General Remarks related to the Bank App2App implementation

May 2021

- The Bank provides TPPs with an App2App flow for both iOS and Android.

- This flow must be implemented by TPPs who provides PSUs with Mobile Applications which allow consents creation and/or payments initiation.

- The Mobile App provided by the Bank is "**Eurobank Lux Banking**".

  - Available on Apple App Store at:
    https://apps.apple.com/us/app/eurobank-lux-banking/id1473957458
    (This URL is subject to change without prior notice).

  - Available on Android Google Play at:
    https://play.google.com/store/apps/details?id=lu.eurobankpblux.ebanking&hl=en&gl=US
    (This URL is subject to change without prior notice).

To enable and leverage on the Bank App2App flow, there is only two changes to perform at TPP level.

As a TPP, when you will create a resource either *(Initial step)*:

- **a consent** – for example:
  POST https://erbklull-psd2api.luxhub.com/bg/v1/consents

- **or a payment** – for example:
  POST https://erbklull-psd2api.luxhub.com/bg/v1/payments/standard-credit-transfers

Then you will create the related authorisation(s) *(step 2)*:

- **on a consent** – for example:
  POST https://erbklull-psd2api.luxhub.com/bg/v1/consents/{consentId}/authorisations

- **or on a payment** – for example:
  POST https://erbklull-psd2api.luxhub.com/bg/v1/payments/standard-credit-transfers/{paymentId}/authorisations

CHANGE #1 *(initial step & step 2)*

The header "TPP_REDIRECT_URI" becomes mandatory.

You will have to provide the Bank with universal link related to the TPP App in this header.

You will have to request the authorise endpoint, as per the following examples *(step 6)*:

- **for a consent** – performing the following call (cf. Luxhub documentation for additional details):
  GET https://erbklull-psd2api.luxhub.com/api/oauth/authorize?response_type=code&scope=AIS%3A{consentId}%20AIS_EXT&client_id={TppApplicationId}&redirect_uri={TPP_Redirect_URI}&state={generatedState}&code_challenge={code_challenge}&code_challenge_method=S256&resource={authorisationId}

Eurobank Private Bank Luxembourg SA
534, Rue de Neudorf L-2220 Luxembourg
P.O. Box 897 L-2018 Luxembourg
Tel: (+352) 42 07 24-1
Fax: (+352) 42 07 24-65
www.eurobankpb.lu Swift ERBKLULL

- **for a payment** – performing the following call (cf. Luxhub documentation for additional details):

  GET https://erbklull-psd2api.luxhub.com/api/oauth/authorize?response_type=code&scope=PIS%3A{paymentId}&client_id={TppApplicationId}&redirect_uri={TPP_Redirect_URI}&state={generatedState}&code_challenge={PKCE_code_challenge}&code_challenge_method=S256&resource={authorisationId}

In return, a form auto-submit will be received, for example:

```
<html>
 <body onload="document.forms['form1'].submit();">
  <h1>Redirecting to https://access.eurobankpb.lu/xs2a/sca Authentication Server</h1>
  <form id="form1" action="https://access.eurobankpb.lu/xs2a/sca" method="POST">
   <input name="callback_uri" type="hidden" value="https://erbklull-cb.luxhub.com/api/oauth/callback" />
   <input name="LH_token" type="hidden" value="{JWT_TOKEN}" />
  </form>
 </body>
</html>
```

CHANGE #2 *(steps 10 & 11)*

You will have to:

1. Prevent the redirection triggered by this form

2. Extract the following information contained in it:

   - The **Bank universal link**: https://access.eurobankpb.lu/xs2a-app2app/sca

   - The **callback_uri** present in the form "input"

   - The **LH_token** present in the form "input"

3. Create the following request to provide the Bank with the related information. For example:

   GET {Bank universal link}?callback_uri={callback_uri}&LH_token={LH_token}

Finally, [*step 30*] the following call will be emitted from the Bank instead of Luxhub toward the TPP:

GET {TPP_Redirect_URI}?code={authorization_code}&state={state}

The rest of the flow is unchanged from a TPP perspective (*cf. The Network Flow overview below*).

For more details regarding:

- The Bank Main Berlin Group Swagger & The Bank CBPII Extension Berlin Group Swagger

- Additional information regarding the Bank implementation and Supported Flows

- LUXHUB - Get Started – Website & Send email to LUXHUB - Get Started

- Full Berlin Group Documentation of NextGenPSD2 Access to Account Interoperability Framework
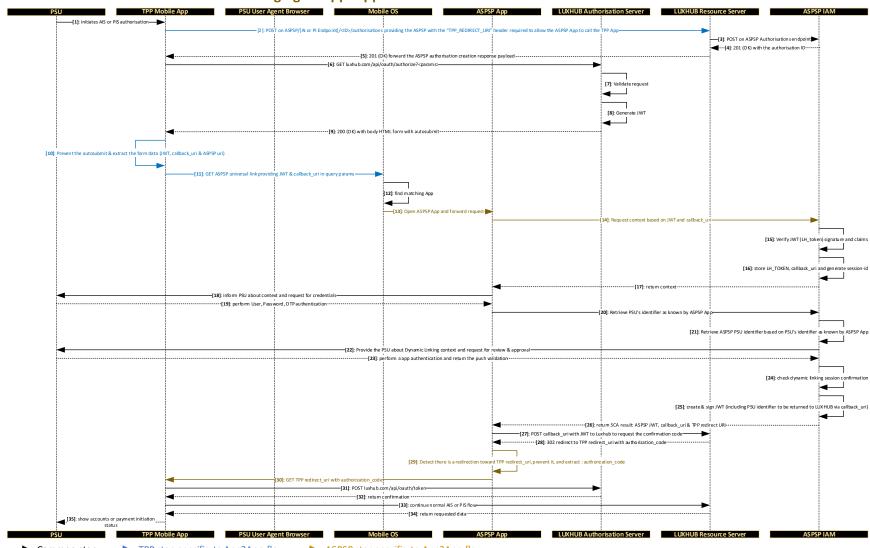
Overview of the Network Flow when Leveraging on App2App Mechanism and Universal Link

Sequence diagram with actors: PSU, TPP Mobile App, PSU User Agent Browser, Mobile OS, ASPSP App, LUXHUB Authorisation Server, LUXHUB Resource Server, ASPSP IAM

[1]: initiates AIS or PIS authorisation
[2]: POST on ASPSP/[AI or PI Endpoint]/<ID>/authorisations providing the ASPSP with the "TPP_REDIRECT_URI" header required to allow the ASPSP App to call the TPP App
[3]: POST on ASPSP Authorisation endpoint
[4]: 201 (OK) with the authorisation ID
[5]: 201 (OK) forward the ASPSP authorisation creation response payload
[6]: GET luxhub.com/api/oauth/authorize?<params>
[7]: Validate request
[8]: Generate JWT
[9]: 200 (OK) with body HTML form with autosubmit
[10]: Prevent the autosubmit & extract the form data (JWT, callback_uri & ASPSP uri)
[11]: GET ASPSP universal link providing JWT & callback_uri in query params
[12]: find matching App
[13]: Open ASPSP App and forward request
[14]: Request context based on JWT and callback_uri
[15]: Verify JWT (LH_token) signature and claims
[16]: store LH_TOKEN, callback_uri and generate session-id
[17]: return context
[18]: inform PSU about context and request for credentials
[19]: perform User, Password, OTP authentication
[20]: Retrieve PSU's identifier as known by ASPSP App
[21]: Retrieve ASPSP PSU identifier based on PSU's identifier as known by ASPSP App
[22]: Provide the PSU about Dynamic Linking context and request for review & approval
[23]: perform a app authentication and return the push validation
[24]: check dynamic linking session confirmation
[25]: create & sign JWT (including PSU identifier to be returned to LUXHUB via callback_uri)
[26]: return SCA result: ASPSP JWT, callback_uri & TPP redirect URI
[27]: POST callback_uri with JWT to Luxhub to request the confirmation code
[28]: 302 redirect to TPP redirect_uri with authorization_code
[29]: Detect there is a redirection toward TPP redirect_uri, prevent it, and extract : authorization_code
[30]: GET TPP redirect_uri with authorization_code
[31]: POST luxhub.com/api/oauth/token
[32]: return confirmation
[33]: continue normal AIS or PIS flow
[34]: return requested data
[35]: show accounts or payment initiation status

→ : Common step   → : TPP step specific to App2App flow   → : ASPSP step specific to App2App flow
→ : Call   ⇢ : Response