PSD2 API Solution - Documentation for TPPs

Addressees:

Business Analysts, Project Managers, Developers, Architects, IT

Authors:

David Schneider, Lars Kieffer, Gerald Haase

Version, date

Version 2.1.1, April 08th, 2021

Copyright

CREALOGIX (Germany) AG
Breitscheidstraße 10
70174 Stuttgart

www.crealogix.com

Table of Contents

List of figures

Document history

Version	Description (remarks)	Date	Author(s)
0.9	First draft	March 13st, 2019	David Schneider, Lars Kieffer, Gerald Haase, Jörg Flade, Ludwig Volk
1.0	Version 1.0	March 13st, 2019	David Schneider, Lars Kieffer, Gerald Haase, Jörg Flade, Ludwig Volk, Martin Bierkoch
1.1	Available Consents added and further clarification in CMM, Clarification TX Data older 90 days, Consent for Available Accounts	November 18th, 2019	David Schneider
1.2	Standing orders, Account owner, Minor corrections	March 19th, 2020	David Schneider
2.0	New Workflows for new TPP process especially Pre-Authentication. TPP registration and API subscription removed. TPP must provide PSD2 Certificate on each API call.	November 20th, 2020	David Schneider
2.1	Minor clarification for only first access. Call Berlin Group API first before calling Pre-Auth API.	February 15th, 2021	David Schneider

1 Introduction

This document describes how TPPs can connect to the PSD2 API Solution.

The document assumes that you have basic knowledge about Payment Services Directive 2 (PSD2) regulation of the European Union, its terminology and use cases. Please refer to the References section below for an overview and detailed information about the regulation. In addition, you will also find a Glossary below with the most important PSD2 terms.

TPPs can use the PSD2 API solution to connect their services. A TPP will use the integrated API Management tool for the PSD2 requests. After login the TPP can subscribe to respective API. PSD2 API Solution will rely on NextGenPSD2 *Access to Account Interoperability Framework* specified by *The Berlin Group* version 1.3.6. The subscription is necessary to allow TPPs to consume the API. This process is explained in this document.
The PSD2 API Solution will follow the Berlin Group Specification for the comply-only features. Thus, it is possible for an end customer - amongst other functions - to get a balance or transactions of the customer's payment accounts, make a payment initiation and check the availability of funds via a TPP. However, not all methods and fields will be available. For further details see chapter *Comply only (MVP).*

Having received a request from a TPP the PSD2 API Solution will then identify the TPP before executing the request. If the ASPSPs backend enforces a SCA via the PSU, the OTP needs to be entered.

A description of the workflows will be given in the workflow section. For the same reasons given above consecutive requests will not be possible in the sandbox. If this function is possible the TPP needs to remember some information like Consent-ID, Payment-ID etc. See Berlin Group Implementation Guidelines.

A TLS-connection between TPP and ASPSP has to be established always including client (i.e. TPP) authentication. For this authentication the TPP will use a qualified certificate for website authentication (QWAC). TPP must provide certificate on each API call. This qualified certificate is issued by a Qualified Trust Service Provider (QTSP) according to the eIDAS regulation. The certificate of the TPP will indicate all roles the TPP is authorized to use. And the QWAC has to be fully compliant to the official standard ETSI TS 119 495.

The TPP will always be identified on Transport Layer (with qualified digital certificate). Additional identification of the TPP at application level (with electronic seal) is not part of the solution. TPP requests data, ASPSP only responses to requests.

Representational state transfer (REST) is used for communication through requests and responses.

2 Sandbox

The sandbox is used to document the API and offer the TPPs the possibility to view the methods. This is compliant to article 30 (3) of the RTS document. In addition, the first test calls can already be made.

It is possible that TPPs can test some calls of the API and receive corresponding demo responses. Note that the APIs are not connected to the backend and therefore return generated mock data. Therefore, consecutive calls and two-factor authentications are not possible.

In the sandbox won't be any certificate check, role check, authorization on the APIs according to the role. This will be later part of the solution for the official TPP test according to Article 30 (5) of the RTS document.

2.1 API Store

The API store enables TPPs to browse the API offerings of the ASPSP, test them via a Sandbox with mock services.

2.2 Using the sandbox

The following workflow shows how a TPP can connect to the ASPSP via the PSD2 API Solution.

Now you can start testing the API, e.g .Account Information. Scroll down the Berlin Group API, for example to the point "/v1/accounts".

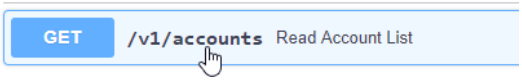

Figure 1 - Get accounts 1/2

After a click on the colored background, the respective item opens for a larger description.

At the end of the description you will see a button "try it out" on the right. Please click on it.

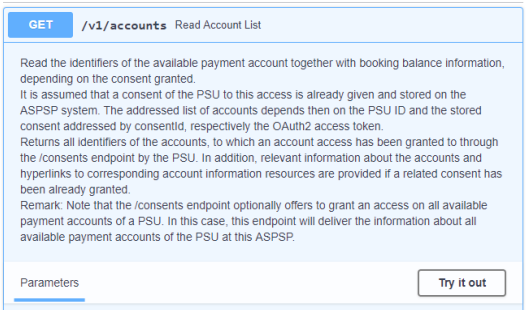

Figure 2 - Get accounts 2/2

The dialog expands again, and you can see the necessary input fields. These contain all parameters that you have to transfer to the API in the later application.

To test the API you can fill in these parameters manually here in the dialog. For example, enter "123" as "Consent-ID".

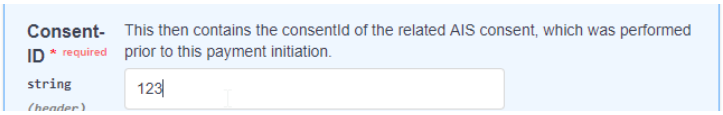

Figure 3 - Consent-ID

If all mandatory parameters are filled, scroll further down. There you will find the "Execute" button. Via this button you initiate a communication with the API and then receive feedback or replies.

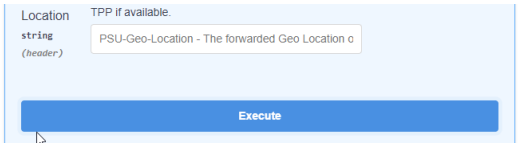

Figure 4 - Execute

Your entries are not checked in the sandbox but answered with random generated sample data.

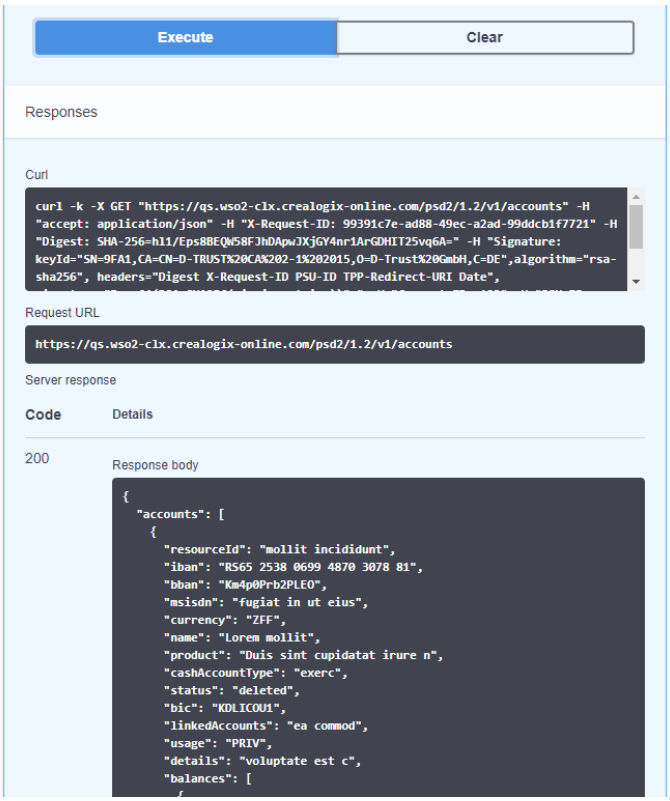

Figure 5 - Response

3 Architecture and Workflows

The TPP will call the APIs via WSO2 API Management. The API call contains the QWAC certificate. Only QWAC certificates with ETSI Standard TS 119 495 can be connected. TPPs must obtain a PSD2 QWAC from any of the participating qualified trust centers.

3.1 TPP uses the service

3.1.1 Usage of TPP application by end customer

If the PSU initiates a request via the TPP application, OAuth2 Server will be used for login. Here the TPP certificate and the credentials of the PSU are necessary for login. The OAuth2 Server calls the TPP Management module to validate the given certificate. If TPP Management approves the certificate, the OAuth2 Server can provide a token to the TPP for this user. If TPP Management denies the certificate, OAuth2 Server will also deny the login request.

A TLS-connection between TPP and ASPSP has to be established always including TPP authentication. For this authentication the TPP has to use a qualified certificate for website authentication (QWAC). This qualified certificate needs to be issued by a Qualified Trust Service Provider (QTSP) according to the eIDAS regulation. The certificate of the TPP will indicate all roles the TPP is authorized to use.

The TPP has to be always identified on Transport Layer (with qualified digital certificate). Additional identification of the TPP at application level (with electronic seal) is not part of this solution.

All secure connections are handled via TLS-Protocol over HTTPS. Only the TPP will be able to establish a connection. ASPSP only makes a response to that request. Representational state transfer (REST) for communication will be used through requests and responses.

3.1.2 ASPSP manages TPP

There are various reasons for deactivating a TPP, for instance if NA has canceled admission, the certificate of TPP has been revoked, if TPP behaves inappropriate or if TPP has abnormal API usage. Therefore, the ASPSP has the possibility to deactivate the TPP with immediate effect.

3.1.3 Strong Customer Authentication and Consent of PSU

When the PSU performs a Strong Customer Authentication (SCA) the integrated OAuth2 Server of the solution is supposed to use the already existing authentication technology for 2FA that is also used for the ASPSP's Online Banking. This results in a consistent user experience for the PSU.

After a PSU's SCA for accessing account information the ASPSP can omit SCA for account information for up to 90 days. This is managed via 90 days consent validity in the CMM.

OAuth2 Server stores the user consents into Consent Management Module (Database). Then the API Management queries the Consent Management Module to check if the PSU Consent for the TPP is already given.

By default, all payment initiations of a TPP require a SCA of the PSU.

4 Consent Management Module

The Consent Management Module (CMM) is necessary for managing consents, for example checking if TPP has PSU consent for accessing a certain account and if the consent is valid. This can be done by the ASPSP (Bank). CMM stores consents into a database.

First, the PSU provides the account data (e.g. IBANs) to the TPP or the TPP request an "Available Accounts Consent" in order get access to the Account details including IBANs, BBANs etc. The TPP will then explicitly request a consent for these specific accounts at once.
Note: it is also possible to grant consent to a single account. But it will be more comfortable for the PSU to request multiple accounts at once as a 2FA is necessary for each consent request. Be aware that by regulation of PSD2 it is not allowed, that the consent requested by the TPP is adapted by PSU or ASPSP during consent creation. This implies, that PSU and TPP need to agree on consent scope before TPP requests the consent.

For each consent creation a Consent-ID will be generated from the CMM and handed over to the TPP. The TPP will need this Consent-ID for each account information request later.
With the Consent-ID the CMM shows the connection between the account of the PSU and the TPP.

It can also be used for disabling a TPP from service (e.g. for a single account or for all accounts).

If a certificate becomes invalid the TPP MM will reject the request of the TPP. At every request the Consent-ID will get validated against the CMM. The CMM will change the status of a consent if it is not valid anymore.

The maximum number of accesses to an account will be determined and communicated by the ASPSP. In each case it will be valid from 0 - 24 hours. The last access time will be saved in the CMM.
The CMM monitors its own status and changes it if necessary, e.g. it sets the consent status to expired.

Consent will be valid for 90 days (obligated by Berlin Group) if no shorter time period is given via consent request from TPP. The consent will automatically expire and can NOT be extended (obligated by Berlin Group).

Consents cannot be changed via Berlin Group API by the TPP. They can only be deleted and newly created.

4.1 Consent iterator

Get accounts	No counting
Get accounts with balance	Count to frequencyPerDay as balance (max. 4)
Get account	No counting
Get account with balance	Count to frequencyPerDay as balance (max. 4)
Get balance	Count to frequencyPerDay as balance (max. 4)
Get transactions	Count to frequencyPerDay as transaction (max. 4)

We do not count for:

- Get accounts
- Get account

because this is the only way for the TPP to get the IBANs.

Balance counter is the one and only counter for:

- Get accounts with balance
- Get account with balance
- Get balance

We count per:
- TPP-ID
- PSU-ID
- IBAN

Info:
We do not count per consent!

If the balance counter = frequencyPerDay (max. 4) then

- Get accounts
- Get account

is still possible! Balance will be empty if balance counter for all accounts = frequencyPerDay. It can happen that Get balance was queried for an account so that this IBAN counter is higher than the IBAN counter. In this case the response will only deliver balance where the frequencyPerDay was not reached. We can still deliver accounts.

For the ASPSP it does not matter how many TPP applications or consents exist. The TPP will simply get the data from the bank until frequencyPerDay (max. 4) per consent is reached.

A TPP can have multiple consents with different frequencyPerDay. If the TPP queries data with a consent A (frequencyPerDay = 1) and the TPP has already queried 2 times with a different consent B (frequencyPerDay = 4) then the request will be denied because IBANCounter (=2) > frequencyPerDay (=1) from Consent A.

4.2 Consent status (Berlin group)

Code	Description
received	The consent data have been received and are technically correct. The data is not authorized yet.
rejected	The consent data have been rejected e.g. since no successful authorization has taken place.
valid	The consent is accepted and valid for GET account data calls and others as specified in the consent object.
revokedByPsu	The consent has been revoked by the PSU towards the ASPSP.
expired	The consent expired.
terminatedByTpp	The corresponding TPP has terminated the consent by applying the DELETE method to the consent resource.

5 Pre-Authentication

Login will be necessary as first step. Berlin Group named this scenario pre-authentication. After pre-authentication (login at ASPSP) TPPs can call Account information, Payment initiation and Funds APIs. We will follow the first pre-authentication approach of Berlin Group ("if OAuth2 has been used as PSU authentication").

A pre-step authentication is used to enable access to the system (login). However, the API calls required for this exist outside the PSD2 API. Hence the name "pre-step". This means that login has to be always the first step. After the login the TPP can request AIS, PIS or PIIS services.

The system therefore offers multiple processes:

- Embedded pre-authentication
- Authorization Code Flow ("GUI")

Please refer to the ASPSP specific part of the documentation which processes are supported.

Embedded approach:

- The PSU provides the TPP his credentials and TAN and the TPP can login via Embedded pre-authentication flow (like embedded approach).

Authorization code flow:

- The login can be either performed by the Authorization code flow (like redirect approach) where the PSU enters the credentials and TAN via the interface at the ASPSP. Thus, the relevant data traffic takes place between PSU and OAuth without having to supply the TPP with confidential information.

As mentioned above the first call is on Pre-Step Authorization API in order to login. However, there is only one exception: if it is the very first call from a TPP, then this call must be against any Berlin Group API endpoint before calling the Pre-Step Authorization API.

After successful pre-step authentication the TPP will receive a session based PSD2 access token. This PSD2 access token is needed for every Berlin Group API call. The TPP must provide the token in the header field "PSD2-AUTHORIZATION".

To consume the API the TPP must add it's PSD2 QWAC certificate in the first call POST psd2-auth/v1/auth/token. A mutual SSL connection between TPP and ASPSP will be established.

The generated psd2-access-token will be valid for a short time (depending on the backend, usually five minutes) like in the online banking. This token will automatically extend its validity if the TPP continues to consume the API. A TPP should be able to process several calls at once if the idle time stays below the token-lifetime.

Using the delete call of the API, a TPP can disconnect the connection if wished. Otherwise it is terminated automatically via the timeout.

5.1 OAuth methods (independent from Berlin Group)

5.1.1 Embedded pre-authentication

The PSU enters its credentials at the TPP interface. The TPP send the credentials then via API to the ASPSP.

POST auth/token → *login request with credentials*

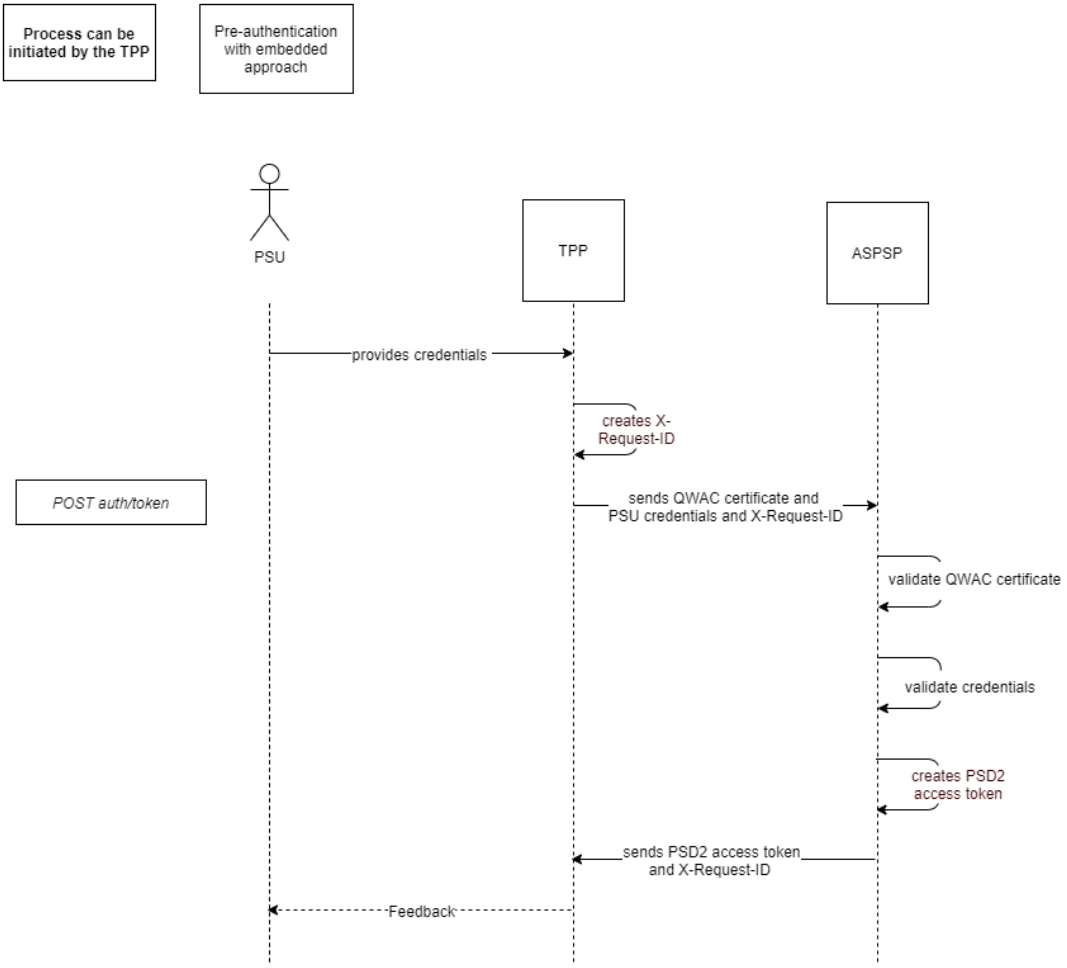

Figure 6 - Embedded pre-authentication

5.1.2 Authorization code flow

Here, the PSU enters its credentials directly at the OAuth server, which significantly increases security. If a SCA is required, it is also mapped via the OAuth server and the TAN recorded there.

5.1.2.1 Authorization code flow - SCA necessary

- Only necessary if SCA have to be performed
- User enters credentials in OAuth GUI
- User can choose device and authentication method (optional functionality)
- User enters TAN in OAuth GUI

POST auth/token → *request without credentials, response with PSD2 authorization code*

GET auth/login → *calls login page for PSU to enter his credentials*

PSU interacts with GUI

POST auth/token → *response with PSD2 token after successful PSU login*

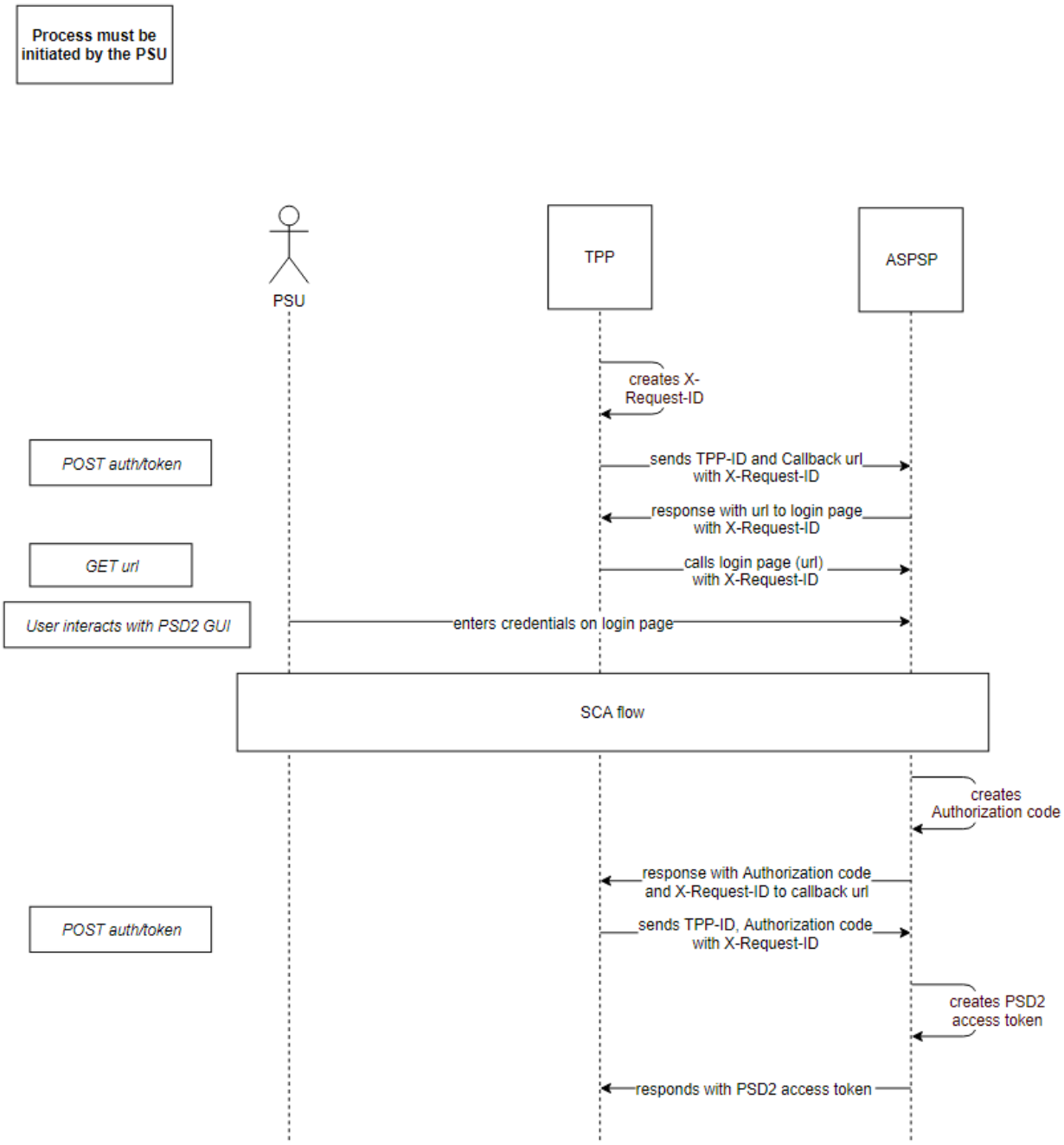

Figure 7 - Pre-auth Authorization code flow (SCA necessary)

The blackbox "SCA flow" is described in chapter *5.1.3*

5.1.2.2 Authorization code flow - SCA not necessary

- not available at first login
- within the exemption period (e.g. 90 days) from SCA.

The TPP has to send the Bank a Callback URL. Callback URL of TPP is necessary to deliver PSD2 access token after successful login.

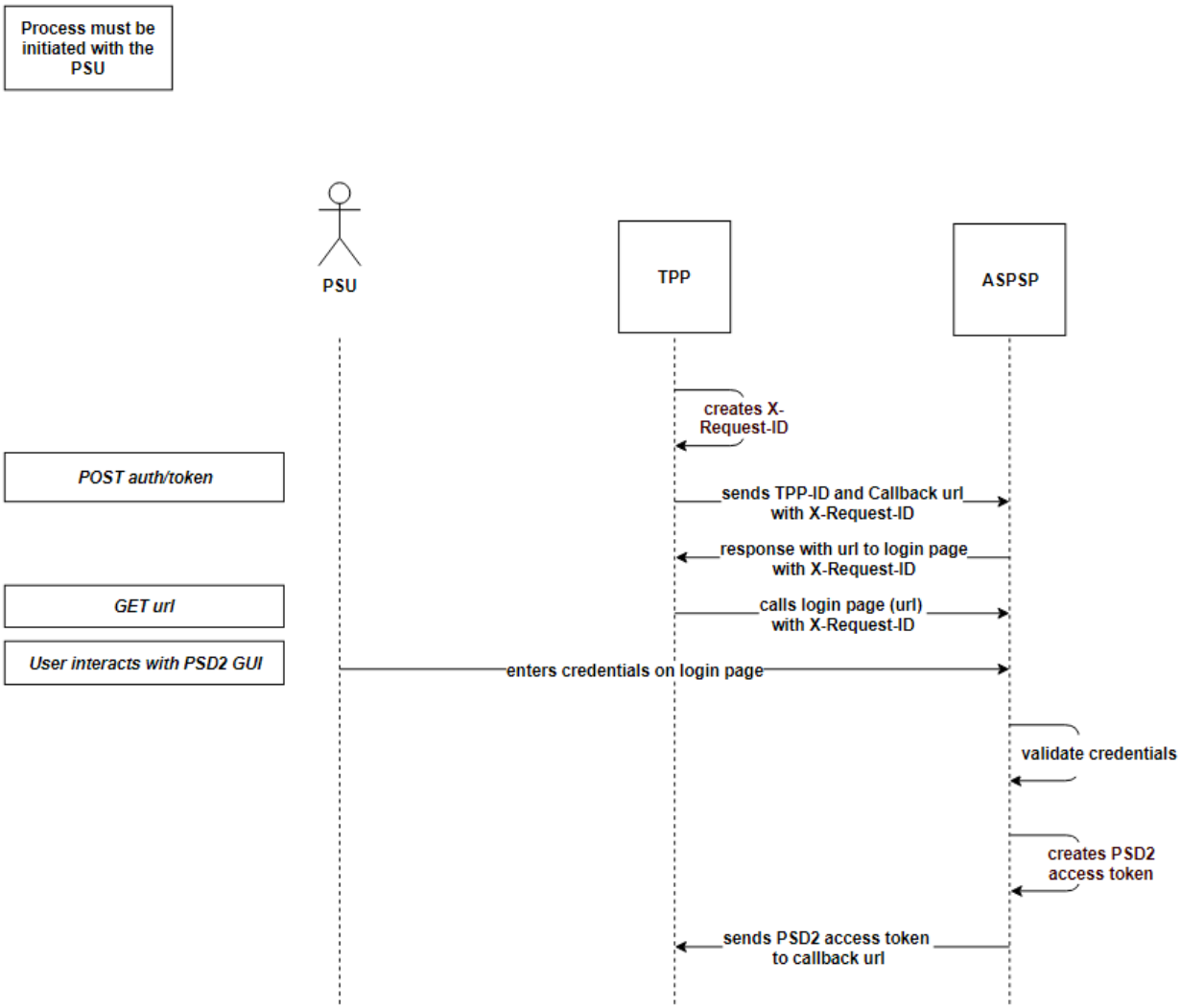

Figure 8 - Pre-auth Authorization code flow (SCA not necessary)

.

5.1.3 SCA flow

This flow shows the execution of the Strong customer authentication.

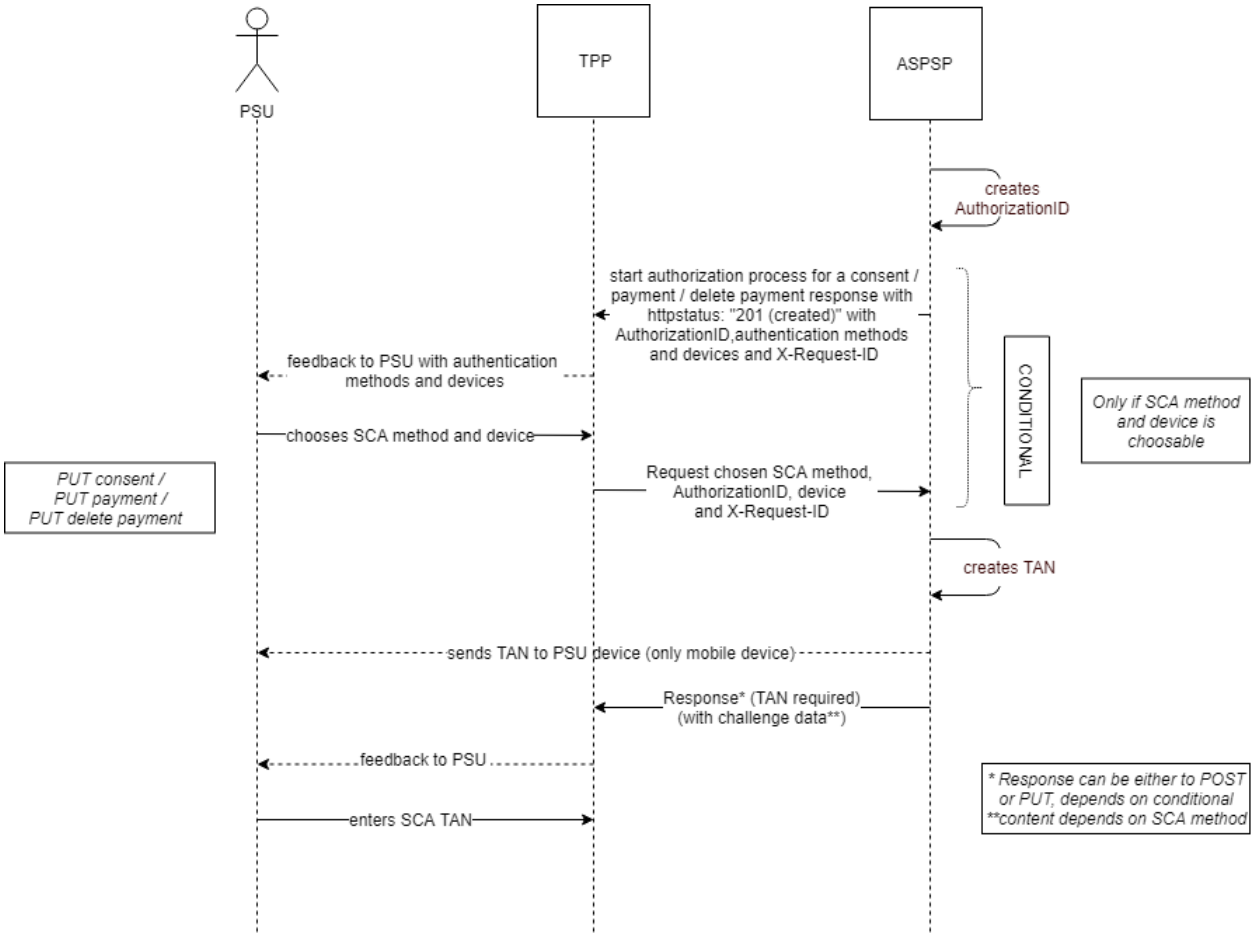

Figure 9 - SCA flow

6 Workflows with Berlin Group API

The following workflows show the process of creating a consent for requesting account information, initiating a payment and funds confirmation via the Berlin Group API.
It is possible to create a detailed consent and global consent according to Berlin Group. Furthermore, the TPP can request a list of available accounts before submitting a detailed consent.

"allPsd2": "allAccounts"

Method	Allowed?
listAccounts	yes
listAccountsWithBalances	yes
readAccountDetails	yes
readAccountDetailsWithBalances	yes
balances	yes
transactions	yes

"availableAccounts": "allAccounts"

Method	Allowed?
listAccounts	yes
listAccountsWithBalances	no
readAccountDetails	yes
readAccountDetailsWithBalances	no
balances	no
transactions	no

"availableAccounts": "allAccountsWithBalances"

Method	Allowed?
listAccounts	yes
listAccountsWithBalances	yes
readAccountDetails	yes
readAccountDetailsWithBalances	yes
balances	yes
transactions	no

Figure 10 - Consent options

6.1 Create Consent

For creating a consent, a SCA will always be necessary.

These are the steps:

Pre-authentication

Then:

POST /v1/consents

POST /v1/consents/{consentId}/authorisations

PUT /v1/consents/{consentId}/authorisations/{authorisationId}

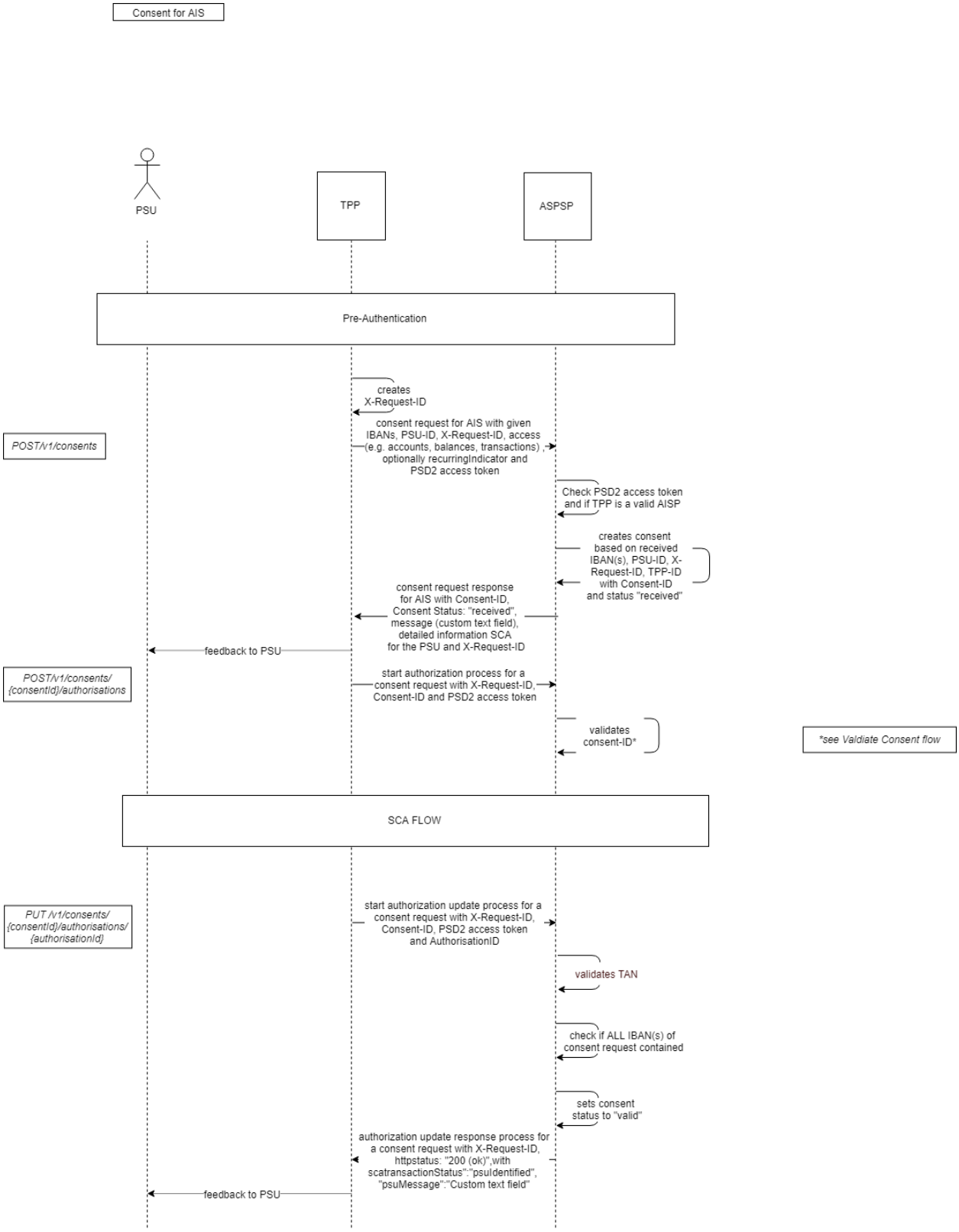

Figure 11 - Consent for AIS

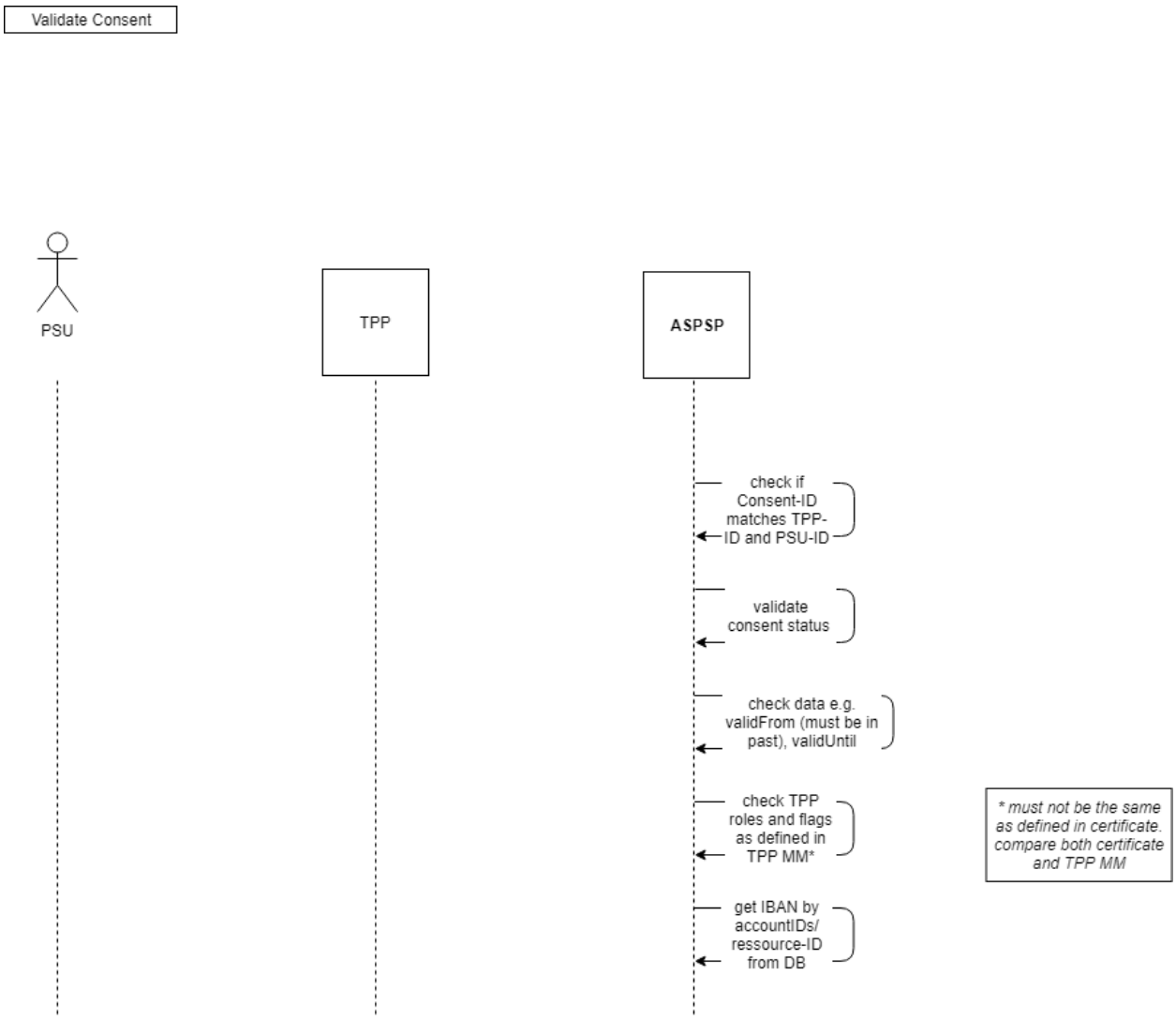

Figure 12 - Validate consent

6.2 Account information

A consent is necessary to perform this AIS request since a consent-ID is needed in the call.

GET/v1/accounts → *initially the first call. But only first time because after the call the account-ids are set. From now on this call is optional. If the field withBalance is missing, the method offers no added value afterwards since the data will always be the same.*

Now the following is possible:

GET/v1/accounts/{account-id}/balances

GET/v1/accounts/{account-id}/transactions/

6.2.1 Get accounts

Here is the field "withBalance" explicitly allowed though it is not a mandatory field.

GET/v1/accounts

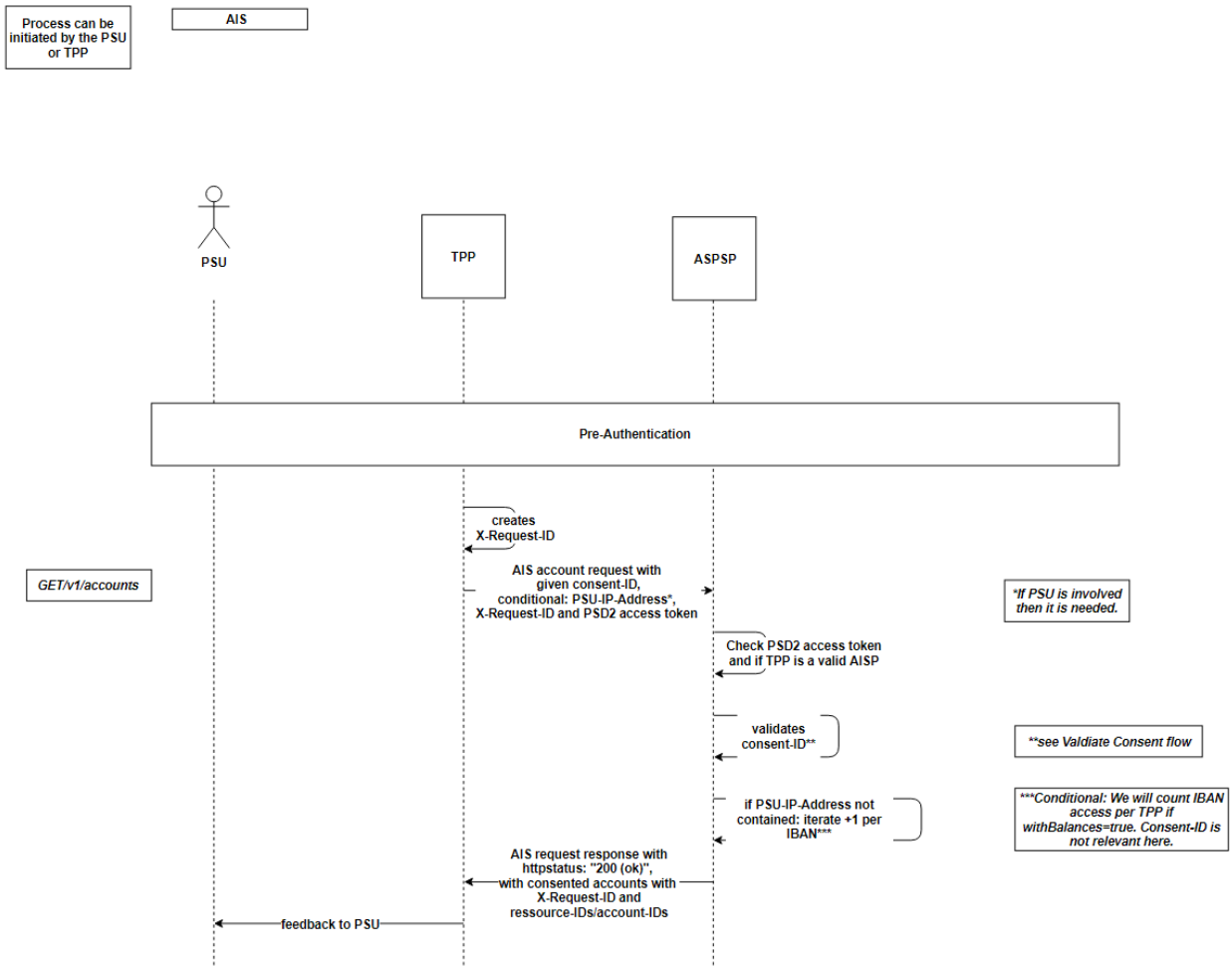

Figure 13 - Get accounts

6.2.2 Get Balances

GET/v1/accounts/{account-id}/balances

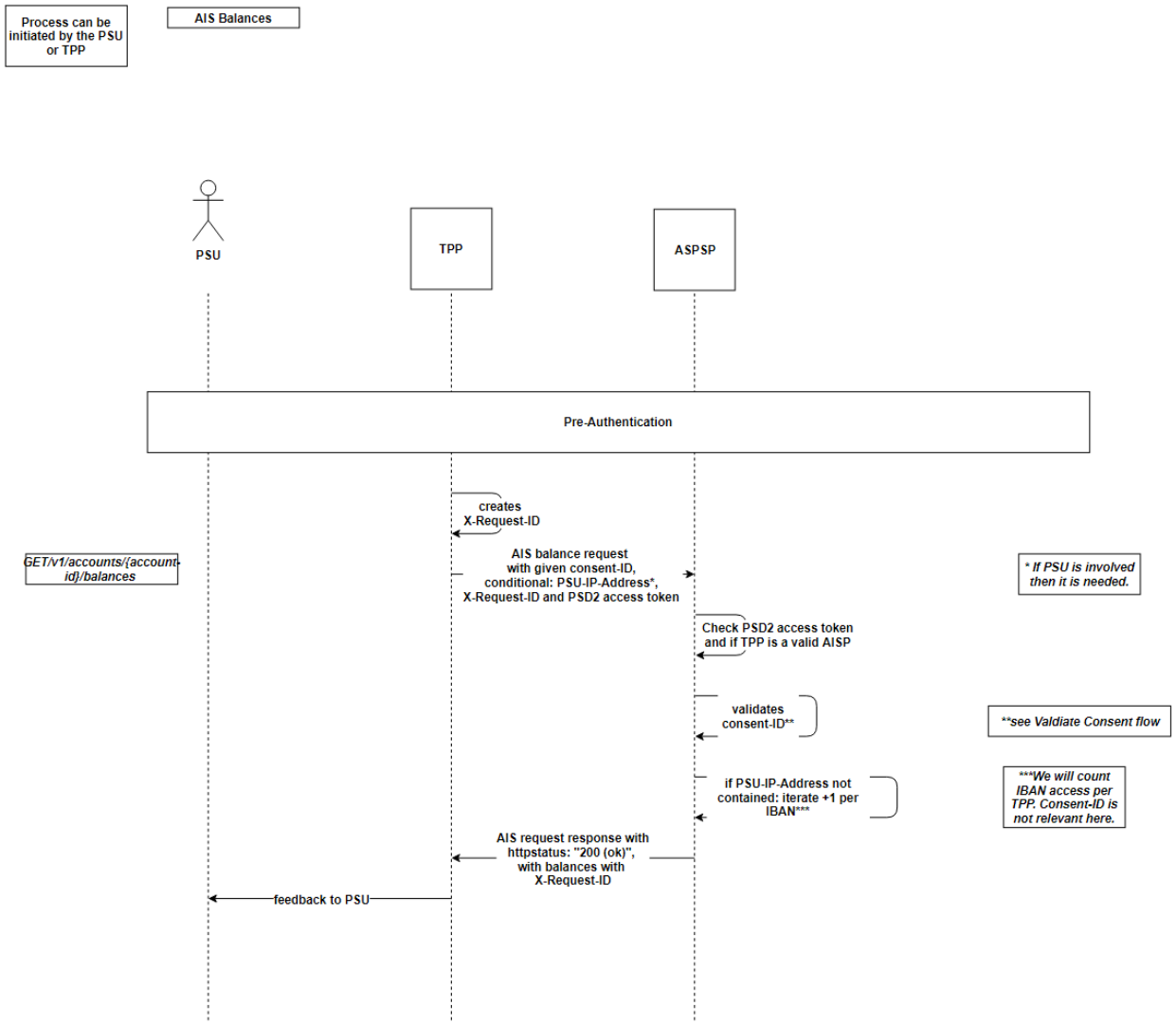

Figure 14 - Get Balances

6.2.3 Get Transactions

GET/v1/accounts/{account-id}/transactions/

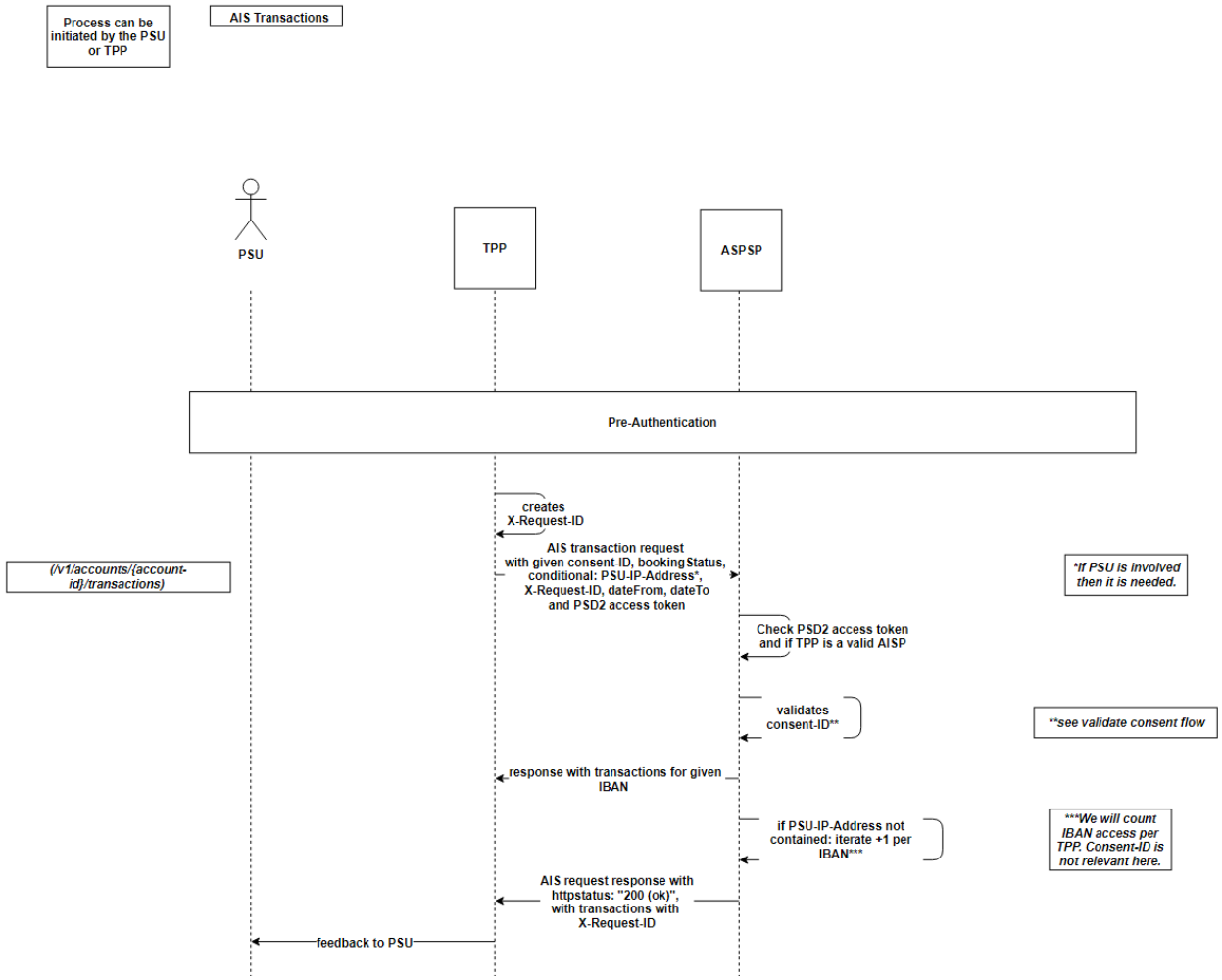

Figure 15 - Get transactions

6.2.4 Get Transactions older 90 days

SCA is necessary if transactions are older than 90 days. If data is requested that is older than 90 days, then a tx_consentId must be transmitted. This tx_consentId will be in the response of the Transactions request. Therefore, the tx_consentId must be used instead of the common consentId. Then this request must be confirmed via SCA. After confirmation the request will be possible. This tx_consentId is unique and is only relevant for this specific request.

6.3 Payments

There will always be a SCA necessary for payments.

6.3.1 Payment initiation

First step: Pre-Step Authentication

POST/v1/{payment-service}/{payment-product} → *initiate payment*

POST/v1/{payment-service}/{paymentId}/authorisations

Last step: SCA flow

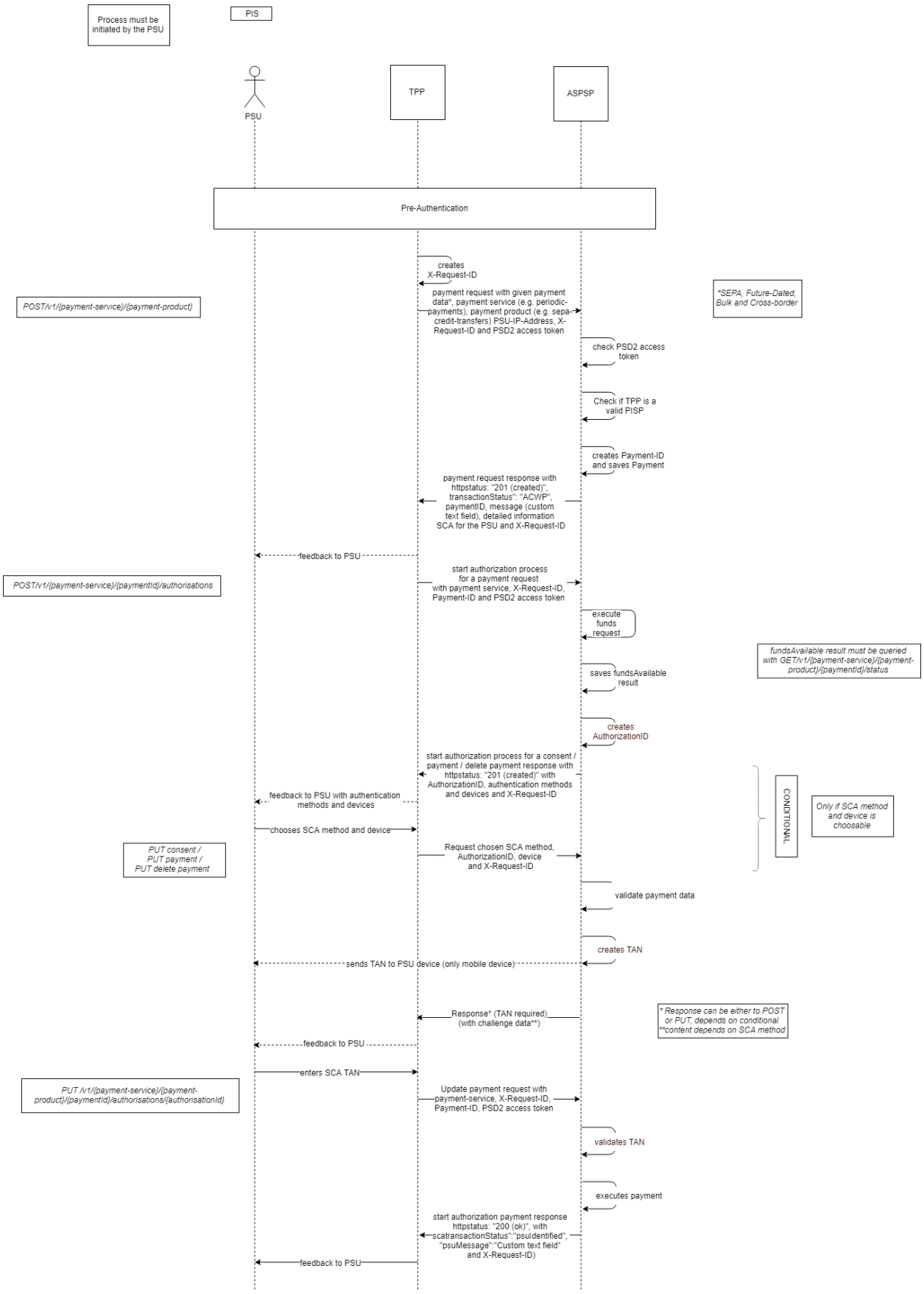

Figure 16 - Payment initiation

6.3.2 Get Payment

Returns the content of a payment object.

First step: Pre-Step Authentication

GET/v1/{payment-service}/{payment-product}/{paymentId}

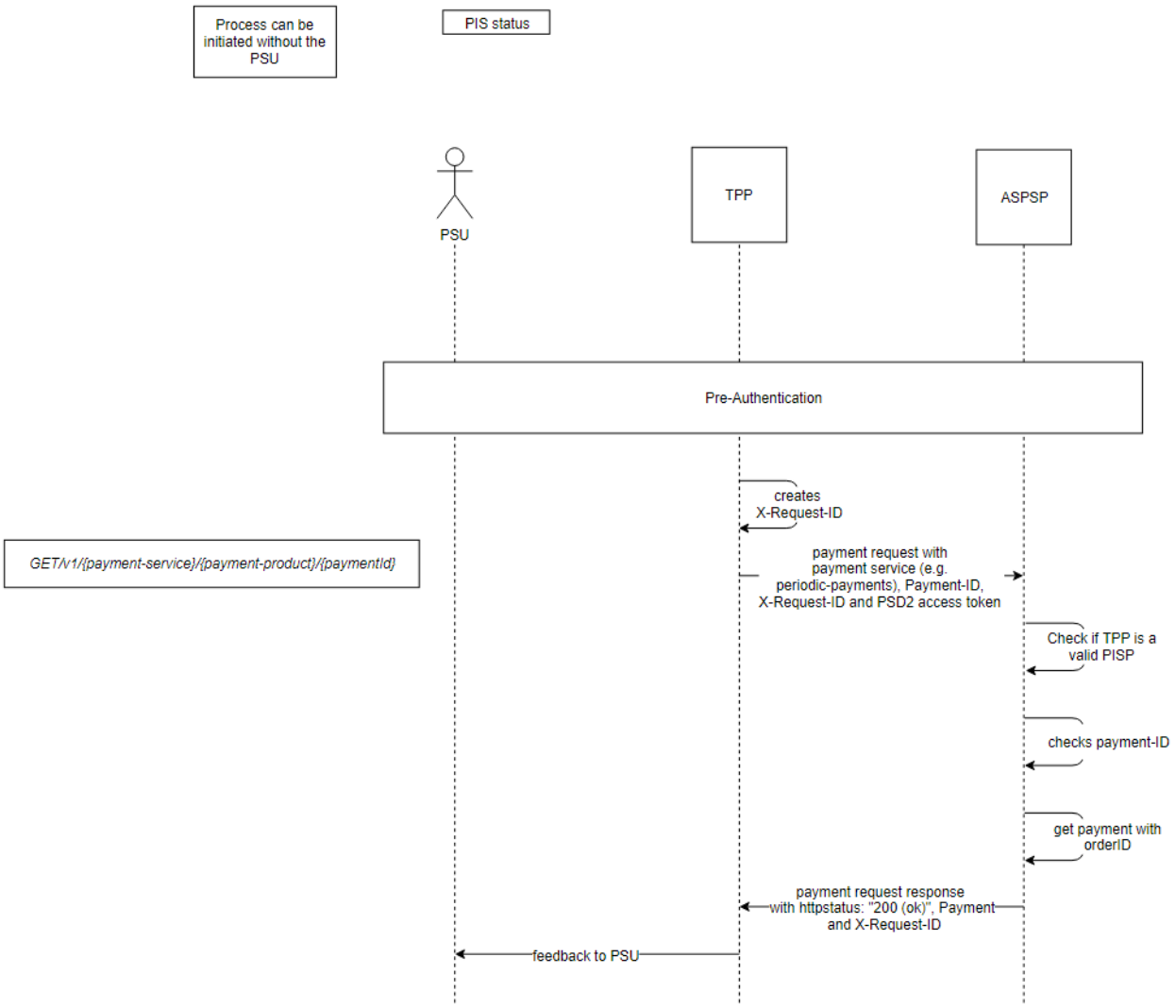

Figure 17 - Get Payment

6.3.3 Get payment status

First step: Pre-Step Authentication

GET/v1/{payment-service}/{payment-product}/{paymentId}/status

A TPP with the role PIS can use this method to get the funds status of the payment.

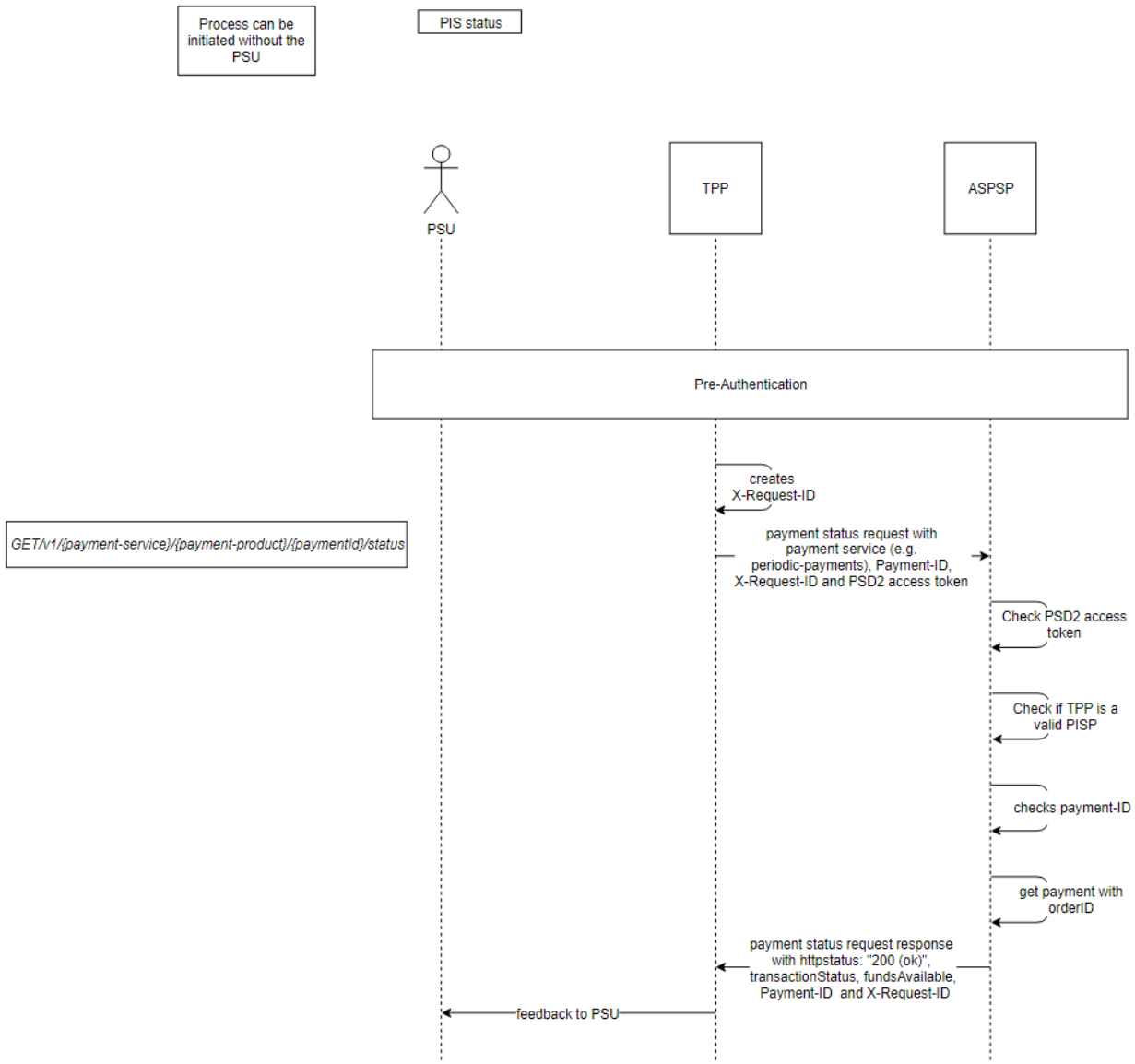

Figure 18 - Get Payment status

6.4 Funds confirmation

Consent will be created by ASPSP like suggested from Berlin Group. TPP has to be registered at API store and provide certificate as the first step. Then the TPP can ask the PSU to request a consent for his IBAN. The PSU will then grant consent at ASPSP with the unique TPP-ID (Authorization number) of the TPP. The TPP provides the PSU the TPP-ID in advance.

No Consent-ID is transported to TPP. Because for the funds request the TPP can NOT enter a consent-ID.

Process must be initiated by the TPP

Consent for PIIS

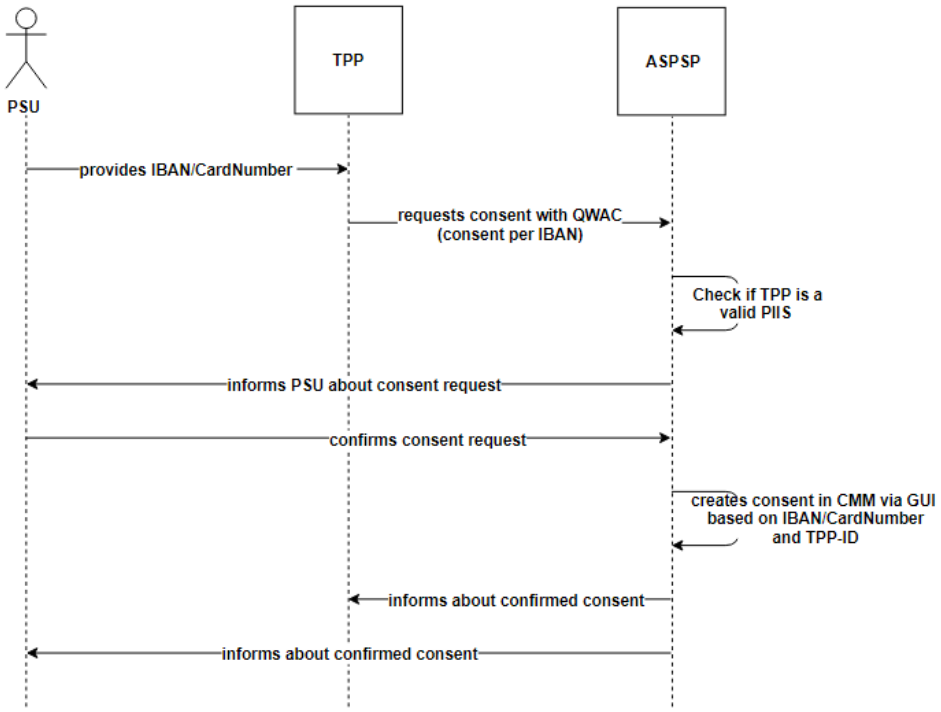

Figure 19 - Consent for PIIS

First step: Pre-Step Authentication

POST /v1/funds-confirmations → *Get "Yes" or "No" answer*

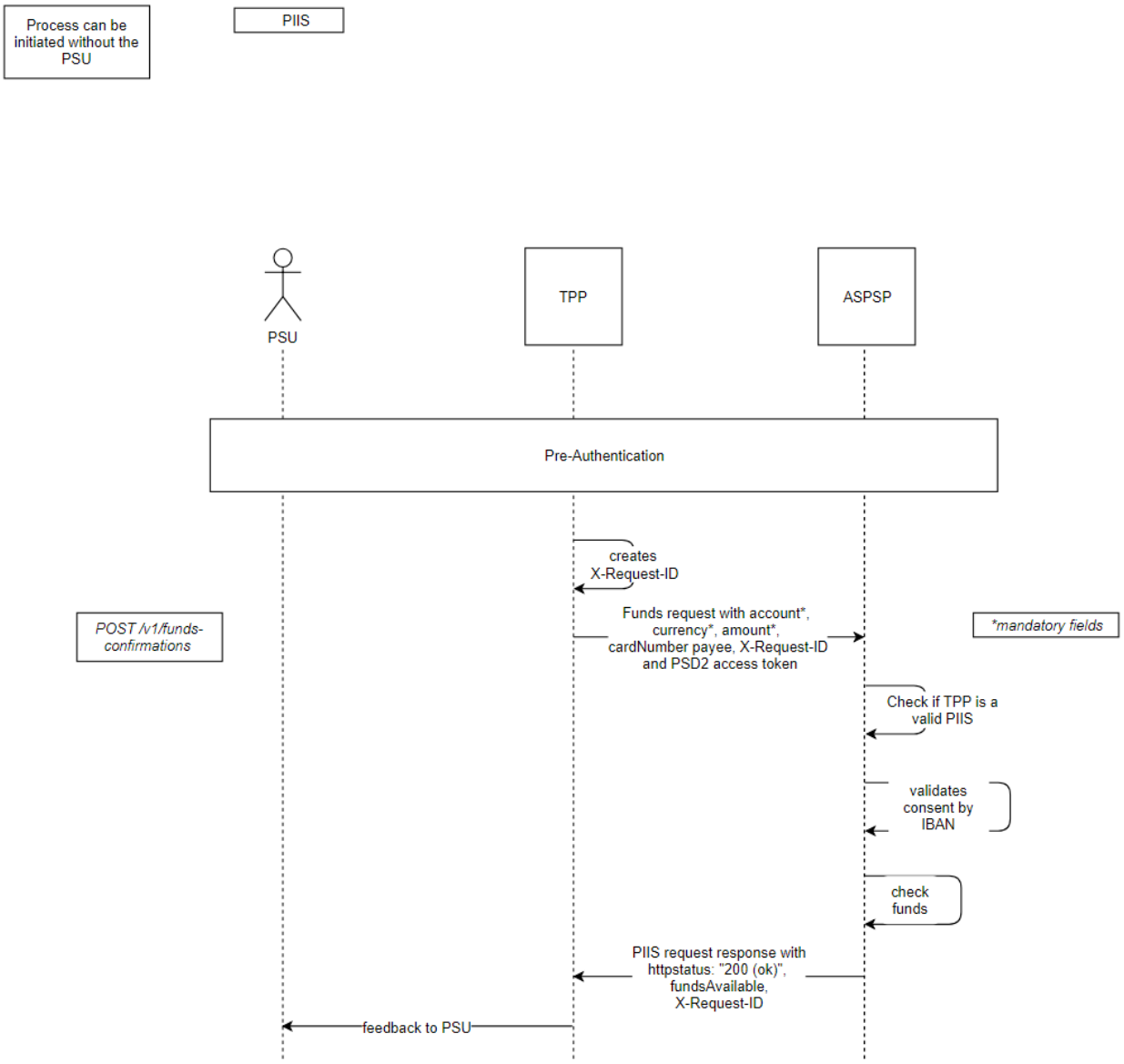

Figure 20 - Funds confirmation

Differences to Consent for AIS

- only 1 IBAN/CardNumber possible for Consent
- ASPSP (Bank) creates consent via CMM GUI
- no expiration date

6.5 New Berlin Group Features

6.5.1 Standing orders

We implemented standing orders via Transaction endpoint. Please refer to Berlin Group Documentation.

6.5.2 Account owner

We use first approach of Berlin Group: “An ASPSP may deliver the account owner name without any extension to the consent model defined above.” Therefore, no special consent is needed.

Account owner will only be delivered for AIS purpose. Account owner will not be delivered with consent on available accounts

7 Comply only (MVP)

For the PSD2 comply only solution that is defined as Minimum Viable Product we offer a shortened Berlin Group API as it is not necessary/possible to offer all endpoints and fields that are defined in Berlin Group. We can only offer functions that are contained in their specific online banking.

7.1 Not included in Berlin Group API

Item	Description
No Signing baskets see also → Signing Baskets	NOT neccessary according to EBA/NA (BaFin) big overhead for PSD2 Solution • e.g. combination of different payment products e.g. a combination of periodic payment and single payment or SEPA und SWIFT → this will be maybe a problem for the core banking or connection to the core banking like FinTS and Group API (also the Development of the GroupAPI) • e.g. matching payment IDs to signing basket • e.g. multiple consents
no Optional fields of Berlin Group API	we will simply ignore these fields in a request
no instant-sepa-credit-transfers	
no target-2-payments	
no pain.001-instant-sepa-credit-transfers	
no pain.001-target-2-payments	
no bbans, card-numbers etc.	ONLY IBANs
no multiple SCA authorisation in a corporate context	
no combined service indicator	will be ignored with FinTS / Group API
GET /v1/accounts/{account-id}/transactions/{resourceId}	No added value to GET /v1/accounts/{account-id}/transactions/. Problem is that every transactions needs an id. In best case that must be provided by the backend of the bank. How to send via FinTS/GroupAPI?
transactionID	for the reason see above

Figure 21 - Not included (Berlin Group)

7.2 Not included endpoints Berlin Group API

Account Information Service (AIS)

Typ	Call	Description
get	/v1/card-accounts	not MVP
get	/v1/card-accounts/{account-id}	not MVP
get	/v1/card-accounts/{account-id}/balances	not MVP
get	/v1/card-accounts/{account-id}/transactions	not MVP

Signing Baskets

Typ	Call	Description
post	/v1/signing-baskets	not MVP
get	/v1/signing-baskets/{basketId}	not MVP
delete	/v1/signing-baskets/{basketId}	not MVP
get	/v1/signing-baskets/{basketId}/status	not MVP
post	/v1/signing-baskets/{basketId}/authorisations	not MVP
get	/v1/signing-baskets/{basketId}/authorisations	not MVP
put	/v1/signing-baskets/{basketId}/authorisations/{authorisationId}	not MVP
get	/v1/signing-baskets/{basketId}/authorisations/{authorisationId}	not MVP

Figure 22 - Not included endpoints

Note: There is still some discussion about what it is contained and what is not contained in the Berlin Group API and in the comply only MVP in general. This can change later.

7.3 Not included in general

Item	Description
Qualified Electronic Seal Certificates (QSealC)	
Restrictionon specific account types (e.g. cash accounts in general, but no saving accounts)	Comment: CLX delivers no functionality to restrict account-types
SCA exemption for small amounts	For payments always SCA
Whitelist	MVP does not have a whitelist for trusted payees. SCA always neccessary!

Figure 23 - Not included in general

8 References

Description	Hyperlink
Short introduction to PSD2 by Berlin Group Initiative	https://docs.wixstatic.com/ugd/c2914b_c6a8a0dca83e4af8859be266415d3d79.pdf
Directive (EU) 2015/2366 of the European parliament and of the council on payment services in the internal market (PSD2) of 25 November 2015	English: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32015L2366 German: https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:32015L2366
Regulatory Technical Standards on strong customer authentication and secure communication under PSD2 (RTS)	English: https://www.eba.europa.eu/regulation-and-policy/payment-services-and-electronic-money/regulatory-technical-standards-on-strong-customer-authentication-and-secure-communication-under-psd2
Commision delegated regulation (EU) 2018/389 of 27 November 2017 supplementing Directive (EU) 2015/2366 of the European Parliament and of the Council with regard to regulatory technical standards for strong customer authentication and common and secure open standards of communication	English: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2018:069:TOC German: https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=OJ:L:2018:069:TOC
Consultation on RTS specifiying the requirements on strong customer authentication and common and secure communication under PSD2	English: https://www.eba.europa.eu/regulation-and-policy/payment-services-and-electronic-money/regulatory-technical-standards-on-strong-customer-authentication-and-secure-communication-under-psd2/-/regulatory-activity/consultation-paper
Discussion on RTS on strong customer authentication and secure communication under PSD2	https://www.eba.europa.eu/regulation-and-policy/payment-services-and-electronic-money/regulatory-technical-standards-on-strong-customer-authentication-and-secure-communication-under-psd2/-/regulatory-activity/discussion-paper
EBA Fallback document	https://eba.europa.eu/-/eba-publishes-final-guidelines-on-the-exemption-from-the-fall-back-mechanism-under-the-rts-on-sca-and-csc

NextGenPSD2 Access to Account Interoperability Framework (Berlin Group Standard) • Documentation • Technial documentation / API description • OpenAPI File	https://www.berlin-group.org/nextgenpsd2-downloads

9 Glossary

PSD2 abbreviation	Meaning	Usage
2FA	Two Factor Authentication	
AIS	Account Information Service according to article 4 (16) of [PSD2] and as regulated by article 67 of [PSD2].	This service may be used by an AISP to request information about the account of a PSU. The account is managed by the ASPSP providing the XS2A Interface. Functionality and restrictions of this service comply with the requirements defined by article 67 of [PSD2].
AISP	Account Information Service Provider offering an AIS to its customer. See article 4 (19) of [PSD2].	
ASPSP	Account Servicing Payment Service Provider providing and maintain a payment account for a payer. See article 4 (17) of [PSD2]. For example a bank.	
FCS	Fund confirmation service	This service may be used by a PIISP to request a confirmation of the availability of specific funds on the account of a PSU. The account is managed by the ASPSP providing the XS2A Interface. Functionality and restrictions of this service comply with the requirements defined by article 65 of [PSD2].
eIDAS	**e**lectronic **ID**entification, **A**uthentication and trust **S**ervices is an EU regulation on electronic identification and trust services for electronic transactions in the internal market. It is a set of standards for electronic identification and trust services for electronic transactions in the European Single Market. It was established in EU Regulation 910/2014 of 23 July 2014 on electronic identification and repeals directive 1999/93/EC from 13 December 1999.	
MVP	Minimum Viable Product	Focus on scope in agile development
NA/NCA	National (Competent) Authority. Holds a list of TPPs registered in that particular country.	
PIS	Payment Initiation Service according to article 4 (15) of [PSD2] and as regulated by article 66 of [PSD2].	This service may be used by a PISP to initiate a single payment on behalf of a PSU using a given

		account of that PSU. The account is managed by the ASPSP providing the XS2A Interface. Functionality and restrictions of this service comply with the requirements defined by article 66 of [PSD2].
PISP	Payment service provider offering a PIS to its customer. See article 4 (18) of [PSD2].	
PIISP	Payment Instrument Issuer Service Provider according to article 4 (14) and 45) of [PSD2]. A PIISP can use the service "Confirmation on the availability of funds" as regulated by article 65 of [PSD2].	
PSU	Payment Service User according to article 4 (10) of [PSD2].	
QTSP	Qualified Trust Service Provider, e. g. a trust centre issuing qualified certificates. German: Vertrauensdiensteanbieter (eIDAS)	
SCA	Strong Customer Authentication – authentication procedure based on two factors compliant with the requirements of [PSD2] and [EBA-RTS].	
TPP	Third Party Provider – generic term for AISP/PIISP/PISP.	
TSP/QTSP	Trust Service Provider according to [eIDAS]. Within the context of the XS2A interface specification only qualified TSPs (QTSPs) according to section 3 of [eIDAS] issuing qualified certificates for electronic seals and/or qualified certificates for website authentication which are compliant with the requirements of [EBA-RTS] are relevant.	
XS2A	Access to account interface – interface provided by an ASPSP to TPP for accessing accounts.	
QSealC	Qualified Electronic Seal Certificates	
QWAC	Qualified Website Authentication Certificates	