Entwicklerhilfen zur Anbindung eines Fachdienst an den IDP-Dienst

Auf dieser Seite finden sich Informationen zu Umgebungen, Referenzimplementierungen und Codebeispiele

GitHub-Projekt

Eine Referenzimplementierung des IDP-Dienstes sowie ein IDP-Server-Dockerimage, Testsuiten und Codebeispiel liegen im github-Projekt. Über die r bel-logs gelangt man zu beispielhaften Requests und Responses für Authentifizierungsabläufe mit HBA bzw. eGK.

- README https://github.com/gematik/ref-idp-server#readme
- IDP-Server als Docker-Image https://github.com/gematik/ref-idp-server#idp-server-as-docker-image
- rbel-logs https://github.com/gematik/ref-idp-server#tokenflow-sites
  - HBA (Primärsysteme) https://gematik.github.io/ref-idp-server/tokenFlowPs.html - Beschreibt den Kompletten Authentisierungsprozess vom Initialen Request bis zum erhalt der Token bei Verwendung eines Heilberufsausweis
  - eGK (Versichertensysteme, z.B. E-Rezept-App) https://gematik.github.io/ref-idp-server/tokenFlowEgk.html - Beschreibt den Kompletten Authentisierungsprozess vom Initialen Request bis zum erhalt der Token bei Verwendung einer elektronischen Gesundheitskarte

Anforderungen aus gematik-Spezifikationen an den Fachdienst

Die Spezifikationsdokumente, in denen die relevanten Anforderungen formuliert sind, finden sich im Fachportal der gematik. Über die Suchfunktion im Portal gelangt man durch Suche nach dem relevanten Spezifikationsdokument zur aktuellen Version der gesuchten Spezifikation.

Konkret gilt für den Fachdienst Kapitel 5.1 "Registrierung des Fachdienstes beim IDP-Dienst" aus gemSpec_IDP_FD (Spezifikation Identity Provider – Nutzungsspezifikation für Fachdienste) wobei beim Zugriff auf den IDP-Dienst in seiner Rolle als Identity Provider anstelle der ACCESS-TOKEN die ID -TOKEN verwendet werden.

Dies deckt die folgenden Anforderungen ab

Anforderungsnummer	Kurzbeschreibung	relevantes Spezifikationsdokument	Anmerkung
A_20739	Registrierung der Claims des Fachdienstes	gemSpec_IDP_FD	RISE_ServiceID_001 (RU) RISE_ServiceID_002 (TU) RISE_ServiceID_003 (PU)
A_20676	Nutzer-Informationen im Claim	gemSpec_IDP_FD	
A_20505-02	Inhalte der Claims für Leistungserbringer (HBA)	gemSpec_IDP_FD	
A_20506-03	Inhalte der Claims für Leistungserbringerinstitutionen (SMC-B)	gemSpec_IDP_FD	
A_21521	Fachdienst: Prüfung der Signatur des Discovery Document	gemSpec_IDP_FD	
A_20362	"ACCESS_TOKEN" generelle Struktur	gemSpec_IDP_FD	
A_20364	Unverschlüsselt eingehende ACCESS_TOKEN sind ungültig	gemSpec_IDP_FD	
A_20365-01	Die Signatur des "ACCESS_TOKEN" ist zu prüfen	gemSpec_IDP_FD	

A_20504	Reaktion bei ungültiger oder fehlender Signatur des "ACCESS_TOKEN"	gemSpec_IDP_FD	
A_20373	Prüfung der Gültigkeit des "ACCESS_TOKEN" für den Zugriff auf Fachdienste	gemSpec_IDP_FD	
A_20369-01	Abbruch bei unerwarteten Inhalten	gemSpec_IDP_FD	
A_20370	Abbruch bei falschen Datentypen der Attribute	gemSpec_IDP_FD	
A_21520	Prüfung des "aud" Claim des ACCESS_TOKEN mit der vom Fachdienst registrierten URI	gemSpec_IDP_FD	

Im präferierten Fall in dem der Authorization Server selbst den Request an den IDP-Dienst parametrisiert und auch mit dem Authorization Code das ID_Token abruft gelten für den Fachdienst selbst zusätzlich folgende Anforderungen aus gemSpec_IDP_Frontend Kapitel 8 "Funktionsmerkmale Anwendungsfrontend des IDP Dienstes". In dieser Situation ist es nicht notwendig am IDP-Dienst zusätzlich zum Fachdienst auch noch ein dediziertes Anwendungsfrontend zu Registrieren.

Anforderungsnummer	Kurzbeschreibung	relevantes Spezifikationsdokument	Anmerkung
A_20603	Organisatorische Registrierung des Anwendungsfrontends	gemSpec_IDP_Frontend	Erfolgt im Rahmen der Registrierung des Fachdienstes
A_20740	Bekanntgabe der Redirect-URI des Anwendungsfrontend	gemSpec_IDP_Frontend	
A_20512	Regelmäßiges Einlesen des Discovery Document	gemSpec_IDP_Frontend	
A_20623	Anwendungsfrontend: Prüfung der Signatur des Discovery Document	gemSpec_IDP_Frontend	
A_20309	Bildung von "CODE_VERIFIER" und "CODE_CHALLENGE"	gemSpec_IDP_Frontend	
A_20483	Formulierung und Inhalte der Anfrage zum "AUTHORIZATION_CODE" für einen "ACCESS_TOKEN"	gemSpec_IDP_Frontend	
A_21323	Erzeugung des "Token-Key"	gemSpec_IDP_Frontend	
A_21324	Erzeugen des "KEY_VERIFIER"	gemSpec_IDP_Frontend	
A_20529-01	Senden von "AUTHORIZATION_CODE" und "KEY_VERIFIER" an den Token-Endpunkt	gemSpec_IDP_Frontend	
A_19937	Fehlermeldungen des Token-Endpunktes Anzeige	gemSpec_IDP_Frontend	
A_20085	Fehlermeldungen des Anwendungsfrontends	gemSpec_IDP_Frontend	
A_20079	Ausfall der Fehlermeldung des Token-Endpunktes	gemSpec_IDP_Frontend	
A_19938-01	Annahme des ID_TOKEN	gemSpec_IDP_Frontend	
A_20625	Anwendungsfrontend: Prüfung der Signatur des ID_TOKEN	gemSpec_IDP_Frontend	
A_21327	Löschung von "ID_TOKEN"	gemSpec_IDP_Frontend	

Anforderungen aus gematik-Spezifikationen an ein integriertes Authenticator-Modul

Wenn nicht der gematik Authenticator für die Authentisierung verwendet wird sondern, analog zum E-Rezept, das Authenticator-Modul in ein Primärsystem oder eine andere Anwendung integriert wird so kommen dort auch die Vorgaben aus dem Kapitel 5.1.4 "Authentifizierung der LEI" des gemILF_PS_eRp zur Anwendung.

Im präferierten Fall, in dem der Authorization Server selbst den Request an den IDP-Dienst parametrisiert und nach der Authentisierung auch mit dem Authorization Code das ID_Token abruft, entfällt jedoch eine Reihe von Anforderungen, weil diese durch den Authorization Server des Fachdienst übernommen werden.

Anforderungsnummer	Kurzbeschreibung	relevantes Spezifikationsdokument	Anmerkung
~~A_20654~~	~~Registrierung des Primärsystems~~	gemILF_PS_eRp	übernimmt der Fachdienst mittels A_20603/A_20739
A_20655	Regelmäßiges Einlesen des Discovery Document	gemILF_PS_eRp	Wird zwar ebenfalls durch den Fachdienst mittels A_20512 durchgeführt aber notwendig für den Authentisierungsprozess
A_20656-01	Prüfung der Signatur des Discovery Document	gemILF_PS_eRp	Wird zwar ebenfalls durch den Fachdienst mittels A_20623 durchgeführt aber notwendig für den Authentisierungsprozess
A_20657	Prüfung der Signatur des Discovery Document	gemILF_PS_eRp	Wird zwar ebenfalls durch den Fachdienst mittels A_20623 durchgeführt aber notwendig für den Authentisierungsprozess
~~A_20659~~	~~Erzeugen des CODE_VERIFIER~~	gemILF_PS_eRp	übernimmt der Fachdienst mittels A_20309

~~A_20660~~	~~Erzeugen des Hash-Werts des CODE_VERIFIER~~	gemILF_PS_eRp	übernimmt der Fachdienst mittels A_20309
~~A_21333~~	~~Erzeugung des "Token Key"~~	gemILF_PS_eRp	übernimmt der Fachdienst mittels A_21323
~~A_21334~~	~~Erzeugung des "KEY_VERIFIER"~~	gemILF_PS_eRp	übernimmt der Fachdienst mittels A_21324
~~A_20671-01~~	~~Einreichen des AUTHORIZATION_CODE beim Token-Endpunkt~~	gemILF_PS_eRp	übernimmt der Fachdienst mittels A_20529-01
~~A_20672-01~~	~~Annahme des ID_TOKEN~~	gemILF_PS_eRp	übernimmt der Fachdienst mittels A_19938-01
~~A_20673-01~~	~~Annahme des "ACCESS_TOKEN"~~	gemILF_PS_eRp	nicht relevant
~~A_20674~~	~~Formale Prüfung der Signatur des ID_TOKEN~~	gemILF_PS_eRp	übernimmt der Fachdienst mittels A_20625
~~A_20675~~	~~Gültigkeitsprüfung der Signatur des ID_TOKEN innerhalb der TI~~	gemILF_PS_eRp	übernimmt der Fachdienst mittels A_20625
A_20658 A_21337 A_21338	Sicheres Löschen der Token Löschung von TOKEN bei zeitlichem Ablauf Sichere Speicherung der Token	gemILF_PS_eRp	gelten für weitergeleitete Auth_Codes und generell auch für eigene Token des Fachdienstes
A_20661	Anfrage des "AUTHORIZATION_CODE" für ein "ACCESS_TOKEN"	gemILF_PS_eRp	Den Request führt das integrierte Authenticator-Modul aus, die Parameter dazu werden aber vom Fachdienst bestimmt und zuvor übermittelt.
A_20662	Annahme des "user_consent" und des "CHALLENGE_TOKEN"	gemILF_PS_eRp	Muss das Authenticator-Modul durchführen
A_20663-01	Prüfung der Signatur des CHALLENGE_TOKEN	gemILF_PS_eRp	Muss das Authenticator-Modul durchführen
A_20664	Bestätigung des Consent	gemILF_PS_eRp	Muss das Authenticator-Modul durchführen
A_20665-01	Signatur der Challenge des IdP-Dienstes	gemILF_PS_eRp	Muss das Authenticator-Modul durchführen
A_20666-01	Auslesen des Authentisierungszertifikates	gemILF_PS_eRp	Muss das Authenticator-Modul durchführen
A_20667-01	Response auf die Challenge des Authorization-Endpunktes	gemILF_PS_eRp	Muss das Authenticator-Modul durchführen
A_20668	Annahme des "AUTHORIZATION_CODE"	gemILF_PS_eRp	Muss das Authenticator-Modul durchführen und leitet den Code dann weiter zum Fachdienst

Adressen des IDP-Dienstes

Umgebung	URL "issuer"	URL Discovery Dokument "uri_disc"
RU Internet	idp-ref.app.ti-dienste.de	https://idp-ref.app.ti-dienste.de/.well-known/openid-configuration
RU TI	idp-ref.zentral.idp.splitdns.ti-dienste.de	
PU Internet	idp.app.ti-dienste.de	https://idp.app.ti-dienste.de/.well-known/openid-configuration
PU TI	idp.zentral.idp.splitdns.ti-dienste.de	

Abholen des Discovery Dokument

Das Discovery Document der IDP-Dienst-Referenzimplementierung kann z. B. über den Browser durch Aufruf der Schnittstelle https://idp-ref.app.ti-dienste.de/.well-known/openid-configuration erfolgen. Das signierte JWT sieht in etwa so aus

```
eyJhbGciOiJCUDI1NlIxIiwia2lkIjoicHVrX2Rpc.....VR4R1lnQ1ZJNFFnPT0iXSwidHlwIjoiSldUIn0.
eyJpYXQiOjE2NzUzMzc0OTEsImV4cCI6MTY3NTQyM.....VzX3N1cHBvcnRlZCI6WyJwYWlyd2lzZSJdfQ.
T5r4MkQRhxqmMfcrDXaKoPGZqhWWQQ9R-tPHbQI6mf9JIlWnWVCen689U0JT5-jxoXYbLmzQEtWxIAcRKS7U8A
```

Nach der Decodierung bekommt man in der payload das Discovery-Dokument des IDP-Dienstes

```
{
  "iat": 1675337491,
  "exp": 1675423891,
  "issuer": "https://idp-ref.app.ti-dienste.de",
  "jwks_uri": "https://idp-ref.app.ti-dienste.de/certs",
  "uri_disc": "https://idp-ref.app.ti-dienste.de/.well-known/openid-configuration",
  "authorization_endpoint": "https://idp-ref.app.ti-dienste.de/auth",
  "sso_endpoint": "https://idp-ref.app.ti-dienste.de/auth/sso_response",
  "token_endpoint": "https://idp-ref.app.ti-dienste.de/token",
  "auth_pair_endpoint": "https://idp-ref.app.ti-dienste.de/auth/alternative",
  "uri_pair": "https://idp-pairing-ref.zentral.idp.splitdns.ti-dienste.de/pairings",
  "kk_app_list_uri": "https://idp-ref.app.ti-dienste.de/directory/kk_apps",
  "third_party_authorization_endpoint": "https://idp-ref.app.ti-dienste.de/extauth",
  "uri_puk_idp_enc": "https://idp-ref.app.ti-dienste.de/certs/puk_idp_enc",
  "uri_puk_idp_sig": "https://idp-ref.app.ti-dienste.de/certs/puk_idp_sig",
  "code_challenge_methods_supported": ["S256"],
  "response_types_supported": ["code"],
  "grant_types_supported": ["authorization_code"],
  "id_token_signing_alg_values_supported": ["BP256R1"],
  "acr_values_supported": ["gematik-ehealth-loa-high"],
  "response_modes_supported": ["query"],
  "token_endpoint_auth_methods_supported": ["none"],
  "scopes_supported": ["openid",   ....,   ....],
  "subject_types_supported": ["pairwise"]
}
```


Öffentliche Schlüssel und Zertifikate

Discovery Document, Access- und ID-Token sind signiert und teilweise für den Transport verschlüsselt. Für die Prüfung der Signaturen, und für die Schlüsselaushandlung zur Verschlüsselung benötigt eine Fachanwendung entsprechende öffentliche Schlüssel und Zertifikate. Diese können unter der URL, welche im Discovery Document unter dem Parameter "jwks_uri" angegeben ist, abgerufen werden.

So liefert der Aufruf von https://idp-ref.app.ti-dienste.de/certs drei Einträge

1. puk_idp_sig - öffentlicher Schlüssel und Zertifikat zur Validierung der vom IDP-Dienst ausgestellten Signaturen
2. puk_idp_enc - öffentlicher Schlüssel zum Verschlüsseln von für den IDP-Dienst bestimmten Daten
3. puk_idp_sig_sek - öffentlicher Schlüssel des IDP-Dienst zur Authentisierung gegenüber sektoralen Identity Providern.


```
{"keys":[
{"kid":"puk_idp_sig",
 "use":"sig",
 "kty":"EC",
 "crv":"BP-256",
 "x":"pogLhoK59j_BX7OKqZWQ0GkEckCbr2IJ5HZLRLkXyn8",
 "y":"qBNddqxoOK_2Vd5ocnuQtP1q_PuRslxfAQjv4E4dReA",
 "x5c":["MIIC+jCCAqCgAwIBAgICG3wwCgYIKoZIzj0E.......
d0zgqqbskiLAbmwbxMOjrtClRS6xK2J61BATOj20w=="]},
{"kid":"puk_idp_enc",
 "use":"enc",
 "kty":"EC",
 "crv":"BP-256",
 "x":"pkU8LlTZsoGTloO7yjIkV626aGtwpelJ2Wrx7fZtOTo",
 "y":"VliGWQLNtyGuQFs9nXbWdE9O9PFtxb42miy4yaCkCi8"},
{"kid":"puk_idp_sig_sek",
 "use":"sig",
 "kty":"EC",
 "alg":"ES256",
 "crv":"P-256",
 "x":"sk8Cig9IjJqATxrJkWRdw2gJ7Qut7ygToC8o3z2C_IU",
 "y":"LGXTzotnGJuMThRp0QWa2HldCfNoxbMh-PownRgAKko"}]}
```


Prüfung IDP-Dienst Zertifikate

OCSP-Prüfung

Im Discovery-Dokument des IDP-Dienstes ist im *claim* "uri_puk_idp_sig" ein JWK Objekt hinterlegt. Der Parameter "x5c" enthält das verwendete Signer-Zertifikat als Base64 ASN.1 DER-Encoding. Hier kommt ausnahmsweise NICHT URL-safes Base64-Encoding zum Einsatz!].

So liefert der Aufruf von https://idp-ref.app.ti-dienste.de/certs/puk_idp_sig für die IDP-Dienst-Referenzumgebung:

```
{"kid":"puk_idp_sig",
"use":"sig",
"kty":"EC",
"crv":"BP-256",
"x":"pogLhoK59j_BX7OKqZWQ0GkEckCbr2IJ5HZLRLkXyn8",
"y":"qBNddqxoOK_2Vd5ocnuQtP1q_PuRslxfAQjv4E4dReA",
"x5c":["MIIC+jCCAqCgAwIBAgICG3wwCgYIKoZIzj0 ... lRS6xK2J61BATOj20w=="]}
```

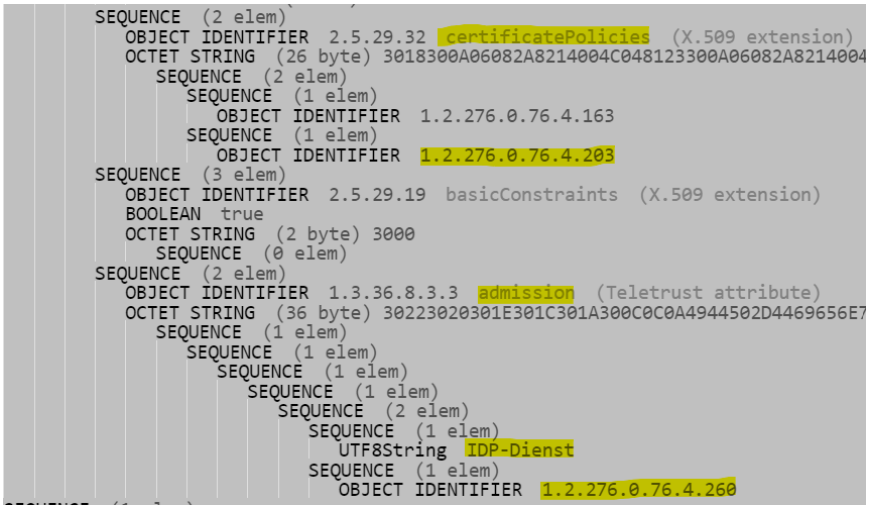

Das Zertifikat sollte stündlich über OCSP geprüft werden. Es stehen sowohl im Internet wie auch aus der TI erreichbare OCSP-Responder der Komponenten-PKI zur Verfügung.

Prüfung Zertifikatsinhalte

Im Discovery-Dokument des IDP-Dienstes ist im *claim* "id_token_signing_alg_values_supported" festgelegt, welche Algorithmen vom IDP-Dienst unterstützt werden. Für den IDP-Dienst in der Referenzumgebung wird z. B. nur der Algorithmus *BP256R1* für die Signatur der ID-Token unterstützt. ("id_token_signing_alg_values_supported": ["BP256R1 "]). Das Signaturzertifikat für die Signatur der ID-Token ist von der Komponenten PKI der gematik ausgestellt.

Zur Prüfung der Zertifikate muss "admission" und "certificatePolicies" im Zertifikat analysiert werden, um Zertifikatstyp und technische Rolle zu ermitteln. Die Auflösung des ObjectIdentifier erfolgt anhand der Spezifikation Festlegung von OIDs.

	ObjectIdentifier	Bedeutung	Auflösung des ObjectIdentifier
admission	1.2.276.0.76.4.260	technische Rolle bzw. ProfessionOID	"IDP-Dienst"
certificatePolicies	1.2.276.0.76.4.203	Zertifikatstyp bzw. Zertifikatstyp-OID	"C.FD.SIG"



Authorization Request

Der Endpunkt für Authorization-Request gegen den IDP-Dienst ist die URL welche im Discovery Document unter dem Parameter "authorization_endpoint" angegeben ist. Die Transportverschlüsselung (TLS) ist mit normalen Serverzertifikaten für diese URL abgesichert.

Request Parameter	Wert / Beispiel	Beschreibung
scope	openid+e-rezept	Der *scope* entspricht dem zwischen Fachdienst und IDP-Dienst festgelegten Wert. Der *scope* ist im Response auf einen Token-Request als *claim* im vom IDP-Dienst ausgestellten Access-Token enthalten.
response_type	code	Referenziert den erwarteten Response-Type des Flows. Der Wert ist immer *code*. Damit wird angezeigt, dass es sich hierbei um einen Authorization Code Flow handelt.
redirect_uri	https://redirect.gematik.de/erezept	Die URL wird vom Fachdienst beim Registrierungsprozess im IDP-Dienst hinterlegt und leitet die Antwort des Servers an diese Adresse um.
state	AcYxMQ5MZMpRh6WOBjs8	Der *state* - Parameter wird vom Client zufällig generiert, um CSRF zu verhindern. Indem der Server mit diesem Wert antwortet, werden Redirects legitimiert. *state* kann eine maximale Länge von 512 Zeichen haben.

code_challenge_method	S256	Der Fachdienst generiert einen Code-Verifier und erzeugt darüber einen Hash im Verfahren SHA-256, hier abgekürzt als S256. Das Ergebnis ist die *code_challenge*. *code_challenge_method* und *code_challenge* sind Bestandteile des PKCE-Flow.
nonce	nN4LkW1moAwg1tofYZtf	String zur Verhinderung von CSRF-Attacken. Dieser Wert ist optional. Wenn er mitgegeben wird, muss der gleiche Wert als *claim* im abschließend ausgegebenen ID-Token gesetzt sein. *nonce* kann eine maximale Länge von 512 Zeichen haben.
client_id	eRezeptApp	Die Client-ID des Fachdienstes wird bei Registrierung des Fachdienstes beim IDP-Dienst durch den Anbieter des IDP-Dienstes festgelegt.
code_challenge	SU8xsVcUypYGUi2g-mzs7rvR2lMtQ9vyj_9Hxs0WcII	Der mit der unter *code_challenge_method* angegebenen Methode (S256) erzeugte Hashwert des Code-Verifier wird zum IDP-Dienst als *Code-Challenge* gesendet. *code_challenge_method* und *code_challenge* sind Bestandteile des PKCE-Flow.

Token

Token-Request

Der Endpunkt für den Token-Request gegen den IDP-Dienst ist die URL, welche im Discovery Document unter dem Parameter "token_endpoint" angegeben ist (z. B. : https://idp-ref.app.ti-dienste.de/token). Die Transportverschlüsselung (TLS) ist mit normalem Serverzertifikaten für diese URL abgesichert.

Request Parameter	Wert / Beispiel	Beschreibung
client_id	eRezeptApp	Die Client-ID des Fachdienstes wird bei Registrierung des Fachdienstes beim IDP-Dienst durch den Anbieter des IDP-Dienstes festgelegt.
code	code	Der vom IDP-Dienst ausgestellte Authorization-Code
grant_type	authorization_code	Der Grant-Type ist immer *authorization_code*
key_verifier	{ "token_key": "T0hHOHNKOTFaREcxTmN0dVRKSURraTZxNEpheGxaUEs", "code_verifier": "W91A37hQ8oeDRVpnkYgpYthjl4LqYy95A87ISy9zpUM" }	Der Parameter key_verifier wird mit dem Verschlüsselungsschlüssel "puk_idp_enc" verschlüsselt. Der verschlüsselter Key-Verifier enthält • token_key symmetrischer Schlüssel (AES256) zur Verschlüsselung des ID-Token durch den IDP-Dienst und Entschlüsselung des ID-Token durch den Fachdienst • code_verifier vom Fachdienst vor dem Authorization-Request generiert
redirect_uri	https://redirect.gematik.de/erezept	Die URL wird vom Fachdienst beim Registrierungsprozess im IDP-Dienst hinterlegt und leitet die Antwort des Servers an diese Adresse um.

Token-Response

Folgende *claims* sind immer oder Kontext abhängig Bestandteil des Token:

claim	Wert / Beispiel	Token	Beschreibung	Kontext
at_hash	5AZmDxrYImUa6-kjMNAL3g	ID-Token	Erste 16 Bytes des Hash des Authentication Token Base64(subarray(Sha256(authentication_token), 0, 16)	Claim ist immer im Token enthalten
sub	ez4D403gBzH1IhnYOXA4aUU-7spqPbWUyUELPoA79CM	ID-Token / Access-Token	subject. Base64(sha256(audClaim + idNummerClaim + serverSubjectSalt)	Claim ist immer im Token enthalten
professionOID	1.2.276.0.76.4.49	ID-Token / Access-Token	professionOID gemäß Festlegung von OIDs	Abhängig vom durch den Fachdienst angefragten *scope* ist dieser *claim* im Token enthalten.
idNummer	X114428530	ID-Token / Access-Token	unveränderlicher Teil der KV-Nummer bei Versicherten Telematik-ID bei Leistungserbringern und SMC-B	Abhängig vom durch den Fachdienst angefragten *scope* ist dieser *claim* im Token enthalten.

amr	["mfa", "sc", "pin"]	ID-Token / Access-Token	Gemäß des übertragenen Werts des Authenticator-Moduls in der Datenstruktur "Signed_Authentication_Data"	Claim ist immer im Token enthalten
iss	https://idp-ref.app.ti-dienste.de	ID-Token / Access-Token	URL, unter welchem der IDP-Dienst erreichbar ist	Claim ist immer im Token enthalten
aud	https://erp-ref.zentral.erp.splitdns.ti-dienste.de/	Access-Token	beim IDP-Dienst registrierter Identifier der Fachanwendung	Claim ist immer im Access-Token enthalten
	eRezeptApp	ID-Token	beim IDP-Dienst registrierte client_id des anfragenden Client	Claim ist immer im ID-Token enthalten
acr	gematik-ehealth-loa-high	ID-Token / Access-Token	Authetisiertes Vertrauensniveau	Claim ist immer im Token enthalten
azp	eRezeptApp	ID-Token / Access-Token	beim IDP-Dienst registrierte client_id des anfragenden Client	Claim ist immer im Token enthalten
auth_time	1618243993	ID-Token / Access-Token	Timestamp der Authentisierung	Claim ist immer im Token enthalten
scope	"openid e-rezept"	Access-Token	Liste der angefragten scopes (i. d. R. openid + bei Registrierung des Fachdienstes vergebener Identifier)	Claim ist immer im Token enthalten
exp	1618244294	ID-Token / Access-Token	Zeitpunkt des Gültigkeitsende des Token.	Claim ist immer im Token enthalten
iat	1618243994	ID-Token / Access-Token	Zeitpunkt der Ausstellung des Token.	Claim ist immer im Token enthalten
organizationName	AOK Plus	ID-Token / Access-Token	Herausgeber-ID bei Versicherten Identitäten Organisationsbezeichnung bei SMC-B	Abhängig vom durch den Fachdienst angefragten *scope* ist dieser *claim* im Token enthalten.
given_name	Max	ID-Token / Access-Token	Vorname bei Versichterten und Leistungserbringern Vorname des Verantwortlichen/Inhabers bei SMC-B	Abhängig vom durch den Fachdienst angefragten *scope* ist dieser *claim* im Token enthalten.
family_name	Mustermann	ID-Token / Access-Token	Nachname bei Versichterten und Leistungserbringern Nachname des Verantwortlichen/Inhabers bei SMC-B	Abhängig vom durch den Fachdienst angefragten *scope* ist dieser *claim* im Token enthalten.
nonce	nN4LkW1moAwg1tofYZtf	ID-Token	Beliebig vom Fachdienst generierter Wert zum Schutz vor Angriffen.	Die *nonce* kann im Authorization Request des Fachdienstes an den IDP-Dienst als Parameter mitgegeben werden. Dieser *claim* darf nur dann im ID-Token an den Fachdienst zurück übermittelt werden, wenn er im Authorization Request des Fachdienstes an den IDP-Dienst als Parameter tatsächlich mitgegeben wurde. Dann jedoch müssen die Werte exakt übereinstimmen.
jti	c1c760ca67fe1306	ID-Token / Access-Token	Eindeutiger Token-Bezeichner, verhindert die Wiederverwendung des Tokens.	Claim ist immer im Token enthalten

Prüfung der erhaltenen ID-Token

Fachdienste nutzen am besten nur die ID-Token des IDP-Dienstes. Access-Token werden für das E-Rezept verwendet, hier ist der IDP-Dienst der Authorization-Server.

- Entschlüsselung der Token mit dem vorgesehenen *token_Key*

  Der Fachdienst erzeugt einen symmetrischen Schlüssel, den er im Token-Request im Parameter *key_verifier* als *token_key* an den IDP-Dienst übermittelt. Das abgerufene ID-Token muss mit diesem Schlüssel verschlüsselt sein. Der Fachdienst muss bei Erhalt des ID-Token vom IDP-Dienst den ID-Token mit dem *token_key* entschlüsseln können.

- Prüfung der zeitlichen Gültigkeit des Token

  Zur Prüfung der Gültigkeit des ID-Token müssen die *claims iat* und *exp* aus dem ID-Token ausgewertet werden. Dabei ist *iat* der Ausstellungszeitpunkt und *exp* der Ablaufzeitpunkt des ID-Token. Beide *claims* haben das Format [Sekunden seit 1970, RFC 7519 Sect.2] . Der Zeitraum der Verwendung des Tokens muss zwischen den Werten der Attribute *iat* und *exp* liegen.

- Prüfung der Signatur des Token
  Um die Signatur der Token zu prüfen muss diese mathematisch mittels des "puk_idp_sig" aus dem jwks-Schlüsselsatz des IDP_Dienstes verifziert werden können. Das dort im x5c-Claim enthaltene Zertifikat ist nach Möglichkeit mindestens Täglich mittels OCSP auf seine Gültigkeit zu prüfen.

- Prüfung des audience-Parameters

  Der im Token mitgelieferte *claim aud* muss mit der beim IDP-Dienst registrierten client_id übereinstimmen.

- Prüfung des nonce-Parameters

  Optional kann der Fachdienst zur Absicherung gegen Attacken ein *nonce* generieren und diese im Authorization-Request an den IDP-Dienst mit übermitteln. In diesem Fall muss der ID-Token den *claim nonce* enthalten. Der Wert von *nonce* muss exakt dem vom Fachdienst übermittelten Wert entsprechen. Gibt der Fachdienst im Authorization-Request an den IDP-Dienst keinen Parameter nonce mit, so darf das vom IDP-Dienst ausgestellte ID-Token den *claim nonce nicht* enthalten.