OAuth Browser-less App2App

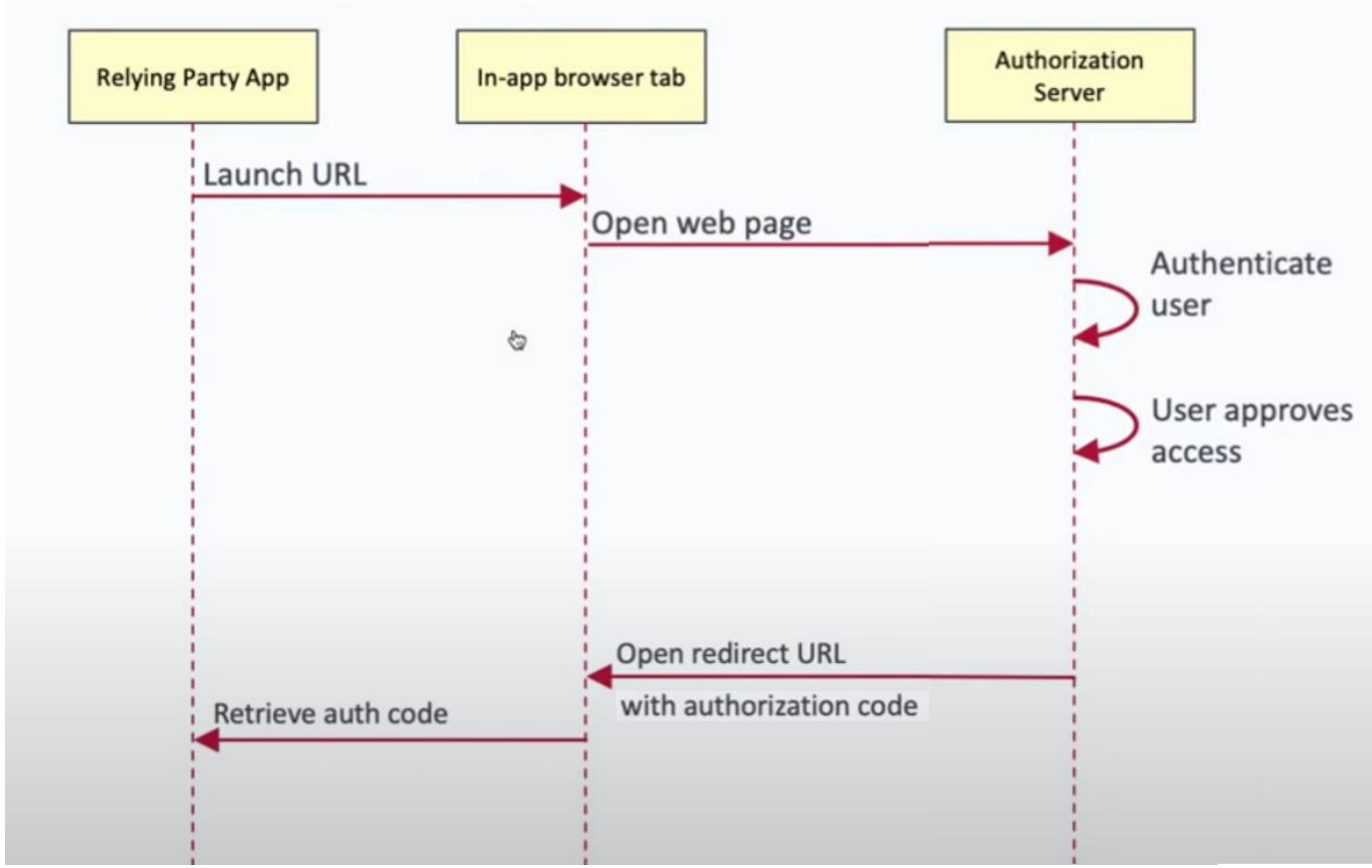
https://datatracker.ietf.org/doc/draft-zehavi-oauth-app2app-browserless/

IETF 124: OAuth WG
Nov 2025 Montreal

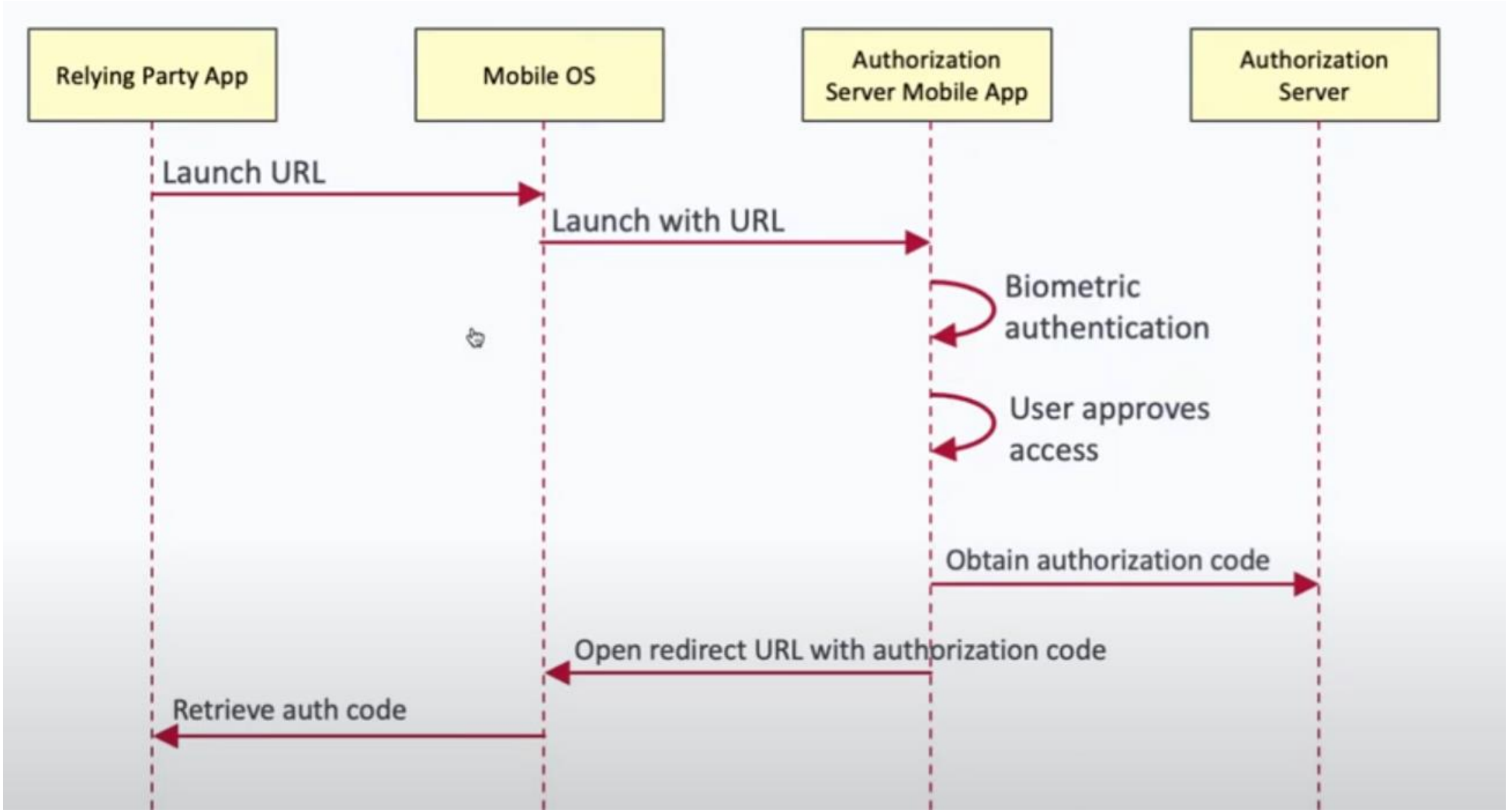
Yaron Zehavi
Raiffeisen Bank International

Challenge

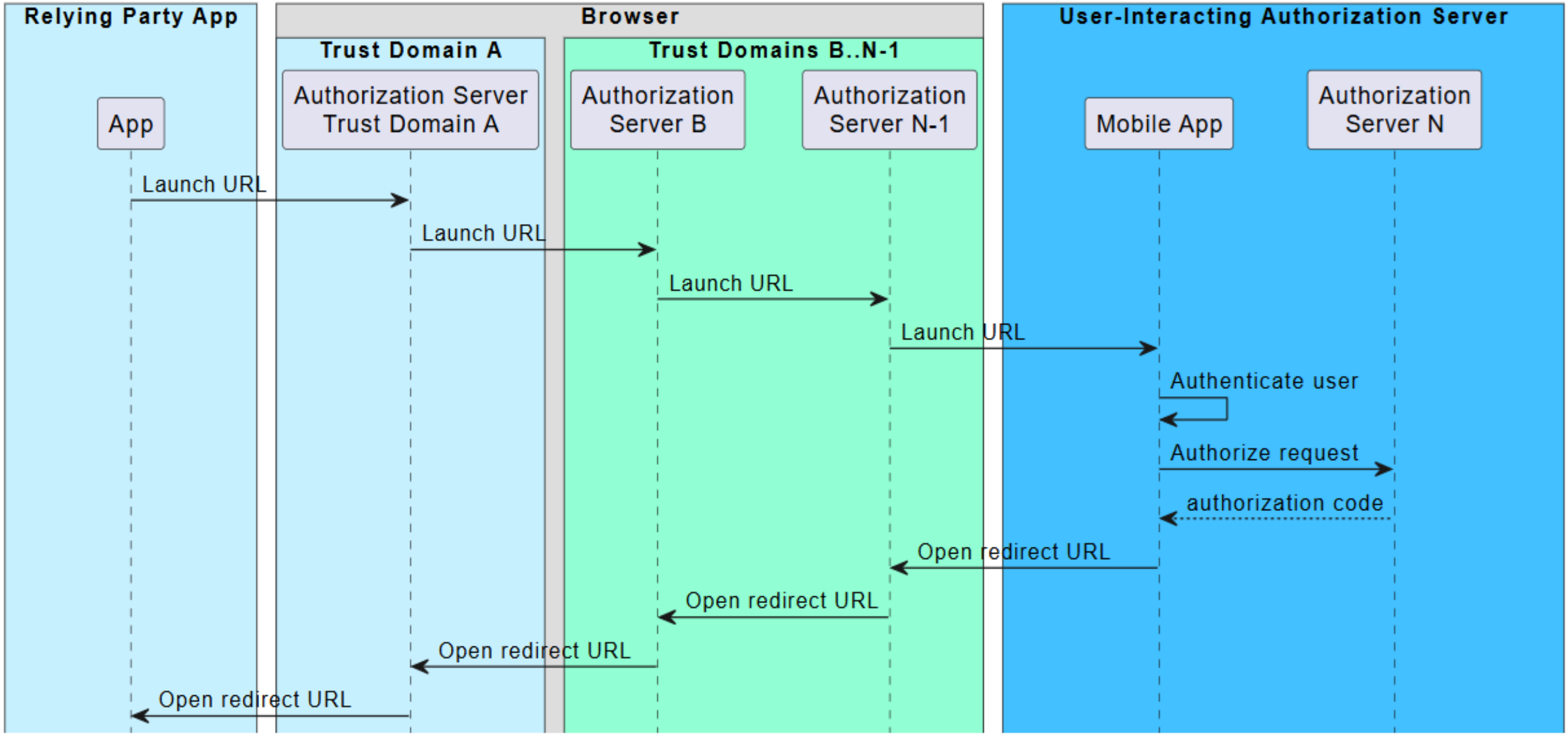
rfc 8252: App2Web



2020: App2App (Single trust domain)



App2App across non-app-claimed urls requires browser

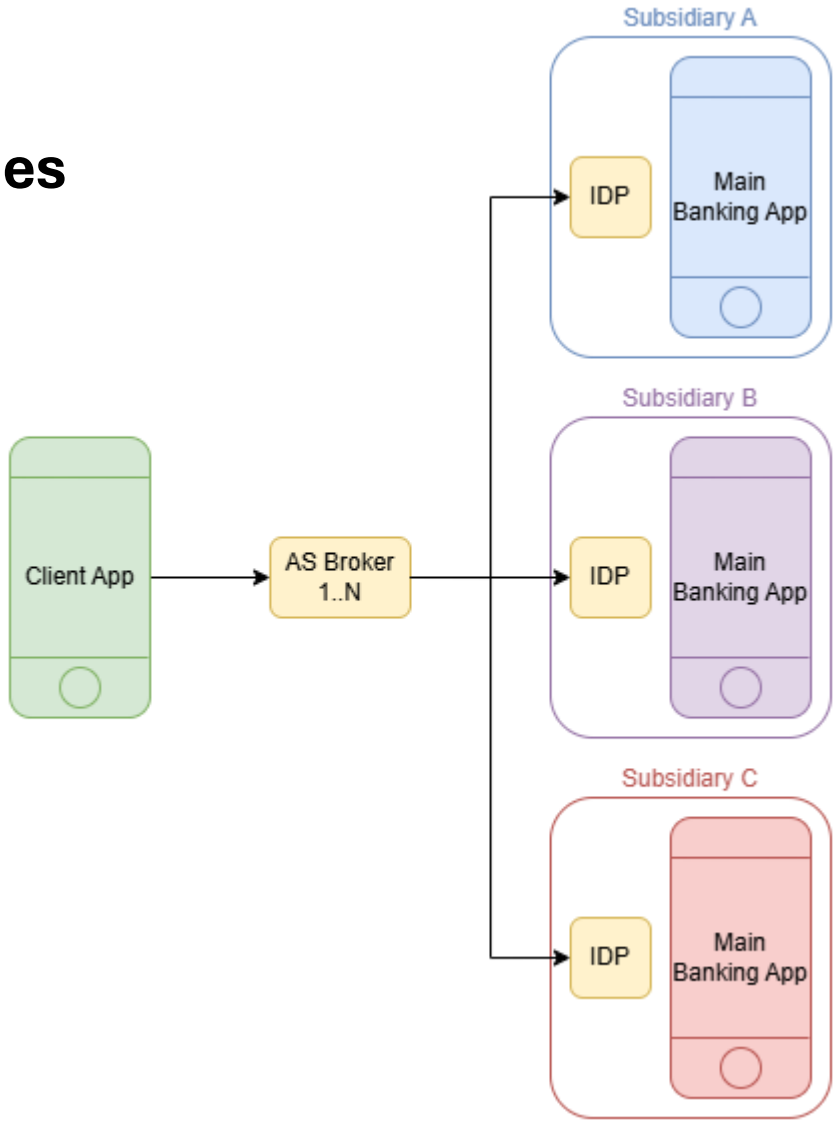


What’s wrong with browsers in app2app?

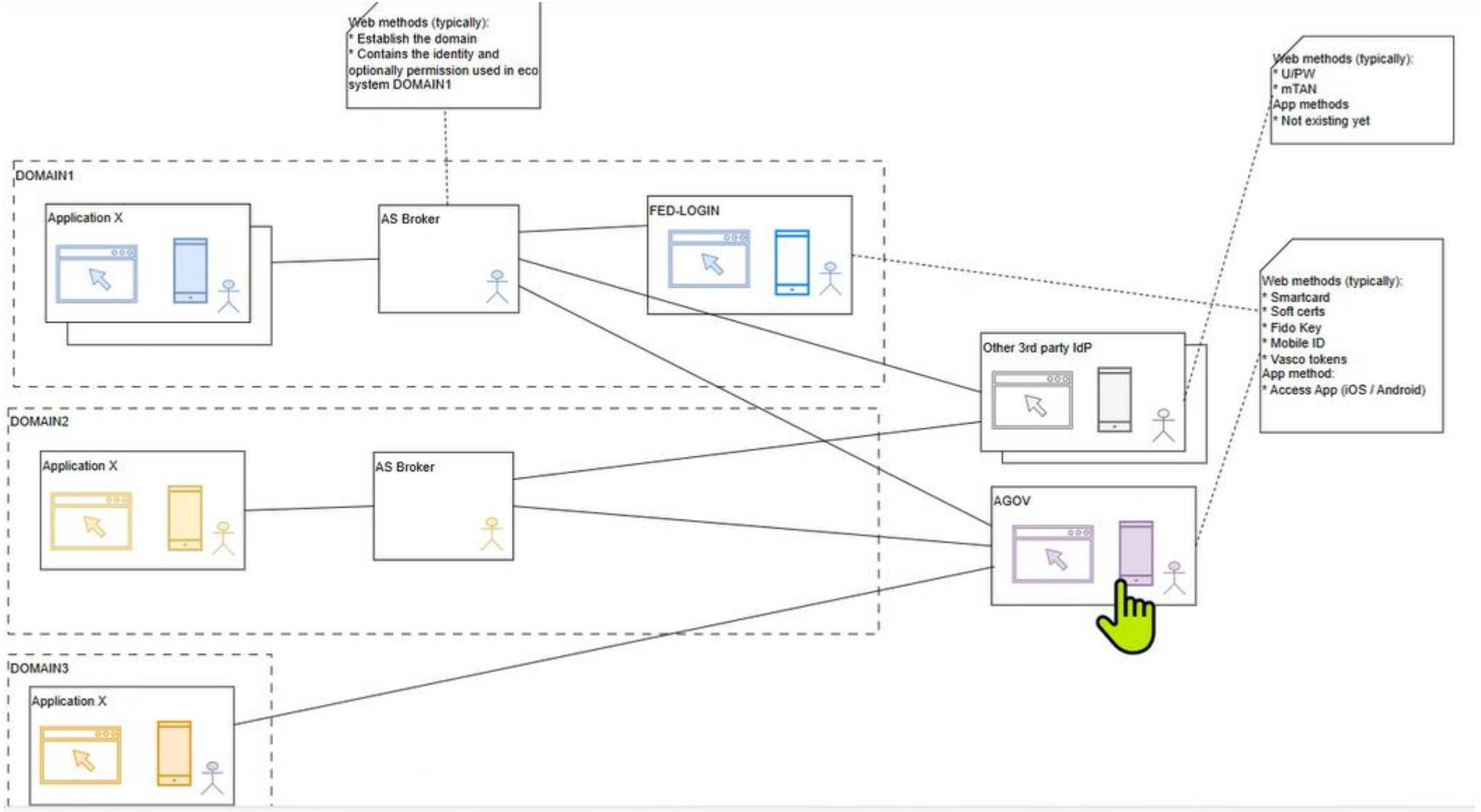
- May break flow (missing cookies)
- May prompt user to open deep links
- Adds friction and slows down UX
- Orphan tabs remain
- Business stakeholders object

App2App federation use case: Raiffeisen

11 Subsidiaries

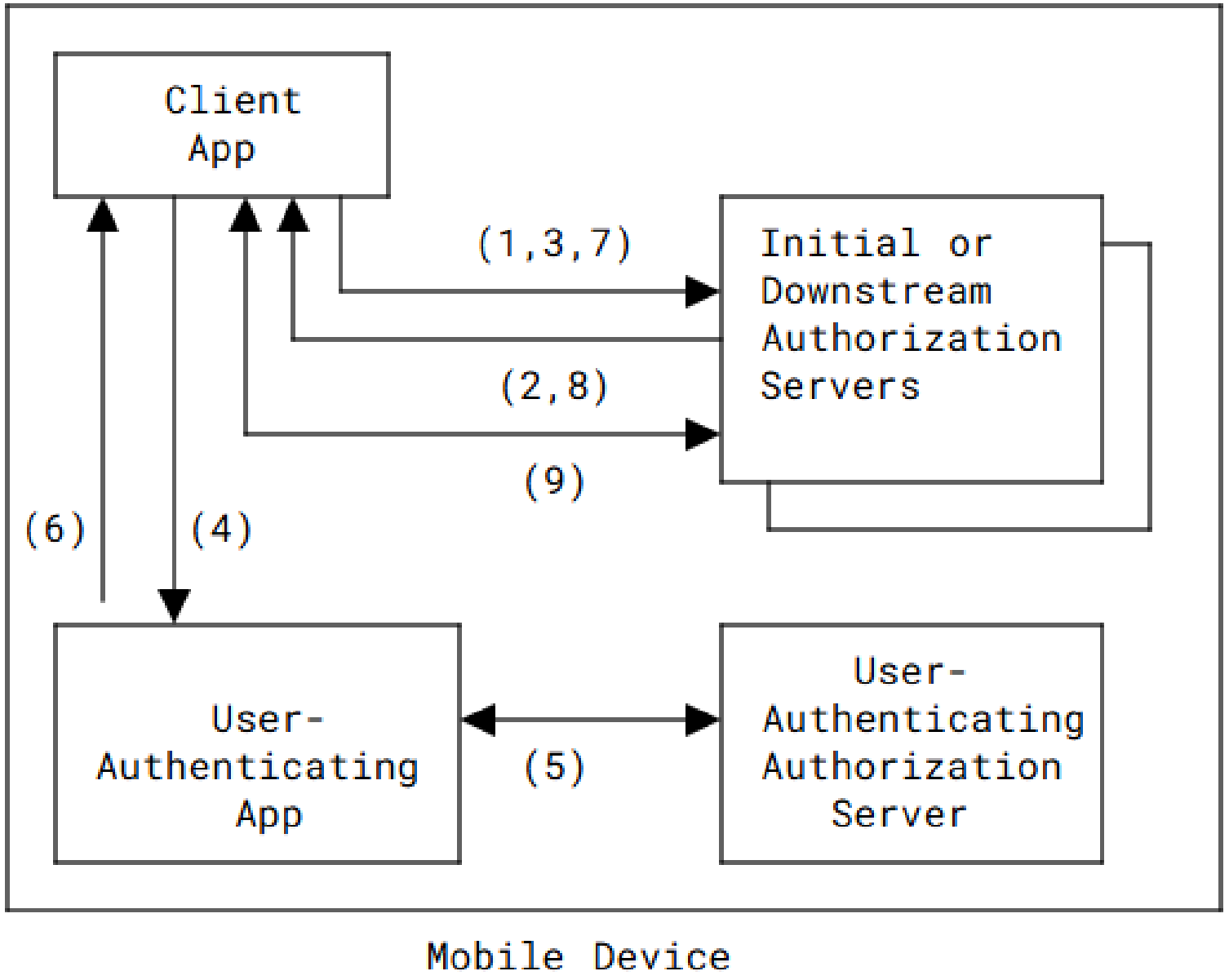


App2App federation use case: Swiss eID



Protocol Overview

App2App Browser-less draft: Client App as User-Agent



Client app calls native_authorization_endpoint

- OAuth 2.0 as REST API:
  - Returns application/json
  - Avoids HTTP 302
  - Avoids Bot detection challenges

- OAuth 2.0 endpoint interoperable with all rfc's
- Accepts native_callback_uri (registered redirect_uri)

native_authorization_endpoint response

Federating response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "action": "call",
    "url": "https://next-as.com/authorization/native?
           client_id=s6BhdRkqt3&
           request_uri=urn%3Aietf%3Aparams%3Aoauth%3Arequest_uri%
           3AR3p_hzwsR7outNQSKfoX"
}
```

native_authorization_endpoint response

Deep link response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "action": "deep_link",
    "url": "uri of native authorization request handled by *User-Interacting App*",
}
```

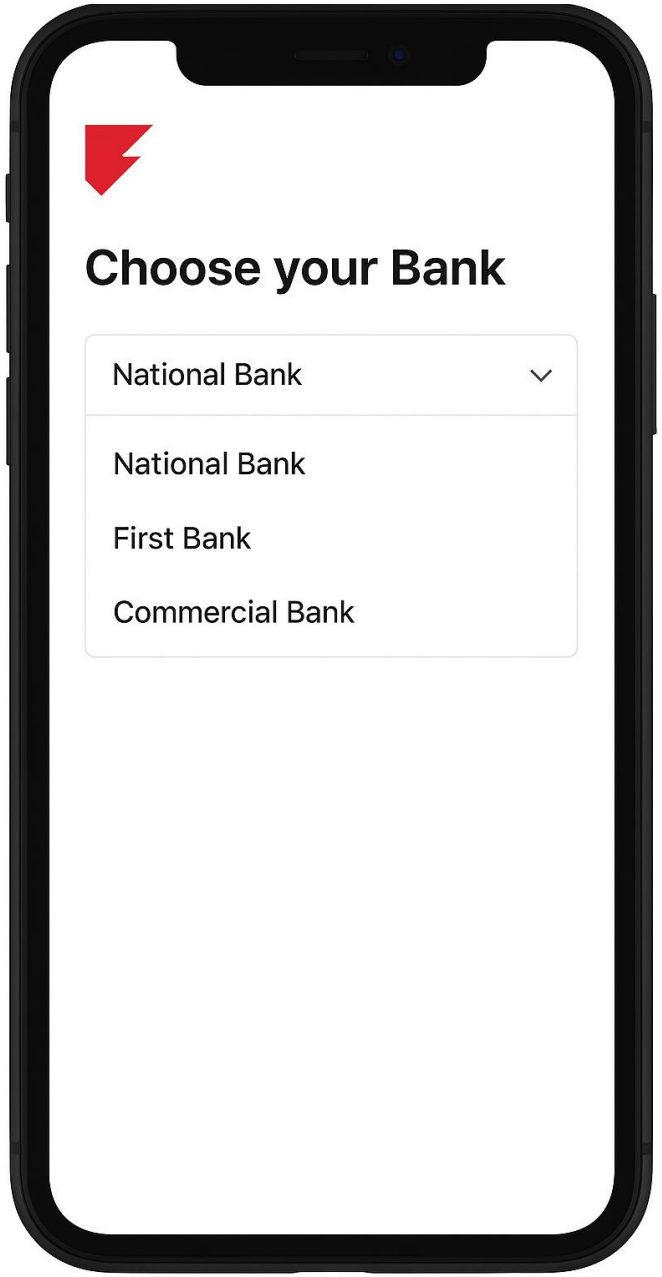
native_authorization_endpoint response

Routing guidance required (multiple choice)

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "action": "prompt",
    "id": "request-identifier-1",
    "logo": "uri or base64-encoded logo of Authorization Server",
    "userPrompt": { "options": {
        "bank": {
            "title": "Bank",
            "description": "Choose your Bank",
            "values": {
                "bankOfSomething": {
                    "name": "Bank of Something",
                    "logo": "uri or base64-encoded logo"
                },
                "firstBankOfCountry": {
                    "name": "First Bank of Country",
                    "logo": "uri or base64-encoded logo"
                }}}}},
    "response": { "post": "url to POST to using
                           application/x-www-form-urlencoded" }
}
```

native_authorization_endpoint response



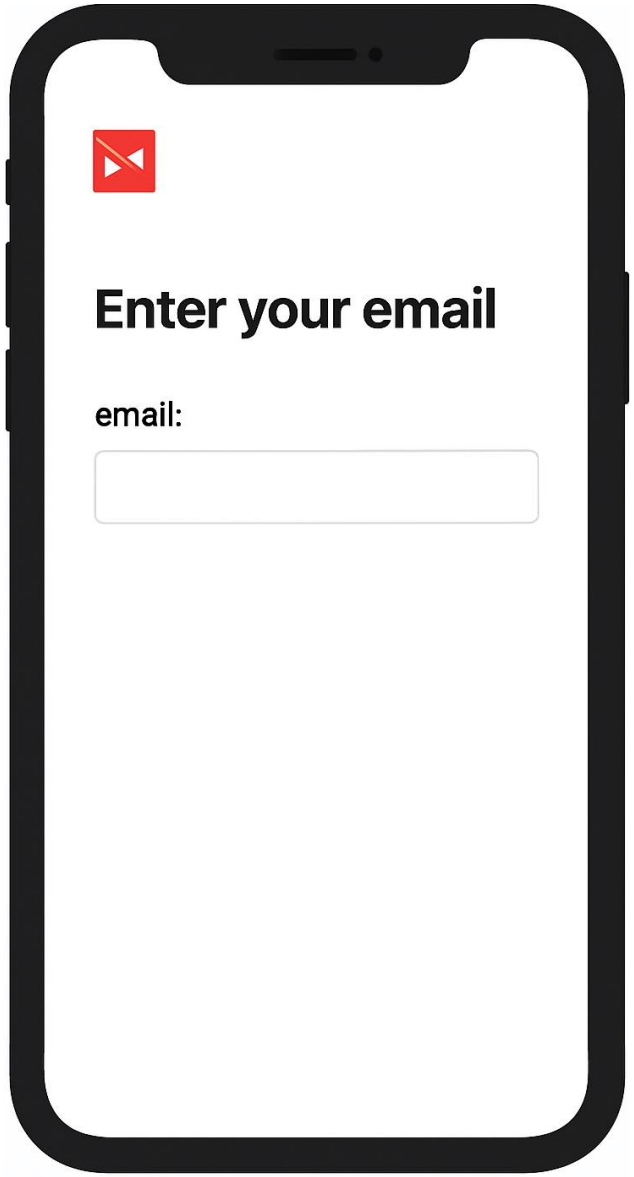
native_authorization_endpoint response

Routing guidance required (email)

```
HTTP/1.1 200 OK
Content-Type: application/vnd.oauth.app2app.routing+json

{
    "action": "prompt",
    "id": "request-identifier-2",
    "logo": "uri or base64-encoded logo of Authorization Server",
    "userPrompt": { "inputs": {
        "email": {
            "hint": "Enter your email address",
            "title": "E-Mail",
            "description": "Lorem Ipsum"
        }}},
    "response": { "get": "url to use for a GET with query params" }
}
```

native_authorization_endpoint response



Target app called via deep link

- Validates request
- Authenticates user
- Authorizes request
- Prepares response to redirect_uri
- Natively invokes native_callback_uri with redirect_uri as parameter
  - Trust establishment thru OpenID Federation Client Metadata, or allowList

Client App calls redirect_uri

- **And all subsequent uris**
- **Response syntax follows native_authorization_endpoint's response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "action": "call",
    "url": "redirect_uri of an OAuth Client, including response parameters",
}
```

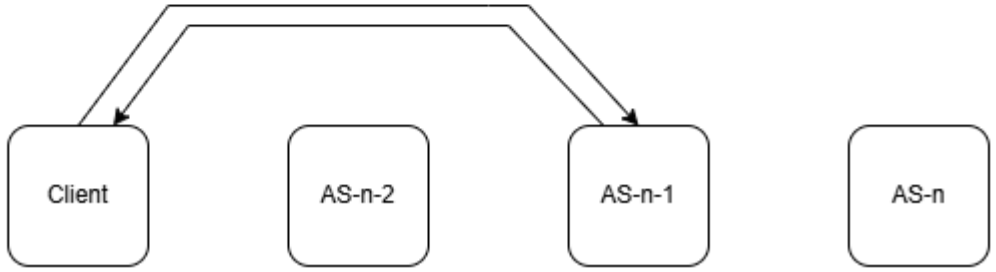
- **Until its own redirect_uri is reached and the protocol ends**

Feedback & Discussion

Binding to user agent (George Fletcher)

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "action": "call",
    "url": "https://as-n-2.com/redirect?code=SplxlOBeZQQYbYS6WxSbIA
           &state=xyz"
}
```



- How does as-n-2 lookup state (e.g pkce code_verifier) while binding to original user against theft?
- Draft uses cookies which are less common in REST
- Is there a better way?

Additional Feedback

By: George Fletcher / Arndt Schwenkschuster / Filip Skokan

- **Make PKCE mandatory**
- **Recommend par: Mitigate request leakage and manipulation**
- **Routing response privacy, not sure how to handle**

Next Steps

Requesting Feedback & Collaboration

- Presented in OSW 2025 and IETF 123
- Wish to use in Raiffeisen, Swiss eID
- Would like to ask for WG adoption in IETF 125
- Please provide additional feedback