Login - Übersicht

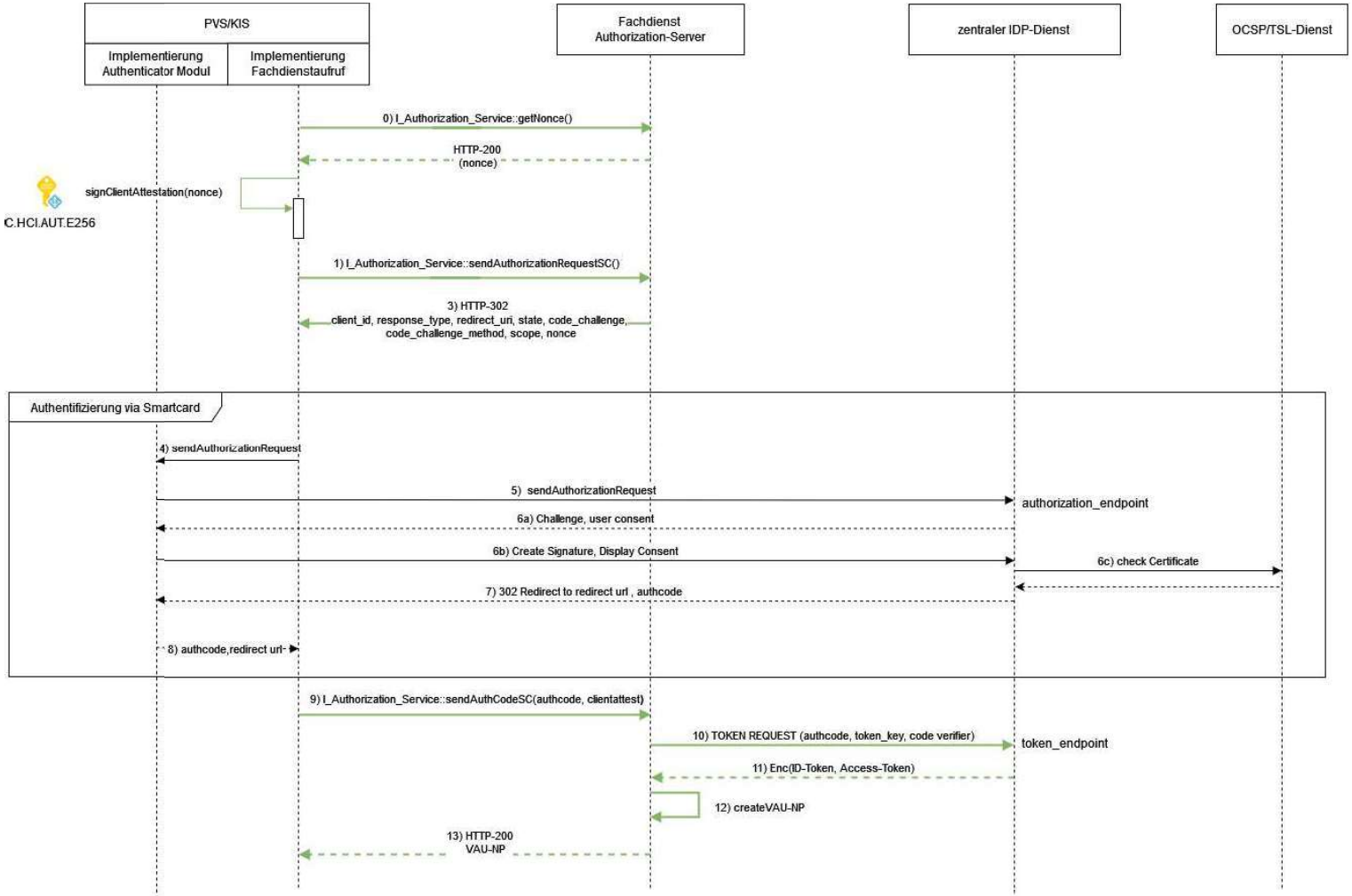
- Lokalisierung der Service-Endpunkte
- VAU Kanal Aufbau
- Authentisierung und Nutzung des IDP-Dienstes

➔ User Session ist aufgebaut

Lokalisierung der Service-Endpunkte der ePA

Change C_11843

Authentisierung - Überblick



Authentisierung - Erfahrungsbericht
Wie sind wir bei der Implementierung des Login Flows vorgegangen?

- In unserer PS Simulation (Mock) wurde der Login eines LEI auf Basis des OIDC Flows -umgesetzt (wie in der Spec beschrieben)
- Wie auch im Sequenzdiagramm aus der Spec zu sehen ist, erfolgt eine Kommunikation mit IDP und Authorization Server
- Der Mock für den Authorization Server wurde von unserem IDM Team umgesetzt (https://github.com/gematik/app-asforepa)
- Zusätzlich haben wir vom IDM Team eine intern entwickelte Client Library benutzt, welches die Kommunikation mit dem IDP kapselt
- In der PS-SIM wurde aus der API Spec (auf openAPI Basis: https://github.com/gematik/ePA-Basic/blob/ePA-3.0.1/src/openapi/I_Authorization_Service.yaml) für den Authorization Server der Client generiert. Hier gibt es die Möglichkeit für verschiedene Programmiersprachen und Frameworks Clients generieren zu lassen. Mit dem "API first" Ansatz spart man viel Zeit und entwickelt damit direkt gegen den Contract aus der Spezifikation

Authentisierung - Erfahrungsbericht
Details zum Flow

- **nonce** holen vom AuthorizationServer → getNonce
- initialisieren AuthorizationServer → sendAuthzRequestSc
  - response = 302
  - redirectUrl in Location Header
  - Extrahieren von **scope, redirect_uri, code_challenge, client_id und state**
- **X509 certificate** vom Konnektor holen (AUT Zertifikat vom Typ ECC)
- External Authenticate am Konnektor zur Signierung (Typ urn:bsi:tr:03111:ecdsa)
  - der zu signierende Inhalt ist Base64 encodiert
- Aufruf des Logins am IDP anhand der zur Verfügung stehenden Client Library (Kommunikation mit dem IDP)
  - alle Parameter von oben plus
    - certificate
    - external_authenticate
    - code_challenge
    - state
    - nonce
  - Im Erfolgsfall ist die Response vom IDP ist ein **authorization_code**

Authentisierung - Erfahrungsbericht
Details zum Flow

- **clientAttest** erstellen
  - JWT/JWS
  - header.payload.signature
- header = typ, x5c, alg
- payload = claims wie die nonce
- signature
  - per external authenticate signiert
  - JWS
- JWT is payload of a JWS
- header (base64UrlEncoded).payload (base64UrlEncoded).signature
  (signing mit external authenticate (header '.' payload))
- Schicke **clientAttest** und **authorization_code** an AuthorizationServer → sendAuthCodeSC
  - AuthorizationServer validiert u.a. den clientAttest und
    schickt im Erfolgsfall Statuscode 200 und Nutzerpseudonym (VAU-NP) zurück
    → für die Nutzung im erneuten VAU Handshake in Nachricht 1