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

DRAFT 03-05-2018

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

Inhalt

1. Problemstellung 3
2. Anwendungsszenarien für das Offline Token 3
2.1. Offline Token erstellen 3
2.2. Einlösen eines Offline Tokens 4
2.3. Deaktivieren eines Offline Tokens 5
3. Informationsmodell und Transaktionen 6
3.1. Informationsmodell 7
3.1.1. offlineTokenInfo 7
3.1.2. offlineTokenPolicy 7
3.1.3. offlineTokenPolicyAssertion 8
3.1.4. offlineTokenPolicyRef 8
3.2. Transaktionen 8
3.2.1. EFA Ressource Manager 8
3.2.2. EFA Policy Provider 12
4. Technische Umsetzung 14
4.1. Technische Umsetzung Informationsmodell 14
4.2. Technische Umsetzung Transaktionen 20
5. Technische Abläufe 22
5.1. Token registrieren 22
5.2. Token einlösen 22
5.3. Token invalidieren 22
6. ATNA Audit Trail 23
7. Fehlermeldungen und Warnungen 24

1. Problemstellung

Die technische Spezifikation der EFA 2.0 unterstützt in der aktuellen Version die Vergabe von Berechtigungen auf eine Fallakte über ein strukturiertes Berechtigungsmanagement. Ein bereits berechtigter EFA-Teilnehmer erteilt im Auftrag des Patienten nach schriftlicher Einwilligung des Patienten bei der Anlage der Fallakte oder bei der Anpassung der Berechtigungen einer Fallakte weiteren Einrichtungen oder Personen Zugriff auf die Fallakte. Um Berechtigungen auf eine Akte zu vergeben, muss der Arzt bereits zugriffsberechtigt auf die Fallakte 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 in beispielsweise in Notfallszenarien relevant, schnell neue Leistungserbringer für eine Fallakte 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 Fallakte zu berechtigen. Das Offline Token gilt immer für eine bestimmte Fallakte und ermöglicht EFA-Teilnehmern nach Überreichung des Tokens durch den Patienten, sich selbst ein Zugriffsrecht auf diese Fallakte 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 Fallakte auf, um den Entwicklungsaufwand möglichst gering zu halten und auf bereits etablierte Verfahren aufzusetzen

2. Anwendungsszenarien für das Offline Token

2.1.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 Token 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 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 (z. B. EFA Portal) aus
- Das EFA-Teilnehmersystem erzeugt eine Offline Token Kennung

- Das EFA-Teilnehmersystem erzeugt eine Einlöse-Policy, welches die Kennung und die Fallakte verknüpft
- Das EFA-Teilnehmersystem stellt die Einlöse-Policy in die Fallakte ein
- Das EFA-Providersystem speichert die Einlöse-Policy und deaktiviert bereits bestehende Offline Tokens sofern vorhanden
- Das EFA-Teilnehmersystem generiert einen QR-Codes aus der Kennung
- Der EFA-Teilnehmer aktiviert das Drucken des QR-Codes
- Der EFA-Teilnehmer übergibt das gedruckte Offline Token an den Patienten

2.2.Einlösen eines Offline Tokens

Durch das Aushändigen des QR-Code kann der Patient einem beliebigem EFA-Teilnehmer den Zugriff auf die entsprechende Fallakte ermöglichen. Dazu wird der QR-Code im EFA-Teilnehmersystem eingelesen. Mit der im QR-Code enthaltenen Kennung kann das Client System die notwendigen Metadaten und die Zugriffsberechtigung in Form einer Einlöse-Policy beim EFA-Providersystem abrufen. Zusammen mit der Benutzeridentität (EFA Identity Assertion) hat der einlösende EFA-Teilnehmer vollen Zugriff auf die Fallakte. Jetzt 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.

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

- Der Patient übergibt dem zu berechtigendem EFA-Teilnehmer das Offline Token
- 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 Kennung aus dem QR-Code
- Das EFA-Teilnehmersystem sendet eine Anfrage inkl. der Kennung aus dem Offline Token sowie der Benutzeridentität des EFA-Teilnehmers an das EFA-Providersystem das Offline Token einzulösen
- Das EFA-Providersystem sendet nach Prüfung der Benutzeridentität eine temporäre Einlöse-Policy an das EFA-Teilnehmersystem
- Das EFA-Teilnehmersystem fragt mit der temporären Einlöse-Policy und der Benutzeridentität des EFA-Teilnehmers die aktuellen Berechtigungen der Fallakte beim EFA-Providersystem ab
- Das EFA-Providersystem prüft die Einlöse-Policy und die Benutzeridentität und liefert im Erfolgsfall die aktuellen Berechtigungen an das EFA-Teilnehmersystem zurück
- Das EFA-Teilnehmersystem öffnet die Berechtigungskonfiguration und zeigt diese dem EFA-Teilnehmer an
- Der EFA-Teilnehmer editiert die Berechtigungen und gibt sich oder seiner gesamten Einrichtung Zugriffsrechte auf die Fallakte
- Der EFA-Teilnehmer bestätigt die Anpassung der Berechtigungen nach schriftlicher Einwilligung durch den Patienten
- Das EFA-Teilnehmersystem sendet eine Anfrage inkl. der Einlöse-Policy sowie der Benutzeridentität des EFA-Teilnehmers zur Anpassung der Berechtigungen
- Das EFA-Providersystem prüft die Einlöse-Policy und die Benutzeridentität und speichert im Erfolgsfall die neuen Berechtigungen

2.3.Deaktivieren eines Offline Tokens

Das Deaktivieren eines Offline Tokens erfolgt automatisiert beim Erstellen eines neuen Offline Tokens. Eine Deaktivierung losgelöst vom Erstellen ist ebenfalls möglich.

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 Fallakte zugriffsberechtigten EFA-Teilnehmer mit, dass er seinen Offline Token deaktivieren möchte
- Der EFA-Teilnehmer öffnet die Fallakte und wählt die Funktion zum Deaktivieren des Offline Tokens aus
- Das EFA-Teilnehmersystem sendet eine Anfrage mit der Benutzeridentität des EFA-Teilnehmers an das EFA-Providersystem das Offline Token zu deaktivieren
- Das EFA-Providersystem prüft die Benutzeridentität, deaktiviert das Offline Token und löscht die Einlöse-Policy
- 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.

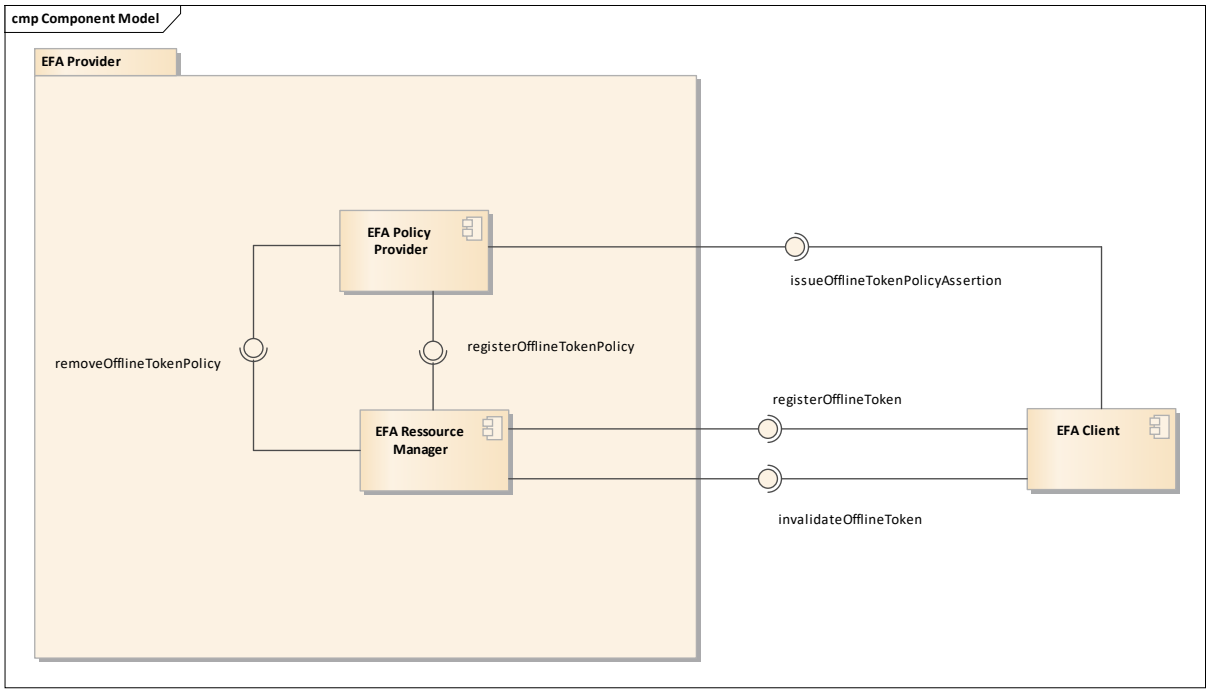


Abbildung 1 Komponentenmodell zum Offline Token

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 Interaktionen zwischen den Komponenten werden in Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** detailliert beschrieben.

Die nachfolgende Tabelle listet die EFA-Akteur 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 Ressource 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 im Folgendem beschriebenen Klassen erweitert.

3.1.1. offlineTokenInfo

Diese Klasse repräsentiert eine strukturierte Abbildung einer Zugriffsregel für eine konkrete EFA in einem Dokument. Die Zugriffsregel ist für jeden EFA-Teilnehmer anwendbar. Zu jeder Zeit hat eine EFA maximal eine gültige offlineTokenInfo Instanz. Eine neue Instanz invalidiert die bestehende Instanz.

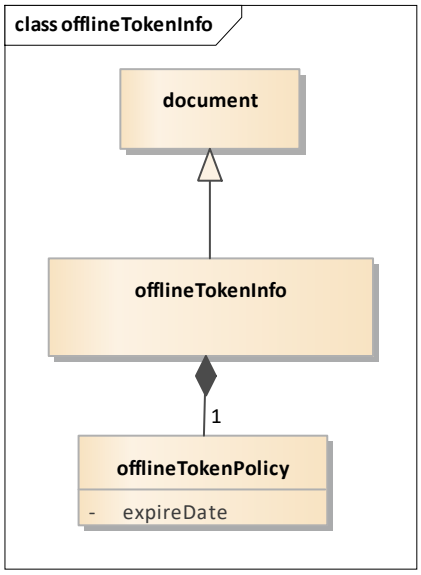


Abbildung 2 Klasse offlineTokenInfo

3.1.2. offlineTokenPolicy

Dieses Sicherheitsobjekt beschreibt eine Zugriffsberechtigung auf eine EFA. Sie ist an einer zufällig erzeugten Kennung gebunden und nicht an einen konkreten EFA-Teilnehmer. Das Endes es Gültigkeitszeitraums eine Offline Token (expireDate) ist optional und darf die aktuelle Gültigkeitsdauer der Fallakte überschreiten. Ist dieses nicht gesetzt, greift immer die Gültigkeit der Fallakte.

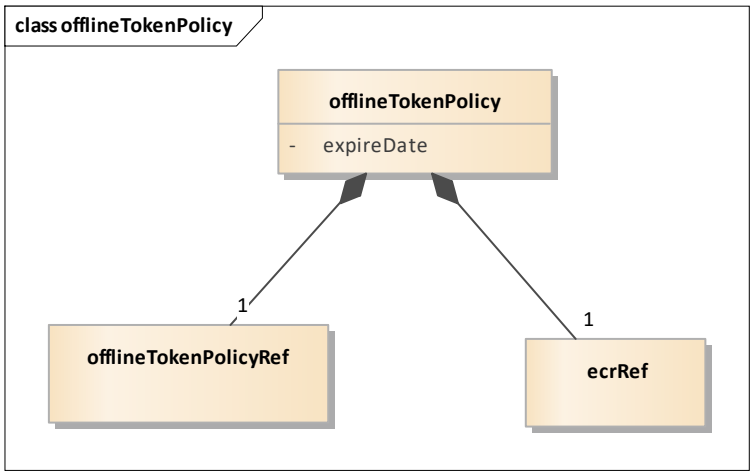


Abbildung 3 Sicherheitsobjekt offlineTokenPolicy

3.1.3. offlineTokenPolicyAssertion

Dieses Sicherheitsobjekt beschreibt eine signierte Form der offlineTokenPolicy und wird als Nachweis für eine Zugriffsberechtigung verwendet. Die Anwendung erfolgt im Rahmen des Policy-Push Verfahrens und ist nur in Verbindung mit der Klasse subjectIdentity gültig. Eine Verifizierbarkeit der Daten einer offlineTokenPolicyAssertion ist nur gegeben, wenn die offlineTokenPolicyAssertion selbst integer und authentisch ist. Daher muss jede offlineTokenPolicyAssertion von der ausstellenden Stelle signiert werden.

3.1.4. offlineTokenPolicyRef

Die Klasse beschreibt eine Kennung an die eine offlineTokenPolicy gebunden ist. Die Kennung setzt sich aus einer kryptografisch sicher erzeugtem Zufallswert und einer Repräsentation der homeCommunityId zusammen. Die Kennung muss gegenüber dritten geheim gehalten werden, da diese einem EFA-Teilnehmer direkten Zugriff auf eine Fallakte ermöglicht.

3.2.Transaktionen

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

3.2.1. EFA Ressource Manager

Der EFA Ressource Manager wird um die beiden folgenden Funktionen erweitert:

- registerOfflineToken
- invalidateOfflineToken

3.2.1.1. registerOfflineToken

Operation	*registerOfflineToken*
Funktionalität	Registriert ein neues offlineTokenInfo für eine bestehende Fallakte. Ein zuvor gültiges offlineTokenInfo-Document wird invalidiert.
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*.
	ecrRef	Eindeutige Identifizierung der Fallakte,
	offlineTokenInfo docRelationship	Das EFA-Teilnehmersystem erzeugt eine offlineTokenInfo. Die darin enthaltene offlineTokenPolicy enthält die homeCommunityID dieses EFA-Providers und eine randomisierte, eindeutige Kennung. offlineTokenInfo MUSS über docRelationship (Wert "ersetzt") mit dem gültigen offlineTokenInfo-Dokument in der Fallakte assoziiert werden wenn ein solches bereits existiert.
Rückgabe	statusInfo	Informationen zur Durchführung der Operation (z.B. aufgetretene Fehler oder für die weitere EFA-Nutzung potenziell relevante Warnungen)
Vorbedingungen	Das übergebene offlineTokenInfo Dokument ist konsistent	
Ablauf	1. Trage im offlineTokenInfo enthaltene offlineTokenPolicy beim Policy Provider ein und lösche vorhanden offlineTokePolicy wenn ein bestehndes offlineTokenInfo ersetzt wird. 2. Speichere und registriere offlineTokenInfo bei Document Repository und Document Registry. 3. Gib statusInfo an den 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 7) werden in die allgemeinen Fehlermeldungen und Warnungen der EFA eingepflegt.	

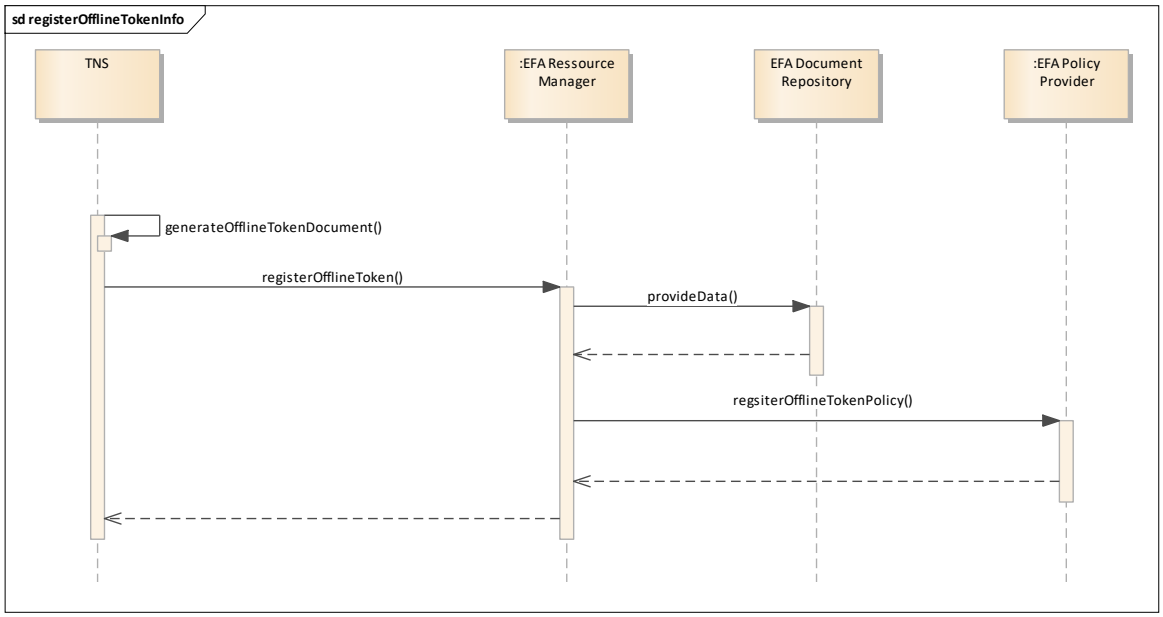


Abbildung 4 Sequenzdiagramm registerOfflineTokenInfo

3.2.1.2. invalidateOfflineToken

Operation	invalidateOfflineToken	
Funktionalität	Invalidieren eines Offline Token Documents in einer Fallakte und zurückziehen der offlineTokenPolicy beim EFA Policy Provicer.	
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*.
	documentID	Eindeutige Identifizierung des zu invalidierenden offlineTokenInfo-document.
Rückgabe	statusInfo	Informationen zur Durchführung der Operation (z.B. aufgetretene Fehler oder für die weitere EFA-Nutzung potenziell relevante Warnungen)
Vorbedingungen		
Ablauf	1. Ändert den Wert des Attributs *Status* von offlineTokenInfo auf *ungültig*.	

	2. Die Bestehenden offlineTokenPolicy wird beim EFA Policy Provider mit removeOfflineTokenPolicy zurückgezogen. 3. Gibt statusInfo an den 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 7) werden in die allgemeinen Fehlermeldungen und Warnungen der EFA eingepflegt.

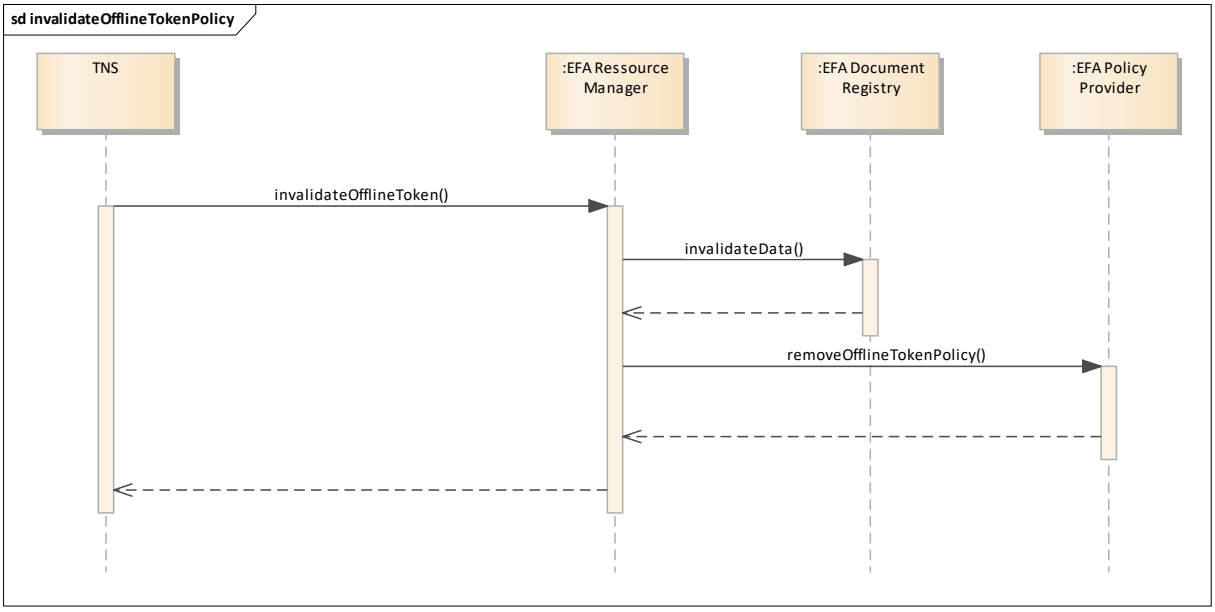


Abbildung 5 Sequenzdiagramm invalidateOfflineTokenPolicy

3.2.2. EFA Policy Provider

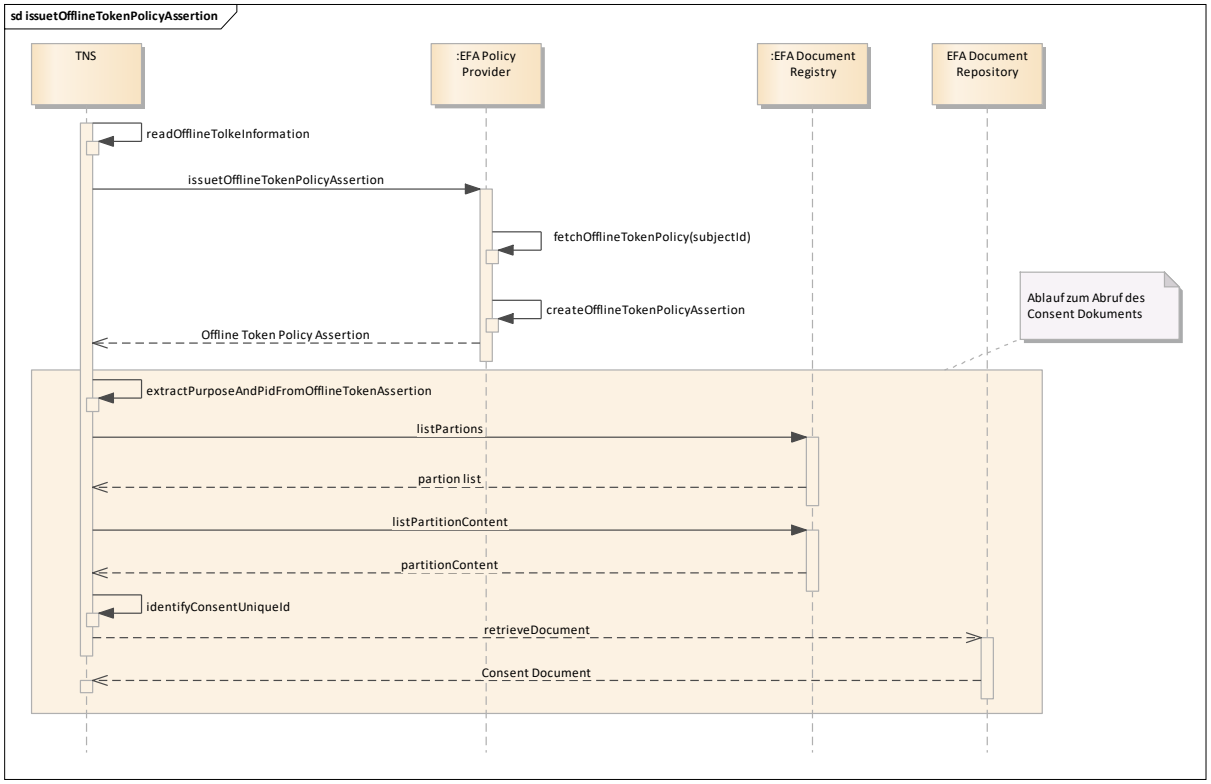
Der EFA Policy Provider wird um die folgenden Funktionen erweitert:

- issueOfflineTokenPolicyAssertion
- registerOfflineTokenPolicy
- removeOfflineTokenPolicy

3.2.2.1.issueOfflineTokenPolicyAssertion

Operation	issueOfflineTokenPolicyAssertion	
Funktionalität	Diese Operation stellt eine offlineTokenAccessPolicy für eine Fallakte 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	offlineTokenPolicyAssertion	Ein Berechtigungstoken, das den Leistungserbringer, der es einreicht, zum Zugriff auf die Fallakte berechtigt.
Vorbedingungen		
Ablauf	1. Suche offlineTokenPolicy anhand der offlineTokenPolicyRef. 2. Erzeuge offlineTokenPolicyAssertion. (Wenn die offlineTokenPolicyRef auf eine andere Affinity-Domain verweist, muss die Anfrage an alle in Frage kommenden Policy Provider weitergeitet werden.) 3. Gebe offlineTokenPolicyAssertion 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.



3.2.2.2. registerOfflineTokenPolicy

Diese Transaktion registriert eine offlineTokenPolicy beim EFA Policy Provider.
Die Transaktionen registerOfflineTokenPolicy ist in Ihrer Umsetzung nicht spezifiziert. Die Implementierung ist dem Hersteller überlassen.

3.2.2.3. removeOfflineTokenPolicy

Diese Transaktion zieht eine Offline Token Policy beim EFA Policy Provider zurück
Die Transaktionen removeOfflineTokenPolicyist in Ihrer Umsetzung nicht spezifiziert. Die Implementierung ist dem Hersteller überlassen.

4. Technische Umsetzung

Die technische Umsetzung orientiert sich an der Lösung der ELGA[1] zur e-medikation. Die Gesamtarchitektur der ELGA sieht vor, dass Apotheker dazu berechtigt sind auf mit einem Rezept verknüpfte Dokumente zuzugreifen und die verordnete Medikation einzutragen. Zu diesem Zweck befindet sich eine eindeutige Identifikationsnummer (e-med-id) auf dem Rezept. Dieses Szenario entspricht im Wesentlichen dem eines Offline Tokens.

4.1. Technische Umsetzung Informationsmodell

4.1.1. offlineTokenPolicyRef

Technisch wird die offlineTokenPolicyRef (Kennung) in der Form UUID^(HomeCommunityId abgebildet. Die homeCommunityId ist für ein Peer 2 Peer Szenario relevant.

Beispiel: 063ee249-7e96-4ab3-8c01-4dcb85366869^1.222.3333

4.1.2. offlineTokenPolicy

Die offlineTokenPolicy (Einlöse-Policy) wird durch eine XACML PolicySet mit genau einer Policy abgebildet. Die Struktur der Policy ist in Abschnitt „Policy Attachment for offline token“ definiert.

Element or Attribute	Opt.	Constraints
PolicySet	R	
@PolicySetId	R	Shall be of type UUID or OID. Shall not be URN encoded.
@PolicyCombiningAlgId	R	Shall be *urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides*.
Target	R	
Resources	R	
Resource	R	Shall contain at least: • ResourceMatch for EFA Folder classification, • ResourceMatch for purpose classification, • ResourceMatch for patientId.
Actions	R	May contain Action elements that qualify the use of specific operations in the context of an EFA.

	Policy	R	Shall be the policy for the subject stated in the ECR Policy Assertion. This element shall conform the Policy Attachment for offline token.

4.1.2.1. Policy Attachment for offline token

Element or Attribute	Opt.	Constraints
Policy	R	
@PolicyId	R	Shall be of type UUID or OID. Shall not be URN encoded.
@RuleCombiningAlgId	R	Shall be *urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides*
Target	R	
Subjects	R	
Subject	R	
SubjectMatch	R	Shall contain at leat one of the following SubjectMatch element SubjectMatch for EFA Identity Assertion NameID. The offlineTokenPolicyRef MUST be used.
Resources	R	
Resource	R	
ResourceMatch	R	Restricts access to open ECRs. This match relates to ecrStatus. It applies the IHE-D Cookbook XACML binding of DocumentEntry.availabilityStatus.

@MatchId	R	Shall be *urn:oasis:names:tc:xacml:1.0:function:anyURI-equal*
AttributeValue	R	Shall be *urn:oasis:names:tc:ebxml-regrep:StatusType:Approved*
@DataType	R	Shall be *http://www.w3.org/2001/XMLSchema#anyURI*
ResourceAttributeDesignator	R	
@AttributeId	R	Shall be *urn:ihe:iti:xds-b:2007:availability-status*
@DataType	R	Shall be *http://www.w3.org/2001/XMLSchema#anyURI*
Environments	O	If relevant for the use case, the expiration time can be defined. Otherwise the access is restricted by the ECR expiration time.
Environment	O	
EnvironmentMatch	R	Verifies, that the current date is before the date of expiry, i. e. the grace period has not started.
@MatchId	R	Shall be *urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than-or-equal*
AttributeValue	R	Shall be the point in time when offline token should expire.
@DataType	R	Shall be *http://www.w3.org/2001/XMLSchema#dateTime*
EnvironmentAttributeDesignator	R	

						@AttributeId	R	Shall be *urn:oasis:names:tc:xacml:1.0:environment:current-dateTime*
						@DataType	R	Shall be *http://www.w3.org/2001/XMLSchema#dateTime*

4.1.3. offlineTokePolicyAssertion

Die offlineTokenPolicyAssertion wird technisch als signierte Form eines XACML 2.0 PolicySet abgebildet.

Assertion Element	Opt	Usage Convention
@Version	R	MUST be “2.0”
@ID	R	URN encoded unique identifier (UUID) of the assertion
@IssueInstant	R	Time instant of issuance in UTC
Issuer	R	Address URI that identifies the endpoint of the issuing service
Subject	R	This element defines the subject confirmation method of the user in order to use the Policy Assertion as a supporting token. Moreover, it defines the subject name identifier that accords with the user identity from an Identity Assertion.
NameID	R	Identifier of the HP given in the Identity Assertion encoded as an X.509 subject name, an e-Mail address or as a string value (unspecified format). Only identifiers must be used that can be long-term tracked back to an individual person.
@Format	R	MUST be *urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified* or *urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName* or *urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress* For providing an OID as a subject identifier

			the *unspecified* format must be used. The OID must be provided as a string encoded in ISO format.
SubjectConfirmation		R	
	@Method	R	This element MUST hold a URI reference that identifies a protocol to be used to authenticate the subject.[SAML2.0core] The value of this element MUST be set to *urn:oasis:names:tc:SAML:2.0:cm:holder-of-key*
	SubjectConfirmationData	R	
		ds:KeyInfo	R	The XML Signature [XMLSignature] element MUST embed a cryptographic key that is only held by the user. This can be the user's public key (ds:KeyValue/ds:RSAKeyValue), the complete user's X.509 certificate (ds:X509Data/ds:X509Certificate), or an encrypted symmetric key (xenc:EncryptedKey [XMLEncryption]). This symmetric key MUST be encrypted by using the public key of the consumer service's certificate [eFA PKI 1.2].
Conditions		R	
	@NotBefore	R	time instant from which the assertion is useable. This condition MUST be assessed by the assertion consumer to proof the validity of the assertion.
	@NotOnOrAfter	R	Time instant at which the assertion expires. This condition MUST be assessed by the assertion consumer to proof the validity of the assertion. The maximum validity timespan for a Policy Assertion MUST NOT be more than 4 hours.
XACMLPolicyStatement		R	
	PolicySet	R	PolicySet that expresses the given authorization (offlineTokenPolicy MUST be used).
ds:Signature		R	See Assertion Signature: http://wiki.hl7.de/index.php?title=cdaefa:EFA_Policy_Assertion_SAML2_Binding

4.1.4. offlineTokenInfo

Die technische Umsetzung von offlineTokenInfo durch Advanced Patient Privacy Consents Content Module (IHE ITI Supplement APPC[2], Section 5.6). Das im Dokument verwendete PolicySet entspricht der offlineTokenPolicy.

Für das DocumentEntry wird noch ein TypeCode benötigt, damit das OfflineToken-Dokument auch als solches identifiziert werden kann. Diese Abstimmung wird mit der ValueSet-Gruppe von IHE-D durchgeführt

[2] http://ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_Suppl_APPC.pdf

4.2. Technische Umsetzung Transaktionen

4.2.1. registerOfflineToken

Technisch wird die Transaktion registerOfflineToken IHE-XDS Provide and Register Document Set[3] (ITI-41) umgesetzt. Es wird ein Offline Token Dokument in die Elektronische Fallakte eingestellt.

registerOfflineToken	ITI-41	Constraints
context	SAML Identity Assertion within the SOAP Security Header	see IHE Cookbook
ecrRef	XDS Folder Attribute: patientID XDS Folder Attribute: codeList	The codeList shall contain all purpose codes as assigned to the ECR that is to be aligned
offlineTolkenInfo	XDS Document	see OfflineTokenInfo
docRelationship	RPLC-Association	The given consentInfo SHALL be associated with the approved offlineTokenInfo-DocumentEntry of the case record.

4.2.2. invalidateOfflineTokenInfo

Die technische Umsetzung von invalidateOfflineTokenInfo entspricht den EFA XDS Binding von invalidateData (http://wiki.hl7.de/index.php?title=cdaefa:EFA_XDS_DocumentRegistry). Zusätzlich muss die mit dem Dokument verknüpfte offlineTokenPolicy vom EFA Provider gelöscht werden.

4.2.3. issueOfflineTokenPolicyAssertion

Die Anfrage wird an einen WS-Trust 1.3[4] konformen Security Token Service gesendet. Um einen Nutzer als EFA-Teilnehmer zu identifizieren muss eine gültige EFA Identity Assertion im Security Header der Anfrage mitgesendet werden. Die Anfrage hat die Struktur einer *RequestSecurityToken* Nachricht wie sie in *WS-Trust 1.3 definiert ist*. Es sollte *SOAP Version 1.2 verwendet werden.* Die Übermittlung der offlineTokenPolicyRef erfolgt als Claim wie in WS-Federation 1.2 spezifiziert.

---

[3] http://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol2b.pdf#nameddest=3_41_Provide_and_Register_Docum

[4] http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-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	Claim URI muss in Form von URI festgelegt werden, z.B. "urn:ihe:efa:offlinetoken:id".
/auth:ClaimType/auth:Value	The offlineTokenPolicyRef

Beispiel:

```
<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:ihe:efa:offlinetoken:id">
            <auth:Value>063ee249-7e96-4ab3-8c01-4dcb85366869^1.222.3333</auth:Value>
        </auth:ClaimType>
    </trust:Claims>
</trust:RequestSecurityToken>
```

5. Technische Abläufe

5.1.Token registrieren

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

- Das EFA-Teilnehmersystem erzeugt einen kryptografisch starken Zufallswert und kombiniert diesen mit der eigenen homeComunityId zu einer Kennung.
- Das EFA-Teilnehmersystem erzeugt ein Offline Token Dokument mit einer Zugriffs-Policy für eine EFA und der generierten Kennung als subject-id der Policy.
- Das EFA-Teilnehmersystem stellt das Offline Token Dokument in die entsprechende EFA ein.
- Das EFA-Providersystem liest die Policy aus dem Offline Token Dokument aus und speichert diese beim Policy Provider.
- Das EFA-Teilnehmersystem druckt die Kennung als QR-Code aus.

5.2.Token einlösen

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

- Das EFA-Teilnehmersystem liest die Kennung aus dem QR-Code ein.
- Das EFA-Teilnehmersystem fragt mit der Kennung die Policy Assertion bei einem dedizierten Security Token Service des Policy Providers ab. Die EFA Identity Assertion des EFA-Teilnehmers wird dabei im Security Header und die Kennung als Claim in einer Request Security Token Nachricht mitgesendet.
- Der Security Token Service sucht die passende Policy für die gesendete Kennung und sendet diese in Form einer signierten Policy Assertion zurück an das EFA-Teilnehmersystem. Entspricht die mitgesendete homeCommunityId nicht der des Security Token Service wird die Nachricht an den entsprechenden Security Token Service weitergeleitet.
- Das EFA-Teilnehmersystem extrahiert den Zweck sowie die Patienten-ID aus der Policy Assertion und kann somit eine EFA abrufen. Mit der Policy Assertion und der EFA Identity Assertion hat das EFA-Teilnehmersystem Zugriff auf die EFA.
- Das EFA-Teilnehmersystem sucht das Consent Dokument aus der EFA und öffnet dieses.
- Das EFA-Teilnehmersystem passt das Consent Dokument 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 ermittelt die EntryUUID des Offline Token Dokuments.
- Das EFA-Teilnehmersystem invalidiert den DocumentEntry des Offline Token Dokuments.
- Das EFA-Providersystem entfernt die verknüpfte Offline Token Policy beim Policy Provider.

6. ATNA Audit Trail

Für das ATNA Audit Trail Binding werden zusätzliche Events spezifiziert.

EFA Service	**eventTypeCode. codeSystemName**	**eventTypeCode. code**	**eventTypeCode. displayName**
registerOfflineToken	EFAv2 Transaction	EFA-12	registerOfflineToken
invalidateOfflineToken	EFAv2 Transaction	EFA-13	invalidateOfflineToken
issueOfflineTokenPolicyAssertion	EFAv2 Transaction	EFA-14	issueOfflineTokenPolicyAssertion
registerOfflineTokenPolicy	EFAv2 Transaction	EFA-15	registerOfflineTokenPolicy
removeOfflineTokenPolicy	EFAv2 Transaction	EFA-16	removeOfflineTokenPolicy

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:

OfflineTokenInfo	**Fault: Invalid SubjectID** Die subjectID entspricht nicht den Anforderungen.	registerOfflineToken
	Fault: Inconsistent PID Der im *offlineTokenInfo* benannte Patient kann nicht auf die an die bestehende Akten gebundene patientID abgebildet werden.	
	Fault: Invalid Lifespan (nur relevant, wenn Gültigkeitsdauer gesetzt) Die angegebene Gültigkeitdauer des Offline Tokens (und damit der daran hängenden Akte) ist nicht gültig, da sie sie die aktuelle Lebenszeit der EFA überschreitet.	
	Fault: Inconsistent Offline Token Info Im *offlineTokenInfo* enthaltene Angaben sind inkonsistent oder gar widersprüchlich zu anderen Argumenten des Operationsaufrufs.	