TurkishBank UK

TURKISH BANK UK
OPEN BANKING API SPECIFICATIONS

V 0,1

Version History

VERSION	AUTHOR	CHANGES & NOTES	DATE
0.1	Bora Uzun, Solution Architect	initial version	11.07.2019

Review Panel

NAME	ROLE	ORGANISATION	DATE
Omer Ayan	Head of IT and Projects	TBUK	
Huseyin Ergun	IT Director	Turkish Bank HQ	
Cem Tezer	Development Team Lead	Turkish Bank HQ	

Approval Panel

VERSION	AUTHOR	CHANGES & NOTES	DATE
Omer Ayan	Head of IT and Projects	TBUK	
Soner Ersoy	Vice President, CTO	Turkish Bank HQ	

1. Introduction

1.1 Backgroud

Turkish Bank UK started to work on open-banking concept within the PSD2 context from early 2019. This is a very first initial draft version to cover technical data for the integration. Initial draft focuses more on the APIs then other informational, guidance and compliance data. There might be slight changes with the information given within the following versions along with more information on non-technical, non API related data.

Todo: fill little bit of history and intro here on the next version

1.2 Objectives

The objectives of this specification document are

- To provide and introduction on open banking and PSD2
- To define all the flows , sequences and messages used across the system
- To give details of the connectivity and on-boarding
- API definitions and specs and usage samples
- Test Strategy plan introduction for staging from Sandbox to Live requirements

1.2 Version notes(v0.1)

This version focuses on technical integration spec, later versions will cover the areas mentioned. Service definition is given with a yml file. However This spec details the solution and api services as well as messages accros the system. Messaging samples and postman collections will be available with next version

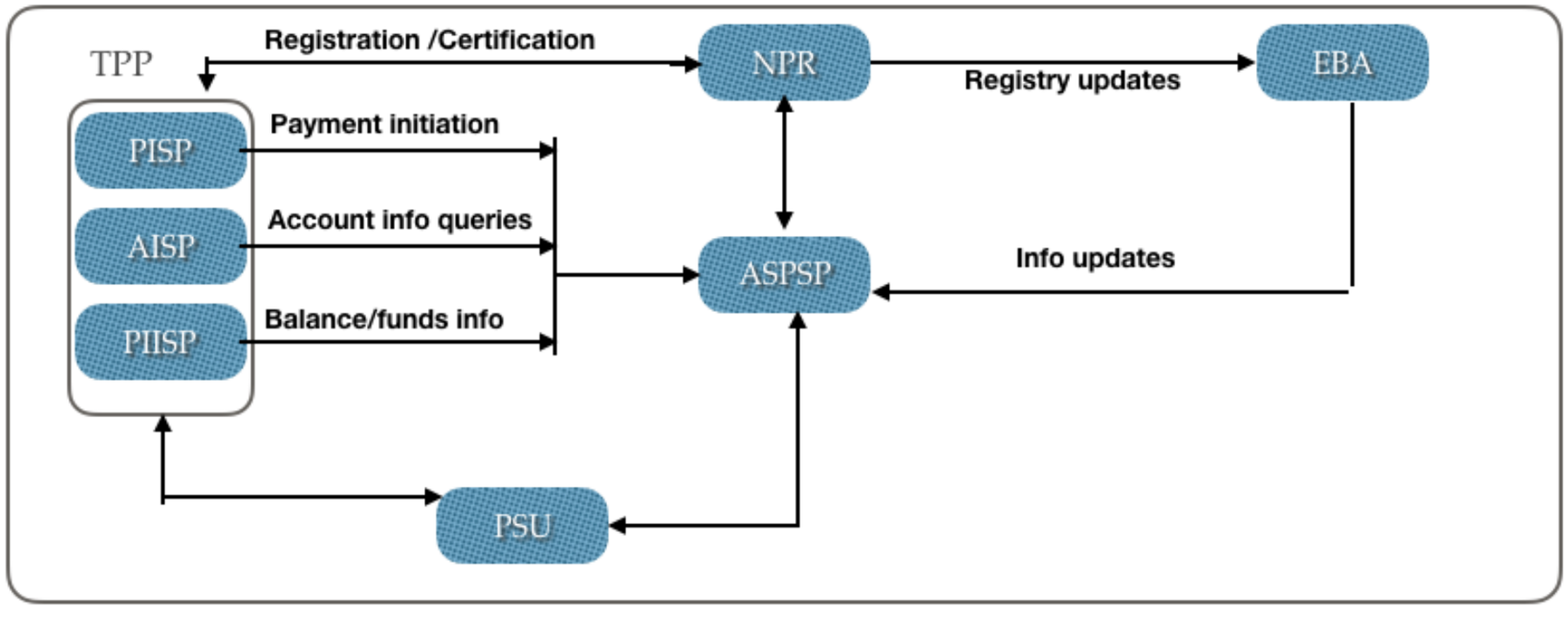
2. High Level Architecture

2.1 PSD2 at a glance

The Payment Services Directive is an EU Directive, administered by the European Commission to regulate payment services and payment service providers throughout the European Union and European Economic Area

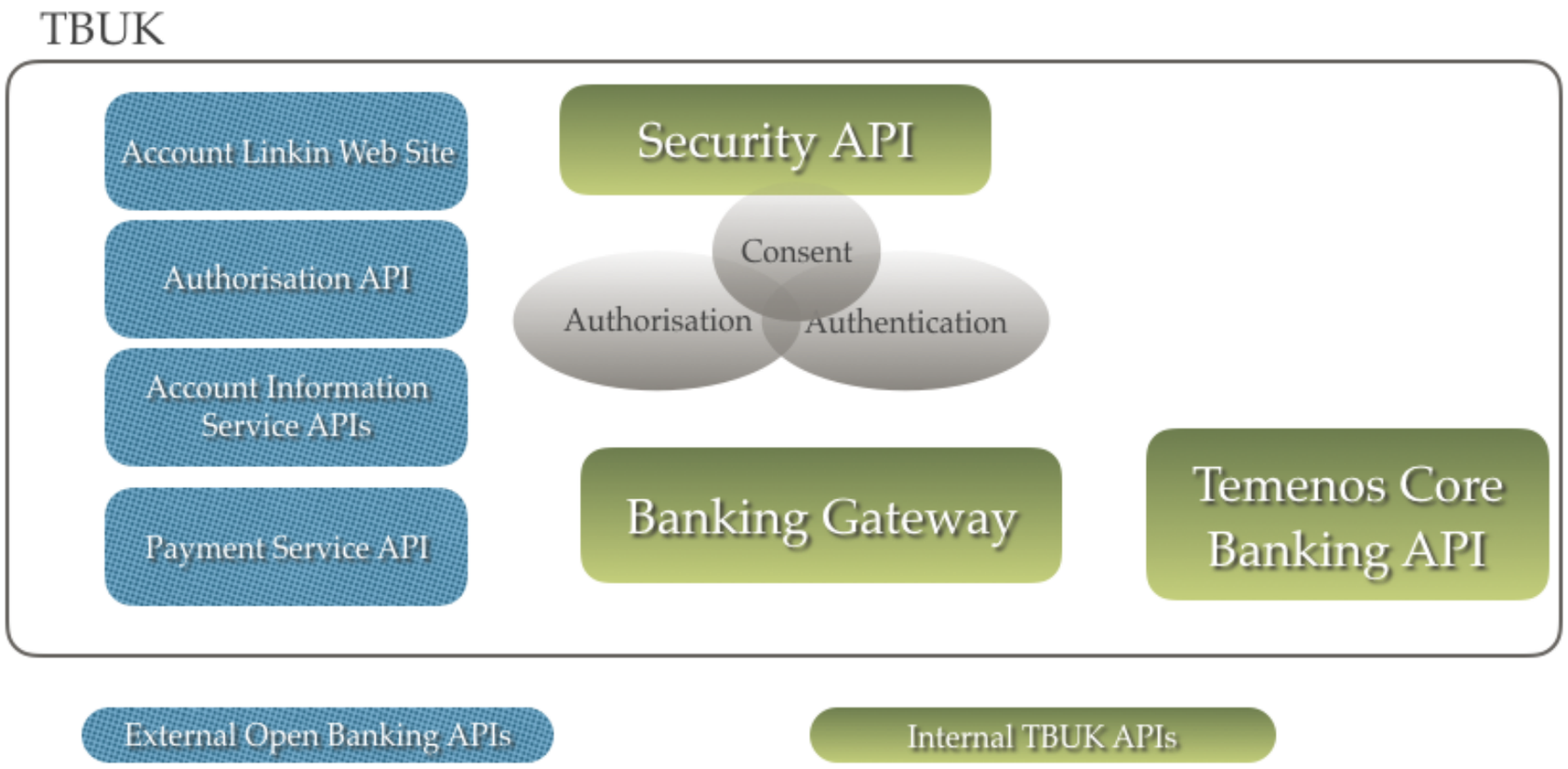
PSD2 is the second Payment Services Directive, designed by the countries of the European Union. It could revolutionise the payments industry.

PSD2 actors at a glance



todo : a quick explanation of each actor

2.1 Conceptual Diagram



Blue coloured APIs are TPP consumable PSD2/Open Banking APIs, Green coloured APIs are internal APIs that have been used in order to provide secure and reliable APIs. Grey coloured circles are the concepts that the particular API deals with

2.2 External APIs

Account Linking web site: This is the landing page to link the PSU with TPP and Turkish Bank UK. TPP redirects PSU to account linking web site where PSU give his/her consents. Turkish Bank UK redirects to TPP link with Access Code. This plays the Auth Server role on Oath2 Implicit Grant model.

Authorisation API: This is the API that TPP gets the list of consented accounts of PSU with the exchange of access token

AIS API: *Account Information Service API,* the API that will return the information for the PSU accounts, account transaction, balance and other account informations

Payment Service API: will receive the payment orders by PSU. It can also be queried for payment

2.3 Security

Turkish Bank UK Open banking APIs implements Oath2 Implicit Grant procedure. However there is an optional additional level of security required for longer life access tokens for the payments only.

Connectivity between TPP and TBUK is SSL/TLS based on certificates

2.4 Additional Security for Payments API

Why is it required ?

Oath2 Implicit Grant could be weakening security factor for long-timed access tokens. In case the public certificate and access token is captured by unauthorised malware applications it could let to unauthorised payments. Therefore TBUK has two solutions for that:

- Option one is short living access tokens for payments, so user needs to be redirected to Account Linking
- Implementing our additional authentication flow

3. Sequences

Sequences can be grouped into five categories (four fundamental and one optional)

- Authentication (aka Account Linking Sequence, Consent Sequence)
- Authorisation sequence
- Payment Authentication Sequence (Optional)
- Payment Initiation Sequence
- Account info retrieve sequence

Authentication sequence (aka Account Linking Sequence, aka Consent sequence):

When a new PSU registers with the TPP. TPP must redirect user into the Account Linking Web site with the required parameters. This is where PSU will login with TBUK internet banking two factor credentials and listed all available accounts. PSU will be able to select account he/she wants to give consent for.

If the TPP is

- certified as **AISP** only, then the **scope** in the request can be only ais (account information service)
- certified as **PISP** only, then the **scope** in the request can be only ps (payment service)
- If certified as **both** then , the **scope** in the request can be ps or ais or ais+ps(both account info and payment)

A bearer Access Token is returned back to TPP's redirect link. *Validity of the access token* depends on the scope. If it is account information services, it can be up to a month. For payment services it can have much less validity. In addition, for the payment services depending on the high risk payment requests(such as a new payee, continuous payments in short time, high amount transactions, etc) TBUK can invalidate the access token and requires OTP sign in again via Account Linking page

We recommend TPPs to have 2 clientId from us when registering. One clientId is AISP of the TPP, and another clientId for PISP. Due to implicit grant we follow, the access token given for PISP will be issued as valid for much less time, where AISP only clientId can enjoy longer period of validity.

Authorisation Sequence

Authorisation for the given consent by PSU for the particular TPP can be retrieved by utilising the Auth API by sending to access token key. In return, Turkish Bank UK will return the list of consented account details

Account Information Sequence

Sequence to gather information about Accounts, transactions and statements. Requires a valid access token which must be gathered by following Authentication sequence beforehand.

Payment Sequence

Sequences to initiate all possible payments. Requires a valid access token which must be gathered by following Authorisation sequence beforehand**.**

Payment Authentication Sequence (Optional)

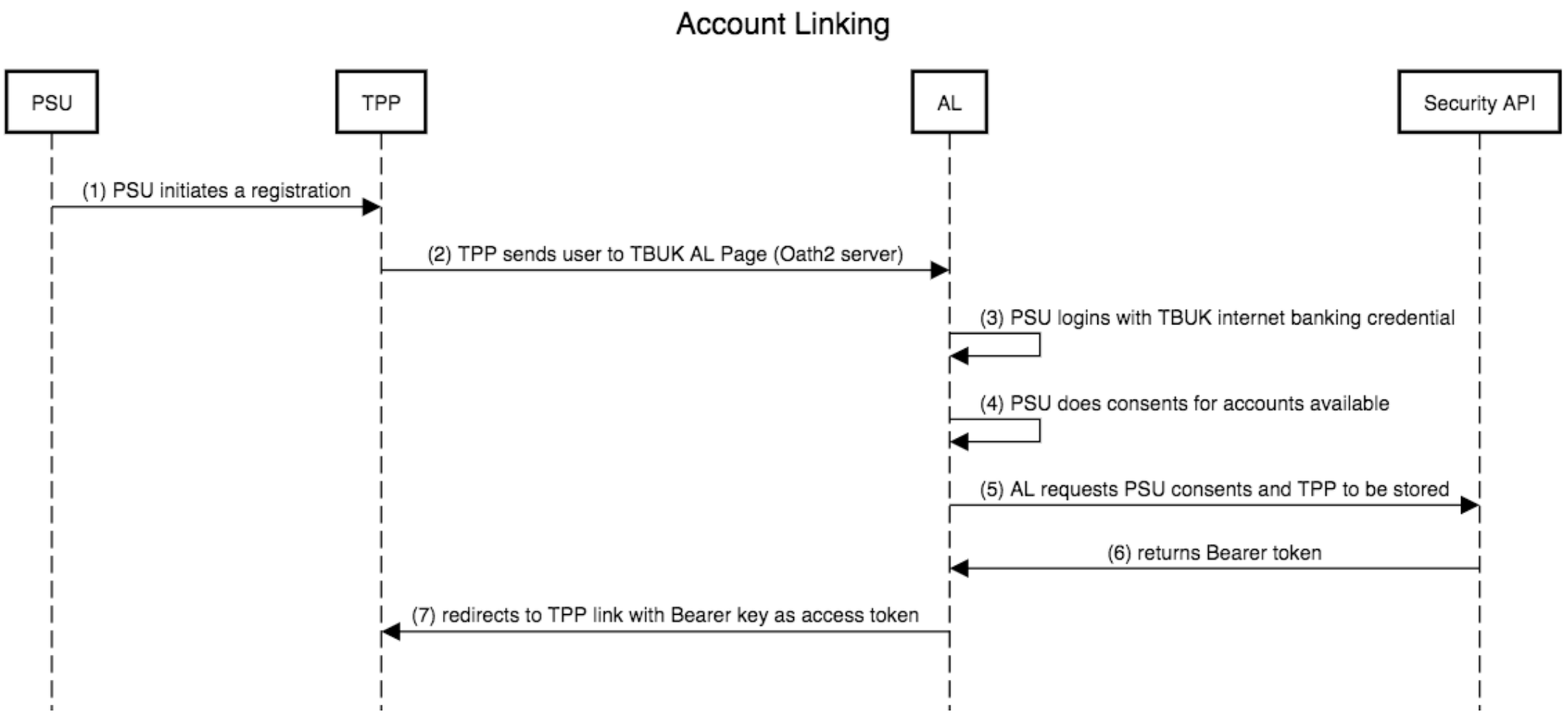
If TPP wants retrieve longer living access tokens for payments. This sequence has to be followed before every payment initiation request.

TPP will request a unique PaymentUID from the Auth API with using the access token, TBUK Auth API will return an OTP(One Time Password) which is PaymentUID. TPP and TBUK will use the same secret private key. TPP will sign/encrypt the PaymentUID with that private key and send as a parameter to Payment API along with the access token. TBUK will decode the encryption by using the same private key and check the PaymentUID and will validate the authenticity of the request in this way. This is not required, but optional. However, without this access token for PISPs will be very short living access tokens.

Please see the following pages for the sequence diagrams

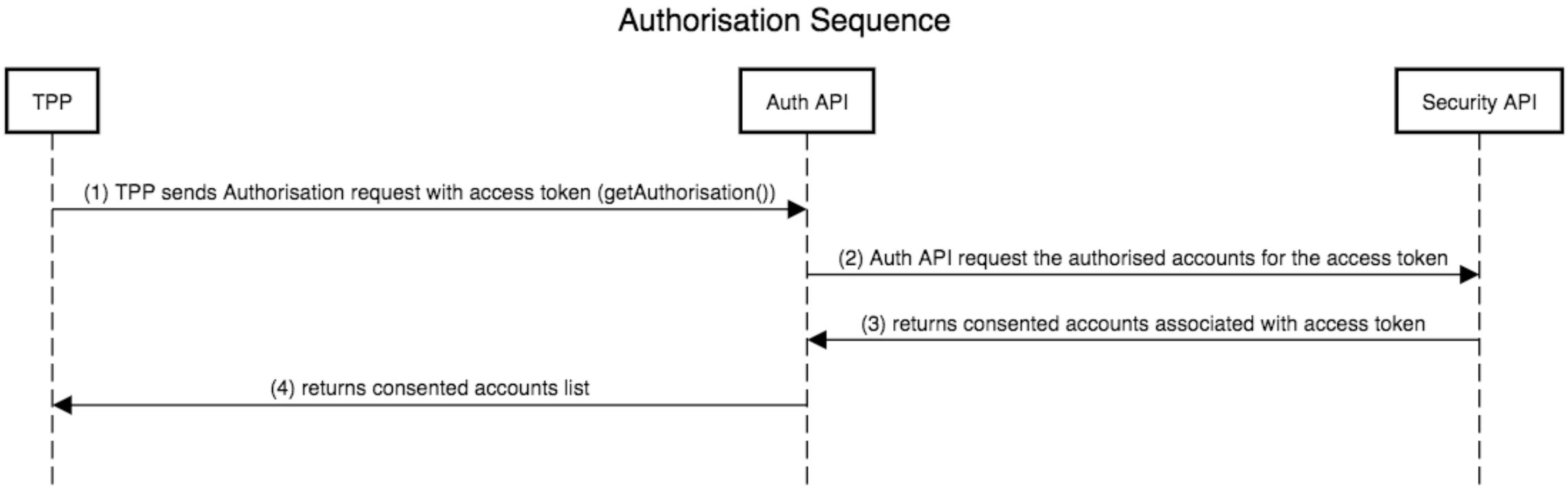
Sequence : Account Linking & Consent

1. PSU initiates an account link action on a TPP
2. TPP redirects the user to TBUK Account Linking Page
3. PSU logins with TBUK Internet Banking login details (using OTP)
4. PSU sees the authentication request's scope and selects from the accounts available to give a consent to that particular TPP
5. when PSU confirms, TBUK stores the consent
6. Security API returns a bearer token that associates the user consent details for later uses
7. Turkish Bank AL page redirects to TPP redirect URL with bearer token as access code



Sequence : Authorisation Sequence

1. TPP request the authorisation by using /getAuthorisation with required params including access token
2. Auth API sends the internal Security API
3. Security API does the access token validation and returns the consented accounts list
4. TBUK Auth API returns the list of accounts to TPP

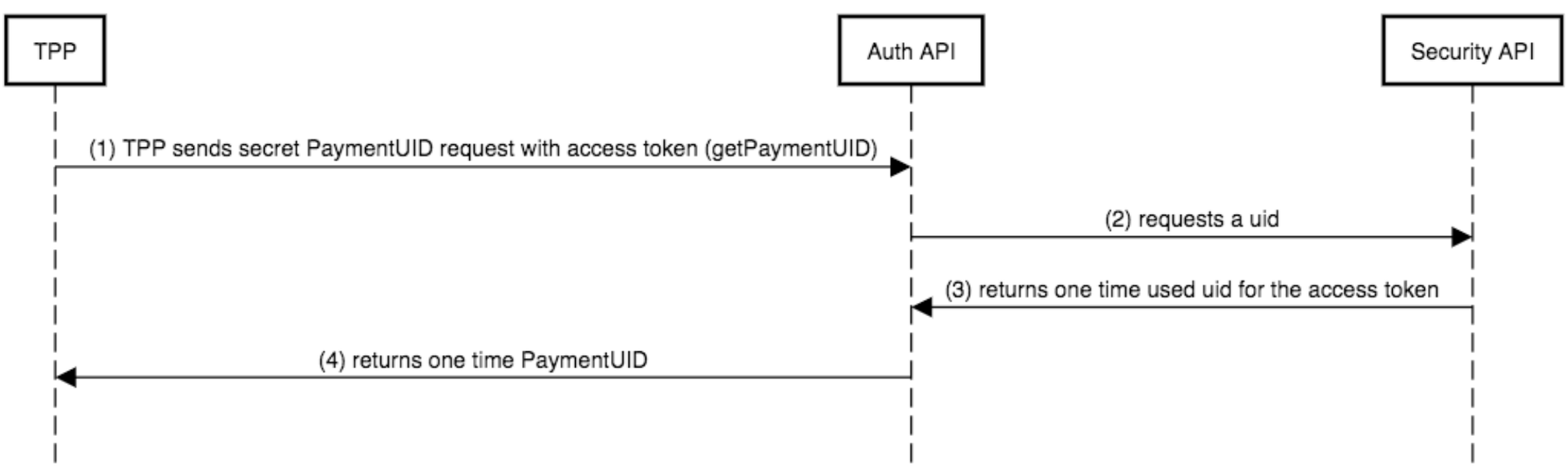


Sequence : Payment Authentication Sequence (OPTIONAL) ! Important

1. TPP requests OTP(one time password) which is PaymentUID with a access token to Auth API /getPaymentUID
2. TBUK Auth API makes an internal call to Security API and issues a PaymentUID as an OTP for the access token
3. Security API returns the PaymentUID
4. TBUK Auth API returns PaymentUID (as plain text, not encrypted)

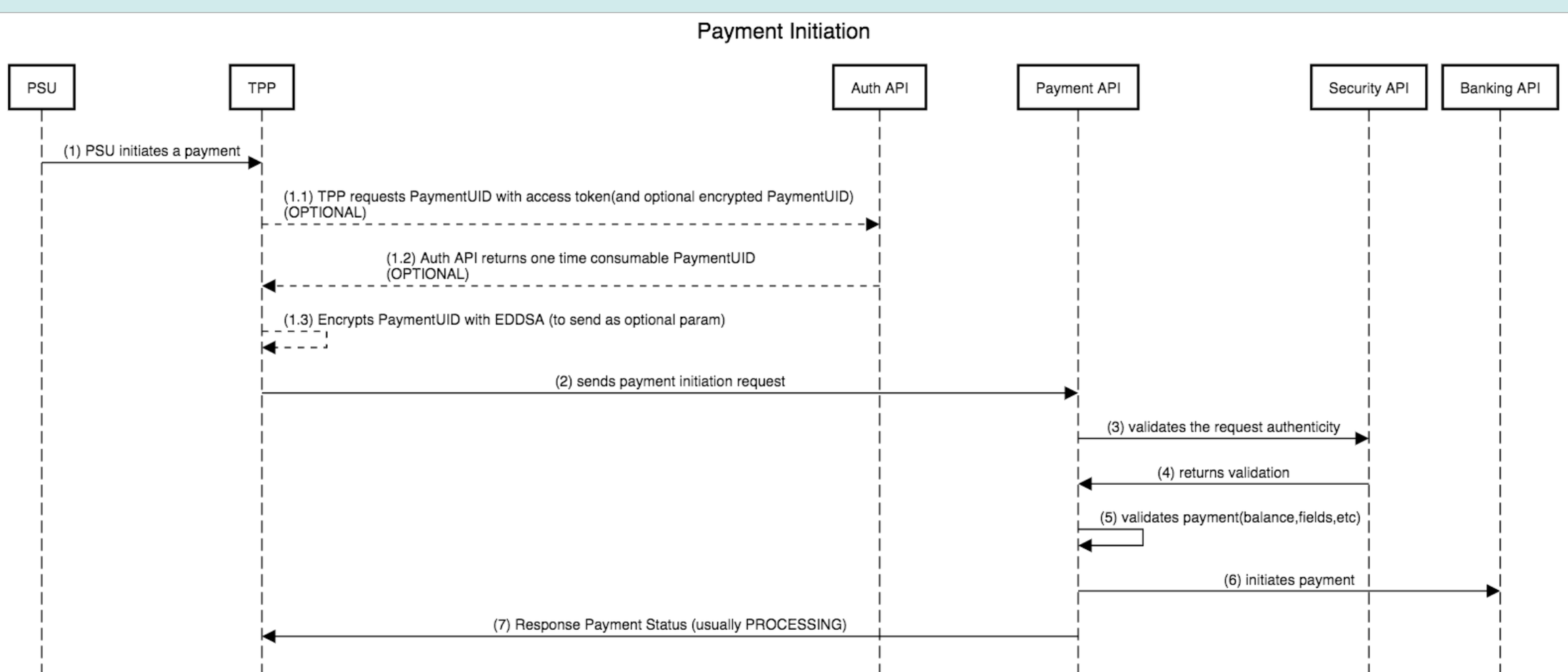
Note, TBUK will return PaymentUID as plain text, TPP will encrypt this plain text when making a call to Payment API. TBUK and TPP will be sharing the same private key for the encryption.

This is an OPTIONAL sequence flow if TPP(PISP) prefers to have longer living access tokens for Payments API



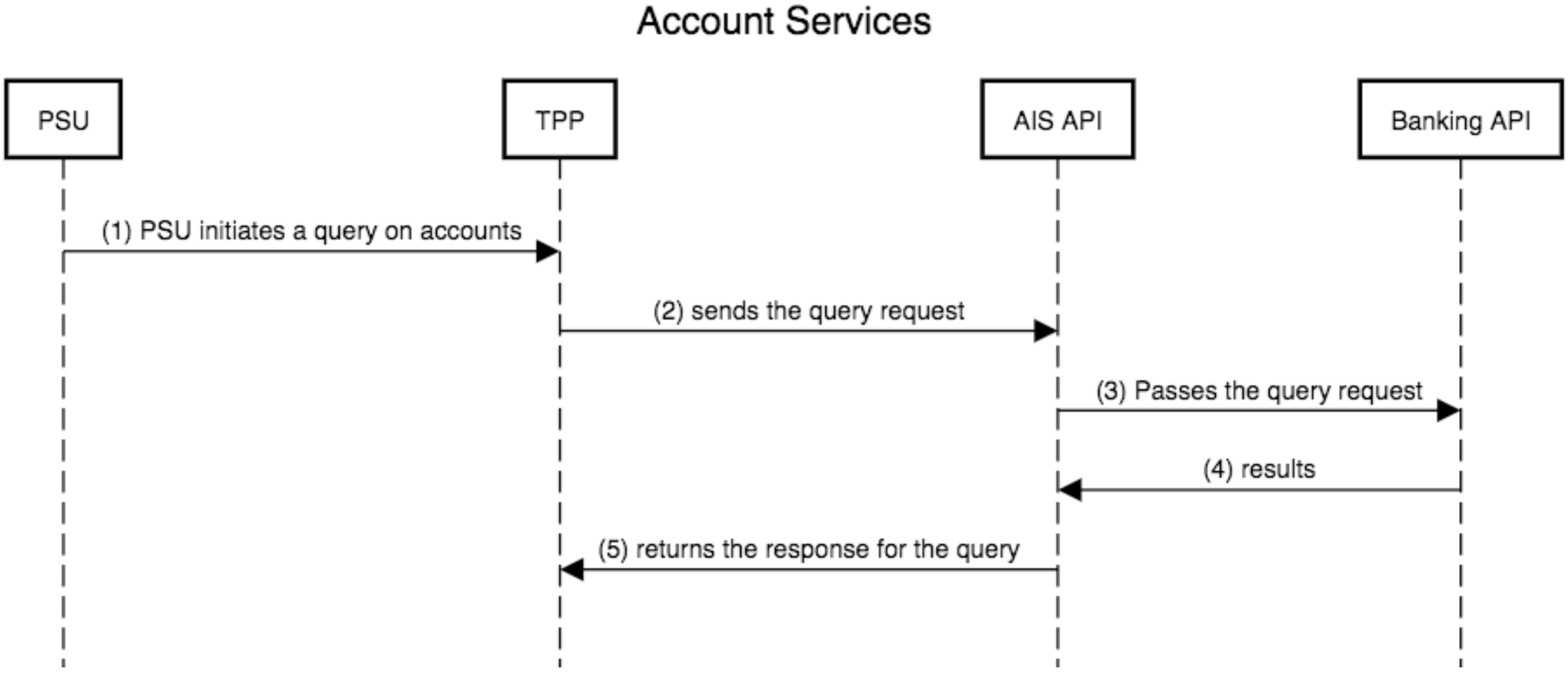
Sequence : Payment Sequence

1. PSU initiates an action on a TPP
   1.1 (Optional) TPP request the authorisation by using /getPaymentUID with required params and access token
   1.2 (Optional) Auth API returns a PaymentUID as one time password (not encrypted, this is a plain text)
   1.3 (Optional) TPP signs/encrypts the PaymentUID with a private key. (TBUK has the same private key too)
2. TPP sends Payment Initiation request to Payment API
3. Payment API passes the params and access token to Security API
4. Security API validates and sends the validation back to Payment API
5. Payment API does validation on the request (required fields such as bank account etc, as well as balance check, etc)
6. Payment API initiates the payment using the internal Payment API
7. Payment API returns back the Payment Status (This is usually confirmation of receipt)



Sequence : Account Information Sequence

1. PSU initiates an action on a TPP
2. TPP request the query on AIS(Account Information Service) API
3. AIS API validates the access token with internal APIs (Security API) and queries the Banking API
4. Banking API returns the query results
5. AIS API returns the results to TPP



4. Service Details

In this chapter we explain all the services available to TPPs. a swagger compatible TBUK-psd2-services.yml file can be found for all the service definitions on github.

Account Linking Service (web page):

base url (sandbox): https://openbanking.turkishbank.co.uk/accountlinking/servletcontroller
query parameters

- *audience:* This is the TPP Name when registered in Turkish Bank UK systems.
- *scope:* can be “ais” , “ps”, “ais+ps”
- *audience:* This is the TPP Name when registered in Turkish Bank UK systems.
- *client_id*: a TPP can have multiple clients for different scopes such as 1 is for “ais” and 2 is for “ps” scope. In such case particular client_id refers to AISP or PISP (so a certified TPP can have a PISP and AISP client)
- *memberId:* This is the unique identifier of the PSU on TPP system.

Sample Request:

https://openbanking.turkishbank.co.uk/accountlinking/servletcontroller?audience=TPPNAME&scope=ais&response_type=token&
tppMemberId=
client_id=TPP_CLIENT_ID_FOR AISP&
redirect_uri=https://YOURLINK/callback&

Auth Service

Method name: **GetAuthorisation**

base url (sandbox): https://openbanking.turkishbank.co.uk:6135/Auth/getAuthorisation

http method: GET

http header params :

- *Authorization:* ACCESS_TOKEN issued by Account Linking
- audiance:TPPNAME
- clientId:TPP_CLIENT_ID

Sample Postman for request :

https://github.com/tbukdev/sandbox.openbanking/blob/master/TBUK-OpenBanking-API.postman-collection-client-AUTH-.json

Method name: GetPaymentUID

base url (sandbox): https://openbanking.turkishbank.co.uk:6135/Auth/getPaymentUID

http method: POST

http header params :

- *Authorization:* ACCESS_TOKEN issued by Account Linking
- audiance:TPPNAME
- clientId:TPP_CLIENT_ID

Sample Postman for request :

https://github.com/tbukdev/sandbox.openbanking/blob/master/TBUK-OpenBanking-API.postman-collection-client-AUTH-.json

Note : postman collections will be available from 13.07.2019

Account Information Service(1)

Method name: **getAccountList**

base url (sandbox): https://openbanking.turkishbank.co.uk:6135/AccountInformation/getAccountsList

http method: GET

http header params :

- *Authorization:* ACCESS_TOKEN issued by Account Linking
- audiance:TPPNAME
- clientId:TPP_CLIENT_ID

Sample Postman for request :
https://github.com/tbukdev/sandbox.openbanking/blob/master/TBUK-OpenBanking-API.postman-collection-client-AIS-.json

Method name: **getAccountDetails**

base url (sandbox): https://openbanking.turkishbank.co.uk:6135/AccountInformation/getAccountDetail/{accountId}

http method: GET

http header params :

- *Authorization:* ACCESS_TOKEN issued by Account Linking
- audiance:TPPNAME
- clientId:TPP_CLIENT_ID

Sample Postman for request :
https://github.com/tbukdev/sandbox.openbanking/blob/master/TBUK-OpenBanking-API.postman-collection-client-AIS-.json

Note : postman collections will be available from 13.07.2019

Account Information Service(2)

Method name: **getAccountTransactions**

base url (sandbox):
https://openbanking.turkishbank.co.uk:6135/AccountInformation/getAccountTransactions/{accountId}/{dateFrom}/{dateTo}

http method: GET

http header params :

- *Authorization:* ACCESS_TOKEN issued by Account Linking
- audiance:TPPNAME
- clientId:TPP_CLIENT_ID
- tpp*MemberId*

Sample Postman for request :
https://github.com/tbukdev/sandbox.openbanking/blob/master/TBUK-OpenBanking-API.postman-collection-client-AIS-.json

Method name: **getStatement**

base url (sandbox): https://openbanking.turkishbank.co.uk:6135/AccountInformation/getStatement/{accountId}/{month}/{year}

http method: GET

http header params :

- *Authorization:* ACCESS_TOKEN issued by Account Linking
- audiance:TPPNAME
- clientId:TPP_CLIENT_ID
- tpp*MemberId*

Sample Postman for request :
https://github.com/tbukdev/sandbox.openbanking/blob/master/TBUK-OpenBanking-API.postman-collection-client-AIS-.json

Note : postman collections will be available from 13.07.2019

Payment Initiation Service(1)

Method name: **domesticPayment** - for UK transfers

base url (sandbox): https://openbanking.turkishbank.co.uk:6135/PaymentInitiation/domesticPayment

http method: POST

http header params :

- *Authorization:* ACCESS_TOKEN issued by Account Linking
- audiance:TPPNAME
- clientId:TPP_CLIENT_ID
- *nonce:* encrypted PaymentUID (Optional see Payment Auth Sequence)

Sample Postman for request :

https://github.com/tbukdev/sandbox.openbanking/blob/master/TBUK-OpenBanking-API.postman-collection-client-PAYMENT-.json

Method name: **internationalPayment**

base url (sandbox): https://openbanking.turkishbank.co.uk:6135/PaymentInitiation/internationalPayment

http method: POST

http header params :

- *Authorization:* ACCESS_TOKEN issued by Account Linking
- audiance:TPPNAME
- clientId:TPP_CLIENT_ID
- *nonce:* encrypted PaymentUID (Optional see Payment Auth Sequence)

Sample Postman for request :

https://github.com/tbukdev/sandbox.openbanking/blob/master/TBUK-OpenBanking-API.postman-collection-client-PAYMENT-.json

Note : postman collections will be available from 13.07.2019

Payment Initiation Service(2)

Method name: **transferBetweenAccounts** - for Transfering between TBUK accounts or PSU Accounts

base url (sandbox): https://openbanking.turkishbank.co.uk:6135/PaymentInitiation/transferBetweenAccounts

http method: POST

http header params :

- *Authorization:* ACCESS_TOKEN issued by Account Linking
- audiance:TPPNAME
- clientId:TPP_CLIENT_ID
- *nonce:* encrypted PaymentUID (Optional see Payment Auth Sequence)

Sample Request: <github url of sample>

Sample Response: <github url of sample>

Method name: **paymentStatusEnquiry**

base url (sandbox): https://openbanking.turkishbank.co.uk:6135/PaymentInitiation/paymentStatusEnquiry/{transactionId}

http method: GET

http header params :

- *Authorization:* ACCESS_TOKEN issued by Account Linking
- audiance:TPPNAME
- clientId:TPP_CLIENT_ID
- *nonce:* encrypted PaymentUID (Optional see Payment Auth Sequence)

Sample Postman for request :

https://github.com/tbukdev/sandbox.openbanking/blob/master/TBUK-OpenBanking-API.postman-collection-client-PAYMENT-.json

Note : postman collections will be available from 13.07.2019

5. Messages & Models

In this section, you will find explanations for the request and response messages. As well as details of model classes

All the model classes are available on the github. It requires lombok. The details of setting up lombok on IDEs(Eclipse, IntelliJ) can be found on this link https://www.baeldung.com/lombok-ide

todo: push models into github repo

Payment Initiation Messages

TBUK uses a canonical payment initiation message for all payment methods. However, expected fields are different by methods.

Payment Initiation Request message has 4 parts

```
private GrpHdr grpHdr; // Header -> refrenceId (reference Id for TPP), CreDt (Date Time of message)
private BankAccount debtor; // from account
private BankAccount creditor; // to account
private InstructedAmount instructedAmount; // Amount, Currency
private RemittanceInfo remittanceInfo; // unstructeredText for description
```

Bank Account has following possible fields

```
private String accountId; // unique TBUK ID for the account
private String name;      // Account name
private String sortCode;
private String accountNo;
private String bicCode;
private String iban;
private Currency currency;
private AccountSubType accountSubType; // CURRENT_ACCOUNT , NOTICE ACCOUNT, TIME DEPOSIT
private boolean statementAvailable;
private Address address;
```

```
{
  "accountId": "08351342",
  "name": "JAMES BOND",
  "sortCode": "010101",
  "accountNo": "12345678",
  "statementAvailable": false
}
```

JSON representation can ignore non-required fields. For example domestic payment initiation doesn't require bicCode and Iban number. The JSON representation of a BankAccount at the right is valid.

prepare and push postman samples to github

todo: next version to include request, response and models