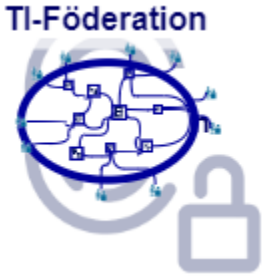
Web-App Flow (1 Gerät)

Diese Seite beschreibt die Nutzung einer Web-Anwendung über einen Browser *auf einem mobilen Endgerät* im Kontext föderierter IDPs.



Einführung

Der OAuth 2.0 Flow ist im Detail abhängig von der konkreten Anwendungsarchitektur. Die Spezifikation OAuth 2.0 for Browser-Based Apps unterscheidet hier drei mögliche Architekturansätze, mit unterschiedlichen Auswirkungen auf den OAuth 2.0 Flow und den damit verknüpften Bedingungen.

- Die Fälle OAuth 2.0 for Browser-Based Apps - 6.1, OAuth 2.0 for Browser-Based Apps - 6.2 müssen in unserer Betrachtung nicht unterschieden werden.
- Der Fall, dass eine Fachanwendung nur im Browser läuft (OAuth 2.0 for Browser-Based Apps - 6.3) - also eine reine Browseranwendung darstellt - wird nicht als relevanter UseCase für eine Anwendung des Gesundheitswesens mit Bedarf einer Nutzerauthenitisierung betrachtet.

Bei Web-Anwendungen sollten auf Seiten des jeweiligen Fachdienstes die in OAuth 2.0 Security Best Current Practice beschriebenen Sicherheitsaspekte berücksichtigt werden.

Der Ablauf des OIDC Flow ist prinzipiell identisch mit dem App-zu-App-Flow.

Terminologie

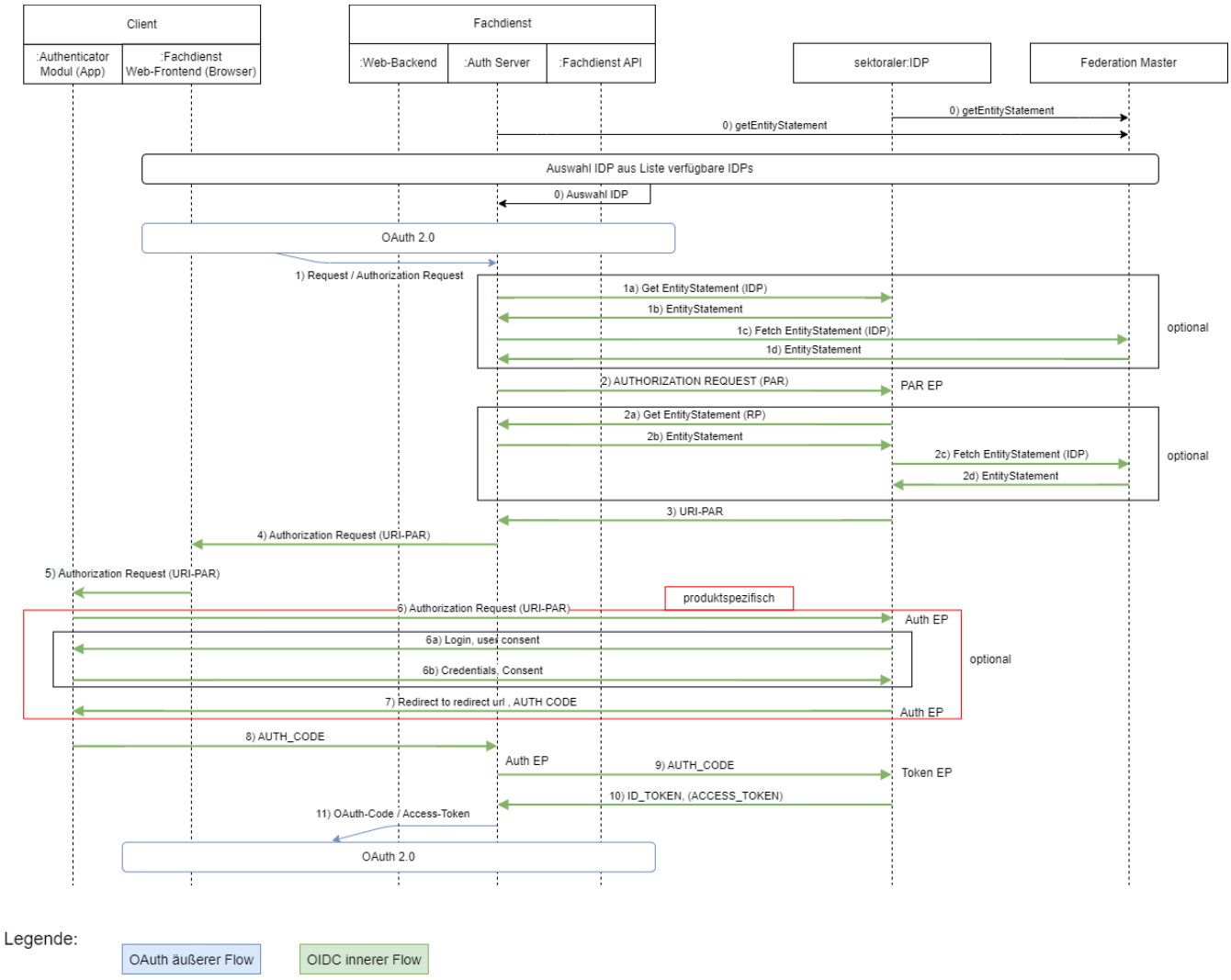
Erläuterungen zur verwendeten Terminologie sind zentral für alle Flows hier abgelegt.

Vorbedingungen

- Registrierung der Fachanwendung als RP beim Federation Master
- Registrierung des App-Link/Universal-Link für das Authenticator-Modul des IDP auf dem Gerät des Nutzers (auf Adresse des IDP) - oder anfragen über Web.
- Aktueller Signaturschlüssel des Federation Master ist bekannt und vertrauenswürdig bei IDP und Fachdienst eingebracht worden.
- Sektorale IDP ist Teil des TI-Vertrauensraums und beim Federation Master registriert.
- Der Fachdienst besitzt ein Web-Backend welches Anwendungslogik realisiert.

Flow - OIDC

Flow Diagramm



Legende:

OAuth äußerer Flow

OIDC innerer Flow

Textuelle Beschreibung der Aktivitäten je Transaktion sowie in den Übergängen

Schritt		Beschreibung

0		• Abruf der Schlüssel des Federation Master
		• Flow zur Auswahl des IDP siehe "Flow - Ermittlung und Auswahl IDP"
		  ◦ Die Auswahl des richtigen IDP ist optional. Ist der IDP bekannt (z.B. durch eine frühere Autorisierung) entfällt der Schritt
		  ◦ Es gibt unterschiedliche Möglichkeiten den Ablauf der IDP-Ermittlung zu gestalten. Spätestens zum Schritt (1a) muss der Ziel-IDP bekannt sein

1		Abweichend vom APP/APP-Flow kommt der Request vom Web-Backend der Anwendung und nicht von einem Anwendungsfrontend (App)
	1-a	Schnittstellendetails analog App-zu-App Flow (1a)
	1-b	Schnittstellendetails analog App-zu-App Flow (1b)
	1-c	Schnittstellendetails analog App-zu-App Flow (1c)
	1-d	Schnittstellendetails analog App-zu-App Flow (1d)
2		Schnittstellendetails analog App-zu-App Flow (2)
	2-a	Schnittstellendetails analog App-zu-App Flow (2a)
	2-b	Schnittstellendetails analog App-zu-App Flow (2b)
	2-c	Schnittstellendetails analog App-zu-App Flow (2c)

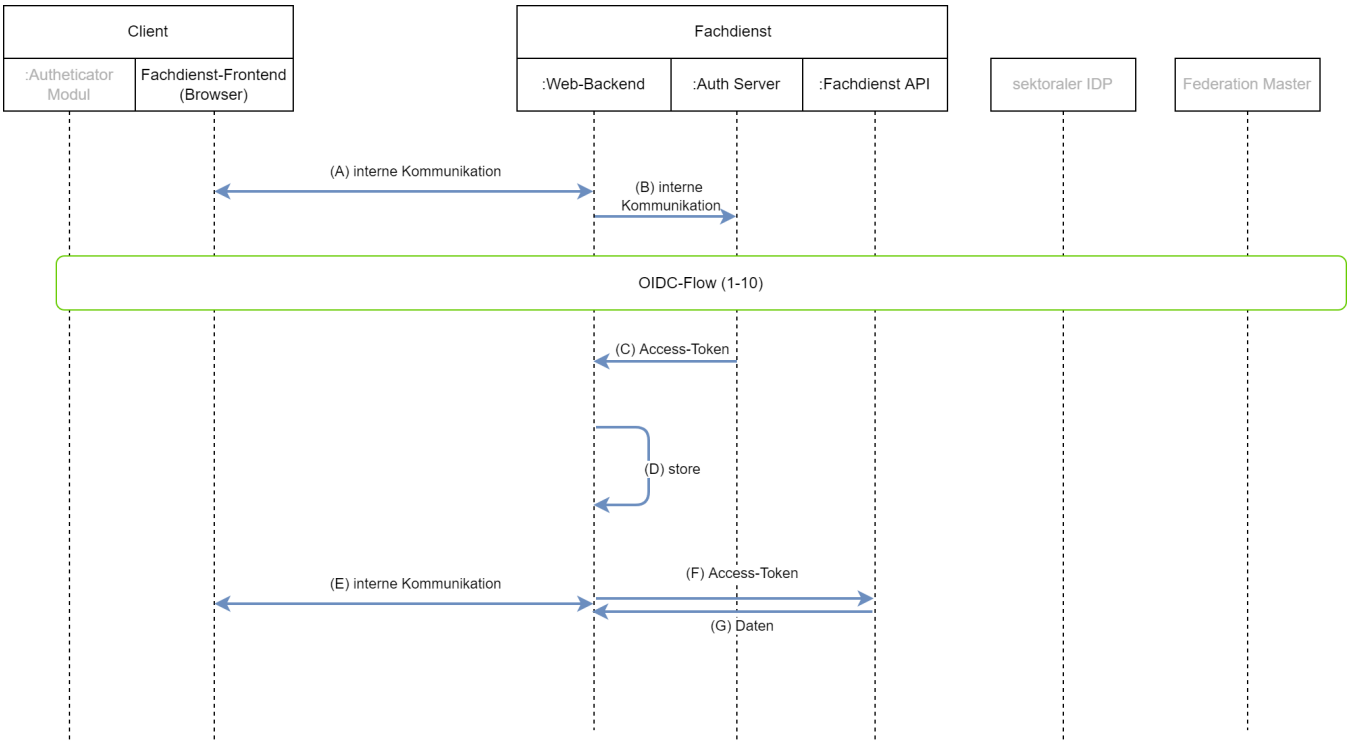
	2-d	Schnittstellendetails analog App-zu-App Flow (2d)
3		Schnittstellendetails analog App-zu-App Flow (3)
4		Abweichend vom APP/APP-Flow läuft der Redirect über das Web-Backend zum Web-Frontend. Schnittstellendetails analog App-zu-App Flow (4)
5		Schnittstellendetails analog App-zu-App Flow (5)
6		Schnittstellendetails analog App-zu-App Flow (6)
	6-a	Schnittstellendetails analog App-zu-App Flow (6a)
	6-b	Schnittstellendetails analog App-zu-App Flow (6b)
7		Schnittstellendetails analog App-zu-App Flow (7)
8		Abweichend vom APP/APP Flow führt das Authenticator Modul des IDP den Redirect zum Authorization-Service des Fachdienst aus und übergibt den "AUTHORIZATION_CODE". Schnittstellendetails analog App-zu-App Flow (9)
9		Schnittstellendetails analog App-zu-App Flow (10)
10		Schnittstellendetails analog App-zu-App Flow (11)
11		Der Autorisierungsserver des Fachdienst reicht ACCESS_TOKEN und REFRESH_TOKEN an das Web-Backend der Anwendung weiter. Diese liegen zu keiner Zeit im Browser des Nutzers.
12		Der Access-Token (Refresh-Token) wird im Web-Backend der Anwendung persistiert. Die Kommunikation zwischen Web-Frontend und Web-Backend ist implementierungsspezifisch. Der Zugriff auf das Fachdienst-API erfolgt über das Web-Backend. Der Access-Token muss bei jedem Zugriff mitgegeben werden.

Flow - OAuth 2.0 Flow mit Web-Backend in Fachdienst-Domäne

Der OAuth 2.0 Flow basiert auf der Spezifikation https://tools.ietf.org/id/draft-ietf-oauth-browser-based-apps-07.html#name-browser-based-apps-that-can.

Werden Web-Backend, Authentifizierungsservice und Resourcenserver eines Fachdienst in einer Domäne betrieben, so ist gemäß OAuth 2.0 for Browser-Based Apps - 6.1 OAuth konformes Redirect und das Ausstellen von Authorization-Code nicht notwendig. Für die Kommunikation zwischen Web-Backend und Browser muss ein entsprechend sicheres Pattern verwendet werden (z.B. Http-only cookie, Split Access Token Cookie Pattern), um die Anwendung gegen potentielle Arttacken abzusichern.

Flow Diagramm



Textuelle Beschreibung der Aktivitäten je Transaktion sowie in den Übergängen

Schritt		Funktion	Beschreibung	Standard
A			Die Kommunikation zwischen Web-Frontend (Browser) und Web-Backend ist implementierungsspezifisch und muss den Standards für Web-Anwendungen genügen.	• HTTP 1.1 https://tools.ietf.org/html/rfc2616/ • HTML https://html.spec.whatwg.org/
B		• Erzeugung eines Request zur Autorisierung des Anwenders	Das Web-Backend erzeugt einen Request gegen den Authorization-Service des Fachdienstes. Dieser Request kann anwendungsspezifisch sein und muss nicht zwingend ein Standard-konformer Authorization-Request sein. Das Erzeugen eines Codeverifier und einer Codechallenge ist nicht notwendig, da es sich hier um eine gemeinsame Domäne handelt, bestehend aus Web-Backend, Authorization-Service und Fachdienst-API.	-
C		• Erzeugung und Speicherung eines Access-Token	Der Authorization-Service stellt dem Web-Backend ein Access-Token aus, welches dieser für die sichere Kommunikation des Fachdienstes verwendet.	• OAuth 2.0 for Browser-Based Apps https://tools.ietf.org/id/draft-ietf-oauth-browser-based-apps-07.html#name-javascript-applications-wit
D		• Speicherung des Access-Token	Das Web-Backend speichert sich das Access-Token für folgende Zugriffe auf Services des Fachdienst (Fachdienst-API)	-

E			Datenanfragen und -aufbereitung in der Kommunikation zwischen Web-Frontend (Browser) und Web-Backend ist implementierungsspezifisch und muss den Standards für Web-Anwendungen genügen.	• HTTP 1.1 https://tools.ietf.org/html/rfc2616/ • HTML https://html.spec.whatwg.org/
F		• Request mit Access-Token erzeugen	Datenanfragen des Web-Backend an das Fachdienst-API müssen das Access-Token enthalten.	• The OAuth 2.0 Authorization Framework https://datatracker.ietf.org/doc/html/rfc6749#section-7.1
G		• Access-Token prüfen • Datenanfrage beantworten	Der Fachdienst prüft den Access-Token in Datenanfragen des Web-Backend gegen den Authorization-Service und stellt die Daten bereit.	• OAuth 2.0 Authorization Framework: Bearer Token Usage https://datatracker.ietf.org/doc/html/rfc6750#section-2

Flow - Ermittlung und Auswahl IDP

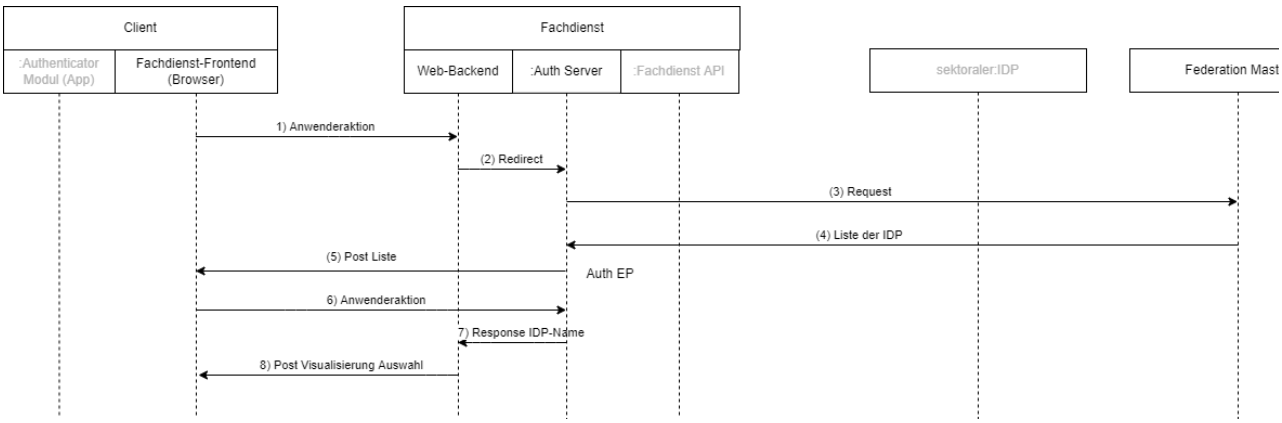
Die konkrete Umsetzung ist hier Anwendungsspezifisch.
Zum einen kann der Autorisierungsserver die Auswahlliste der IDPs erst bei Eingang eines AUTHORIZATION_REQUEST durch das Web-Backend zur Anzeige beim Nutzer bringen lassen.

- Alternativ ruft das Web-Frontend beim Autorisierungsserver die Liste aller IDPs ab. Die Ermittlung der registrierten IDPs erfolgt über den Federation-Master. Beim Federation-Master sind die Entity-Statements aller registrierten IDP hinterlegt. Die Bereitstellung der Liste kann über zwei Wege erfolgen:

  a) Der Client verwendet das OIDC Federation API. Der Client muss dann aus dem Response die für eine Auswahl notwendigen Informationen extrahieren.

  b) Der Federation-Master stellt ein zusätzliches API neben dem Standard-API bereit und liefert hier nur die für eine Auswahl notwendigen Informationen (Name der Organisation/Kasse, Icon, weitere Informationen für Folge-Request zur Ermittlung des vollständigen Entity-Statement). Die Adresse des API kann z.B. als custom-metadata im Entity-Statement des Federation Master hinterlegt werden.

Flow Diagramm (Abruf der IDP-Liste durch Web-Frontend)



Textuelle Beschreibung der Aktivitäten je Transaktion sowie in den Übergängen

Schritt	Beschreibung	Standard
1	Anwenderaktion am Web-Frontend der Anwendung zur Auswahl eines IDP	

2	Redirect des Web-Backend zum Authorization-Service des Fachdienstes zur Bereitstellung der Auswahl	
3	Anfrage des Authorization-Service des Fachdienstes am Federation Master nach Liste der in der Föderation registrierten IDPs	○ Entity Listings Request https://openid.net/specs/openid-connect-federation-1_0.html#rfc.section.7.3.1 ○ OP-Metadata *organisation_name* https://openid.net/specs/openid-connect-federation-1_0.html#OP_metadata ○ Metadata Erweiterung https://openid.net/specs/openid-connect-federation-1_0.html#metadata
4	Rückantwort des Federation Master mit der signierten Liste der in der Föderation registrierten IDPs	
5	Signaturprüfung der Liste und Darstellung der verfügbaren IDP am Web-Frontend des Fachdienstes	
6	Übertragung des durch den Nutzer am Web-Frontend des Fachdienstes ausgewählten IDP (aus der Liste der verfügbaren IDPs)	
7	Response auf den Redirect-Aufruf mit den Informationen zum ausgewählten IDP	
8	Visualisierung des ausgewählten IDP im Web-Frontend der Anwendung	

Schnittstellenbeschreibung

Exemplarische Beschreibung des Ablaufs einer Dienstnutzung sowie der Vor- und Nachbedingungen

(0) Abruf der Schlüssel des Federation Master

Der Abruf der Schlüssel des Federation Master erfolgt analog dem App-zu-App Flow (Federation Master)

IDP Liste

Beschreibung zur IDP-Liste ist im App-zu-App Flow (IDP-Liste).

(1) Authorization Request von Web-Backend zum Authentication Endpunkt (Auth ES) des Autorisierungsserver des Fachdienstes

Die Kommunikation zwischen Web-Frontend und Web-Backend ist anwendungsspezifisch. Das Web-Backend des Fachdienstes sendet einen Request an den Autorisierungsserver des Fachdienstes. Dieser Request ist ebenfalls anwendungsspezifisch. Damit der weitere Ablauf OIDC konform und weitest gehend identisch zum App-zu-App Flow ablaufen kann, muss der Request einigen Festlegungen genügen.

Das Web-Backend sendet ein HTTP-GET an den AS des Fachdienstes.

Die folgenden GET-Parameter werden im query string verwendet:

Name	Werte	Beispiel	Anmerkungen
client_id	VSCHAR	`"digaxy"`	kein ";" und kein "" (definiert gem. Unicode U+253C (9532)), kein Leerzeichen
state	VSCHAR	af0ifjsldkj	optional
redirect_uri	URL	"https://Fachdienst007.de"	Adresse des Fachdienst weil da soll der ACCESS_TOKEN am Ende landen.
code_challenge	Hash über code_verifier	K2-ltc83acc4h0c9w6ESC_rEMTJ3bww-uCHaoeK1t8U	PKCE optional weil Kommunikation innerhalb der Anwendung und nichts zum Browser fließt oder Redirects folgt.
code_challenge_method	S256	<-	PKCE optional, siehe oben
response_type	code	<-	CODE Flow optional wenn andere Mechanismen die Verbindung schützen
scope	"e-rezept"	<-	anwendungsspezifisch zu definieren kein open-id
weiterere Claims			weitere claims können vereinbart werden

idp_iss	URL	"https://idp4711.de"	nicht Standard Parameter iss URL des IDP den der Nutzer für die Authentisierung ausgewählt hat. optional - nötig wenn Auswahl des IDP im Frontend passiert.

(1 a) Falls der Autorisierungsserver des Fachdienstes das EntityStatement des IDP noch nicht kennt, lädt er dies herunter

Request analog App-zu-App Flow (1a):

(1 b) Der IDP sendet sein EntityStatement zurück

Response analog App-zu-App Flow (1b)

signed_jwks

Die Werte sind analog zu App-zu-App Flow (1-signed_jwks)

(1 c) Der Autorisierungsserver des Fachdienstes ruft das Entity Statement zum IDP beim Federation Master ab

Request analog App-zu-App Flow (1c)

(1 d) Der Federation Master sendet sein EntityStatement über den angefragten sektoralen IDP zurück

Response analog zu App-zu-App Flow (1d)

(2) Der Autorisierungsserver des Fachdienstes sendet ein (Pushed) Authorization Request an den Authentication Endpunkt (Auth ES) des sektoralen IDP

HTTP-POST analog App-zu-App Flow (2) inclusive TLS Clientauthentisierung.

(2 a) Falls der IDP das EntityStatement des Autorisierungsserver des Fachdienst noch nicht kennt, lädt er dies herunter.

Request analog App-zu-App Flow (2a)

(2 b) Der Autorisierungsserver des Fachdienst sendet sein EntityStatement zurück und der IDP registriert ihn als Client

Response analog App-zu-App Flow (2b)

signed_jwks

Die Werte sind analog zu App-zu-App Flow (2b-signed_jwks)

(2 c) Abruf des Entity Statement zum Fachdienst beim Federation Master

Request analog App-zu-App Flow (2c)

(2 d) Der Federation Master sendet sein Entity Statement über den Fachdienst zurück

Response analog App-zu-App Flow (2d)

(3) Der Authentication-Endpunkt (Auth EP) des sektoralen IDP antwortet dem AS des Fachdienst mit einer Request URI

Response analog App-zu-App Flow (3)

(4) Der Authorization Server des Fachdienst antwortet dem Frontend mit einem redirect und seiner Request URI

Abweichend vom APP/APP-Flow läuft der Redirect zum Web-Frontend.

Redirect analog App-zu-App Flow (4)

(5) Das Web-Frontend sendet den Authentication Request an die URI des IDP und leitet ihn somit an das Authenticator Modul weiter

HTTP-GET analog App-zu-App Flow (5)

(6) Das Authenticator Modul leitet den Authentication Request an den IDP weiter. (proprietär)

Die Schritte zur Nutzer-Authentifizierung und zur Erstellung des AUTHORIZATION_CODE durch den IDP sind anwendungsspezifisch und werden hier nicht weiter spezifiziert.

(7) Der Authorization-Endpunkt des sektoralen IDP antwortet dem Authenticator Modul mit einem Redirect zum Fachdienst. (proprietär)

Redirect analog App-zu-App Flow (7)

(8) Das Authenticator Modul des IDP ruft über die Redirect-URL den Autorisierungsserver des Fachdienstes auf und übergibt den "AUTHORIZATION_CODE"

Abweichend vom App/App Flow führt das Authenticator-Modul des IDP den Redirect zum Authorization-Server des Fachdienstes aus und übergibt den AUTHORIZATION_CODE. Der Request wird mit einem HTTP-OK quittiert.

HTTP-POST analog App-zu-App Flow (9)

(9) Der Autorisierungsserver reicht den "AUTHORIZATION_CODE" und den "Code_Verifier" beim Token-Endpunkt des IDP ein.

HTTP POST analog App-zu-App Flow (10) inclusive TLS Clientauthentisierung.

(10) Der Autorisierungsserver erhält vom Token-Endpunkt des IDP einen ID_TOKEN und ACCESS_TOKEN mit den gewünschten Claims, der mit dem öffentlichen Schlüssel aus der Registrierung verschlüsselt ist.

Response analog App-zu-App Flow (11)

(11) Der Autorisierungsserver des Fachdienst reicht das ACCESS_TOKEN und REFRESH_TOKEN a n das Web-Backend der Anwendung weiter.

HTTP-200

- Content-Type=application/json
- Cache-Control=no-store
- Pragma=no-cache

Die JSON-Struktur sieht so aus:

{

"access_token": <ACCESS_TOKEN>,

"refresh_token": <REFRESH_TOKEN>,

"token_type": "Bearer",

"scope": "e-rezept",

"expires_in": 300, (Gültigkeit des ACCESS_TOKEN in Sekunden, https://tools.ietf.org/html/rfc6749 section 4.2.2)

}

(12) Kommunikation Web-Frontend, Web-Backend der Anwendung und Fachdienst-API

Das Web-Backend persistiert Access-Token und Refresh-Token. Das Web-Backend benötigt diese für die autorisierte Kommunikation mit dem Fachdienst-API. Die Kommunikation zwischen Web-Frontend und Web-Backend ist implementierungsspezifisch. Access-Token und/oder Refresh-Token werden nicht an das Frontend weitergereicht.

Das Web-Backend verwendet das Access-Token für die Kommunikation mit dem Fachdienst-API. Das Fachdienst-API prüft den Access-Token bevor Anfragen entsprechend quittiert werden.

```
GET /resource/1 HTTP/1.1
     Host: example.com
     Authorization: Bearer <ACCESS_TOKEN>
```