gematik

C_12349_Anlage

Inhaltsverzeichnis

1 Änderungsbeschreibung....3

2 Änderungen in gemKPT_TI-Föderation....4

2.1 Anpassungen in Kapitel "5.6 Unterstützung Single-Sign-On auf Anwendungsebene"....4

2.2 Neues Kapitel "5.6.2 IDP gesteuertes Single-Sign-On (idp_controlled)"....4

2.3 Neues Kapitel "5.6.2.1 Prinzipieller Ablauf bei IDP controlled SSO"....5

2.4 Neues Kapitel "5.6.2.2 Technischer Ablauf bei IDP controlled SSO"....8

2.5 Neues Kapitel "5.6.2.3 Ablaufbeschreibung bei IDP controlled SSO"....10

3 Änderungen in gemSpec_IDP_Sek....19

3.1 Anpassungen im Kapitel 4.2 "Entity Statement des sektoralen IDP"....19

3.2 Anpassungen Kapitel "5.2.2 Rahmenbedingungen"....24

3.3 neues Kapitel "5.2.4 IDP-controlled-SSO auf Anwendungsebene"....25

3.3.1 neues Kapitel 5.2.4.1 "PAR-Endpunkt"....25

3.3.2 neues Kapitel 5.2.4.1.1 "PAR-Endpunkt Eingangsdaten"....25

3.3.3 neues Kapitel 5.2.4.1.2 "PAR-Endpunkt Ausgangsdaten"....30

3.3.4 Neues Kapitel "5.2.4.2 Code Retrieval Endpunkt"....30

3.3.5 neues Kapitel "5.2.4.2.1 Code Retrieval Endpunkt Eingangsdaten"....31

3.3.6 neues Kapitel "5.2.4.2.2 Code Retrieval Endpunkt Ausgangsdaten"....31

3.3.7 neues Kapitel 5.2.4.3 Erweiterung Token-Endpunkt....32

3.3.8 neues Kapitel 5.2.4.4 "SSO Session Autorisierungen"....32

3.3.9 neues Kapitel 4.2.4.5 "Nutzerspezifische Einstellungen"....35

4 Änderungen in der gemSpec_IDP_FD....37

4.1 Anpassungen in Kapitel "3.4 Entity Statements"....37

4.2 Anpassungen in Kapitel 3.5 "Anfrage von ID_TOKEN beim sektoralen Identity Provider"....42

4.3 Anpassungen in Kapitel 3.8 ACCESS_TOKEN....44

5 Änderungen in gemSpec_IDP_FedMaster....45

5.1 Änderungen in Kapitel 4.3 "Anforderungen an den Trust Anchor der TI-Föderation (Federation Master)"....45

6 Änderungen in gemSpec_Perf....46

6.1 Änderungen in Kapitel 3.1.2 "Betriebsdatenerfassung v2 Spezifika Identity Provider"....46

1 Änderungsbeschreibung

Die bisher geschaffene Möglichkeit zur Verwendung von SSO auf Anwendungsebene setzt voraus, dass das Authenticator Modul auf dem Endgerät eine laufende Authentisierungssitzung hat, dass SSO Autorisierungsanforderungen aus der Anwendung auch ohne erneute Authentisierung durchlaufen werden.

Diese Art von SSO auf Anwendungsebene hat mit der Herausforderung zu kämpfen, dass bei einer Mehr-App-Strategie, also wenn das Authenticator Modul nicht in der FdV-Anwendung enthalten ist, sowie bei der Cross-device Authentisierung, App Sprünge die Benutzererfahrung negativ beeinflussen. Nicht nur weil zwischen den Anwendungen hin und her gesprungen wird, sondern auch weil der alleinstehende Authenticator nach inaktivität die Anmeldung beendet, ein Nutzer also beim Wechsel in ein anderes Modul innerhalb der Anwendung nach wenigen Minuten sich doch erneut authentisieren muss.

Motiviert durch diese identifizierten Defizite, soll mit diesem Änderungseintrag ein alternatives SSO auf Anwendungsebene unter der Bezeichnung "IDP-controlled-SSO" spezifiziert werden, dass durch Zusammenspiel von Anwendung, Authorization Server und sektoralen IDP eine bessere Benutzererfahrung ermöglichen soll.

2 Änderungen in gemKPT_TI-Föderation

2.1 Anpassungen in Kapitel "5.6 Unterstützung Single-Sign-On auf Anwendungsebene"

Der folgende Abschnitt wird in das neues Kapitel "5.6.1 Frontend-des-Versicherten gesteuertes Single-Sign-On (epfdv_controlled)" verschoben

Beim Single-Sign-On (SSO) authentifiziert sich der Nutzer spätestens beim erstmaligen Aufruf eines TI-Fachdienstes innerhalb einer Anwendung gegenüber dem sektoralen Identity Provider (IDP). Anschließend kann der Nutzer für einen definierten Zeitraum weitere TI-Fachdienste sowie anwendungsspezifische Funktionen der Krankenversicherung nutzen, ohne für jeden einzelnen Fachdienst erneut eine aktive Authentifizierung durchführen zu müssen.
Der sektorale Identity Provider stellt dabei sicher, dass für jeden TI-Fachdienst ein eigenes ID-Token ausgestellt wird. Dadurch bleiben die Vertrauensbeziehungen und die Identitätsinformationen der einzelnen Fachdienste voneinander getrennt.
Der Verzicht auf eine aktive Nutzerinteraktion bei jeder Authentifizierung eines TI-Fachdienstes erfordert zusätzliche Sicherheitsmaßnahmen. Diese Maßnahmen müssen sicherstellen, dass die für den jeweiligen Fachdienst geforderte Widerstandsfähigkeit gegenüber dem relevanten Angriffspotenzial weiterhin gewährleistet bleibt.
Eine mögliche spezifikationskonforme Umsetzung besteht in der Verwendung einer vom sektoralen Identity Provider erzeugten Session-ID sowie eines kryptographischen Schlüsselpaars, das sowohl an die Session-ID als auch an die konkrete Instanz des Anwendungskontexts gebunden ist.
Informationen zu den SSO-Präferenzen des Nutzers sowie zur Identität der anfordernden Anwendung werden standardkonform über den Request-Parameter "authorization_details"[1] übermittelt (siehe [RFC9396] sowie [OAuth 2.0 Rich Authorization Requests]).
Ein Single-Sign-On ist derzeit ausschließlich für Fachdienste zulässig, die innerhalb des ePA-FdV eingebunden sind. Die nachfolgende Beschreibung erläutert die Umsetzung daher exemplarisch anhand eines ePA-FdV mit integrierten TI-Fachdiensten.

Die Unterkapitel 5.6.1-5.6.4 werden in das neue Kapitel 5.6.1 verschoben:

- 5.6.1 zu 5.6.1.1
- 5.6.2 zu 5.6.1.2
- 5.6.3 zu 5.6.1.3
- 5.6.4 zu 5.6.1.4

2.2 Neues Kapitel "5.6.2 IDP gesteuertes Single-Sign-On (idp_controlled)"

Beim IDP-gesteuerten Sigle-Sign-On erfolgt die Steuerung der in der Anwendung nutzbaren Fachdienste nicht durch die Anwendung selbst, sondern wird außerhalb der Anwendung im sektoralen IDP und dessen Authenticator Modul durch den authentisierten Nutzer gesteuert.

Dazu wird von der Anwendung bei jedem Start ein neues kurzlebiges Schlüsselpaar erzeugt, dass die Authentizität der Anwendung innerhalb einer Sitzung gegenüber dem IDP, über die einzelnen Authentisierungananfragen hinweg, absichert. Dabei wird der in der ersten Authentisierung mitgesendete öffentliche Schlüssel an die SSO Session gebunden, so dass bei folgenden, aus der selben Anwendungssitzung heraus gestarteten, Authentisierungen nachgewiesen werden kann, dass es noch die selbe laufende SSO-Sitzung ist. Technisch wird dies mittels einem an DPoP (demonstrating proof of posession) angelehnten Verfahren gegenüber dem IDP realisiert.

DPoP nach RFC9449 wird hier absichtlich nicht verwendet, da dies bereits für Abläufe im Zusammenhang mit der Zero Trust Architektur für die Telematikinfrastruktur [gemKPT_Zero_Trust] reserviert ist.

Da die Anwendung nicht direkt mit dem sektoralen IDP kommuniziert, ist hier das Zusammenspiel von allen beteiligten Komponenten von Nöten:

- die Anwendung übermittelt im Authorization Request die Anwendungsinformationen, welche Fachdienste in der Anwendung gebündelt authentisiert werden können sollen
- der Fachdienst Authorization Server authentisiert den Client
- der Fachdienst Authorization Server leitet die von der Anwendung in den authorization_details im Authorization Request gelieferten Anwendungsinformationen im PAR an den sektoralen IDP weiter und ergänzt diese Informationen um die Attestierung der Client-Authentizität
- der IDP prüft ob der Fachdienst zur angegebenen Client-Attestierung berechtigt ist
- der IDP holt vom Nutzer die Einwilligung zur Nutzung der angeforderten Module innerhalb der Anwendung ein
- der Fachdienst Authorization Server kann optional das ID-Token ohne AuthCode Übermittlung vom IDP abrufen
- der Nutzer kann im Authenticator Modul SSO Präferenzen verwalten

Die volle Nutzbarkeit von IDP gesteuertem SSO ist also nur gegeben, wenn alle daran beteiligten Komponenten dies unterstützen. Daher sollten Anwendungen dieses Verfahren nur zur Nutzung anbieten, sofern alle beteiligten Systeme Kompatibilität mit diesem Feature im openid-federation Entity Statement signalisiert haben, da es sonst zu unerwarteten Benutzererfahrungen oder gar schlimmstenfalls zu Fehlermeldungen führen kann.

2.3 Neues Kapitel "5.6.2.1 Prinzipieller Ablauf bei IDP controlled SSO"

Im folgenden werden die Abläufe zum Start und der Verwendung einer SSO Session angerissen um das in den darauffolgenden Kapiteln enthaltene Ablaufdiagramm und Ablaufschritte im Gesamtkontext leichter verstehen zu können.

Bei Start der Anwendung

Beim Start der Anwendung findet die Initialisierung statt, dazu werden die folgenden Schritte ausgeführt:

- Erzeugung eines kurzlebigen Schlüsselpares zu Identifikation der SSO-Sitzung, das nach längerer Inaktivität von der Anwendung verworfen werden muss.

- Laden des öffentlichen Enc-Schlüssels zum Zweck des verschlüsselten Transportes zum sektoralen IDP
- die Anwendungsinformationen werden zusammengestellt, diese umfassen Name der Anwendung, Name des Gerätes und welche Fachdienste innerhalb dieser Anwendung technisch nutzen können sollen

Bei dem ersten Funktionaufruf eines Fachdienstes aus der Anwendung durch den Nutzer

Beim ersten Funktionsaufruf eines Fachdienstes (oder eines Kostenträger-eigenen Dienstes, der in Vertrauensstellung zum sektoralen IDP dieses Kostenträgers steht) erfolgen die folgenden Schritte:

- die authorization_details für den Authorization Request zum Fachdienst werden mit den Anwendungsinformationen befüllt
- die Anwendung sendet den Authorization Request an den Authorization Server
- der Authorization Server kopiert die client_information aus den authorization_details in die authorization_details, die er im Pushed Authorization Request (PAR) an den sektoralenm IDP mitsenden wird. Er ergänzt diese authorization_details um die client_assertion, in der er Auskunft darüber gibt wie sicher der Client (die Anwendung) gegenüber dem Authorization Server authentifiziert wurde
- der IDP validiert den PAR Aufruf und die darin enthaltenden SSO-Informationen, liefert dem Authorization Server die URI-PAR zum IDP zurück
- der Authorization Server liefert diese URI-PAR an die Anwendung zurück
- die Anwendung wechselt durch Aufruf der URI-PAR in das Authenticator-Modul
- das Authenticator Modul ruft die zur URI-PAR gehörigen Authentisierungsanfrage auf und holt vom Nutzer die Einwilligung zum Umfang der innerhalb der Anwendung zulässigen SSO-Nutzung zur Authentisierung ein
- der IDP bindet die erteilten Einwilligungen an die SSO Session und liefert dem Authenticator den Authorization Code für den Authorization Server zurück
- das Authenticator Modul übergibt den Authorization Code per Intent zurück an die Anwendung
- die Anwendung sendet den Authorization Code an den Authorization Server des Fachdienstes
- der Authorization Server übermittelt den Authorization Code an den IDP, der im Gegenzug ID-Token, Access-Token und authorization_details mit session_information an den Authorization Server zurück liefert
- der Authorization Server verarbeitet die im ID-Token gelieferten Nutzerinformationen und kopiert die in den authorization_details enthaltenden session_information und liefert der Anwendung der zu diesem Aufrufkontext gehörigen Authentication Code zurück
- die Anwendung ruft die zum Authorization Code gehörigen Access-Token, Refresh-Token und authorization_deatils vom Authorization Server ab
- aus den empfangenen authorization_details werden die session_information extrahiert und der SSO-Anwendungskontext damit vervollständigt
- das Modul der Fachanwendung greift mit dem Access Token auf den Fachdienst zu
- nach Ablauf des Access Token erlangt das Fachmodul gegen Vorlage des Refresh Token ein neues Access Token

Weitere Aufrufe von Fachdiensten innerhalb der selben Sitzung

Bei der Nutzung weiterer Fachmodule innerhalb der Anwendung finden die folgenden Schritte statt:

- die authorization_details für den Authorization Request zum Fachdienst werden mit den Anwendungsinformationen befüllt
- die Anwendung sendet den Authorization Request an den Authorization Server
- der Authorization Server kopiert die client_information aus den authorization_details in die authorization_details, die er im Pushed Authorization Request (PAR) an den sektoralenm IDP mitsenden wird ergänzt diese authorization_details um die client_assertion, in der er Auskunft darüber gibt wie sicher der Client (die Anwendung) gegenüber dem Authorization Server authentifiziert wurde
- der IDP validiert den PAR Aufruf und die darin enthaltenden SSO-Informationen, liefert dem Authorization Server die URI-PAR zum IDP und gegebenenfalls einen preauthorization-hint zurück
- wird das preauthorization Feature unterstützt, so kann der Authorization Server direkt selbst den Authorization Code vom IDP abrufen und der Anwendung den Anwendungs Authorization Code zurücksenden
- wird das preauthorization Feature nicht unterstützt, findet ein App-Sprung in das Authenticator Modul und zurück zur Anwendung statt, an dessen Ende die Anwendung den Anwendungs Authorization Code erhält
- die Anwendung sendet den Authorization Code an den Authorization Server des Fachdienstes
- der Authorization Server übermittelt den Authorization Code an den IDP, der im Gegenzug ID-Token, Access-Token und authorization_details mit session_information an den Authorization Server zurück liefert
- der Authorization Server verarbeitet die im ID-Token gelieferten Nutzerinformationen und kopiert die in den authorization_details enthaltenden session_information und liefert der Anwendung der zu diesem Aufrufkontext gehörigen Authentication Code zurück
- die Anwendung ruft die zum Authorization Code gehörigen Access-Token, Refresh-Token und authorization_deatils vom Authorization Server ab
- aus den empfangenen authorization_details werden die session_information extrahiert und der SSO-Anwendungskontext aktualisiert
- das Modul der Fachanwendung greift mit dem Access Token auf den Fachdienst zu
- nach Ablauf des Access Token erlangt das Fachmodul gegen Vorlage des Refresh Token ein neues Access Token

Kontrollmöglichkeiten durch den Nutzer am sektioralen IDP

Da die SSO-Präferenzen nun (persistent) im IDP gehalten werden, haben diese Vorrang vor eventuell in der Anwendung gesetzten Präferenzen. Daher stehen dem Anwender nach Authentisierung in seinem Authenticator Modul folgende Einstellungen zur Verfügung:

- einsehen und beenden von laufendenden SSO-Sitzungen
- einsehen, ändern und löschen von persistierten SSO-Präferenzen je Anwendung

- einsehen und ändern der Nutzerabhängigen Einstellungen zu den maximal gleichzeitig laufenden SSO-Sitzungen, deren Beendigung bei Inaktivität und Verhalten überschreiten der maximal gleichzeitig gültigen Sitzungen (z. B. beenden ältester Sitzung, anzeigen aller laufenden Sitzungen, beenden aller Sitzungen oder Fehlermeldung)

2.4 Neues Kapitel "5.6.2.2 Technischer Ablauf bei IDP controlled SSO"

Im folgenden wird der technische Ablauf exemplarisch an einer Anwendung gezeigt, die nach Start der Anwendung einen Kasseneigenen Dienst aufruft und anschließend mit der gestartetet SSO-Sitzung auf die ePA zugreift.

Der besseren Lesbarkeit halber, wurde das Sequenzdiagramm so gestaltet, dass die Lifelines für den 2. Abschnitt wiederholt wurden. Im Einzelnen wird kurz angerissen, welche Abwecihungen zu breücksichtigen sind, wenn der Nutzer SSO nicht zu beginn gestartet hat oder vorher einen anderen Fachdienst in der Anwendung benutzt hat, der keine SSO Session etablieren konnte.

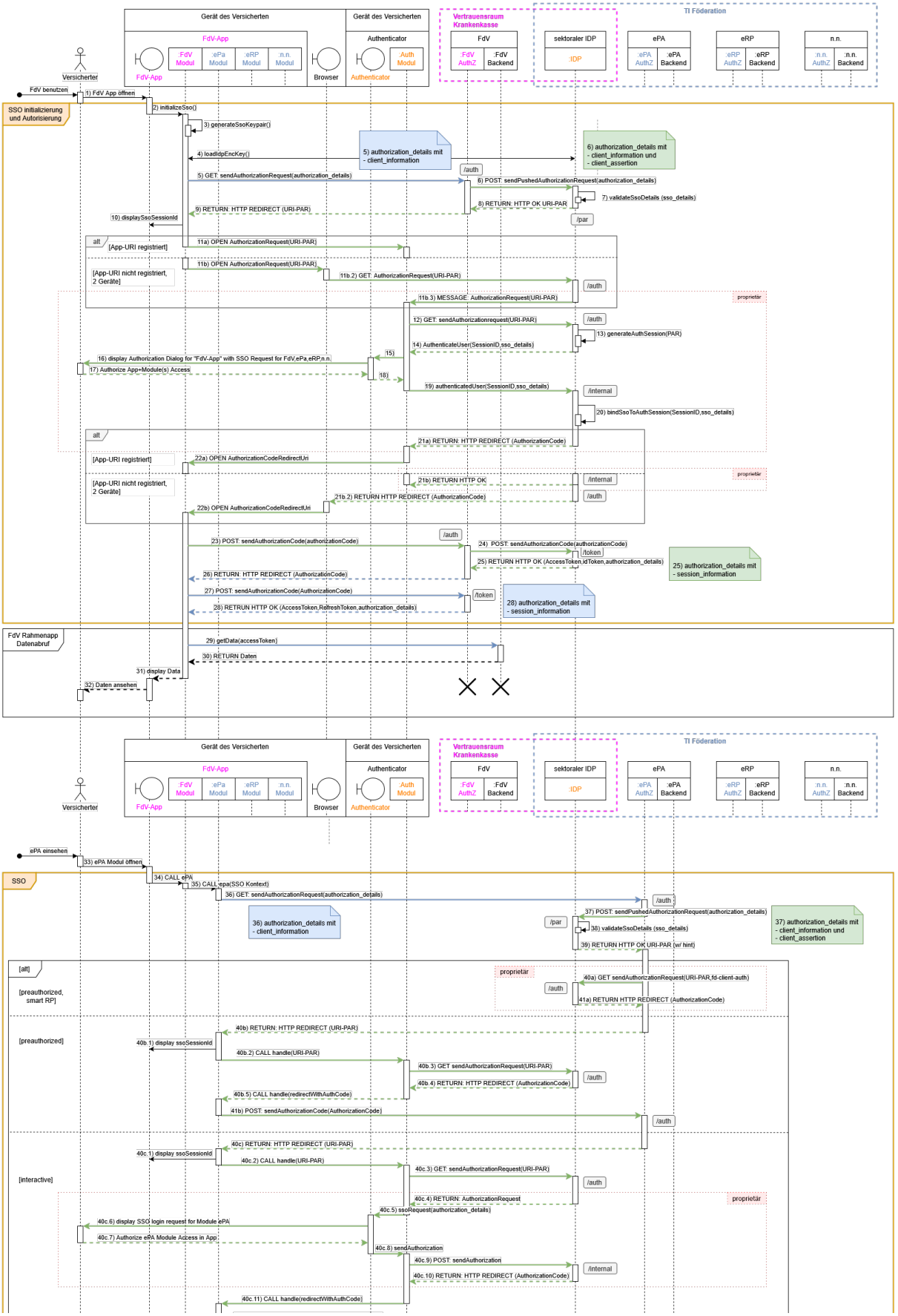

Gerät des Versicherten
Gerät des Versicherten
Vertrauensraum Krankenkasse
TI Föderation
FdV-App
Authenticator
FdV
sektoraler IDP
ePA
eRP
n.n.
Versicherter
Browser
FdV benutzen
1) FdV App öffnen
SSO initializierung und Autorisierung
2) initializeSso()
3) generateSsoKeypair()
4) loadIdpEncKey()
5) authorization_details mit - client_information
6) authorization_details mit - client_information und - client_assertion
5) GET: sendAuthorizationRequest(authorization_details)
6) POST: sendPushedAuthorizationRequest(authorization_details)
7) validateSsoDetails (sso_details)
8) RETURN: HTTP OK URI-PAR
9) RETURN: HTTP REDIRECT (URI-PAR)
10) displaySsoSessionId
/auth
/par
alt
[App-URI registriert]
[App-URI nicht registriert, 2 Geräte]
11a) OPEN AuthorizationRequest(URI-PAR)
11b) OPEN AuthorizationRequest(URI-PAR)
11b.2) GET: AuthorizationRequest(URI-PAR)
11b.3) MESSAGE: AuthorizationRequest(URI-PAR)
proprietär
12) GET: sendAuthorizationrequest(URI-PAR)
13) generateAuthSession(PAR)
14) AuthenticateUser(SessionID,sso_details)
15)
16) display Authorization Dialog for "FdV-App" with SSO Request for FdV,ePa,eRP,n.n.
17) Authorize App+Module(s) Access
18)
19) authenticatedUser(SessionID,sso_details)
/internal
20) bindSsoToAuthSession(SessionID,sso_details)
21a) RETURN: HTTP REDIRECT (AuthorizationCode)
22a) OPEN AuthorizationCodeRedirectUri
21b) RETURN HTTP OK
21b.2) RETURN HTTP REDIRECT (AuthorizationCode)
22b) OPEN AuthorizationCodeRedirectUri
23) POST: sendAuthorizationCode(authorizationCode)
24) POST: sendAuthorizationCode(authorizationCode)
/token
25) RETURN HTTP OK (AccessToken,idToken,authorization_details)
25) authorization_details mit - session_information
26) RETURN: HTTP REDIRECT (AuthorizationCode)
27) POST: sendAuthorizationCode(AuthorizationCode)
28) RETRUN HTTP OK (AccessToken,RefreshToken,authorization_details)
28) authorization_details mit - session_information
FdV Rahmenapp Datenabruf
29) getData(accessToken)
30) RETURN Daten
31) display Data
32) Daten ansehen
ePA einsehen
33) ePA Modul öffnen
SSO
34) CALL ePA
35) CALL epa(SSO Kontext)
36) GET: sendAuthorizationRequest(authorization_details)
36) authorization_details mit - client_information
37) POST: sendPushedAuthorizationRequest(authorization_details)
38) validateSsoDetails (sso_details)
39) RETURN HTTP OK URI-PAR (w/ hint)
37) authorization_details mit - client_information und - client_assertion
[alt]
[preauthorized, smart RP]
[preauthorized]
[interactive]
40a) GET sendAuthorizationRequest(URI-PAR,fd-client-auth)
41a) RETURN HTTP REDIRECT (AuthorizationCode)
40b) RETURN: HTTP REDIRECT (URI-PAR)
40b.1) display ssoSessionId
40b.2) CALL handle(URI-PAR)
40b.3) GET sendAuthorizationRequest(URI-PAR)
40b.4) RETURN: HTTP REDIRECT (AuthorizationCode)
40b.5) CALL handle(redirectWithAuthCode)
41b) POST: sendAuthorizationCode(AuthorizationCode)
40c) RETURN: HTTP REDIRECT (URI-PAR)
40c.1) display ssoSessionId
40c.2) CALL handle(URI-PAR)
40c.3) GET: sendAuthorizationRequest(URI-PAR)
40c.4) RETURN: AuthorizationRequest
40c.5) ssoRequest(authorization_details)
40c.6) display SSO login request for Module ePA
40c.7) Authorize ePA Module Access in App
40c.8) sendAuthorization
40c.9) POST: sendAuthorization
40c.10) RETURN: HTTP REDIRECT (AuthorizationCode)
40c.11) CALL handle(redirectWithAuthCode)

Abbildung 1: Beispiel für SSO-Anmeldung an der ePa in einem FdV nach Erstanmeldung an einem Kassendienst

2.5 Neues Kapitel "5.6.2.3 Ablaufbeschreibung bei IDP controlled SSO"

Im folgenden werden die relevanten einzelnen Schritte erläutert.

Tabelle 1: Ablauf der Aufrufe zum Beispiel für SSO-Anmeldung an der ePa in einem FdV nach Erstanmeldung an einem Kassendienst

Schritt	Teilschritt	"In-App-Authenticator-Modul"	"externe-Authenticator-Modul APP" (nur bei Abweichendem Verhalten befüllt)
1		Der Versicherte startet sein von der Krankenkasse bereitgestelltes FdV, dass mehrere TI-Fachdienste (z.B. ePA, E-Rezept und TIM) und Kassendienste beinhaltet.	
2		Der Anwendungskontext wird neu initialisiert. Dazu werden die anwendungsspezifischen Informationen (Anwendungsname, Gerätename, Fachdienste, die in der Anwendung gemeinsam mittels SSO authentisiert werden können sollen) für den SSO-Kontext konfiguriert. Der zu nutzende IDP ist bekannt.	
3		Es wird ein neues kurzlebiges Schlüsselpaar erzeugt, anhand dessen die SSO-Sitzung Fachdienstübergeifend wiedererkannt werden kann. Die Teile des Schlüssels werden im folgenden `sso_signing_key`, `sso_signing_key_public` und `sso_signing_key_private` bezeichnet. Aus dem`sso_signing_key_public` wird nach RFC7638 mittels SHA256 die `kid` des des `sso_signing_key` abgeleit.	
4		Die Anwendung lädt die IDP Enc-Schlüssel des IDP.	Die Anwendung lädt die vom IDP veröffentlichten Enc-Schlüssel.

5		Das FdV-Modul sendet einen Authorization Request an den Authorization Server des Kassen-Fachdienst, welcher im Parameter `authorization_details` ein Objekt vom Typ `urn:ti:gematik:auth:sso:client_information` gemäß dem Schema nach A_30652* enthält. Das in (3) erzeugte SSO Schlüsselpaar wird für die Signatur des darin enthaltenen sso.jwt benutzt. Für die Verschlüsselung des sso.jwe jedoch ein zufälliger ephemeral public key und dem public key des IDP Enc Schlüssels verwendet wird.	
6		Der Kassen-Fachdienst Authorization Server stellt sicher, dass es sich bei dem aufrufenden Client (App) um ein zugelassenes ePa-FdV handelt. Ist dies der Fall, wird ein Pushed Authorization Request an den IDP gestellt, der in den authorization_details das `urn:ti:gematik:auth:sso:client_information` Objekt aus dem Authorization Request des Clients enthält und das vom Authorization Server dazu erzeugt`urn:ti:gematik:auth:sso:client_assertion` Objekt.	
7		Der IDP validiert die im PAR übermittelten SSO Informationen und speichert diese nach Prüfung zu URI-PAR zwischen. Dazu wird die Integrität und Authentizität der Assertion geprüft. Anschließend wird sichergestellt dass der Authorization Server in seiner Rolle als Client des IDP berechtigt ist, den client_type "epafdv" zu attestieren. Konnte das erfolgreich sichergestellt werden, wird anschließend die Client Information entschlüsselt und auf Binding an die Assertion geprüft.	
8		Der IDP beantwortet den PAR aus (6) mit der URI-PAR.	
9		Der Kassen-Fachdienst Authorization Server beantwortet den Authorization Request aus (5) mit einem Redirect auf die in (8) empfangene URI-PAR.	
10		-	Die Anwendung zeigt die zu authentisierende SSO Session ID an.
11a		Es wird in das Authenticator Modul gewechselt.	Es wird von der

			Anwendung in den lokal auf dem Gerät installierten Authenticator gesprungen.
alt. 11b		-	Es wird von der Anwendungin die für den Authenticator registrierte Webadresse im Browser des Gerätes gesprungen.
	11b.2	-	Der Browser ruft die übergebene PAR-URI am IDP auf.
	11b.3	-	Über einen nicht näher spezifizierten Weg (z.B. Notification Service, dessen Zielgerät der IDP anhand eines gesetzten Cookies identifiziert) wird die PAR-URI an den Authenticator übermittelt.
12		Der Authenticator ruft unter Verwendung von Client Authentifizierung die übergebene URI-PAR am IDP auf.	
13		Der IDP startet einen Authenticator Session und assoziiert diese mit der URI-PAR	
14		Der IDP antwortet dem Authenticator auf den Aufruf der URI-PAR (12) mit allen notwendigen Informationen, dass der Nutzer eine Autorisierung erteilen kann.	
15		Der Authenticator stellt dem Nutzer den Autorisierungsdialog dar.	

16		Die UI des Authenticators zeigt dem Nutzer einen Autorisierungsdialog an, aus dem hervorgeht dass er die Anwendung für den Zugriff auf die angefragten Fachdienste autorisieren soll.	Die UI des Authenticators zeigt dem Nutzer einen Autorisierungsdialog an, aus dem hervorgeht dass er die Anwendung mit Anwendungsname, Gerätename und Session-ID für den Zugriff auf die angefragten Fachdienste autorisieren soll.
17		Der Nutzer autorisiert die Anwendung. (Vereinfachte Darstellung, die weitere Kommunikation zwischen Authenticator und IDP hier nicht einbezieht)	
18		Der Authenticator verarbeitet die Nutzereingaben.	
19		Der Authenticator übermittelt die Autorisierungsinformationen, u.a. für welche Fachdienste die Anwendung ohne weitere Nutzerinteraktion vorab autorisiert ist an den IDP.	
20		Der IDP startet die SSO Session mit der erteilten Autorisierung in dem vom Nutzer gewählten Umfang (Sitzungsdauer, vorab autorisierte Fachdienste).	
21a (zu 11a)		Der IDP antwortet dem Authenticator mit dem zu (12) korrelierenden Autorization Code.	
22a		Der Authenticator übergibt die Authorization Code redirect-url an die Anwendung	Der Authenticator wechselt per Systemaufruf der Authorization Code redirect-url zurück in die Anwendung.
alt. 21b (zu		-	Der IDP meldet dem Authenticator

11b)			die erfolgreiche Annahme der Authentisierung.
	21b.2	-	Der Browser, der auf den in (11b.2) abgesetzten Aufruf wartet, erhält den Authorization Code Redirect.
22b		-	Der Browser öffnet die Redirect url mit Authorization Code, wodurch der Nutzer wieder in die Awendung geleitet wird.
23		Die Anwendung sendet den Autorization Code an den Authorization Server des Kassen-Fachdienstes.	
24		Der Authorization Server des Kassen-Fachdienstes, in seiner Rolle als Client des IDP, sendet den von der Anwendung empfangenden Authorization Code an den IDP. Dabei authentifiziert er sich mittels mTLS.	
25		Der IDP antwortet dem Authorization Server mit Herausgabe von ID-Token und authorization_details, das ein Objekt vom Typ`urn:ti:gematik:auth:sso:client_information` enthält, in dem der kid des in (3) erzeugten Schlüssel bestätigt wird, wenn der Nutzer SSO zugestimmt hat.	
26		Der Authorization Server gibt nach Prüfung der erforderlichen Claims des vom IDP empfangenen ID-Token, einen Access Code an die Anwendung zurück.	
27		Die Anwendung übermittelt den Authorization Code (und PKCE relevante Parameter) an den Token Endpunkt des Authorization Server.	
28		Der Authorization Server antwortet mit einer Tokenresponse, die Access-Token, Refresh-	

		Token und die vom IDP empfangenen`urn:ti:gematik:auth:sso:client_information` enthält. Ist hier nicht die Session ID zur Bestätigung enthalten, weiß die Anwendung dass kein SSO-Kontext aufgebaut werden konnte.	
29		Die Anwendung fragt Daten am Kassen-Backend unter Verwendung des in (28) bereitgestellten Access-Token ab.	
30		Das Backend liefert die angefragten Daten zurück, mit dem im Kontext des Access-Token zugegriffen werden konnte.	
31		Die Anwendung stellt die Daten dar.	
32		Der Nutzer betrachtet die Daten in der Anwendung.	
Der Nutzer entscheidet sich weitere Daten innerhalb der Anwendung ansehen zu wollen, in diesem Fall Daten der ePA.			
33		Der Nutzer wählt den ePA-Teil in der Anwendung.	
34		Die Anwendung weist das FdV-Modul an, das ePA-Modul mit SSO-Kontext zu starten.	
35		Das FdV-Modul übergibt den Anwendungs-SSO-Kontext an das eingebettete ePA-Modul.	
36		Das ePA-Modul bereitet den Authorization Request zum ePa Authorization Server vor, wählt die zu sendende code_challenge. Mit dem SSO-Kontext wird das mit dem Authorization Request mit zu sendende `urn:ti:gematik:auth:sso:client_information` Objekt erzeugt und schließlich im Authorization Request in den `authorization_details` mitgesendet. (vgl. 5)	
37		Der ePa Authorization Server stellt sicher, dass es sich bei dem aufrufenden Client (App) um ein zugelassenes ePa-FdV handelt. Ist dies der Fall, wird ein Pushed Authorization Request an den IDP gestellt, der in den authorization_details das `urn:ti:gematik:auth:sso:client_information` Objekt aus dem Authorization Request des Clients enthält und das vom Authorization Server dazu erzeugt`urn:ti:gematik:auth:sso:client_assertion` Objekt. Ein E-Rezept Authorization Server könnte an	

		dieser Stelle nicht attestieren, dass der aufrufende Client ein ePA-FdV ist, würde daher hier in der Client Attestation keinen client_type angeben.	
38		Der IDP prüft die übermittelten SSO-Informationen und entscheidet nach A_30666* ob der PAR bereits pre-autorisiert ist, der anfragende Authorization Server also bereits durch den Nutzer innerhalb der SSO Sitzung pre-autorisiert worden ist.	
39		Der IDP gibt die PAR-URI an den Authorization Server zurück. Ist der PAR nach Prüfung in (38) bereits pre-autorisiert, so wird zusätzlich der Hinweis rp_code_retrieval mit ausgegeben.	
Alternative: pre-authorized, smart Relying Party unterstützt code_retrieval			
40a		Der Authorization Server fragt unter Verwendung der Clientauthentisierung mittels mTLS an der`relying_party_code_retrieval_endpoint` Schnittstelle des IDP den Authorization Code an (der sonst von der Anwendung nach Authentisierung übergeben werden würde).	
41a		Der IDP übermittelt den Authorization Code, wenn die Bedingungen nach (38) erfüllt sind direkt an den Authorization Server.	
Alternative: pre-authorized, Relying Party unterstützt code_retrieval nicht			
40b		Der ePA Authorization Server liefert dem Authorization Request des ePA-Moduls aus (36) die vom IDP in (39) empfangende URI-PAR zurück.	
	40b.1	-	Die Anwendung zeigt die SSO Session ID an. (vgl. 10)
	40b.2	Die Anwendung übergibt den Kontrollfluss per Systemaufruf in das Authenticator Modul mit Übergabe der URI-PAR. Der Einfachheit halber wird hier nur der same-device Flow beschrieben.	
	40b.3	Das Authenticator Modul ruft die URI-PAR, mittels Clientauthentisierung gesichert, am IDP auf.	
	40b.4	Der IDP übermittelt seinem Authenticator Modul direkt den Authorization Code, weil	

		eine Authentisierung für den PAR nicht mehr nötig ist.	
	40b.5	Das Authenticator Modul übermittelt den Authorization Code mittels Systemaufruf zurück an das ePA-Modul.	
41b		Das ePA-Modul übermittelt den Authorization Code an den Authorization Server der ePA.	
Alternative: pre-authorized, Relying Party unterstützt code_retrieval nicht, IDP entscheidet Risikobasiert dass Nutzerinteraktion notwendig ist			
40c		*siehe 40b*	
	40c.1	-	*siehe 40b.1*
	40c.2	*siehe 40b.2*	
	40c.3	*siehe 40b.3*	
	40c.4	Der IDP signalisiert dem Authenticator, dass eine Nutzerbestätigung für den Login an dem ePA-Fachdienst notwendig ist, übermittelt dazu die anzuzeigenen Informationen.	
	40c.5	Das Authenticator Modul startet eine neue Nutzeransicht.	
	40c.6	Der Authenticator zeigt dem Nutzer an, dass der den Login am ePA-Fachdienst für die laufende Anwendung mit einem OK bestätigen muss. Der Nutzer muss hierfür nicht authentisiert werden.	Der Authenticator zeigt dem Nutzer an, dass er den Login am ePA-Fachdienst für die Anwendung mit Name, Gerätename und Session ID mittels klick auf OK bestätigen muss.
	40c.7	Nutzer klickt "OK"	
	40c.8	Die Nutzereingabe wird an das Authenticator Modul weitergeleitet.	
	40c.9	Das Authenticator Modul übermittelt dem IDP die Bestätigung durch einen Nutzer.	

	40c.10	Der IDP sendet dem Authenticator Modul den Authorization Code	
	40c.11	*siehe 40b.5*	
41c		*siehe 41b*	
Alternative: nicht-preauthorized, SSO Informationen im PAR enthalten und valide			
40d		*siehe 40b*	
	40d.1	-	*siehe 40b.1*
	40d.2	*siehe 40b.2*	
	40d.3	*siehe 40b.3*	
	40d.4	Der IDP signalisiert dem Authenticator Modul, dass eine Authorization für den ePA-Fachdienst im Kontext einer noch nicht gestarteten SSO Sitzung angefordert wurde. Hier kann nun nachfolgend eine einzelne Autorisierung, als gäbe es kein IDP-Controlled-SSO durchgeführt werden, z.B. wenn der Nutzer sein Limit nach A_30670* auf 0 gesetzt hat. Oder mitten in der Nutzung der Anwendung kann eine SSO-Sitzung etabliert werden (wenn z.B. vorherige Aufrufe der selben Anwendung keine SSO-Sitzung starten konnten, weil die client_type attestation an dem Fachdienst nicht zum Start der Sitzung genügte).	
	40d.5	Das Authenticator Modul startet eine neue Nutzeransicht.	
	40d.6	*siehe 16*	*siehe 16*
	40d.7	*siehe 17*	
	40d.8	*siehe 18*	
	40d.9	*siehe 19*	
	40d.10	*siehe 40b.4*	
	40d.11	*siehe 40b.5*	
41d		*siehe 41b*	
Ende Alternativer Abläufe			

42		Der ePA Authorization Server, in seiner Rolle als Client des IDP, sendet den vom ePA-Modul empfangenden Authorization Code an den IDP. Dabei authentifiziert er sich mittels mTLS. (vgl. 24)	
43		Der IDP antwortet dem ePA Authorization Server mit Herausgabe von ID-Token und authorization_details, das ein Objekt vom Typ`urn:ti:gematik:auth:sso:client_information` enthält, in dem der kid des in (3) erzeugten Schlüssel bestätigt wird, wenn der Nutzer SSO zugestimmt hat. (vgl. 25)	
44		Der ePA Authorization Server gibt nach Prüfung der erforderlichen Claims des vom IDP empfangenen ID-Token, einen Access Code an das ePA-Modul zurück. (vgl. 26)	
45		Das ePA-Modul übermittelt den Authorization Code (und PKCE relevante Parameter) an den Token Endpunkt des ePA Authorization Server. (vgl. 27)	
46		Der ePA Authorization Server Antwortet mit einer Tokenresponse, die Access-Token, Refresh-Token und die vom IDP empfangene`urn:ti:gematik:auth:sso:client_information` enthält. Ist hier nicht die Session ID zur Bestätigung enthalten, weiß die Anwendung dass kein SSO-Kontext aufgebaut werden konnte. (vgl. 28)	
47		Das ePA-Modul fragt Daten am ePA-Aktensystem unter Verwendung des in (46) bereitgestellten Access-Token ab. (vgl. 29) Vereinfachte Darstellung, die Application Security Layer o.ä. hier nicht berücksichtigt.	
48		Das ePA-Aktensystem liefert die angefragten Daten zurück, mit dem im Kontext des Access-Token zugegriffen werden konnte.	
49		Das ePA-Modul erzeugt eine neue Darstellung	
50		Die Anwendung zeigt dem Nutzer die angeforderten Daten aus dem dem Aktzensystem.	
Beispieldaten werden an dieser Stelle noch zu den jeweiligen Schritten ergänzt.			

255

256

257

3 Änderungen in gemSpec_IDP_Sek

3.1 Anpassungen im Kapitel 4.2 "Entity Statement des sektoralen IDP"

Es wird Kapitel 4.2 wie folgt angepasst

alt:

A_22643-02 -Entity Statement des sektoralen IDP
Der sektorale IDP MUSS ein selbst signiertes Entity Statement gemäß[OpenID Federation 1.1] ("Entity Statement") bereitstellen und im Internet verfügbar machen. Das Entity Statement MUSS mindestens die in der folgenden Tabelle aufgeführten Metadaten enthalten:

Tabelle 2: Header des Entity Statement des sektoralen IDP

Name	Werte / Wertebereich
`alg`	string, zulässiger Wert "ES256"
`kid`	string, UUID7-Format [RFC9562#name-uuid-version-7]
`typ`	string, zulässiger Wert "entity-statement+jwt"

Tabelle 3 :*Allgemeine Attribute im well-known-Dokument des sektoralen IDP*

Name	Werte / Wertebereich
`iss`	string, URL nach [RFC1738]
`sub`	string, URL nach [RFC1738]
`iat`	number, Alle time-Werte in Sekunden seit 1970,[RFC7519#section-2]
`exp`	number, Alle time-Werte in Sekunden seit 1970,[RFC7519#section-2]
`jwks`	Set von JWK [RFC7517]
`authority_hints`	[string]

	zulässige Werte gemäß [OpenID Federation 1.1] ("Claims that MUST or MAY Appear in Entity Configurations but Not in Subordinate Statements") - authority_hints
metadata	JSON Object, erforderlicher Wert: "openid_provider"

Tabelle 4 : Attribute des Metadatenblocks openid_provider im well-known-Dokument des sektoralen IDP

Name	Werte
`issuer`	string, URL nach [RFC1738] zulässiger Wert: identisch mit dem Wert des Claims `iss`
`signed_jwks_uri(*)`	string, URL nach [RFC1738]
`jwks (*)`	Set von JWK [RFC7517]
`authorization_endpoint`	string, URL nach [RFC1738]
`token_endpoint`	string, URL nach [RFC1738]
`pushed_authorization_request_endpoint`	string, URL nach [RFC1738]
`organization_name`	string (gemäß [OpenID-Federation 1.1 "Informational Metadata Extensions"] - organization_name) Wertebereich: ^[à-üÀ-Üß\w\ \-\.\+*\/]{1,128}$
`keywords`	[string] (gemäß [OpenID-Federation 1.1 "Informational Metadata Extensions"] - keywords) erforderliche Werte: "product_type_version:<von der gematik zugelassene Produkttyp-Version>" "product_type:<von der gematik zugelassener Produkttyp>"
`contacts`	[string] (gemäß [OpenID-Federation 1.1 "Informational Metadata Extensions"] - contacts) erforderlicher Wert in Liste: "<E-Mail-

	Adresse für Supportanfragen>"
logo_uri	string, URL nach [RFC1738] zulässiger Wert: <URL>*.png
client_registration_types_supported	[string] zulässiger Wert: "automatic"
subject_types_supported	[string] zulässiger Wert: "pairwise"
response_types_supported	[string] zulässiger Wert: "code"
scopes_supported	[string], Wertebereich: "openid", "<weitere Scopes nach A_22989*>"
claims_supported	[string], Wertebereich: "<Claims nach A_22989*>"
claims_parameter_supported	boolean, Wertebereich: true/false
response_modes_supported	[string] zulässiger Wert: "query"
grant_types_supported	[string] zulässiger Wert: "authorization_code"
require_pushed_authorization_requests	boolean, Wertebereich: true/false zulässiger Wert: true
token_endpoint_auth_methods_supported	[string] erforderlicher Wert: "self_signed_tls_client_auth"
id_token_signing_alg_values_supported	[string] erforderlicher Wert: "ES256"
id_token_encryption_alg_values_supported	[string] erforderlicher Wert: "ECDH-ES"
id_token_encryption_enc_values_supported	[string] erforderlicher Wert: "A256GCM"
user_type_supported	[string] zulässiger Wert: "IP" (Insured Person)

ti_features_supported {	
`id_token_version_supported`	[string] zulässige Werte in Liste: "1.0.0", "2.0.0"

(*) - gemäß [OpenID Federation 1.1] - "Usage of jwks, jwks_uri, and signed_jwks_uri in Entity Metadata" darf der Metadatenblock nur entweder `signed_jwks_uri` oder `jwks` `enthalten.`**[**<=, , **]**

neu:

A_22643-03 Tabelle "Attribute des Metadatenblocks openid_provider im well-known-Dokument des sektoralen IDP" ergänzt um

- Anpassung in C_12871
- neuen Endpunkt

	Werte
`relying_party_code_retrieval_endpoint`	string, URL nach [RFC1738]

- neue zulässige Werte für ti_features_supported durch A_27506-01

	Werte
`sso_version_supported`	[string], optional Zulässige Werte in Liste: "idp_controlled"

alt:

A_27506 -Signalisierung der unterstützten TI-Features durch einen sektoralen IDP der TI-Föderation
Ein sektoraler IDP MUSS in seinem Entity Statement im Metadatenblock `openid_provider` in einem Claim `ti_features_supported` signalisieren, welche spezifischen Versionen der TI-Föderation unterstützt werden. Im Claim`ti_features_supported` MUSS der sektoraler IDP aktuell die Unterstützung der in Tabelle "Durch einen sektoralen IDP unterstützte TI-Features" genannten Claims signalisieren.

Tabelle 5 : Durch einen sektoralen IDP unterstütze TI-Features

claim	**Wertebereich**	**Beschreibung**
`id_token_version_supported`	[string], zulässige Werte in Liste: "1.0.0", "2.0.0"	Mit A_22867-* und A_23207-* ändert sich die Syntax des vom sektoralen IDP ausgestellten ID Token nicht abwärtskompatibel. Für einen Übergangszeitraum muss ein sektoraler IDP die beiden Versionen: • 1.0.0 nach A_22867-01 und A_23207-02, • 2.0.0 nach A_27591 und

		A_27592, unterstützen.

[<=, IDP-Sek, funkt. Eignung: Herstellererklärung**]**

neu:

A_27506-01 Signalisierung der unterstützten TI-Features durch einen sektoralen IDP der TI-Föderation

Ein sektoraler IDP MUSS in seinem Entity Statement im Metadatenblock `openid_provider` in einem Claim `ti_features_supported` signalisieren, welche spezifischen Versionen der TI-Föderation unterstützt werden. Im Claim`ti_features_supported` MUSS der sektorale IDP aktuell die Unterstützung der in Tabelle "Durch einen sektoralen IDP unterstützte TI-Features" genannten Claims signalisieren.

Tabelle 6 : Durch einen sektoralen IDP unterstütze TI-Features

claim	Wertebereich	Beschreibung
`id_token_version_supported`	[string], zulässige Werte in Liste: "1.0.0", "2.0.0"	Mit A_22867-* und A_23207-* ändert sich die Syntax des vom sektoralen IDP ausgestellten ID-Token nicht abwärtskompatibel. Für einen Übergangszeitraum muss ein sektoraler IDP die beiden Versionen: • 1.0.0 nach A_22867-01 und A_23207-02, • 2.0.0 nach A_27591 und A_27592, unterstützen.
`sso_version_supported`	[string], optional Zulässige Werte in Liste: "idp_controlled"	Der Wert idp_controlled darf nur angegeben werden, wenn IDP-controlled-SSO auf Anwendungsebene (Kapitel 5.2.4) vollständig umgesetzt ist.

3.2 Anpassungen Kapitel "5.2.2 Rahmenbedingungen"

Text alt:

Unter folgenden Rahmenbedingungen ist ein Single-Sign-On möglich:

- Ein SSO erstreckt sich ausschließlich über die TI-Fachdienste, die innerhalb eines Anwendungskontextes (ePA-FdV) vom Nutzer aufgerufen werden können.
- Ist das Authenticator-Modul des sektoralen IDP, über welchen der Nutzer sich authentisiert, ebenfalls Teil des Anwendungskontextes, so muss sein Frontend im

Ablauf der Nutzerauthentisierung eine vom Authenticator-Modul bereitgestellt API-Schnittstelle aufrufen.

- Ist das Authenticator-Modul des sektoralen IDP, über welchen der Nutzer sich authentisiert, eine Authenticator APP, so muss im Ablauf der Nutzerauthentisierung sichergestellt werden, dass das Authenticator-Modul immer vom gleichen Anwendungskontext (ePA-FdV) aufrufen wird.
- Der Nutzer muss sich im Anwendungskontext mindestens einmal aktiv auf dem Vertrauensniveau "gematik-loa-high" authentifizieren.
- Für die Laufzeit eines Anwendungskontextes (Laufzeit der Anwendung) wird vom sektoralen IDP eine SessionID generiert.
- Zu einem Anwendungskontext wird durch das Authenticator-Modul ein Schlüsselpaar im systemeigenen Schlüsselspeicher erzeugtund der SessionID zugeordnet.
- Der Nutzer muss innerhalb des Anwendungskontextes konfigurieren können, welcher Fachdienst an einem SSO-Verfahren teilnehmen darf.

Die Anforderungen in diesem Kapitel und die zusätzlichen Informationen in [gemKPT-TI-Föderation] Kapitel "5.6 Unterstützung Single-Sign-On auf Anwendungsebene" betreffen nur Hersteller, die ein SSO auf Anwendungsebene implementieren. Für Hersteller, die kein SSO auf Anwendungsebene umsetzen, gelten die Anforderungen dieses Kapitels nicht.

Text neu:

Unter folgenden Rahmenbedingungen ist ein Single-Sign-On möglich:

- Ein SSO erstreckt sich ausschließlich über die TI-Fachdienste, die innerhalb eines Anwendungskontextes (ePA-FdV) vom Nutzer aufgerufen werden können.
- Darüber hinaus dürfen bei idp-controlled-sso weitere Dienste innerhalb des Anwendungskontextes eingebunden werden, sofern es eine Vertrauensbeziehung zwischen IDP und den Diensten gibt.
- Ist das Authenticator-Modul des sektoralen IDP, über welchen der Nutzer sich authentisiert, ebenfalls Teil des Anwendungskontextes, so muss sein Frontend im Ablauf der Nutzerauthentisierung eine vom Authenticator-Modul bereitgestellt API-Schnittstelle aufrufen.
- Ist das Authenticator-Modul des sektoralen IDP, über welchen der Nutzer sich authentisiert, eine Authenticator APP, so muss im Ablauf der Nutzerauthentisierung sichergestellt werden, dass das Authenticator-Modul immer vom gleichen Anwendungskontext (ePA-FdV) aufrufen wird.
- Der Nutzer muss sich im Anwendungskontext mindestens einmal aktiv auf dem Vertrauensniveau "gematik-ehealth-loa-high" authentifizieren.
- Für die Laufzeit eines Anwendungskontextes (Laufzeit der Anwendung) wird vom sektoralen IDP eine SessionID generiert.
- Zu einem Anwendungskontext wird durch das Authenticator-Modul ein Schlüsselpaar im systemeigenen Schlüsselspeicher erzeugtund der SessionID zugeordnet.
- Der Nutzer muss innerhalb des Anwendungskontextes konfigurieren können, welcher Fachdienst an einem SSO-Verfahren teilnehmen darf.

3.3 neues Kapitel "5.2.4 IDP-controlled-SSO auf Anwendungsebene"

Ein IDP, der IDP-controlled-SSO unterstützt, authentisiert eine Anwendung anhand der vom Fachdienst weitergeleiteten SSO-Informationen.

Die von der Anwendung übermittelten Informationen dienen dazu, dass der Nutzer im IDP-Frontend alle in einer Anwendung enthaltenen Fachdienste innerhalb einer Anwendungssitzung gemeinsam autorisieren kann und im besten Fall ohne weitere Authentisierungsdialoge auskommt.

Dazu sind einige Ergänzungen der Schnittstellen des IDP zu berücksichtigen.

3.3.1 neues Kapitel 5.2.4.1 "PAR-Endpunkt"

Im Pushed Authorization Request wird entweder eine neue SSO Sitzung initialisiert, so dass der Nutzer im IDP-Frontend die Anwendung nebst Fachdiensten gemeinsam autorisieren kann, oder eine authorization Anfrage des Fachdienstens innerhalb einer SSO-Sitzung beantwortet wird.

3.3.2 neues Kapitel 5.2.4.1.1 "PAR-Endpunkt Eingangsdaten"

A_30652 -Annahme von SSO Client Informationen
Unterstützt der sektorale IDP das Verfahren IDP-controlled-SSO, so MUSS der sektorale IDP die im Pushed Authorization Request übertragenen SSO-Informationen verarbeiten können.
Dazu ist das im Parameter `authorization_details` enthaltende Objekt vom Typ`urn:ti:gematik:auth:sso:client_information` [RFC9396#section-2] auszulesen.
Die Verarbeitung der SSO Client Information darf nur erfolgen, wenn in den authorization_details auch eine SSO Client Assertion mit übertragen wurde und diese validiert werden konnte.

Tabelle 7: Attribute des sso:client_information Objektes

Name	Werte / Wertebereich
`type`	string, zulässiger Wert: "urn:ti:gematik:auth:sso:client_information"
`client_information_encrypted`	string, mit ENC-Schlüssel des IDP verschlüsseltes sso.jwe im JWE compact serialization [RFC7516#section-3.1] Format

[<=, ,]

A_30653 -Entschlüsseln der SSO Client Information
Unterstützt der sektorale IDP das Verfahren IDP-controlled-SSO, so MUSS der sektorale IDP das nach folgendem Schema im `client_information_encrypted` Attribut übertragene JWE [RFC7516#section-3.1], im folgenden auch als sso.jwe bezeichnet, mit seinem Enc-Schlüssel entschlüsseln können.

Bei nicht entschlüsselbaren oder abgelaufenen JWE MUSS der PAR mit einer HTTP/400 (Bad Request)`invalid_request` Error Response [RFC6749#section-4.1.2.1] mit dem Attribut error_code 79310 gemäß A_22826 abgelehnt werden.

Tabelle 8 : sso.jwe Header

Name	Werte / Wertebereich
`alg`	string, gültiger Wert: "ECDH-ES"
`enc`	string, gültiger Wert: "A256GCM"
`kid`	string, kid des enc Schlüssel des IDP
`cty`	string, gültiger Wert: "ssojwt"
`exp`	number, Unix Timestamp des Ablaufzeitpunktes
`zip`	*darf nicht vorhanden sein*
`epk {`	JSON Objekt, JWK nach RFC7517 (Auflistung abschließend)
`kty`	string, zulässiger Wert: "EC"
`crv`	string, zulässiger Wert: "P-256"
`x`	string
`y`	string

Tabelle 9 : sso.jwe Payload

Name	Werte / Wertebereich
`ssojwt`	string, sso.jws im JWS compact serialization [RFC7515#section-3.1] Format

[<=, ,]

A_30654 -Verarbeitung der SSO Client Informationen
Unterstützt der sektorale IDP das Verfahren IDP-controlled-SSO, so MUSS der sektorale IDP das im `ssojwt` Attribut des Payloads der sso.jwe enthaltene SSO JWT [RFC7519], im folgenden auch sso.jwt genannt, und die darin enthaltenden Informationen verarbeiten können.
Bei JWT die nicht dem angegebenen Schema entsprechen, ungültige Werte enthalten,

notwendige Attribute fehlen oder darüberhinausgehende Attribute aufführen, muss der PAR mit einer HTTP/400 (Bad Request)`invalid_request` Error Response [RFC6749#section-4.1.2.1] mit dem Attribut error_code 79311 gemäß A_22826 abgelehnt werden.

Tabelle 10: sso.jwt Header

Name	Werte / Wertebereich
`alg`	string, zulässiger Wert: "ES256"
`kid`	string, kid des im jwk-Attribut enthaltenen Schlüssel.
`typ`	string, gültiger Wert: "ssojwt+jwt"
`version`	number, optional gültige Werte: 1* (default)
`jwk {`	JSON Objekt, JWK nach RFC7517 (Auflistung nicht abschließend)
`crv`	string, zulässiger Wert: "P-256"
`kty`	string, zulässiger Wert: "EC"
`use`	string, zulässiger Wert: "sig"
`kid`	string, zulässiger Wert: Thumbprint des JWK nach RFC7638 mit SHA256

Tabelle 11: sso.jwt Payload

Name	Werte / Wertebereich
`jti`	string
`iat`	number, Alle time-Werte in Sekunden seit 1970,[RFC7519#section-2]
`exp`	number, Alle time-Werte in Sekunden seit 1970,[RFC7519#section-2]
`nbf`	number, Alle time-Werte in Sekunden seit 1970,[RFC7519#section-2]

sso_instance_group	[string], Liste der Fachdienste, die in der App gebündelt authentifiziert werden können sollen. Gültig sind nur ClientIDs, die am IDP registriert sind (z.B. kasseninterne Anwendungen) oder aufgrund ihrer Mitgliedschaft in der TI-Föderation automatisch nach A_22650-* registriert werden können. Ungültige ClientIDs sind ignorieren.
app_authz_code_challenge	string
app_host_name	string, UTF8, darstellbare Zeichen, 30 Zeichen
app_display_name	string, UTF8, darstellbare Zeichen, 30 Zeichen

[<=, ,]

A_30655 -Annahme von SSO Client Assertion
Unterstützt der sektorale IDP das Verfahren IDP-controlled-SSO, so MUSS der sektorale IDP die im Pushed Authorization Request übertragene SSO Assertion verarbeiten können. Dazu ist das im Parameter authorization_details enthaltende Objekt vom Typ [RFC9396#section-2] `urn:ti:gematik:auth:sso:client_assertion` auszulesen und zu validieren.
Bei Assertions, die nicht dem angegebenen Schema entsprechen, ungültige Werte enthalten, notwendige Attribute fehlen oder darüberhinausgehende Attribute aufführen, muss der PAR mit einer HTTP/400 (Bad Request) invalid_request Error Response [RFC6749#section-4.1.2.1] mit dem Attribut error_code 79301 gemäß A_22826 abgelehnt werden.

Tabelle 12: Attribute des sso:client_assertion Objektes

Name	Werte / Wertebereich
type	string, zulässiger Wert: "urn:ti:gematik:auth:sso:client_assertion"
assertion_version	number, optional gültiger Wert: 1 (default)
client_type	string, optional zulässiger Wert: "epafdv"
client_id	string, OAuth2 client-id der Anwendung gegenüber dem Fachdienst Authorization-Server
authz_code_challenge	string, code challenge, die die Anwendung im Authorization Request gegenüber dem Fachdienst Authorization-Server verwendet hat

`signature`	string, Assertion Signatur, die der Fachdienst nach A_30674* [gemSpec_IDP_FD] erzeugt
`signature_kid`	string, Key ID des signer keys des Fachdienstes der nach A_23034* veröffentlicht ist
`signature_alg`	string, begrenzt auf zulässige Werte für`id_token_signed_response_alg` des Fachdienst nach A_23034*

[<=, , **]**

A_30656 -Verarbeitung der SSO Client Assertion
Unterstützt der sektorale IDP das Verfahren IDP-controlled-SSO, so MUSS der sektorale IDP die SSO Client Assertion wie folgt validieren:

- die Verarbeitung bei client_type "epafdv" darf nur erfolgen, wenn
  - der PAR stellende Fachdienst in seiner Rolle als OpenID-Connect Client das Trustmark "https://gemspec.gematik.de/trustmarks/client_assertion_epafdv" aufweist oder
  - sich der PAR stellende Authorization Server im selben Vertrauensraum mit dem IDP unter Kontrolle des Kostenträgers befindet und dessen Funktionsweise nach den Vorgaben gemäß A_30672*, A_30675*, A_30677* und A_30678* der [gemSpec_IDP_FD] sichergestellt ist
- die Assertion Signatur validiert werden kann
- die authz_code_challenge der Assertion dem Attribut app_authz_code_challenge der sso.jwt entspricht

Schlägt die Validierung fehl, so ist der Pushed Authorization Request mit einer HTTP/400 (Bad Request) invalid_request Error Response [RFC6749#section-4.1.2.1] mit dem Attribut error_code 79301 gemäß A_22826 abgelehnt werden.**[**<=, , **]**

A_30657 -Start einer IDP-controlled-SSO Session
Unterstützt der sektorale IDP das Verfahren IDP-controlled-SSO, so MUSS der sektorale IDP die folgenden SSO Informationen in einer der kid zugeordneten SSO Session speichern und anhand der Session ID bzw. des KID wiedererkennen können, sofern der client_type mit "epafdv" belegt ist und die kid nicht in der Liste der abgelaufenden Session IDs enthalten ist.

- kid
- jwk
- client_type
- client ids der Fachdienste
- app host name
- app display name
- Zeitpunkt des Beginns der SSO Session

Wird die SSO Session nicht innerhalb des Gültigkeitszeitraums der PAR-URI autorisiert, so ist diese wieder zu verwerfen. Um die SSO Session wiederzuerkennen, muss diese intern mit der jeweiligen PAR-URI assoziiert werden.
Abgelaufende SSO Sessions sind unverzüglich zu verwerfen. Die Session IDs abgelaufender Sessions sind 72 Stunden nach Ablauf vorzuhalten.
Ein Pushed Authorization Request, bei dem der kid des sso.jwt in der Liste der abgelaufenden Session IDs enthalten ist,muss mit einer HTTP/400 (Bad Request) invalid_request Error Response [RFC6749#section-4.1.2.1] mit dem Attribut error_code 79312 gemäß A_22826 abgelehnt werden.**[**<=, , **]**

3.3.3 neues Kapitel 5.2.4.1.2 "PAR-Endpunkt Ausgangsdaten"

A_30658 -Signalisierung der Möglichkeit zum Code Abruf
Unterstützt der sektorale IDP das Verfahren IDP-controlled-SSO und sind die Bedingungen der Autorisierungsprüfung A_30666* erfüllt, so MUSS der sektorale IDP die Ausgangsdaten des PAR um folgendes Attribut ergänzen:

Tabelle 13: zusätzliche Attribute des PAR Antwort Objektes

Name	Werte / Wertebereich
`rp_code_retrieval`	boolean, true

[<=, , **]**

3.3.4 Neues Kapitel "5.2.4.2 Code Retrieval Endpunkt"

Der Code Retrieval Endpunkt dient bei IDP-controlled-SSO zum direkten Abruf des `AUTHORIZATION_CODE` durch einen Fachdienst um round trips zum Anwendungsfrontend und App-Sprünge zu vermeiden.

Ist der Fachdienst in einer IDP-Controlled-SSO Sitzung bereits durch den Anwender berechtigt worden, so soll der Fachdienst sich den `AUTHORIZATION_CODE` selbst vom IDP abrufen können, den Code nicht vom Anwendungsfrontend übermittelt bekommen müssen.

3.3.5 neues Kapitel "5.2.4.2.1 Code Retrieval Endpunkt Eingangsdaten"

A_30659 -Prüfung des TLS Clientzertifikates am Code Retrieval Endpunkt
Der Code Retrieval Endpunkt des sektoralen IDP, der IDP-controlled-SSO untertsützt, MUSS das im Rahmen der`self_signed_tls_client_auth`übertragene Zertifikat des Fachdienstes wie folgt überprüfen:

- Das Zertifikat ist über das Entity Statement des Fachdienstes zu verifizieren.
- Das Zertifikat ist zeitlich gültig.

[<=, , **]**

A_30660 -Validierung der PAR URI
Der Code Retrieval Endpunkt des sektoralen IDP, der IDP-controlled-SSO unterstützt, MUSS sicherstellen, dass die übermittelte PAR URI für den anfragenden Client gültig ist und der Client durch den Nutzer in der IDP-controlled-SSO Session gemäß A_30666* autorisiert ist.**[**<=, , **]**

3.3.6 neues Kapitel "5.2.4.2.2 Code Retrieval Endpunkt Ausgangsdaten"

Sind alle in Scope bzw. Claims geforderten Attribute vorhanden, die Gültigkeit der Attribute geprüft, der Client durch den Nutzer für diese SSO Session autorisiert und liegen keine Gründe vor, nach denen ein nicht-interaktives Login unterbunden werden muss, so soll der Client den Authorization Code abrufen können.

A_30661 -Rückgabe Authorization Code an Fachdienst
Der Code Retrieval Endpunkt des sektoralen IDP, der IDP-controlled-SSO unterstützt, MUSS dem berechtigten Client mit einem HTTP/200 Antworten und im application/json Format ein Objekt mit den Authorization Code zum Abruf des ID Token zurück liefern. Berechtigt ist der Client, wenn die pre-authentisierung gemäß A_30666* erfolgreich ist.

Tabelle 14; Attribute Code Retrieval Objektes

Name	Werte / Wertebereich
code	string

[<=, , **]**

3.3.7 neues Kapitel 5.2.4.3 Erweiterung Token-Endpunkt

Die Information, dass ein Authorization Request erfolgreich im Kontext einer SSO Session autorisiert wurde, wird nicht nur im ID_TOKEN im custom claim gemäß A_27592 signalisiert, sondern auch mit einem zusätzlichen Informationsobjekt in der Token-Response beantwortet, dass weitere technische Informationen enthält. Diese Informationen werden außerhalb des ID_TOKEN bereitgestellt, da sie an die Anwendung zurück weitergeleitet werden sollen.

A_30676 -Ausgabe SSO Session Informationen
Ein sektoraler IDP, der IDP-controlled-SSO unterstützt, MUSS in einer erfolgreichen Token-Response gemäß A_22706* den Status der SSO Session beauskunften, wenn im PAR SSO Informationen empfangen wurden.
Dazu ist in der Token-Response in den`authorization_details` [RFC9396#name-token-response] ein Objekt vom Typ`urn:ti:gematik:auth:sso:session_information` [RFC9396#section-2] auszugeben, dass in der `sso_session_id` den `kid` der in der Authorization Anfrage enthaltenden`urn:ti:gematik:auth:sso:client_information` wiederspiegelt.

Tabelle 15: Attribute des sso:session_information Objektes

Name	Werte / Wertebereich
type	string, zulässiger Wert:

	"urn:ti:gematik:auth:sso:session_information"
sso_session_id	string

[<=, ,]

3.3.8 neues Kapitel 5.2.4.4 "SSO Session Autorisierungen"

Die Autorisierung einer SSO Session ist bei IDP-controlled-SSO ein elementarer Grundstein für das Vertrauen in die weiteren Authentisierungen, daher ist hier insbesondere Acht darauf zu legen, dass der Nutzer eine informierte Entscheidung treffen kann, ihm dabei alle relevanten Informationen transparent bereitzustellen sind.

Die Einwilligung zur Verarbeitung der angeforderten Claims ist nicht Bestandteil der SSO Session Autorisierung, hierzu wird auf den bestehenden Ablauf zur Einholung der Erlaubnis zur Verarbeitung der Claims zurückgegriffen. Liegen für einen anfragenden Fachdienst nicht alle notwendigen Einwilligungen vor, wird der Nutzer zur Erteilung der Einwilligung in den Authentisierungsablauf eingebunden. Liegen jedoch alle Einwilligungen aus vorherigen Authentisierungsvorgängen bereits im IDP vor, so können SSO Authentifizierungen für die Laufzeit der SSO Sitzung innerhalb der Anwendung für den Nutzer unsichtbar erfolgen.

A_30662 -Vorbereitung der SSO Session Autorisierung
Ein sektoraler IDP, der IDP-controlled-SSO unterstützt, MUSS sicherstellen, dass die Bedingungen zur Autorisierung einer neuen SSO Session gegeben sind. Dazu ist zu prüfen,

1. dass nicht bereits die maximal zulässige Anzahl an parallelen SSO Sitzungen für den Nutzer erreicht wurde, sofern der Nutzer konfiguriert hat, dass Überschreiten dieser Anzahl zu einer Fehlermeldung führen soll und mehr als 0 Sitzungen zulässig sind, muss der PAR mit mit einer HTTP/400 (Bad Request) invalid_request Error Response [RFC6749#section-4.1.2.1] mit dem Attribut error_code 79313 gemäß A_22826 abgelehnt werden.

[<=, ,]

A_30663 -SSO Session Autorisierung
Ein sektoraler IDP, der IDP-controlled-SSO unterstützt, MUSS vom Nutzer die Einwilligung zum Start einer SSO Sitzung einholen.
Dazu sind ihm im Dialog die folgenden Informationen aufbereitet darzustellen:

1. Name der Anwendung
2. Anzeigenamen der Fachdienste, die mit autorisiert werden sollen
3. Name des Gerätes*
4. Session ID

Die Fachdienste sollen alle vorausgewählt sein. Hat der Nutzer in der Vergangenheit der Speicherung der Entscheidung der zulässigen Fachdienste einer Anwendung zugestimmt, dürfen diese Präferenzen wiederverwendet werden und eine vereinfachte Darstellung angeboten werden.
Die Session ID soll dem Nutzer in einer, mit der Darstellung im ePa-FdV abgestimmten,

Darstellung zur Korrelation präsentiert werden.
Die Lebenszeit der IDP seitig gehaltenen SSO Session startet mit erfolgreicher Authentisierung des Nutzers.

*) *Auf Anzeige des Gerätenamens kann verzichtet werden, sofern für den selben Nutzer nicht die selbe Anwendung von unterschiedlichen Geräten autorisiert wird und dabei auf persistente Nutzerpräferenzen zurückgegriffen wird.*
[<=, , **]**

A_30664 -SSO Session Verwaltung
Ein sektoraler IDP, der IDP-controlled-SSO unterstützt, MUSS im Zuge der erteilten SSO Session Autorisierung sicherstellen, dass die SSO Session Nutzerpräferenzen eingehalten werden.

- Ist die vom Nutzer konfigurierte maximal zulässige parallele Session Zahl erreicht und der Nutzer hat die Auswahl der zu beendenden Session konfiguriert, so muss dem Nutzer nun angeboten werden, welche Session zu beenden ist.
- Ist die vom Nutzer konfigurierte maximal zulässige parallele Session Zahl erreicht und der Nutzer hat die Beendigung der ältesten Session konfiguriert, so ist diese nun zu beenden.
- Der vom Nutzer gewählte Gültigkeitszeitraum ist auf die Session anzuwenden.

[<=, , **]**

A_30665 -Persistente Nutzerpräferenzen je Anwendung
Ein sektoraler IDP, der IDP-controlled-SSO unterstützt, DARF die SSO Präferenzen persistieren und diese bei künftigen SSO Autorisierungen wiederverwenden, wenn die Anwendung zuverlässig wiedererkannt werden kann.
Die Anwendung kann als zuverlässig wiedererkannt betrachtet werden, wenn die client_id der SSO Assertion von einem für den client_type "epafdv" berechtigten Fachdienst stammt und mit dem Name der Anwendung und dem Name des Gerätes korreliert werden kann.**[**<=, , **]**

A_30666 -Nicht interaktive SSO Folgeauthentisierung
Ein sektoraler IDP, der IDP-controlled-SSO unterstützt, MUSS einen PAR als pre-authentisiert betrachten wenn die folgenden Bedingungen Erfüllt sind:

Tabelle 16: Bedingungen für nutzerinteraktionslose Folgeautorisierung

client_type (PAR)	Bedingungen (alle müssen zutreffen)
epafdv, *fehlend*	• die sso Session ist autorisiert, der Nutzer authentisiert worden • client_id des Fachdienstes ist in der Liste der vorab autorisierten Fachdienste dieser Session • für die vom Fachdienst angefragten Claims sind Einwilligungen vorhanden • im PAR ist kein interactive Login (z. B. durch max_age=0, prompt=login o.ä.) erzwungen • IDP Seitig liegen keine Anhaltspunkte vor, dass die client_id der

	Anwendung gegenüber des Fachdienstes von nutzerinteraktionslosem SSO auszuschließen ist • die Sessionhöchstdauer nicht überschritten wurde

[<=, , **]**

A_30667 -Interaktive SSO Folgeauthentisierung
Ein sektoraler IDP, der IDP-controlled-SSO unterstützt, MUSS bei einem PAR, der an eine gültige SSO Session gebunden ist, abweichend von A_30666* interaktiv vom Nutzer die Freigabe anfordern, wenn eine der folgenden Bedingungen erfüllt ist:

Table 1 Bedingungen für Nutzerinteraktion bei Folgeautorisierung

client_type (PAR)	Bedingungen
epafdv, *fehlend*	• im PAR ist ein interactive Login (z. B. durch max_age=0, prompt=login o.ä.) erzwungen • IDP seitig liegen Anhaltspunkte vor, dass die client_id der Anwendung gegenüber des Fachdienstes von nutzerinteraktionslosem SSO auszuschließen ist

Hierzu genügt es dem Nutzer anzuzeigen, mit welcher Anwendung auf welchen Fachdienst zugegriffen werden soll, die mit einem einfachen "OK" durch das Frontend des IDP bestätigt werden kann.**[**<=, , **]**

A_30668 -ID_TOKEN Version 2 oder neuer bei IDP-controlled-SSO
Ein sektoraler IDP, der IDP-controlled-SSO untertützt, MUSS ID_TOKEN in Version 2.0.0 oder neuer ausstellen. Veraltete `amr` nach A_23129* dürfen nicht verwendet werden.**[**<=, , **]**

3.3.9 neues Kapitel 4.2.4.5 "Nutzerspezifische Einstellungen"

Da die Steuerung der SSO Anmeldungen bei IDP-controlled-SSO im Backend erfolgt, werden dem Nutzer Möglichkeiten eingeräumt das Verhalten in gewissem Umfang beeinflussen zu können.

A_30669 -Maximaler Gültigigkeitszeitraum der SSO Sitzung
Ein sektoraler IDP, der IDP-controlled-SSO unterstützt, MUSS dem Nutzer in seinem Frontend nach erfolgreicher Authentifizierung die Möglichkeit einräumen, die höchdauer der SSO Session anpassen zu können.
Die Mindestdauer von 5 Minuten soll dabei nicht unterschreitbar und die Höchstdauer von 6 Stunden darf nicht überschreitbar sein.
Nimmt der Nutzer keine eigene Konfiguration vor, so beträgt diese standardmäßig 60 Minuten.
Wählt der Nutzer eine Höchstdauer über einer Stunde, so muss der Nutzer dies nach einem Warnhinweis mit der Authentifizierung auf den Vertrauensnivau gematik-ehealth-loa-high bestätigen.**[**<=, , **]**

A_30670 -Maximal gleichzeitige SSO Sitzungen
Ein sektoraler IDP, der IDP-controlled-SSO unterstützt, MUSS dem Nutzer in seinem Frontend nach erfolgreicher Authentifizierung die Möglichkeit einräumen, die maximal gleichzeitig möglichen SSO Sessions bis zu einer Höchstanzahl von 5 gleichzeitigen Sessions anpassen zu können.

Nimmt der Nutzer keine eigene Konfiguration vor, so beträgt diese standardmäßig 1.
Wählt der Nutzer 0, so ist IDP-controlled-SSO für seine gesundheitsID deaktiviert.
Wählt der Nutzer eine Anzahl größer 2 so muss der Nutzer dies nach einem Warnhinweis mit der Authentifizierung auf den Vertrauensnivau gematik-ehealth-loa-high bestätigen.
[<=, , **]**

Hinweis: Nach aktueller Spezifikationslage bedeuted diese Konfiguration, auf wie vielen unterschiedlichen Geräten der Nutzer gleichzeitig die Anwendung seiner Kasse, in dem das ePa-FdV eingebettet ist, mit aktivem SSO innerhalb der Anwendung nutzen kann. Sitzungen bleiben bis zu deren Timeout IDP-seitig erhalten. Wird die Anwendung, in dem der Nutzer per SSO Authentisiert wurde auf dem Mobilgerät zu lange nicht verwendet, muss der Nutzer sich erneut Authentifizieren, auch wenn im IDP noch eine nicht abgelaufende Sitzung läuft.

A_30671 -Verhalten bei Überschreitung gleichzeitig zulässiger SSO Sitzungen
Ein sektoraler IDP, der IDP-controlled-SSO unterstützt, MUSS dem Nutzer in seinem Frontend nach erfolgreicher Authentifizierung die Möglichkeit einräumen, das Verhalten bei Überschreitung der maximal zulässigen parallel laufenden SSO Sitzungen anpassen zu können.
Zur Auswahl stehen dabei folgende Optionen (sinngemäß):

1. älteste Sitzung beenden, wenn neue Sitzung authentifiziert wurde
2. neue Sitzungen mit Fehlermeldung ablehnen
3. Auswahldialog zur Beendigung einer auszuwählenden laufenden Sitzung

Nimmt der Nutzer keine eigene Konfiguration vor, so ist der Auswahldialog (3) anzuzeigen.
Die Konfiguration darf durch den Nutzer im Authentisierungsablauf geändert werden können, wenn der Auswahldialog (3) angeboten wird.**[**<=, , **]**

4 Änderungen in der gemSpec_IDP_FD

4.1 Anpassungen in Kapitel "3.4 Entity Statements"

alt:

A_23034-02 -Entity Statement veröffentlichen
Ein Fachdienst als Relying Party der TI-Föderation MUSS seine Entity Configuration in einem selbst-signierten Entity Statement gemäß[OpenID Federation 1.1] ("Entity Statement") bereitstellen und im Internet verfügbar machen. Die Entity Configuration MUSS mindestens die in der folgenden Tabelle aufgeführten Metadaten enthalten:

Tabelle 17: Header des Entity Statement der Relying Party der TI-Föderation

Name	Werte / Wertebereich
`alg`	string, zulässiger Wert: "ES256"
`kid`	string, UUID7-Format [RFC9562#name-uuid-version-7]
`typ`	string, zulässiger Wert: "entity-statement+jwt"

Tabelle 18 : Allgemeine Attribute der Entity Configuration im .well-known-Dokument der Relying Party der TI-Föderation

Name	Werte / Wertebereich
`iss`	string, URL nach [RFC1738]
`sub`	string, URL nach [RFC1738]
`iat`	number, Alle time-Werte in Sekunden seit 1970,[RFC7519#section-2]
`exp`	number, Alle time-Werte in Sekunden seit 1970,[RFC7519#section-2]
`jwks`	Set von JWK [RFC7517], zulässige Werte sind, gemäß [OpenID Federation "Claims that MUST or MAY Appear in both Entity Configurations and Subordinate Statements"] - jwks, nur die öffentlichen Schlüssel zu Schlüsseln,

	mit den das Entity Statement signiert ist.
`authority_hints`	[string], zulässige Werte gemäß [OpenID Federation "Claims that MUST or MAY Appear in Entity Configurations but Not in Subordinate Statements"] - authority_hints
`trust_marks`	[JSON Object], Optional, wenn Trust Marks für die Relying Party ausgestellt wurden, werden sie hier propagiert. zulässige Werte gemäß [OpenID Federation "Claims that MUST or MAY Appear in Entity Configurations but Not in Subordinate Statements"] - trust_marks
metadata	JSON Object, erforderlicher Wert: "openid_relying_party"

684 **Tabelle 19 : Attribute des Metadatenblocks openid_relying_party der Entity**
685 **Configuration im .well-known-Dokument der Relying Party der TI-Föderation**

Name	Werte
signed_jwks_uri (*)	string, URL nach [RFC1738]
jwks (*)	Set von JWK [RFC7517]
client_name	string, Wertebereich: ^[à-üÀ-Üß\w\ \-\.\+*\/]{1,128}$
organization_name	string (gemäß [OpenID-Federation "Informational Metadata Extensions"] - organization_name) Wertebereich: ^[à-üÀ-Üß\w\ \-\.\+*\/]{1,128}$
keywords	[string] (gemäß [OpenID-Federation "Informational Metadata Extensions"] - keywords) erforderliche Werte: "product_type_version:<von der gematik zugelassene Produkttyp-Version>" "product_type:<von der gematik zugelassener Produkttyp>"
contacts	[string] (gemäß [OpenID-Federation "Informational Metadata Extensions"] - contacts) erforderlicher Wert: "<E-Mail-Adresse für Supportanfragen>"
redirect_uris	[string], Wertebereich: Bei der Registrierung des Fachdienstes hinterlegte redirect_uris
response_types	[string], zulässiger Wert: "code"
client_registration_types	[string], zulässiger Wert: "automatic"
grant_types	[string], zulässiger Wert: "authorization_code"
require_pushed_authorization_requests	boolean, zulässiger Wert: true

`token_endpoint_auth_method`	string, zulässiger Wert: "self_signed_tls_client_auth"
`default_acr_values`	[string], zulässiger Wertebereich: "gematik-ehealth-loa-high"
`id_token_signed_response_alg`	string, zulässiger Wert: "ES256"
`id_token_encrypted_response_alg`	string, zulässiger Wert: "ECDH-ES"
`id_token_encrypted_response_enc`	string, zulässiger Wert: "A256GCM"
`scope`	string
ti_features_supported {	
`id_token_version_supported`	[string] zulässige Werte in Liste: "1.0.0", "2.0.0"
}	

(*) - gemäß [OpenID Federation 1.1] darf der Metadatenblock nur entweder`signed_jwks_uri` oder `jwks` `enthalten`.
[<=, , **]**

neu:

A_23034-03 ergänzt um neue zulässige Werte für ti_features_supported

	Werte
`sso_version_supported`	[string], optional Zulässige Werte in Liste: "idp_controlled"

alt:

A_27505 -Signalisierung der unterstützten TI-Feature-Versionen durch einen Fachdienst der TI-Föderation
Ein Fachdienst als Relying Party der TI-Föderation MUSS in seinem Entity Statement im Metadatenblock `openid_relying_party` in einem Claim `ti_features_supported` signalisieren, welche spezifischen Versionen der TI-Föderation unterstützt werden. Im Claim`ti_features_supported` MUSS die Relying Party die Unterstützung der in Tabelle "Durch eine Relying Party unterstütze TI-Features" genannten Claims signalisieren.

Tabelle 20 : Durch eine Relying Partyunterstütze TI-Features

claim	Wertebereich	Beschreibung
`id_token_version_supported`	[string], zulässige Werte in Liste: "1.0.0", "2.0.0"	Mit A_22867-* und A_23207-* ändert sich die Syntax des vom sektoralen IDP ausgestellten ID_TOKEN nicht abwärtskompatibel. Für einen Übergangszeitraum muss eine Relying Party die beiden Versionen: • 1.0.0 nach A_22867-01 und A_23207-02, • 2.0.0 nach A_27591 und A_27592, unterstützen.

[<=, Aktensystem_ePA, Anw_DiGA, TI-M_FD_ePA, extNutz_GID, IDP-D, digi_ID_OGR, funkt. Eignung: Herstellererklärung, organ./betriebl. Eignung: Anbietererklärung, funkt. Eignung: Anbietererklärung**]**

neu:

A_27505-01 Signalisierung der unterstützten TI-Feature-Versionen durch einen Fachdienst der TI-Föderation

Ein Fachdienst als Relying Party der TI-Föderation MUSS in seinem Entity Statement im Metadatenblock `openid_relying_party` in einem Claim `ti_features_supported` signalisieren, welche spezifischen Versionen der TI-Föderation unterstützt werden. Im Claim`ti_features_supported` MUSS die Relying Party die Unterstützung der in Tabelle "Durch eine Relying Party unterstütze TI-Features" genannten Claims signalisieren.

Tabelle 21 : Durch eine Relying Partyunterstütze TI-Features

claim	Wertebereich	Beschreibung
`id_token_version_supported`	[string], zulässige Werte in Liste: "1.0.0", "2.0.0"	Mit A_22867-* und A_23207-* ändert sich die Syntax des vom sektoralen IDP ausgestellten ID_TOKEN nicht abwärtskompatibel. Für einen Übergangszeitraum muss eine Relying Party die beiden Versionen: • 1.0.0 nach A_22867-01 und A_23207-02, • 2.0.0 nach A_27591 und A_27592, unterstützen.
`sso_version_supported`	[string], optional Zulässige Werte in Liste: "idp_controlled"	Der Wert idp_controlled darf nur angegeben werden, wenn IDP-controlled-SSO auf Anwendungsebenedurch den Fachdienst Authorization Server

		unterstützt wird

4.2 Anpassungen in Kapitel 3.5 "Anfrage von ID_TOKEN beim sektoralen Identity Provider"

neu:

nach A_24938

A_30672 -Weiterleitung von SSO Informationen
Fachdienste als Relying Party der TI-Föderation MÜSSEN vom Client im Authorization Request übermittelte SSO Informationen an den sektoralen IDP weiterleiten.
Dazu ist das im Parameter `authorization_details` des Authorization Request enthaltende Objekt vom Typ [RFC9396#section-2] `urn:ti:gematik:auth:sso:client_information` auszulesen und gemeinsam mit dem gemäß A_30673* erzeugten Objekt vom Typ `urn:ti:gematik:auth:sso:client_assertion` in die `authorization_details` des PAR zu übernehmen.
Das Objekt`urn:ti:gematik:auth:sso:client_assertion` ist gemäß dem Schema nach A_30655* zu erzeugen.**[**<=, , **]**

neu:

A_30673 -Erzeugung der SSO Assertion
Fachdienste als Relying Party der TI-Föderation MÜSSEN dem sektoralen IDP im Pushed Authorization Request die SSO Client Assertion übertragen, wenn vom Client im Authorization Request SSO Informationen übermittelt wurden.
Das Objekt`urn:ti:gematik:auth:sso:client_assertion` ist gemäß dem Schema nach A_30655* [gemSpec_IDP_Sek] zu erzeugen.

- der `client_type` claim darf durch den Fachdienst nur mit "epafdv" befüllt werden, wenn die Authentizität des Clients gesichert festgestellt werden kann und der Fachdienst zum Befüllen dieses Wertes berechtigt ist, sonst ist das Attribut weg zu lassen.
  Diese Berechtigung ist gegeben, wenn der Fachdienst das TI-Föderation Trustmark vom Typ `https://gemspec.gematik.de/trustmarks/client_assertion_epafdv` besitzt oder der Fachdienst vom gleichen Anbieter betrieben wird, wie das zugelassene ePa-FdV.
- das `authz_code_challenge` claim ist mit Wert der `code_challenge` des Authentication Requests des Clients zu belegen

[<=, , **]**

neu:

A_30674 -Erzeugung der SSO Assertion Signatur
Fachdienste als Relying Party der TI-Föderation MÜSSEN die integrität der SSO Client Assertion kryptografisch absichern.
Die SSO Assertion `signature` folgt dabei den Bildungsvorschriften zur Erzeugung einer JWS Signatur vom Typ `signature_alg` mit dem unter `signatare_kid` addressierten public key des Fachdienstes ohne protected header.
Die Normalisierung der Payload zur Erzeugung der Signatur erfolgt dabei

fürassertion_version 1 nach folgendem Schema: Die Werte von client_type, client_id und authz_code_challenge sind in der genannten Reihenfolge mit einem Doppelpunkt konkateniert. An diese Zeichenkette ist mittels Komma separiert, der Wert des sso.jwe (client_information_encrypted aus den urn:ti:gematik:auth:sso:client_information) angehängt: "<client_type>:<client_id>:<authz_code_challenge>,<sso.jwe>"
Die Signatur ist mit dem vomsignature_alg gewählten Algorithmus und dem unter signatare_kid addressierten Schlüssel des Fachdienstes von der BASE64URL kodierten Payload zu erzeugen und anschließen BASE64URL kodiert in das signature Attribut zu übernehmen.**[**<=, , **]**

ML-133234 entfällt (sieheC_12871_Anlage)

alt:

ML-133234 -AF_10117 - PAR URI
Die gelieferte Antwort enthält die PAR URI, mittels derer das Anwendungsfrontend oder Web-Backend die Benutzerauthentifizierung initiieren kann.**[**<=**]**

neu:

Erweiterung ML-133233

ML-133233 - AF_10117 - Antwort auf den Pushed Authorization Request

Das Anwendungsfrontend oder Web-Backend hat als Antwort auf den Pushed Authorization Request vo

- entweder HTTP-201 und die URI-PAR
- oder einen HTTP-Error Code

Optional kann die Antwort das Attribut rp_code_retrieval mit dem Wert true beinhalten.
[<=]

nach ML-133237:

neu:

A_30675 -Unterstützung des SSO noninteractive Flow
Fachdienste als Relying Party der TI-Föderation, die App-Sprung freies IDP-controlled-SSO unterstützen, KÖNNEN im AF_10118* von den Schritten 1-3 abweichen, wenn in ML-133234-01 das attributrp_code_retrieval mit dem Wert true empfangen wurde und der Fachdienst alternativ den Abruf des AUTHORIZATION_CODE wie folgt unterstützt:
Der Fachdienst muss die im ML-133234* empfangene PAR URI an den relying_party_code_retrieval_endpoint des IDP senden um den AUTHORIZATION_CODE abzurufen,dabei authentisiert er sich als Client der TLS Verbindung mit einem Zertifikat das über sein Entity Statement validierbar ist.**[**<=, , **]**

nach A_23500

neu:

A_30677 -SSO spezifische Ablehnungen des PAR
Fachdienste als Relying Party der TI-Föderation MÜSSEN Frontend-spezifische SSO Fehlermeldungen des Pushed Authorization Request an den IDP, an das Fachdienst-Frontend zurückgeben.
Dies ist der Fall, wenn der PAR mit einer HTTP/400 (Bad Request) Error Response

[RFC6749#section-4.1.2.1] mit einem error_code 79310 bis einschließlich 79319 abgewiesen wurde.
Dann ist der Authorization Request des Fachdienst-Frontend mit dem gleichen error_code in der Error Response abzulehnen.**[**<=, , **]**

Hinweis: Die Liste der der bereits vorgesehenen Fehlernummern des genannten Bereiches sind der Tabelle "Tab_gemSpec_Perf_Fehlercodes_sektoraler_IDP" in der A_22862* [gemSpec_Perf] zu entnehmen

4.3 Anpassungen in Kapitel 3.8 ACCESS_TOKEN

nach A_23079

neu:

A_30678 -Bereitstellen von SSO Session Informationen
Fachdienste als Relying Party der TI Föderation MÜSSEN bei der Bereitstellung eines Access Token am Endpunkt nach A_23076, die nach der Autorisierung gemäß AF_10118 vom IDP gemäß A_30676* empfangenen SSO Session Informationen dem Client mit bereitstellen.
Dazu sind die SSO Session Informationen in der Token-Response gemäß [RFC9396#name-token-response] in den `authorization_details` auszugeben, wie vom IDP empfangen.**[**<=, , **]**

5 Änderungen in gemSpec_IDP_FedMaster

5.1 Änderungen in Kapitel 4.3 "Anforderungen an den Trust Anchor der TI-Föderation (Federation Master)"

alt

A_30010 -Trust Mark Anbieter
Der Anbieter des Federation Master MUSS die Liste der verfügbaren Trust Marks und welche Superior Entities diese ausstellen dürfen, mittels eines organisatorischen Prozess auf Anforderung durch die Gematik anpassen können.
Initial umfasst diese

Trust Mark	Herausgeber
`https://gemspec.gematik.de/trustmarks/zeta_guard`	• <uri des Anbieters der ZETA-TI-Intermediate>

[<=, Anb_IDP_FedMaster, organ./betriebl. Eignung: Anbietererklärung**]**

neu

A_30010-01 - Trust Mark Anbieter

Der Anbieter des Federation Master MUSS die Liste der verfügbaren Trust Marks und welche Superior Entities diese ausstellen dürfen, mittels eines organisatorischen Prozess auf Anforderung durch die Gematik anpassen können.
Initial umfasst diese

Trust Mark	Herausgeber
`https://gemspec.gematik.de/trustmarks/zeta_guard`	• <iss uri des Anbieters der ZETA-TI-Intermediate>
`https://gemspec.gematik.de/trustmarks/client_assertion_epafdv`	• <iss uri des Anbieters des Federation Master>

6 Änderungen in gemSpec_Perf

6.1 Änderungen in Kapitel 3.1.2 "Betriebsdatenerfassung v2 Spezifika Identity Provider"

alt:

A_22826-01 -Performance - Betriebsdatenlieferung v2 - Spezifika sektoraler IDP - Status
Wenn bei der Durchführung der Operation/des Use Case ein Fehler aufgetreten ist, MUSS der Produkttyp sektoraler IDP bei Betriebsdatenlieferungen bzgl. des "status"-Feldes - den Statuscode gem. Tab_gemSpec_Perf_Fehlercodes_sektoraler_IdP festlegen, sofern ein spezifischer Fehlercode bestimmt werden kann. Ist dies nicht möglich, MUSS der definierte Standardcode für interne bzw. externe Fehler verwendet werden.

Tabelle 22: Tab_gemSpec_Perf_Fehlercodes_sektoraler_IDP

Statuscode	Definition	Beschreibung	Bewertung
302	IDP_REDIRECT	Weiterleitung der Anfrage zum Authenticator	SUCCESS
79000	IDP_ERROR	alle internen Fehler des sektoralen IDP	FAILED_SERVICE
79105	SEK_IDP_ERROR_NOT_ALLOWED_USER	Useragent/Version/ ClientID-Kombination nicht erlaubt	FAILED_OTHER
79106	SEK_IDP_AS_nPA_TIME_OUT	Abbruch der Anfrage nach time-out (online Ausweisfunktion)	FAILED_OTHER
79107	SEK_IDP_AS_nPA_USER_FAILURE	Alle Fehler der third party online Ausweisfunktion	FAILED_OTHER
79108	SEK_IDP_AS_eGK_TIME_OUT	Abbruch der Anfrage nach time-out (eGK)	FAILED_OTHER
79109	SEK_IDP_AS_eGK_USER_FAILURE	Alle Fehler der third party eGK	FAILED_OTHER
79110	SEK_IDP_AS_native_TIME_OUT	Abbruch der Anfrage nach time-	FAILED_OTH

		out	ER
79111	SEK_IDP_AS_native_USER_FAILURE	Alle Fehler der third party	FAILED_OTHER
79113	SEK_IDP_No_commonly_TI_Features_supported_version	Keine gemeinsam unterstütze Feature-Version vorhanden	FAILED_OTHER

[<=, IDP-Sek, funkt. Eignung: Herstellererklärung**]**

neu:

A_22826-03 mit zusätzlichen Werten in der Tabelle

Statuscode	Definition	Beschreibung	Bewertung
79300	SEK_IDP_SSO_ERROR	allgemeiner Fehler bei der Verarbeitung der übermittelten SSO Informationen	FAILED_OTHER
79301	SEK_IDP_SSO_ASSERTION_ERROR	Fehlerhafte SSO Assertion	FAILED_OTHER
79310	SEK_IDP_SSO_JWE_ERROR	Fehlerhafte SSO Client Informationen, Entschlüsselung nicht möglich	FAILED_OTHER
79311	SEK_IDP_SSO_JWT_ERROR	Fehlerhafte SSO Client Informationen	FAILED_OTHER
79312	SEK_IDP_SSO_SESSION_EXPIRED	Die SSO Session ist abgelaufen	FAILED_OTHER
79313	SEK_IDP_SSO_USER_LIMIT_REACHED	Maximal gleichzeitig für den Nutzer zulässige SSO Sitzungen wurden überschritten	FAILED_OTHER