Deputy Ministry of Research Innovation and Digital Policy

Cyprus National eID Framework

AUTHENTICATION AND REMOTE SIGNATURE PROTOCOL

Department of Information and Technology Services

Version: [1.1]

Status: [Final]

Date: [2023-05-15]

Table of Contents

1 Scope

The present document specifies protocols and interfaces that are intended for eIDAS-compliant high level of assurance authentication and eIDAS-compliant remote qualified signatures for the Cyprus National eID scheme. Specifically, between a Service Provider and a QTSP.

Existing standards and open specifications are considered as far as applicable.

Using open standards increases interoperability, compatibility with a range of stakeholders and avoid vendor lock-in.

By implementing their services according to these specifications, QTSP's can ensure plug and play compatibility in the Cyprus National eID framework.

2 Interpretation of Requirement Levels

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

3 References

[1] RFC 2119 - https://www.ietf.org/rfc/rfc2119.txt

[2] OpenID Connect Core 1.0 - https://openid.net/specs/openid-connect-core-1_0.html

[3] OAuth 2.0 - https://tools.ietf.org/html/rfc6749

[4] Cloud Signature Consortium (CSC) – Architectures and protocols for remote signature applications https:/cloudsignatureconsortium.org/wp-content/uploads/2019/07/CSC_API_V1_1.0.4.0.pdf

[5] ETSI TS 119 432 - https://www.etsi.org/deliver/etsi_ts/119400_119499/119432/01.01.01_60/ts_119432v010101p.pdf

[6] SCAL2 - CEN EN 419 241-1 - Trustworthy Systems Supporting Server Signing - Part 1: General System Security Requirements

[7] OAuth 2.0 Security Best Current Practice - https://tools.ietf.org/html/draft-ietf-oauth-security-topics-13

[8] PKCE (Proof Key for Code Exchange) - https://tools.ietf.org/html/rfc7636

[9] OAuth 2.0 for Native Apps - https://tools.ietf.org/html/rfc8252

4 Terms, Definitions and Abbreviations

4.1 Terms and Definitions

Access Token: The Access Tokens are used in token-based authentication to allow an application to access an API.

Identity Broker: An Identity Broker is often part of a Single Sign-On Architecture as an intermediary service that connects multiple Service Providers with different Identity Provider (IDP)s. An Identity Broker maps Identity Attributes, including unique identifiers, across multiple Identity Provider (IDP) to one user entity.

Identity Provider: Authenticates users and provides to Service Providers an Authentication Assertion if successful.

Id Token: The ID Token is a security token that contains Claims about the Authentication of an End-User by an Identity Provider, and potentially other requested Claims. The ID Token is represented as a Json Web Token (JWT).

Json Web Token: JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

Refresh Token: The Refresh Token is a long-lived token that is used to obtain a new Access Token after a previous one has expired.

Service Provider: Web Application or Mobile/Native Applications or Single Page Applications that relies on the Identity Provider in order to authenticate users and request claims.

Signature Activation Data: Set of data used to control a given signature operation, performed by a cryptographic module, on behalf of the signer.

Signature Activation Module: Configured software that uses the Signature Activation Data in order that the signing keys are used under sole control of the signer.

NOTE: As defined in CEN EN 419 241-1 [6]

Signing Broker: A Central service acting as an intermediary signing application.

Single Sign On: Single sign-on (SSO) is an authentication process that allows a user to access multiple applications by logging in only once with one set of login credentials. With SSO, the application or website that the user is trying to access relies on a trusted third party to verify that users are who they say they are.

4.2 Abbreviations

API: Application Programming Interface

CSC: Cloud Signature Consortium

IdP: Identity Provider

OCSP: Online Certificate Status Protocol

PKCE: Proof Key for Code Exchange

QTSP: Qualified Trust Service Provider

SAD: Signature Activation Data

SAM: Signature Activation Module

SCAL 2: Sole Control Assurance Level 2

NOTE: As defined in CEN EN 419 241-1 [6]

SSO: Single Sign On

TLS: Transport Layer Security

5 System Requirements

5.1 General Requirements

The authentication and authorization protocols that are applicable in the Cyprus eID framework SHALL be OpenID Connect 1.0 [2] and OAuth 2.0 [3].

QTSP MUST implement an Identity Provider (IdP) conforming to the above standards.

QTSP MUST implement a signing service using one of the following remote signature protocols:

- Cloud Signature Consortium (CSC) - Architectures and protocols for remote signature applications [4].

  The following API's of the protocol SHALL be implemented:

  - info
  - credentials/list
  - credentials/info
  - credentials/authorize
  - signatures/signHash

- ETSI TS 119 432 - Electronic Signatures and Infrastructures (ESI); Protocols for remote digital signature creation [5]

  The CSC JSON binding of the protocol SHALL be implemented.

  The following API's of the protocol SHALL be implemented:

  - info
  - credentials/list
  - credentials/info
  - signatures/signHash

Regardless of the implemented remote signature protocol the access to the APIs SHALL be authenticated using a valid "access token" from the QTSP IdP (CSC "service" scope equivalent).

For the purpose of the National eID Framework the certificates "authMode" MUST be "Implicit" (credentials/info API).

Accessing a credential for remote signing SHALL require an authorization from the user using SCAL2 (sole control assurance level 2) as defined in CEN EN 419 241-1 [6].

The QTSP MUST use the Online Certificate Status Protocol (OCSP) for Certificate Status validation.

5.2 Authentication / Authorization

Having in mind the latest OAuth 2.0 Security Best Current Practice [7], the following scenarios have been identified:

- Authorization Code Grant + PKCE [8]
- Client Credentials Grant [3 section 4.4]

5.2.1 Authorization Code Grant + PKCE

User Authentication (id token) for Web Application, Mobile/Native Applications [9], Single Page Applications, and access token for the signing service for the specific user and optionally refresh token.

The QTSP IdP SHALL sign an authentication challenge that will be part of the authentication request using the user certificate.

The QTSP IdP MUST request credential authorization from the user using SCAL2 (sole control assurance level 2) as defined in CEN EN 419 241-1 [6].

The method of Signature Activation Data (SAD) verification by the Signature Activation Module (SAM) is up to QTSP.

The following requirement are mandatory:

- The PKCE code challenge SHALL BE SHA-256.
- The PKCE code challenge MAY be used as document to be signed (authentication challenge for signing).
- The Token endpoint MUST support private_key_jwt authentication.
- The following claims in the ID Token MUST be returned for the citizen
    - sub – UserID. The identifier associated to the identity of the credential owner.
    - given_name – First Name
    - family_name – Last Name
    - unique_identifier – Id number
    - birthdate – Date of birth
    - approximate_age – Calculated based on birthdate. Used in case the citizen doesn't want to disclose his/her actual birthdate
    - email – Email of the user
    - acr – Level Of Assurance as specified in section 5.2.3
    - requestID – For audit and forensic purposes the IdP must generate a unique request ID that can track the user interaction

5.2.2 Client Credentials Grant

This will be used to authorize applications rather than a user to get access tokens token to call the signing service. The access token will not have information about a user.

5.2.3 Levels of Assurance (LoA) / ACR

Authentication requests MUST identify the requested Level of Assurance (LoA) for end-user authentication using the acr_values (Authentication Context Class Reference) parameter.

The ID token response MUST specify the actual acr value used to perform authentication. If the IdP can't fulfil any of the requested Authentication Context Class References, it MUST return an error.

The allowed ACR values are the below:

ACR value	Description
http://eidas.europa.eu/LoA/high	eIDAS High Level of assurance

5.2.4 Authentication Request

The QTSP IdP SHALL accept only signed Authentication Requests [2 section 6] by trusted clients. The QTSP IdP that receives Authentication Requests that are not signed, or where the verification of the signature fails, the Identity Provider MUST respond with an error.

5.2.5 Single Sign On

The QTSP IdP SHALL NOT support Single Sign-On (SSO). An authentication SHALL be REQUIRED from the end user on each Authentication Request.

5.3 Security

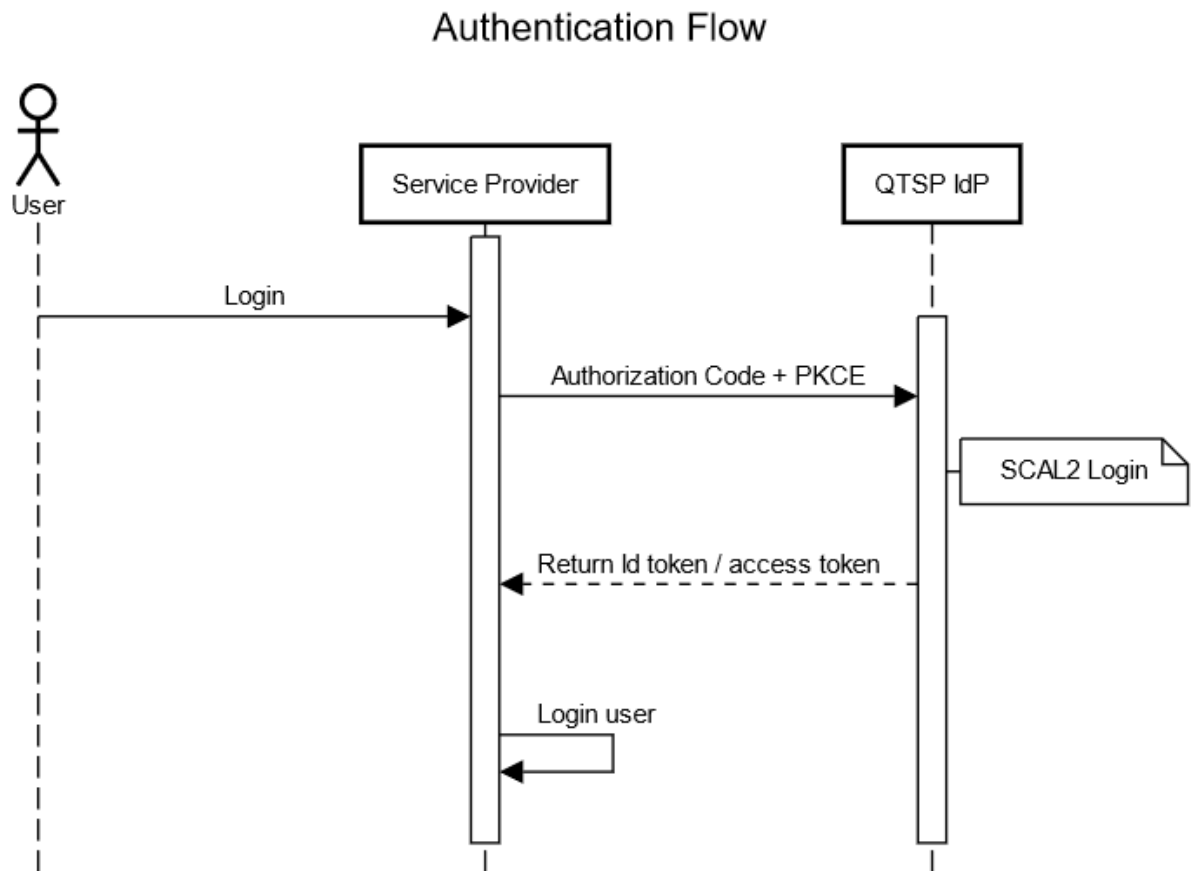
5.3.1 TLS

The provided QTSP endpoints for authentication and signing services MUST be protected by TLS 1.2 or greater.

6 Use Cases

The sample requests and responses that are provided in the diagrams are only a partial representation of complete transactions and are aimed at showing the most important parameters and information.

6.1 Authorization Code + PKCE



Authentication Flow Description

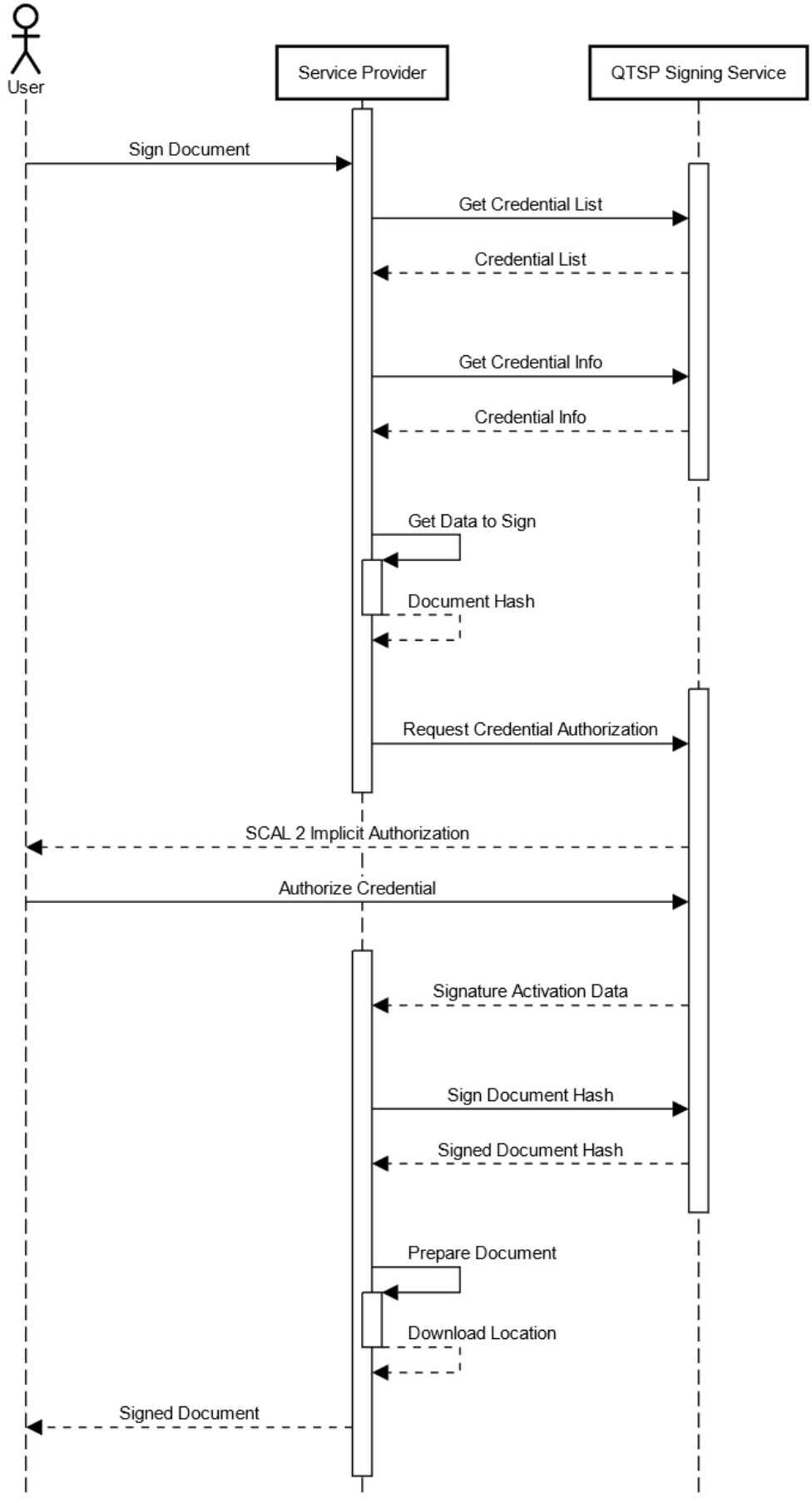
1. User needs to login to Service Provider.
2. The Service Provider submits an Authentication Request to QTSP IdP implementing OpenID Authorization Code + PKCE flow.
3. User authenticates based on QTSP IdP authentication mechanism.

   The QTSP IdP MUST request credential authorization from the user using SCAL2 (sole control assurance level 2) as defined in CEN EN 419 241-1.

   The method of Signature Activation Data (SAD) verification by the Signature Activation Module (SAM) is up to QTSP.

4. QTSP IdP returns an Id Token and Access Token and optionally a refresh token to Service Provider.
5. The Service Provider validates the Id Token.
6. The Service Provider logins the user.

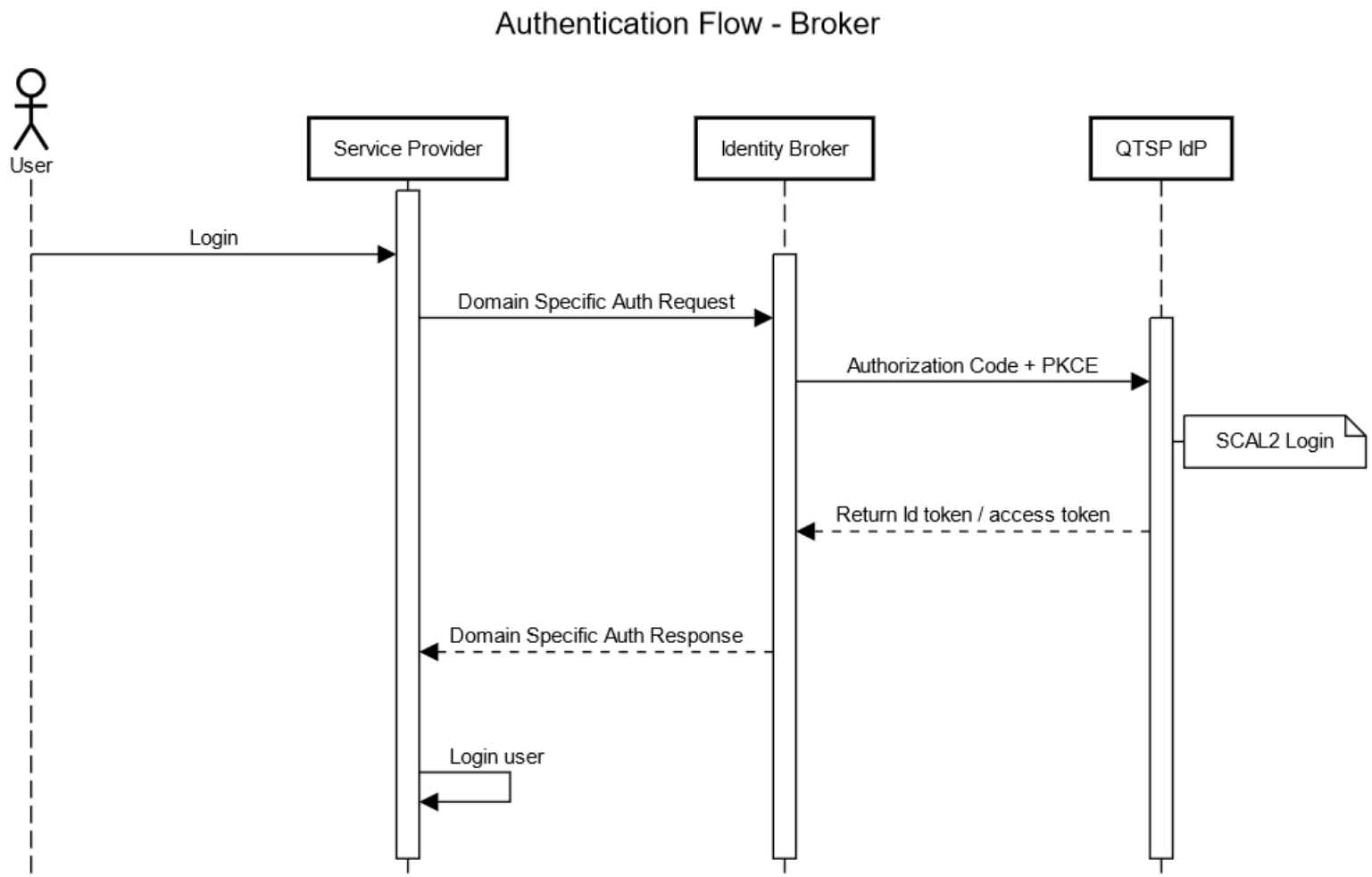
Signing Flow



Signing Flow Description

1. The User needs to sign a document.
2. The Service Provider must request authorization to use the user signing credentials to sign the specific document. The access token acquired during authentication must be used for service authorization.
3. User Authorizes Access.
4. The QTSP Signing Service returns the SAD for the specific document.
5. The Service Provider calls the API for signing the document hash providing the SAD for proof of authorization. The access token acquired during authentication must be used for service authorization.
6. The QTSP Signing Service signs the document hash and returns the signature.

6.2 Client Credential



Authentication Flow Description

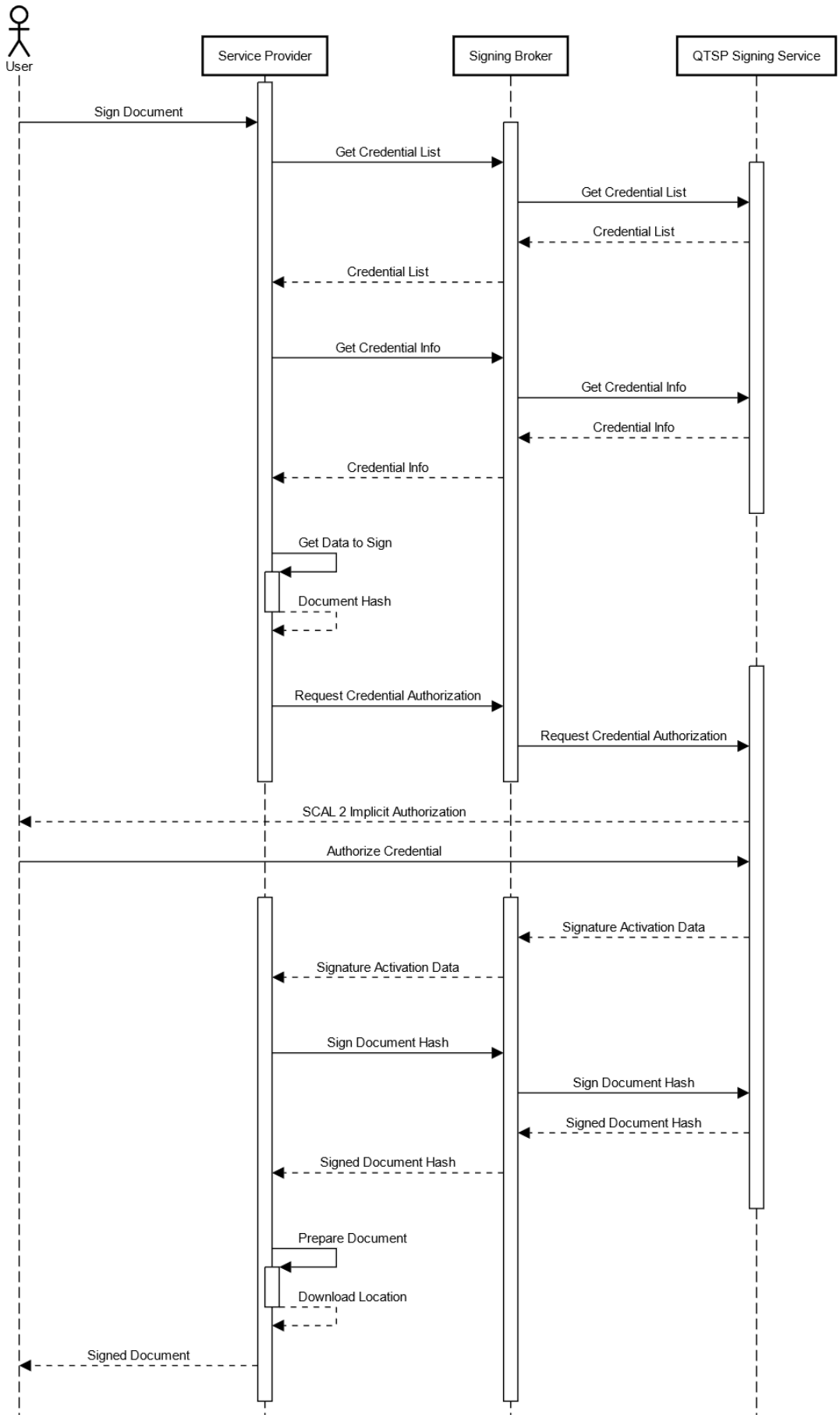
1. User needs to login to Service Provider
2. Service Provider forwards the request to Identity Broker. The authentication request is domain specific.
3. The Identity Broker submits an Authentication Request to QTSP IdP implementing OpenID Authorization Code + PKCE flow.
4. User authenticates based on QTSP IdP authentication mechanism.

   The QTSP IdP MUST request credential authorization from the user using SCAL2 (sole control assurance level 2) as defined in CEN EN 419 241-1.

   The method of Signature Activation Data (SAD) verification by the Signature Activation Module (SAM) is up to QTSP.

5. QTSP IdP returns an Id Token and Access Token to Identity Broker.
6. The Identity Broker validates the Id Token.
7. The Identity Broker returns to Service Provider domain authentication response.
8. Service Provider login the user.

Signing Flow - Broker



Signing Flow Description

1. The User needs to sign a document.
2. The Service Provider requests from Signing Broker to sign the document / document hash.
3. The Signing Broker request Access Token from QTSP Signing Service using client credentials flow.
4. The QTSP Signing service return access token to Signing Broker.
5. The Signing Broker requests authorization to use the user signing credentials to sign the specific document. The access token acquired at step 4 must be used for service authorization.
6. User Authorizes Access.
7. The QTSP Signing Service returns the SAD for the specific document.
8. The Signing Broker calls the API for signing the document hash providing the SAD for proof of authorization. The access token acquired during step 4 must be used for service authorization.
9. The QTSP Signing Service signs the hash document and returns the signature.
10. Signing Broker receives the signed document and forwards it to Service Provider.

7 Conformance Testing Tools

The Department of Information Technologies Services will provide reference client implementations for the Authentication and Signing scenarios that will be able to test the conformance of QTSP IdP and Signing services to this protocol.