen in eine EFA ein. Sofern in der Praxis keine originäre Anbindung an die elektronische Akte existiert, werden die Daten mittels KV-Connect an das Aktensystem gesendet. Die Patientin bzw. der Patient erhält ein „Offline-Token“, mit dem bei Eintritt des Notfalls dem Krankenhaus der Zugriff auf die Daten gewährt werden kann.

Vorgeschichte

Herr Hans Albrecht ist 83 Jahre. Er ist multimorbid und ist seit einigen Jahren Bewohner eines Pflegeheims und erhält aufgrund seiner unterschiedlichen Krankheiten eine regelmäßige Betreuung durch den Hausarzt des Pflegeheims. Die medizinische Dokumentation erfolgt durch den Hausarzt in dessen Praxissystem. Eine potentielle Notfallbehandlung des Pflegebedürftigen ist in einem Vertrag mit dem Pflegeheim geregelt. Dabei ist es möglich, dass Herr Albrecht spät abends oder am Wochenende einen Notfall erleidet und in ein Krankenhaus eingeliefert werden muss.

Beim Hausarzt

Der Hausarzt identifiziert potentielle Notfallpatienten und informiert Herrn Albrecht über die Möglichkeit der EFA für die intersektorale Kommunikation. Herr Albrecht ist von den Vorteilen einer EFA überzeugt und willigt in die Nutzung der EFA ein. Um die Akte für Herrn Albrecht anzulegen, meldet sich der Hausarzt am Primärsystem an und beginnt mit der Neuanlage der Notfallakte. Anhand von personenbezogenen Daten sucht der Arzt den Patienten im System aus. Da Herr Albrecht bereits bei ihm war und eine Fallakte hatte, findet er den Patienten im EFA-System. Er wählt den Patienten aus und erstellt eine neue Fallakte. Dazu legt er die Zweckbestimmung „Notfall“ sowie die initiale Berechtigten-Liste fest. Zusätzlich legt der Arzt fest, dass er für den Patienten ein Offline-Token für seine Notfallakte ausstellen möchte. Aus dem System wird automatisch eine Patienteneinwilligung erzeugt und ausgedruckt. Herr Albrecht unterzeichnet diese Einwilligung. Danach kann die Fallaktenanlage abgeschlossen werden. Der Hausarzt druckt das Offline-Token in Form eines QR-Code aus und überreicht dieses der Pflegeeinrichtung, in der Herr Albrecht stationär versorgt wird. Der Offline-Token-Ausdruck enthält u. A. den Namen, Geburtsdatum (zur einfacheren Verwaltung) sowie einen QR-Code und wird im Pflegeheim in den Begleitpapieren von Herrn Albrecht

hinterlegt. Anschließend lädt der Hausarzt Notfallinformationen in Form eines ärztlichen Kurzberichts in die EFA hoch.

Notfall im Pflegeheim

Einige Wochen später bekommt Herr Albrecht samstags abends Atemnot und drückt den Notrufknopf, wodurch die örtliche Notfallversorgung bzw. der Rettungsdienst gerufen wird. Nach einer mündlichen Befragung des Personals über Zustand und Vorgeschichte des Patienten veranlasst der Rettungsdienst eine Einweisung in ein nahegelegenes Krankenhaus. Die Begleitpapiere von Herrn Albrecht enthalten unter anderem seine elektronische Gesundheitskarte und das vom Hausarzt erstellte Ticket für den Zugriff zur EFA. Herr Albrecht wird anschließend zum Krankenhaus transportiert.

Einweisung in das Krankenhaus

Während der Einweisung im Krankenhaus wird der Patient zunächst über die eGK identifiziert und anschließend wird das Krankenhaus über die vorhandene Notfallakte beim Übergabeprozess aufgeklärt. Für einen schnellen Zugriff auf die Notfallakte überreicht die Rettungskraft das Offline-Token. Die Aufnahmekraft löst das Offline-Token ein. Als Ergebnis erhält das Krankenhaus eine Sicht auf die Berechtigungskonfiguration der Notfallakte und kann die Berechtigung für das Krankenhaus setzen. Dabei ist es wichtig, dass eine Person mit Patientenverfügung eine Einwilligung unterzeichnet.

Nach dem Editieren der Berechtigungen wird der medizinische Inhalt der Notfallakte angezeigt. Durch entsprechende Mechanismen im KIS wird die Notfallakte mit der internen Patientenakte / Patientennummer verknüpft. Die Inhalte können nun automatisiert oder manuell in die Patientenakte im Krankenhausinformationssystem gespeichert werden.

Behandlung im Krankenhaus

Herr Albrecht wird nun im Krankenhaus behandelt, das Team erhält dabei wichtige Informationen durch seine Notfallakte. Nach 2 Tagen ist seine Situation stabil und er kann wieder aus dem Krankenhaus entlassen werden.

Weitere mögliche Anwendungsfälle

Der Vollständigkeit halber werden in der folgenden Tabelle weitere Anwendungsfälle, die nicht Teil des Szenarios sind, aufgeführt.

Anwendungsfall	**Beschreibung**
Erneutes Einlösen des Tokens	Das Offline-Token kann mehrfach eingelöst werden, solange das Offline-Token gültig ist.
Verlust des Offline-Tokens	Bei Verlust des Offline-Tokens kann das Offline-Token invalidiert werden. Danach kann auf Patientenwunsch hin ein neues Offline-Token erstellt werden.
Token kann nicht eingelöst werden	Wenn ein Offline-Token abgelaufen ist oder invalidiert wurde, dann kann es nicht mehr eingelöst werden. Ein Abrufen und Ändern des Berechtigungsdokuments ist dann nicht mehr möglich.

2.1. Anwendungsfall 1: Offline-Token erstellen

Auf Wunsch des Patienten stellt ein auf die Fallakte des Patienten zugriffsberechtigter EFA-Teilnehmer ein Offline-Token aus. Dazu wählt der EFA-Teilnehmer die relevante Fallakte aus und bestätigt die Funktion zum Erstellen eines Offline-Tokens. Nach Erstellung des Offline-Tokens wird dieses ausgedruckt und dem Patienten ausgehändigt. Eine elektronische Übermittlung ist ebenfalls möglich, wenn entsprechende technische Infrastrukturen dafür geschaffen wurden. Die Erzeugung des Offline-Tokens kann standardmäßig in den Prozess der Fallaktenanlage und der Berechtigungskonfiguration eingebettet werden.

Die erfolgreiche Erstellung des Offline-Tokens umfasst die nachfolgend gelisteten Prozessschritte. Im technischen Teil wird der Prozess inkl. der Ausnahmen detailliert beschreiben.

- Der EFA-Teilnehmer wählt die Funktion zum Erstellen des Offline-Tokens im EFA-Teilnehmersystem aus.
- Das EFA-Teilnehmersystem erzeugt eine Offline-Token-Kennung.
- Das EFA-Teilnehmersystem erzeugt eine Offline-Token-Berechtigungsregel, welche die Offline-Token-Kennung und die EFA verknüpft.
- Das EFA-Teilnehmersystem sendet eine Anfrage zum Einstellen der Offline-Token-Berechtigungsregel an das EFA-Providersystem.
- Das EFA-Providersystem speichert die Offline-Token-Berechtigungsregel.
- Das EFA-Teilnehmersystem generiert beispielsweise einen QR-Code (alternativ können in Projekten weitere Formate spezifiziert werden), der die Offline-Token-Kennung enthält. Optional kann je nach Projektkontext auch die Zweckbindung und die Patienten-ID im QR-Code abgelegt werden. Dies kann notwendig sein, wenn das einlösende System keinen Kontext zur EFA mitführt.
- Der EFA-Teilnehmer druckt die Offline-Token-Kennung als QR-Code.
- Der gedruckte QR-Code wird an den Patienten übergeben.

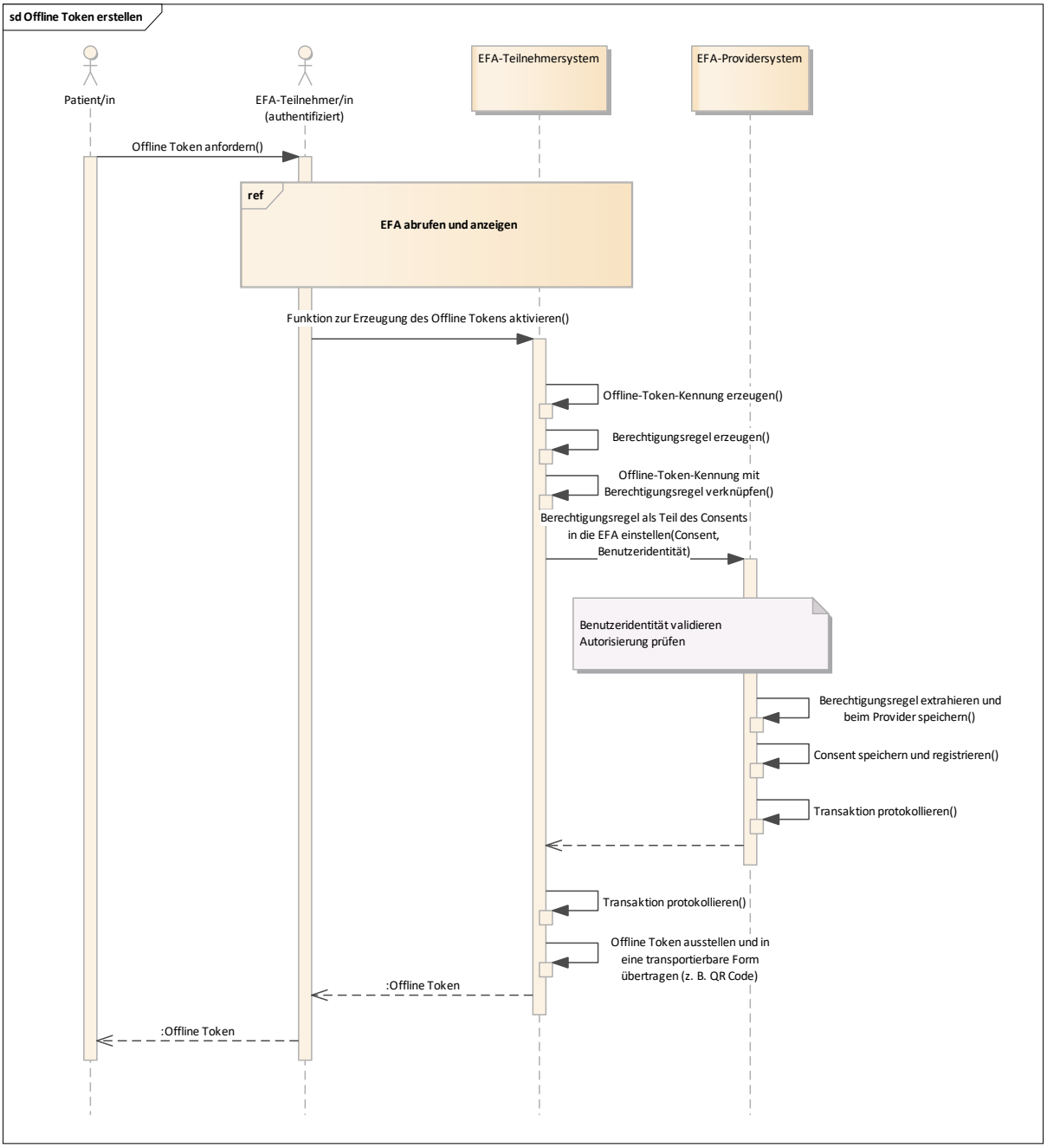

Abbildung 1 Anwendungsfall Offline Token erstellen

2.2. Anwendungsfall 2: Einlösen eines Offline-Tokens

Durch das Aushändigen des QR-Codes kann der Patient einem beliebigem EFA-Teilnehmer den Zugriff auf die entsprechende EFA ermöglichen. Dazu wird der QR-Code im EFA-Teilnehmersystem eingelesen. Mit der im QR-Code enthaltenen Offline-Token-Kennung kann das EFA-Teilnehmersystem eine Offline-Token-Assertion ausstellen lassen. Mit der Offline-Token-Assertion hat der einlösende EFA-Teilnehmer temporär Zugriff auf das Consent-Dokument einer EFA. Danach kann der EFA-Teilnehmer die Berechtigungen gemäß dem Patientenwillen anpassen und erweitern. Prinzipiell kann der EFA-Teilnehmer beliebige Organisationen oder einzelne EFA-Teilnehmer berechtigen, wenn es dem Patientenwillen entspricht. Enthält der QR-Code auch die Zweckbindung und die Patienten-ID,

so können diese vom EFA-Teilnehmersystem genutzt werden, um die EFA aufzufinden. Andernfalls müssen diese Informationen aus dem Behandlungskontext resultieren, bzw. anhand der vorhandenen Daten aus weiteren Quellen ermittelt werden.

Das erfolgreiche Einlösen des Offline-Tokens umfasst die nachfolgenden beschriebenen Schritte. Im technischen Teil wird der Prozess inkl. der Ausnahmen detailliert beschreiben.

- Der Patient übergibt dem zu berechtigenden EFA-Teilnehmer die Offline-Token-Kennung als QR-Code.
- Der EFA-Teilnehmer wählt im EFA-Teilnehmersystem die Funktion zum Einlösen des Offline-Tokens und scannt anschließend den QR-Code ein.
- Das EFA-Teilnehmersystem extrahiert die Offline-Token-Kennung aus dem QR-Code.
- Das EFA-Teilnehmersystem sendet eine Anfrage an das EFA-Providersystem, um das Offline-Token einzulösen. Diese beinhaltet die Offline-Token-Kennung sowie die Benutzeridentität des aktuell angemeldeten Teilnehmers.
- Das EFA-Providersystem sendet nach Prüfung der Benutzeridentität eine Offline-Token-Assertion an das EFA-Teilnehmersystem.
- Das EFA-Teilnehmersystem fragt mit der temporären Offline-Token-Assertion die aktuellen Berechtigungen der EFA beim EFA-Providersystem ab.
- Das EFA-Providersystem prüft auf Basis der Offline-Token-Assertion, ob eine Zugriffsberechtigung auf das Berechtigungsdokument mit der Offline-Token-Assertion vorliegt.
- Das EFA-Teilnehmersystem öffnet die Berechtigungskonfiguration und zeigt diese dem EFA-Teilnehmer an.
- Der EFA-Teilnehmer ändert die Berechtigungen gemäß dem Patientenwunsch.
- Der EFA-Teilnehmer bestätigt die Anpassung der Berechtigungen nach schriftlicher Einwilligung durch den Patienten.
- Das EFA-Teilnehmersystem sendet eine Anfrage zur Anpassung der Berechtigungen an das EFA-Providersystem. Diese beinhaltet die Offline-Token-Assertion und die neuen Berechtigungen.
- Das EFA-Providersystem prüft die Offline-Token-Assertion und speichert im Erfolgsfall die neuen Berechtigungen.

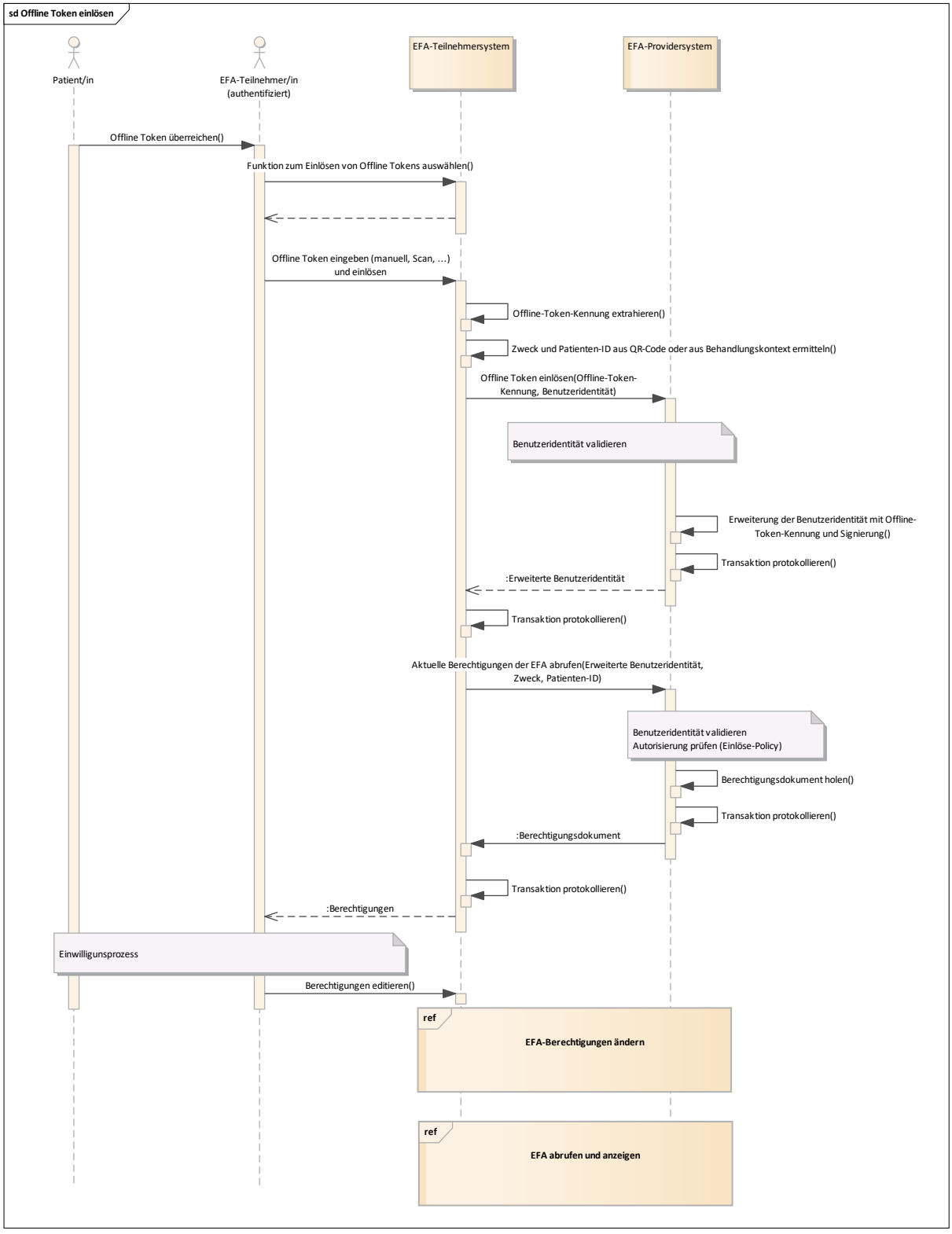

Abbildung 2 Anwendungsfall Offline Token einlösen

2.3. Anwendungsfall 3: Deaktivieren eines Offline-Tokens

Das Deaktivieren eines Offline-Tokens erfolgt durch das Ersetzen oder Entfernen der Offline-Token-Berechtigung aus dem Berechtigungsdokument der EFA.

Das erfolgreiche Deaktivieren des Offline-Tokens umfasst folgende Schritte. Im technischen Teil wird der Prozess inkl. der Ausnahmen detailliert beschreiben.

- Der Patient teilt einem auf seine EFA zugriffsberechtigten EFA-Teilnehmer mit, dass er seinen Offline-Token deaktivieren möchte.
- Der EFA-Teilnehmer öffnet die EFA und wählt die Funktion zum Deaktivieren des Offline-Tokens aus.
- Das EFA-Teilnehmersystem entfernt die Offline-Token-Berechtigungsregel aus dem Berechtigungsdokument.
- Das EFA-Teilnehmersystem sendet ein aktualisiertes Berechtigungsdokument an das EFA-Providersystem.
- Das EFA-Providersystem entfernt die entsprechende Offline-Token-Berechtigungsregel aus seinem Policy-Repository.
- Das EFA-Providersystem sendet eine Statusinformation an das EFA-Teilnehmersystem.
- Das EFA-Teilnehmersystem stellt die Statusinformation dar.

3. Informationsmodell und Transaktionen

Das nachfolgende Modell skizziert die relevanten EFA-Akteure zur Umsetzung des Offline-Tokens.

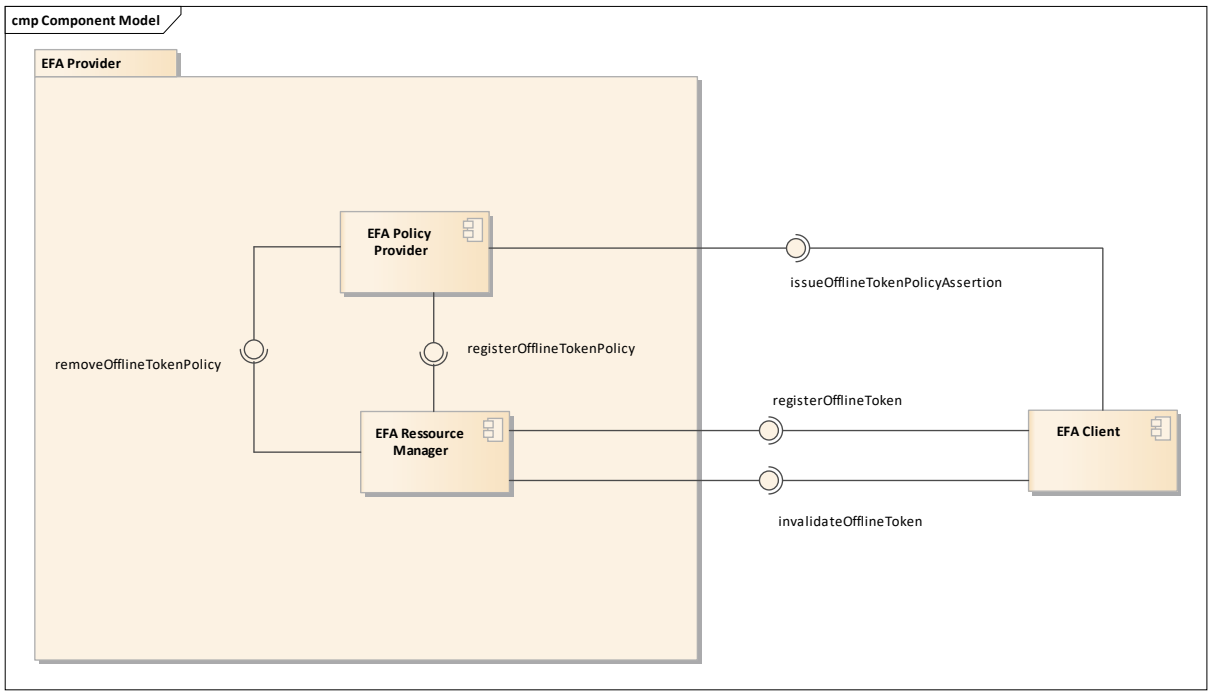

Abbildung 3 Komponentenmodell zum Offline-Token

Für die Umsetzung der Offline-Token Funktionalität sind keine zusätzlichen EFA-Akteure notwendig. Die bestehenden Akteure werden um zusätzliche Funktionalität erweitert, sowie das Informationsmodell und das Sicherheitsmodell ergänzt.

Die nachfolgende Tabelle listet die EFA-Akteure auf, die im Kontext des Offline-Tokens eine Rolle spielen. Auf eine detaillierte Beschreibung der Komponenten wird an dieser Stelle verzichtet. Die Informationen können in der EFA-Spezifikation nachgelesen werden.

EFA-Akteur	**Link**
EFA Policy Provider	http://wiki.hl7.de/index.php?title=cdaefa:EFA_Policy_Provider_SFM
EFA Resource Manager	http://wiki.hl7.de/index.php?title=cdaefa:EFA_XDS_ResourceManager
EFA Client	http://wiki.hl7.de/index.php?title=cdaefa:EFA_Dienste

3.1. Informationsmodell

Das Informationsmodell der EFA-Spezifikation (siehe: http://wiki.hl7.de/index.php?title=cdaefa:EFA_Business_Informationsmodell und http://wiki.hl7.de/index.php?title=cdaefa:EFA_Security_Informationsmodell) wird um die nachfolgend beschriebenen Klassen erweitert.

3.1.1. offlineTokenPolicy

Dieses Sicherheitsobjekt beschreibt eine Berechtigungsregel für eine EFA. Sie ist an eine zufällig erzeugte Kennung gebunden und nicht an einen konkreten EFA-Teilnehmer. Das Ende des Gültigkeitszeitraums muss angegeben werden und darf die aktuelle Gültigkeitsdauer der EFA (ohne grace-period) nicht überschreiten. Dies gilt auch wenn der Gültigkeitszeitraum der EFA verändert wird. Anpassungen des Gültigkeitszeitraum der EFA müssen dazu führen, dass auch die Gültigkeit des Offline-Tokens erneut geprüft wird. Bei einer Verlängerung des Zeitraums kann es sinnvoll auch den Gültigkeitszeitraum des Offline-Tokens zu verlängern. Das Sicherheitsobjekt offlineTokenPolicy ist Teil der consentInfo einer EFA.

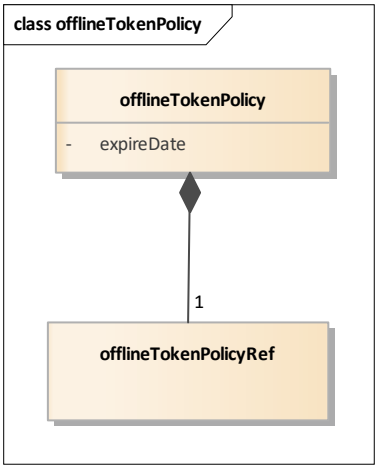

Abbildung 4 Sicherheitsobjekt offlineTokenPolicy

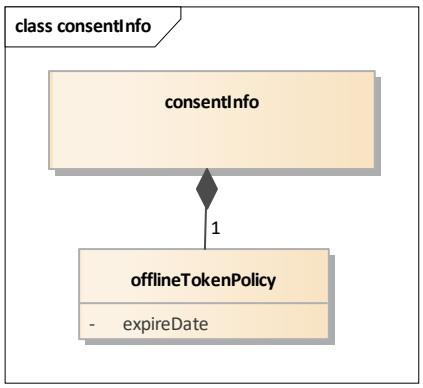

Abbildung 5 Sicherheitsobjekt consetnInfo mit offlineTokenPolicy

3.1.2. offlineTokenAssertion

Dieses Sicherheitsobjekt beschreibt eine um eine Offline-Token-Kennung erweiterte Form von subjectIdentity[1] und wird als Nachweis für eine Zugriffsberechtigung verwendet. Eine Verifizierbarkeit der Daten einer offlineTokenAssertion ist nur gegeben, wenn die offlineTokenAssertion selbst integer und authentisch ist. Daher muss jede offlineTokenAssertion von der ausstellenden Stelle signiert werden.

3.1.3. offlineTokenPolicyRef

Die Klasse beschreibt die Offline-Token-Kennung, an die eine Offline-Token-Berechtigungsregel (offlineTokenPolicy) gebunden ist. Die Offline-Token-Kennung besitzt keine Semantik und kann durch einen Zufallswert repräsentiert bzw. umgesetzt werden. Es muss sichergestellt werden, dass die Offline-Token-Kennung für eine EFA im Kontext eines Patienten eindeutig ist.

3.2.Transaktionen

Die neuen Funktionen der EFA-Akteure werden im Folgenden beschrieben.

3.2.1. EFA Resource Manager

Der EFA Resource Manager nutzt die bestehende Methode registerConsent bzw. bei der Fallaktenanlage die Methode createECR, um Berechtigungsregeln zum Offline-Token zu registrieren.

Für das Einlösen des Offline-Tokens wird der EFA Resource Manager um die folgende Funktion erweitert:

- registerOfflineToken (keine neue Operation, wird eingebettet in registerConsent bzw. createECR)
- invalidateOfflineToken (keine neue Operation, wird eingebettet in registerConsent)

3.2.1.1. registerOfflineToken

Ein Offline-Token wird registriert, indem eine Offline-Token-Berechtigungsregel im Consent einer Fallakte hinterlegt wird. Dazu wird ein bestehendes Consent um diese Offline-Token-Berechtigungsregel erweitert und in die Fallakte eingestellt. Das EFA-Providersystem legt diese Regel im Policy-Repository ab.

Die Spezifikation der Operationen createECR und registerConsent sind Teil der EFA 2.0-Spezifikation und wird an dieser Stelle nicht näher erläutert.

[1] https://wiki.hl7.de/index.php?title=cdaefa:EFA_Security_Informationsmodell#subjectIdentity

3.2.1.2. invalidateOfflineToken

Ein Offline-Token wird invalidiert, indem die dazugehörige Offline-Token-Berechtigungsregel aus dem Consent einer FallAkte entfernt wird. Das EFA-Providersystem entfernt die Berechtigungsregel aus dem Policy-Repository.

3.2.2. EFA Policy Provider

Der EFA Policy Provider wird um die folgenden Funktionen erweitert:

- issueOfflineTokenAssertion
- registerOfflineTokenPolicy
- removeOfflineTokenPolicy

3.2.2.1.issueOfflineTokenAssertion

Operation	issueOfflineTokenAssertion	
Funktionalität	Diese Operation stellt eine signierte offlineTokenAssertion für eine EFA aus.	
Aufrufer	EFA-Teilnehmersystem der gleichen EFA-Provider-Domäne	
Eingabe	context	Gibt den Sicherheitskontext vor, in dem die Operation ausgeführt wird. Bezugsquelle: EFA Kontext Manager *openContext*.
	offlineTokenPolicyRef	Eindeutige Identifizierung der offlineTokenPolicy.
Rückgabe	offlineTokenAssertion	Ein Berechtigungstoken, das den Leistungserbringer, der es einreicht, zum lesenden und schreibenden Zugriff auf das consentInfo einer Fallakte berechtigt. Dieses sollte zeitlich eingeschränkt sein (Empfehlung: 10 Minuten).
Vorbedingungen		

Ablauf	1. Erzeuge offlineTokenAssertion auf Basis der Identity Assertion aus dem Sicherheitskontext. 2. Gebe offlineTokenAssertion an Aufrufer zurück.
Fehler und Warnungen	Folgende Fehler müssen erkannt und rückgemeldet werden: • Gemeinsame Fehlermeldungen und Warnungen Hinweis: die in diesem Dokument aufgeführten spezifischen Fehlermeldungen (siehe Kapitel 0) werden in die allgemeinen Fehlermeldungen und Warnungen der EFA eingepflegt.

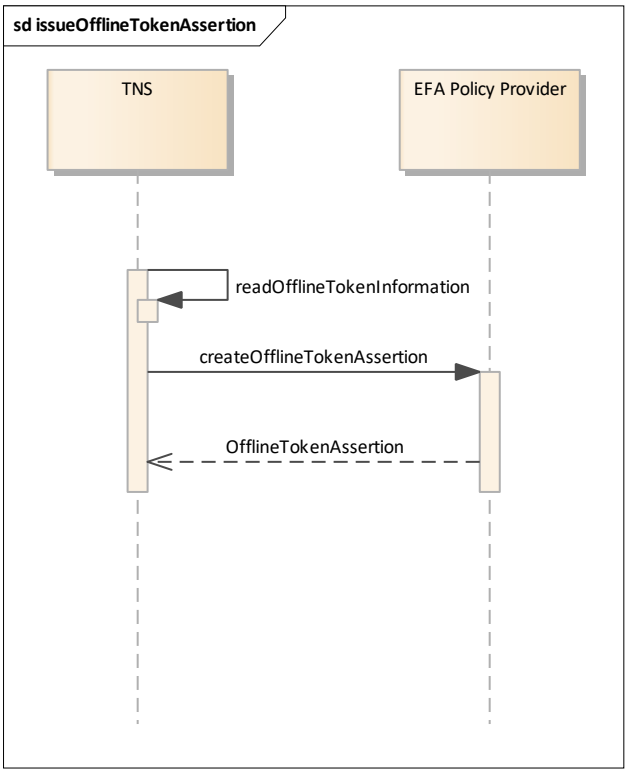

Abbildung 6 Sequenzdiagramm invalidateOfflineTokenPolicy

3.2.2.2. registerOfflineTokenPolicy

Diese Transaktion registriert eine offlineTokenPolicy beim EFA Policy Provider.
Die Transaktion registerOfflineTokenPolicy ist in Ihrer Umsetzung nicht spezifiziert. Die Implementierung ist dem Hersteller überlassen. Da die Policy Teil des Consents ist, empfiehlt sich die Nutzung ähnlicher Mechanismen wie bei der Registrierung der EFA-Berechtigungen.

3.2.2.3. removeOfflineTokenPolicy

Diese Transaktion zieht eine offlineTokenPolicy beim EFA Policy Provider zurück.
Die Transaktionen removeOfflineTokenPolicy ist in Ihrer Umsetzung nicht spezifiziert. Die

Implementierung ist dem Hersteller überlassen. Da die Policy Teil des Consents ist, empfiehlt sich die Nutzung ähnlicher Mechanismen wie bei der Registrierung der EFA-Berechtigungen.

4. Technische Umsetzung

Im Folgenden wird die konkrete technische Umsetzung der Transaktionen beschrieben.

4.1. Technische Umsetzung des Informationsmodells

4.1.1. offlineTokenPolicyRef

Die offlineTokenPolicyRef (Offline-Token-Kennung) ist eine Zeichenkette und muss im Kontext eines Patienten eindeutig sein. Beispielsweise kann hier eine UUID verwendet werden.

Beispiel: 063ee249-7e96-4ab3-8c01-4dcb85366869

Für die Offline-Token-Berechtigungsregel wird ein eigenes XACML Attribut definiert.

SAML Attribute Name	urn:efa:2-0:subject:offlinetoken-id
SAML Beispiel	<saml:Attribute Name="urn:efa:2-0:subject:offlinetoken-id"> <saml:AttributeValue> 063ee249-7e96-4ab3-8c01-4dcb85366869 </saml:AttributeValue> </saml:Attribute>
XACML Target Section	Subject
XACML Attribute ID	urn:efa:2-0:subject:offlinetoken-id
XACML Data Type	http://www.w3.org/2001/XMLSchema#string
XACML Beispiel	<Attribute AttributeId= "urn:efa:2-0:subject:offlinetoken-id" DataType=" http://www.w3.org/2001/XMLSchema#string "> <AttributeValue> 063ee249-7e96-4ab3-8c01-4dcb85366869 </AttributeValue> </Attribute>

4.1.2. offlineTokenPolicy

Die offlineTokenPolicy wird als Erweiterung des Policy-Set im Consent einer EFA abgebildet. Neben der zusätzlichen Offline-Token-Berechtigungsregel macht diese Spezifikation keine weiteren Vorgaben zum Inhalt des Consent Dokuments.

4.1.2.1. Umsetzung als EPPC-G

Für ein Consent nach EPPC-G [2]muss eine Offline-Token-Berechtigungsregel folgende Attribute enthalten.

Die Berechtigungsregel muss genau ein `xacml:Subject` enthalten, welches genau ein `xacml:SubjectMatch` für das Attribut EFA-Offline-Token-Id enthält.

Die Berechtigungsregel muss genau ein `xacml:Resource` enthalten, welches genau ein `xacml:ResourceMatch` für das EPPC-G Attrbut 1.6.21 `Type Code` enthält. Das Attribut muss den Wert 57016-8 für den `code` und 2.16.840.1.113883.6.1 für das `codesystem` enthalten.

Die Berechtigungsregel muss genau ein `xacml:Resource` enthalten, welches genau ein `xacml:ResourceMatch` für das EPPC-G Attribut 1.6.7 `Availability Status` enthält. Das Attribut muss den Wert `urn:oasis:names:tc:ebxml-regrep:StatusType:Approved` enthalten.

Die Berechtigungsregel muss genau ein `xacml:Environment` enthalten, welches genau ein `xacml:EnvironmentMatch` mit der folgenden Ausprägung enthält:

`@MatchId`
urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than-or-equal

`xacml:AttributeValue`
Der Zeitpunkt zu dem die Berechtigungsregel ausläuft

`xacml:AttributeValue/@DataType`
http://www.w3.org/2001/XMLSchema#dateTime.

`xacml:EnvironmentAttributeDesignator/@AttributeId`
Equals *urn:oasis:names:tc:xacml:1.0:environment:current-dateTime*.

`xacml:EnvironmentAttributeDesignator/@DataType`
Equals *http://www.w3.org/2001/XMLSchema#dateTime*.

Beispiel:

```
<Policy PolicyId="2B789DEE-9CB6-11E4-97F9-246A95DB5881"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
    <Target>
        <Subjects>
            <Subject>
                <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">063ee249-7e96-4ab3-8c01-
4dcb85366869</AttributeValue>
                    <SubjectAttributeDesignator AttributeId="urn:efa:2-
0:subject:offlinetoken-id"
                                DataType="http://www.w3.org/2001/XMLSchema#string" />
                </SubjectMatch>
            </SubjectMatch>
        </Subject>
    </Subjects>
    <Resources>
        <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
```

[2] http://wiki.hl7.de/images/EPPC-G_Draft_for_Comment_v04.pdf

```
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI"
                            >urn:oasis:names:tc:ebxml-
regrep:StatusType:Approved</AttributeValue>
                <ResourceAttributeDesignator AttributeId="urn:ihe:iti:xds-
b:2007:availability-status"
                            DataType="http://www.w3.org/2001/XMLSchema#anyURI" />
            </ResourceMatch>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
                <Attribute AttributeId="urn:ihe:iti:xds-b:2007:documententry:type-code"
DataType="urn:hl7-org:v3#CV">
                    <AttributeValue>
                        <hl7:CodedValue code="57016-8"codeSystem="2.16.840.1.113883.6.1"/>
                    </AttributeValue>
                </Attribute>
            </ResourceMatch>
        </Resource>
    </Resources>
    <Environments>
        <Environment>
            <EnvironmentMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:dateTime-
greater-than-or-equal">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#dateTime">2016-
05-08T20:00:00Z</AttributeValue>
                <EnvironmentAttributeDesignator
                            AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-
dateTime"
                            DataType="http://www.w3.org/2001/XMLSchema#dateTime" />
            </EnvironmentMatch>
        </Environment>
    </Environments>
</Target>
<Rule RuleId="2B789DEE-9CB6-11E4-97F9-246A95DB5883" Effect="Permit"/>
</Policy>
```

4.1.2.2. Umsetzung als APPC

Für ein Consent nach APPC [3]muss eine Offline-Token-Berechtigungsregel folgende Attribute enthalten.

Die Berechtigungsregel muss genau ein xacml:Subject enthalten welches genau ein xacml:SubjectMatch für das Attribut EFA-Offline-Token-Id enthält.

Die Berechtigungsregel muss genau ein xacml:Resource enthalten welches genau ein xacml:ResourceMatch für das APPC Attribut 5.6.2.1.5.2.13 Type Code enthält. Das Attribut muss den Wert 57016-8 für den code und 2.16.840.1.113883.6.1 für das codesystem enthalten.

Die Berechtigungsregel muss genau ein xacml:Resource enthalten welches genau ein xacml:ResourceMatch für das APPC Attribut 5.6.2.1.5.1.3 Availability Status enthält. Das Attribut muss den Wert urn:oasis:names:tc:ebxml-regrep:StatusType:Approved enthalten.

Die Berechtigungsregel muss genau ein xacml:Environment enthalten, welches genau ein xacml:EnvironmentMatch mit der folgenden Ausprägung enthält:

@MatchId
urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than-or-equal

xacml:AttributeValue
Der Zeitpunkt zu dem die Berechtigungsregel ausläuft

xacml:AttributeValue/@DataType
http://www.w3.org/2001/XMLSchema#dateTime

xacml:EnvironmentAttributeDesignator/@AttributeId
Equals *urn:oasis:names:tc:xacml:1.0:environment:current-dateTime*

xacml:EnvironmentAttributeDesignator/@DataType
Equals *http://www.w3.org/2001/XMLSchema#dateTime*

Beispiel:

```
<Policy PolicyId="2B789DEE-9CB6-11E4-97F9-246A95DB5881"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
    <Target>
        <Subjects>
            <Subject>
                <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">063ee249-7e96-4ab3-8c01-
4dcb85366869</AttributeValue>
                    <SubjectAttributeDesignator AttributeId="urn:efa:2-
0:subject:offlinetoken-id"
                                DataType="http://www.w3.org/2001/XMLSchema#string" />
                </SubjectMatch>
            </SubjectMatch>
        </Subject>
    </Subjects>
    <Resources>
        <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI"
```

[3] https://wiki.ihe.net/index.php/Advanced_Patient_Privacy_Consents

```
                            >urn:oasis:names:tc:ebxml-
regrep:StatusType:Approved</AttributeValue>
                <ResourceAttributeDesignator
AttributeId="urn:ihe:iti:appc:2016:availability-status"
                            DataType="http://www.w3.org/2001/XMLSchema#anyURI" />
            </ResourceMatch>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
                <Attribute AttributeId="urn:ihe:iti:appc:2016:document-entry:type-code"
DataType="urn:hl7-org:v3#CV">
                    <AttributeValue>
                        <hl7:CodedValue code="57016-8"codeSystem="2.16.840.1.113883.6.1"/>
                    </AttributeValue>
                </Attribute>
            </ResourceMatch>
        </Resource>
    </Resources>
    <Environments>
        <Environment>
            <EnvironmentMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:dateTime-
greater-than-or-equal">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#dateTime">2016-
05-08T20:00:00Z</AttributeValue>
                <EnvironmentAttributeDesignator
                            AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-
dateTime"
                            DataType="http://www.w3.org/2001/XMLSchema#dateTime" />
            </EnvironmentMatch>
        </Environment>
    </Environments>
</Target>
<Rule RuleId="2B789DEE-9CB6-11E4-97F9-246A95DB5883" Effect="Permit"/>
</Policy>
```

4.1.3. offlineTokenAssertion

Eine offlineTokenAssertion erweitert eine EFA Identity Assertion[4] um ein Attribut für offlineTokenRef (siehe. 4.1.2). Die offlineTokenAssertion wird gemäß der ITI-40 verwendet. Sie wird anstelle der EFA-Identity-Assertion geschickt.

Beispiel SAML Assertion (Attributnamen gemäß EPPC-G):

[4] http://wiki.hl7.de/index.php?title=cdaefa:EFA_Identity_Assertion_SAML2_Binding

```
<soap12:Envelope … >
    <soap12:Header … >
        <wsse:Security … >
            <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
                  ID="_2c356d70-1215-42f9-93a0-fc6fab1c966e"
                  IssueInstant="2009-09-21T12:03:28.788Z" Version="2.0">
                  …
                <saml:AttributeStatement>
                    <saml:Attribute
                      FriendlyName=" XSPA Subject"
                      Name="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
                        <saml:AttributeValue
                          xmlns:xs="http://www.w3.org/2001/XMLSchema"
                          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                          xsi:type="xs:string">
                            Dr. Peter Meier
                        </saml:AttributeValue>
                    </saml:Attribute>
                    <saml:Attribute
                      FriendlyName="XSPA Organization"
                      Name="urn:oasis:names:tc:xspa:1.0:subject:organization"
                      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
                        <saml:AttributeValue
                          xmlns:xs="http://www.w3.org/2001/XMLSchema"
                          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                          xsi:type="xs:string">
                            Kreiskrankenhaus Neustadt
                        </saml:AttributeValue>
                    </saml:Attribute>
                    <saml:Attribute
                      FriendlyName="Offlinetoken Id"
                      Name="urn:efa:2-0:subject:offlinetoken-id">
                        <saml:AttributeValue>
                          063ee249-7e96-4ab3-8c01-4dcb85366869
                        </saml:AttributeValue>
                    </saml:Attribute>
                    <saml:Attribute
                      FriendlyName="XSPA Role"
                      Name="urn:oasis:names:tc:xacml:2.0:subject:role"
                      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
                        <saml:AttributeValue
                          xmlns:xs="http://www.w3.org/2001/XMLSchema"
                          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                          xsi:type="xs:string">
                            physician
                        </saml:AttributeValue>
                    </saml:Attribute>
                    <saml:Attribute
                      FriendlyName="XSPA Locality"
                      Name="urn:oasis:names:tc:xspa:1.0:environment:locality"
                      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
                        <saml:AttributeValue
                          xmlns:xs="http://www.w3.org/2001/XMLSchema"
                          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                          xsi:type="xs:string">
                            Kreiskrankenhaus Neustadt
                        </saml:AttributeValue>
                    </saml:Attribute>
                </saml:AttributeStatement>
            </saml:Assertion>
        </wsse:Security>
    <soap12:Envelope … >
<soap12:Header … >
```

4.2. Technisches Binding

4.2.1. registerOfflineToken

Die technische Umsetzung von registerOfflineToken entspricht dem EFA XDS Binding von registerConsent (http://wiki.hl7.de/index.php?title=cdaefa:EFA_Anwendungsdienste_(logische_Spezifikation)#registerConsent).

Die zu registrierende offlineTokenPolicy ist Teil des Consents. Diese muss beim Policy Provider eingetragen werden. Dies findet auch beim Erstellen einer FallAkte Anwendung. Die technische Umsetzung entspricht dann dem EFA XDS Binding vom createEcr (http://wiki.hl7.de/index.php?title=cdaefa:EFA_Anwendungsdienste_(logische_Spezifikation)#createECR)

4.2.2. invalidateOfflineTokenInfo

Die technische Umsetzung von invalidateOfflineTokenInfo entspricht dem EFA XDS Binding von registerConsent (http://wiki.hl7.de/index.php?title=cdaefa:EFA_Anwendungsdienste_(logische_Spezifikation)#registerConsent).
Ist eine bestehende offlineTokenPolicy nicht mehr Teil eines neu eingestellten Consents muss diese beim Policy Provider ungültig werden.

4.2.3. issueOfflineTokenAssertion

Die Anfrage wird an einen *WS-Trust 1.3*[5] konformen Security Token Service gesendet. Um einen Nutzer als EFA-Teilnehmer zu identifizieren, kann eine gültige EFA Identity Assertion im Security zur Authentifizierung im Header der Anfrage übermittelt werden. In diesem Fall kann die EFAIdentity Assertion dazu verwendet die für die OfflineTokenAssertion notwendigen Informationen bereitzustellen. Wenn keine EFA Identity Assertion zur Authentifizierung verwendet wird, müssen die notwendigen Informationen auf andere Weise ermittelt werden (z.B. bei Benutzername+Password Authentifizierung über eine Anfrage am IdP). Die Anfrage hat die Struktur einer *RequestSecurityToken* Nachricht wie sie in *WS-Trust 1.3* definiert ist. Es muss *SOAP Version 1.2* verwendet werden. Die Übermittlung der offlineTokenPolicyRef erfolgt als Claim wie in *WS-Federation 1.2*[6] spezifiziert.

[5] http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html
[6] http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html

Für das *RequestSecurityToken*-Element gelten die Bedingungen in der nachfolgenden Tabelle:

Element or Attribute	Constraints
/wst:RequestSecurityToken/ wst:TokenType	This element is required. The value SHOULD be "http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0".
/wst:RequestSecurityToken/ wst:RequestType	This element is required. The value MUST be "http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue".
/wst:RequestSecurityToken/ wst:Claims/@Dialect	This element is required. The value MUST be http://schemas.xmlsoap.org/ws/2006/12/authorization/authclaims
/auth:ClaimType/@Uri	urn:efa:2-0:subject:offlinetoken-id
/auth:ClaimType/auth:Value	The offlineTokenPolicyRef

Beispiel Anfrage:

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://www.w3.org/2003/05/soap-envelope">
       <Header>
       <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
       ...
     </Security>
  </Header>
<Body>
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
    <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
        <a:EndpointReference>
            <a:Address>https://efaprovider</a:Address>
        </a:EndpointReference>
    </wsp:AppliesTo>
    <trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Bearer</trust:KeyType>
    <trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/Issue</trust:RequestType>
    <trust:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</trust:TokenType>
    <trust:ClaimsDialect="http://schemas.xmlsoap.org/ws/2006/12/authorization/authclaims"
xmlns:auth="http://schemas.xmlsoap.org/ws/2006/12/authorization">
        <auth:ClaimType Uri="urn:efa:2.0:offlinetoken-id">
            <auth:Value>063ee249-7e96-4ab3-8c01-4dcb85366869</auth:Value>
        </auth:ClaimType>
    </trust:Claims>
</trust:RequestSecurityToken>
</Body>
</Envelope>
```

5. Technische Abläufe

Die nachfolgende Tabelle beschreibt ein Mapping der Funktionen auf IHE Transaktionen.

Transaktion	**Binding**
registerOfflineToken (siehe registerConsent, bzw. createECR)	ITI 41
issueOfflineToken	XUA Request Security Token mit Ergänzung Claim zur Darstellung der Offline-Token Kennung
invalidateOfflineToken	ITI 41

5.1.Token registrieren

Der technische Ablauf für die Registrierung des Tokens umfasst die folgenden Schritte:

- Das EFA-Teilnehmersystem erzeugt einen Zufallswert, der im Kontext eines Patienten eindeutig ist.
- Das EFA-Teilnehmersystem erzeugt die offlineTokenPolicy und bettet diese in das consentInfo ein.
- Das EFA-Teilnehmersystem stellt das consentInfo mit der offlineTokenPolicy in die EFA ein.
- Das EFA-Providersystem liest die offlineTokenPolicy aus dem consentInfo aus und speichert diese beim Policy Provider.
- Das EFA-Teilnehmersystem druckt die offlineTokenRef als QR-Code und menschenlesbare Darstellung aus. Hier kann optional der Zweck und die Patienten-ID mit eingebettet werden.

5.1.1. QR Code

Die Offline-Token-Kennung wird dem Patienten in Form eines QR-Code übergeben. Alternativ können projektspezifisch andere Repräsentationsformen gewählt werden. Optional enthält dieser auch die Patienten-ID und den Zweck der FallAkte in folgender Darstellung:

Attribut		Beschreibung
offlineTokenRef	Erforderlich	Die Offline-Token-Kennung aus der entsprechenden Offline-Token-Berechtigungsregel.
purpose	Optional	Zweck der FallAkte aus dem Codesystem 1.2.276.0.76.3.1.81.81.5.6
patientId	Optional	Patienten-ID (siehe Volume 3 IHE ITI TF-3 4.2.3.2.16)

Syntax: offlineTokenRef_patientId_purpose

Beispiel: ed65827f-9273-41ca-ba54-1988c8a63fe_
4c0d5fffb799465^^^&1.3.6.1.4.1.21367.2005.3.7&ISO _Test:Connectathon-2016:Sinusitis-Demo

5.2.Token einlösen

Der technische Ablauf für das Einlösen des Tokens umfasst die folgenden Schritte:

- Das EFA-Teilnehmersystem liest die offlineTokenRef aus dem QR-Code ein.
- Das EFA-Teilnehmersystem fragt mit der offlineTokenRef die offlineTokenAssertion bei einem dedizierten Security Token Service des Policy Providers ab. Die EFA Identity Assertion des EFA-Teilnehmers wird dabei im Security Header und die offlineTokenRef als Claim in einer Request Security Token Nachricht mitgesendet.
- Der Security Token Service ergänzt die Identity Assertion um die Information zur Offline-Token-Kennung.
- Der Security Token Service sendet diese in Form einer signierten offlineTokenAssertion zurück an das EFA-Teilnehmersystem.
- Das EFA-Teilnehmersystem fragt die EFA mit der offlineTokenAssertion, Patienten-ID und Zweck an und erhält eine Liste der Partitionen.
- Das EFA-Teilnehmersystem sucht das consentInfo aus der EFA und öffnet dieses. Ein Zugrif auf andere Dokumente der EFA wird verweigert, da die Berechtigungsregel lediglich einen Zugriff auf das consentInfo erlaubt.
- Das EFA-Teilnehmersystem passt das consentInfo der EFA dem Wunsch des Patienten an.

5.3.Token invalidieren

Der technische Ablauf zum Invalidieren des Tokens umfasst die folgenden Schritte:

- Das EFA-Teilnehmersystem lädt das consentInfo.
- Das EFA-Teilnehmersystem entfernt die offlineTokenPolicy aus dem Consent.
- Das EFA-Teilnehmersystem sendet eine Anfrage zum Aktualisieren des consentInfo an das EFA-Providersystem.
- Das EFA-Providersystem entfernt die verknüpfte offlineTokenPolicy beim Policy-Provider.

6. ATNA Audit Trail

Für das ATNA Audit Trail Binding sind die Events bereits in der EFA 2.0-Spezifikation definiert. Da das Offline-Token Teil des Consents ist, wird die Anlage und das Editieren von Offline-Token über die bestehenden Audit-Codes zur Berechtigungsänderung (Berechtigungen ändern, Berechtigungen initial bei der Fallaktenanlage setzen) protokolliert. Das Einlösen wird mit dem Code EFA-07 (redeemAccessToken) protokolliert.

7. Fehlermeldungen und Warnungen

Die bestehenden Fehlermeldung und Warnungen der EFA (siehe: http://wiki.hl7.de/index.php?title=cdaefa:EFA_Fehlermeldungen_und_Warnungen) werden wie folgt ergänzt:

consentInfo	**Fault: Invalid Lifespan** Die angegebene Gültigkeitsdauer des Offline-Tokens ist nicht gültig, da die aktuelle Lebenszeit der EFA überschritten wird.	registerOfflineToken