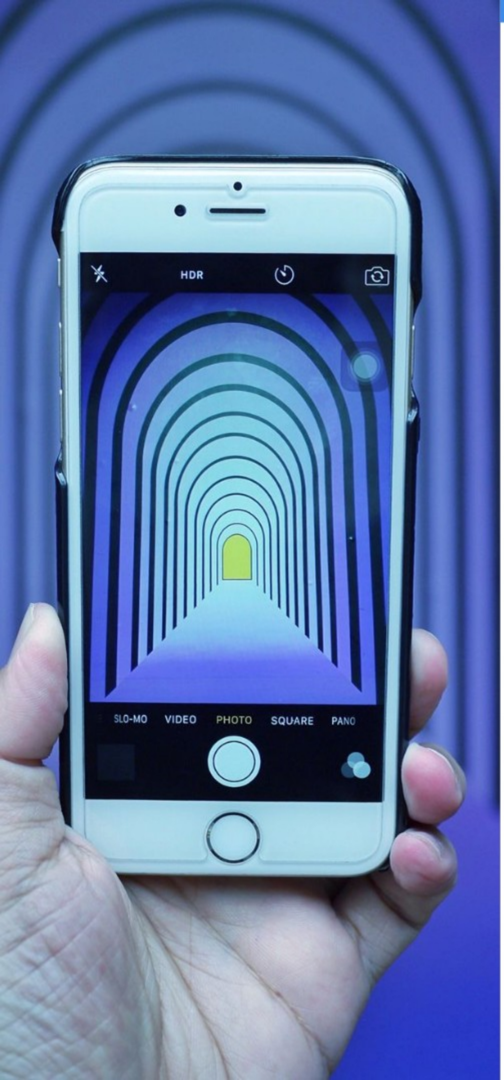


Native SSO for Mobile Apps

George Fletcher
OATH

Nat Sakimura
Nomura Research Institute

Rationale

Session stored in the system browser can be brittle.

It can be reset by customer care agent or simply expired by the platform policy.

“keychain” access is usually granted to the apps from the same developer: suitable for storing shared sessions.

3. Roles

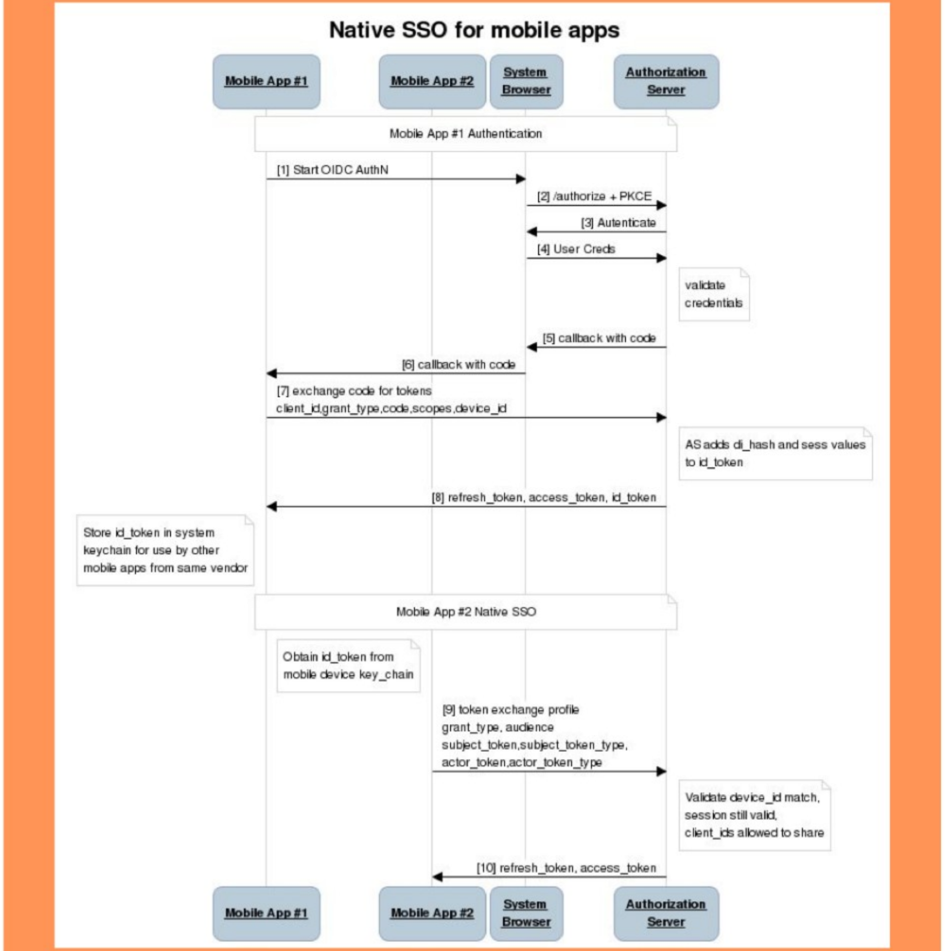
Mobile App #1

Mobile App #2

from the same developer as #1

Authorization Server (AS)

4. Abstract Flow



Native SSO for mobile apps

Mobile App #1

Mobile App #2

System Browser

Authorization Server

Mobile App #1 Authentication

[1] Start OIDC AuthN

[2] /authorize + PKCE

[3] Autenticate

[4] User Creds

validate credentials

[5] callback with code

[6] callback with code

[7] exchange code for tokens
client_id,grant_type,code,scopes,device_id

AS adds di_hash and sess values to id_token

[8] refresh_token, access_token, id_token

Store id_token in system keychain for use by other mobile apps from same vendor

Mobile App #2 Native SSO

Obtain id_token from mobile device key_chain

[9] token exchange profile
grant_type, audience
subject_token,subject_token_type,
actor_token,actor_token_type

Validate device_id match,
session still valid,
client_ids allowed to share

[10] refresh_token, access_token

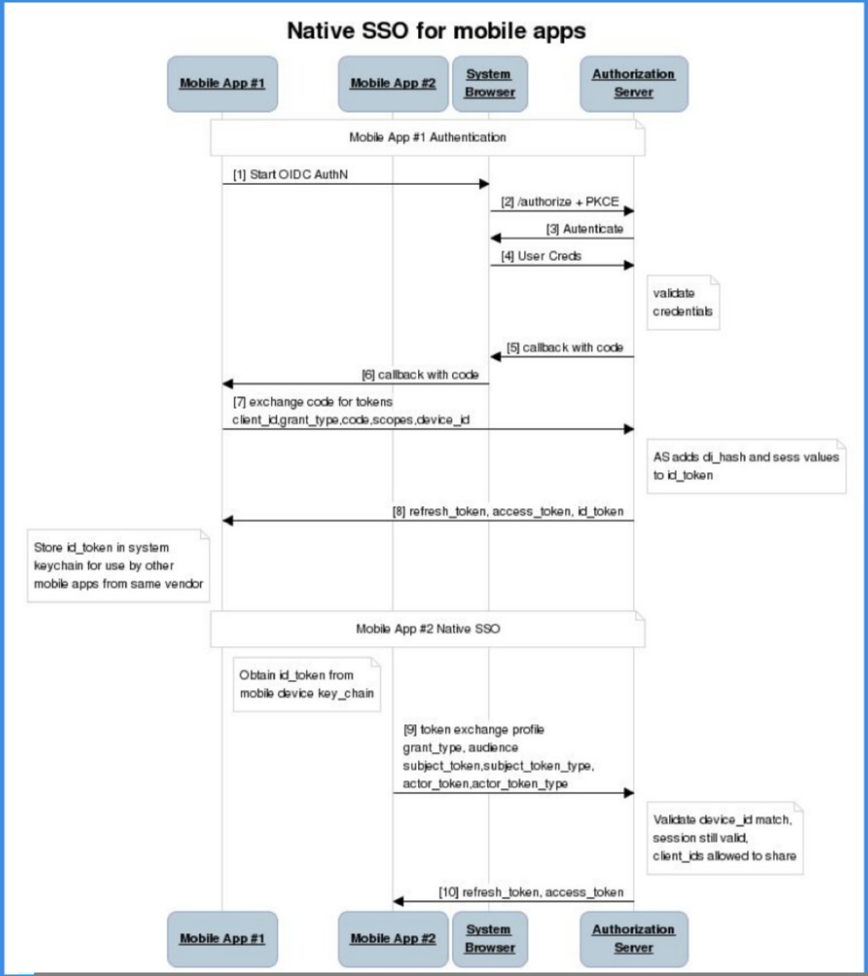
Mobile App #1

Mobile App #2

System Browser

Authorization Server

Mobile App #1



Native SSO for mobile apps

Mobile App #1 | Mobile App #2 | System Browser | Authorization Server

Mobile App #1 Authentication

[1] Start OIDC AuthN

[2] /authorize + PKCE

[3] Autenticate

[4] User Creds

validate credentials

[5] callback with code

[6] callback with code

[7] exchange code for tokens
client_id,grant_type,code,scopes,device_id

AS adds di_hash and sess values to id_token

[8] refresh_token, access_token, id_token

Store id_token in system keychain for use by other mobile apps from same vendor

Mobile App #2 Native SSO

Obtain id_token from mobile device key_chain

[9] token exchange profile
grant_type, audience
subject_token,subject_token_type,
actor_token,actor_token_type

Validate device_id match,
session still valid,
client_ids allowed to share

[10] refresh_token, access_token

[0] Mobile App #1 generates device identifier

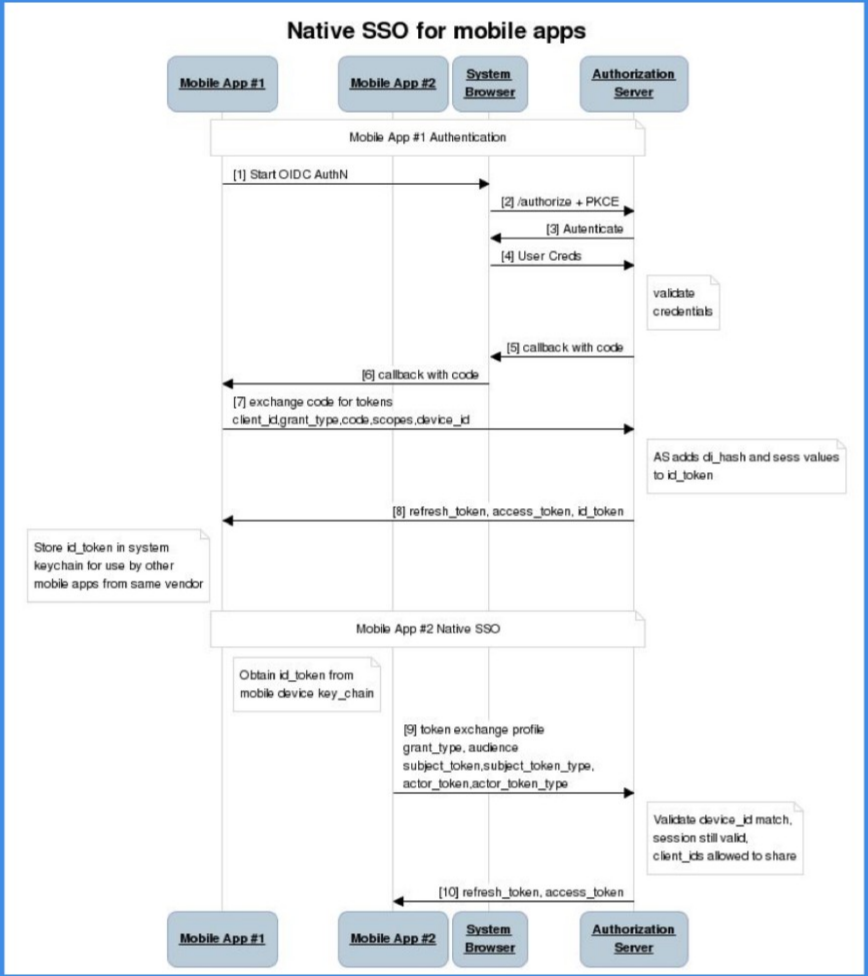
[1]-[6] A regular PKCE sequence.

[7] Sends “device_id” with the token request.

[8] id_token with “di_hash” and “sid” is returned.

App #1 stores the id_token in the system keychain.

Mobile App #2



Native SSO for mobile apps

Mobile App #1

Mobile App #2

System Browser

Authorization Server

Mobile App #1 Authentication

[1] Start OIDC AuthN

[2] /authorize + PKCE

[3] Autenticate

[4] User Creds

validate credentials

[5] callback with code

[6] callback with code

[7] exchange code for tokens
client_id,grant_type,code,scopes,device_id

AS adds di_hash and sess values to id_token

[8] refresh_token, access_token, id_token

Store id_token in system keychain for use by other mobile apps from same vendor

Mobile App #2 Native SSO

Obtain id_token from mobile device key_chain

[9] token exchange profile
grant_type, audience
subject_token,subject_token_type,
actor_token,actor_token_type

Validate device_id match,
session still valid,
client_ids allowed to share

[10] refresh_token, access_token

Mobile App #1

Mobile App #2

System Browser

Authorization Server

Mobile App #2 obtains “id_token” from the system keychain

[9] Sends “id_token” as “subject_token” in Token Exchange Profile.

AS validates “device_id” match, session still valid, “client_id” is allowed to share.

[10] Access Token (+ Refresh Token) returned to the App #2

6.1 OAuth token exchange profile

- grant_type

  REQUIRED. The value MUST be urn:ietf:params:oauth:grant-type:token-exchange

- audience

  REQUIRED. This parameter defines the logical purview of the returnedtokens. For the purposes of this profile, this value MUST be the issuer URI for the OpenID Provider that issued the id_token used in this profile.

- subject_token

  REQUIRED. This parameter MUST contain the id_token obtained bythe first mobile app.

- subject_token_type

  REQUIRED. This parameter MUST contain the value:urn:ietf:params:oauth:token-type:id_token

- actor_token_type

  - REQUIRED. This value MUST be:urn:x-oath:params:oauth:token-type:device-id

- scope

  OPTIONAL. The scopes required by the requesting mobile application

6.2 Token Exchange request for Native SSO

```
POST /token HTTP/1.1
Host: as.example.com
Authorization: Basic ZGZhZGYyMzUyNDU0Og...

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-exchange
&audience=https%3A%3F%3Flogin.aol.com&subject_token=<id_token>
&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Aid-token
&actor_token=95twdf3w4y6wvftw35634t
&actor_token_type=urn%3Ax-oath%3Aparams%3Aoauth%3Atoken-type%3Adevice-id
```

6.3 Native SSO Processing Rules

1 Validate the device_id from the actor_token against the ‘di_hash’ claim in the id_token. If the calculated left-side hash of the device_id does not match the ‘di_hash’ claim in the id_token the AS must return an error of ‘invalid_grant’. See RFC 6749 Section 5.2 [RFC6749] .

2 Check that the session identifier in the id_token is still valid. The AS MUST take the ‘sid’ claim from the id_token and verify that it is still valid for the user identified by the ‘sub’ 5 claim in the id_token. If the session is no longer valid, the AS MUST return an error of ‘invalid_grant’.

3 Validate that the client requesting native SSO is authorized to do so. The AS SHOULD maintain a list of client_ids that can share user authentications. In order to make this check the AS takes the ‘aud’ claim from the id_token and the client_id from the token request and ensures that both client_ids are allowed to share user authentications.

4 The AS SHOULD verify that the scopes requested by the client in the token request (either default scopes or explicitly specified in the optional ‘scope’ parameter) do NOT require explicit user consent. If any requested scopes require explicit user consent the AS SHOULD fail the request and return an error of ‘invalid_scope’.

6.4 Profiled Token Exchange Response

access_token -- REQUIRED. This response field contains the access token issued tothe mobile client identified by the client_id sent in the Authorization header.

issued_token_type -- REQUIRED. This value of this parameter MUST be:

urn:ietf:params:oauth:token-type:access_token

token_type -- REQUIRED. The value of this parameter MUST be “bearer”

expires_in -- REQUIRED. Identifies when the access_token expires.

scope -- OPTIONAL. Follows the token exchange spec definition.

refresh_token -- REQUIRED. A refresh_token that the mobile app can use to obtain additional access_tokens when the access_token expires.

6.4 Profiled Token Exchange Response Example

```
HTTP/1.1 200 OKContent-Type: application/json;charset=UTF-8

Cache-Control: no-store

Pragma: no-cache

{“access_token”:“2YotnFZFEjr1zCsicMWpAA”,

“Issued_token_type”“urn:ietf:params:oauth:token-type:access_token”,

“token_type”:“bearer”,

“expires_in”:3600,

“refresh_token”:“tGzv3JOkF0XG5Qx2TlKWIA”

}
```

7. Conclusion

- **Brittle sessions in System Browser**

  Risk of session deletion in the system browser is real and degrades the user experience.

- **Win-win**

  users can enjoy better user experience and the vendors having less support cost,

- **Security Analysis needed**

  the method introduces a very different flow to [RFC6749] and the [RFC8252].As a result, a separate through security analysis is sought.