PSD2 API Solution - Documentation for TPPs

Addressees:

Business Analysts, Project Managers, Developers, Architects, IT

Authors:

David Schneider, Lars Kieffer, Gerald Haase

Version, date

Version 1.0, March 13th, 2019

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 Introduction

This document describes how TPPs can connect the PSD2 Solution of Aareal Bank.

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 Aareal Bank API solution to connect their services. A TPP can sign up at the integrated API Management tool in order to make use of the Aareal Bank public API. After login the TPP can subscribe to respective API. Aareal Bank will rely on NextGenPSD2 *Access to Account Interoperability Framework* specified by *The Berlin Group* version 1.3. The subscription is necessary to allow TPPs to consume the API. This process is explained in this document.
The Aareal Bank 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 postings 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.

In the first rollout of the sandbox (14th of march, 2019) there will not be any certificate checks because many trust centers like D-Trust (Bundesdruckerei, Germany) currently (March 2019) issue only demo certificates and do not offer any interfaces to validate these certificates. This, and other features like One Time Password checks, will be added in a later stage of the Aareal Bank API solution. For more details on this see Sandbox description.

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 first rollout of the sandbox but on a later stage. 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). 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.

Additionally, 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 and subscribe to certain API packages relevant for PSD2.

2.2 Using the sandbox

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

2.2.1 Workflow

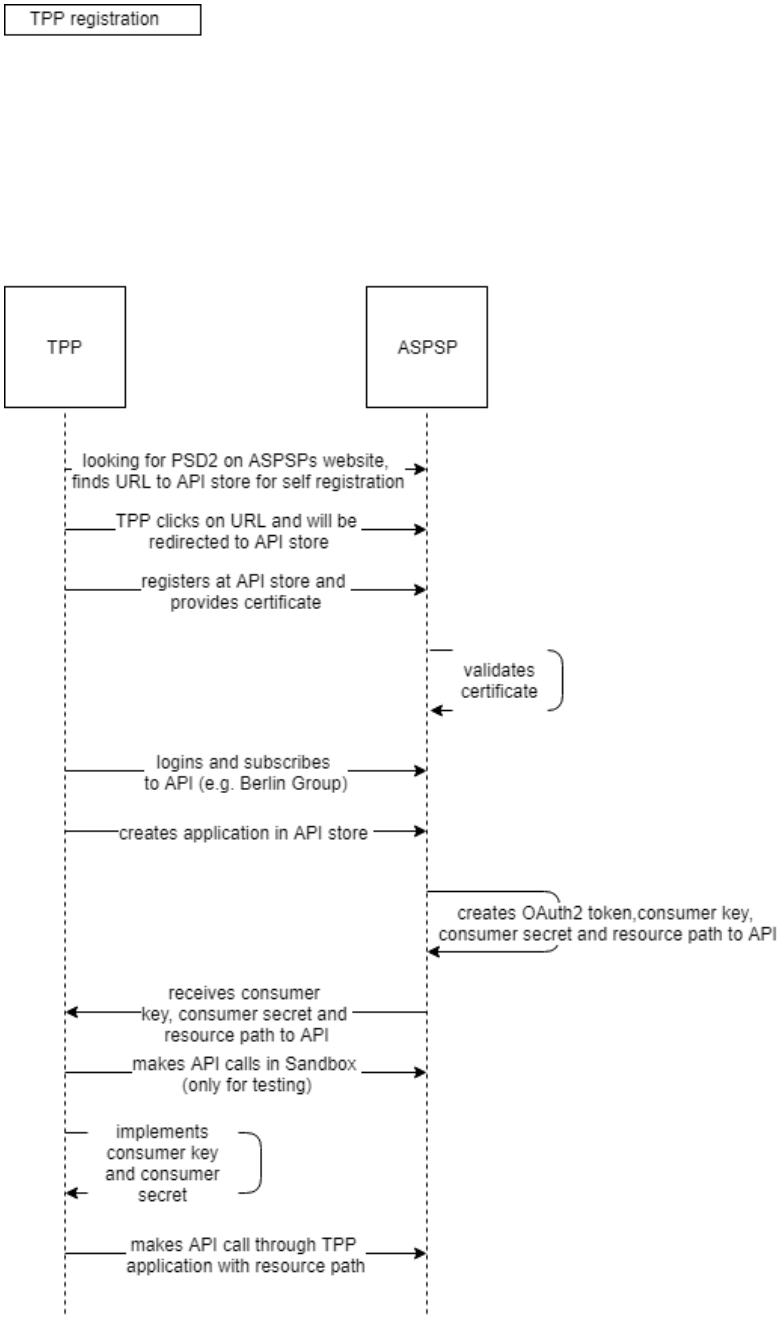

Figure 1 - Workflow TPP registration

2.2.2 Register and Login

To be able to work in the API Store at all, the system has to know the TPPs. This requires a registration, as you know it from other web sites.
To register, please click on the "Sign-up" button at the top of the page. A form will open in which you can enter the necessary data.

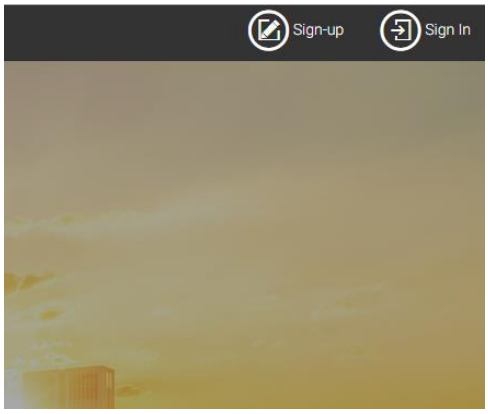

Figure 2 - Register and Login

After the registration has been completed, you can log in to the application. To do this, please use the "Sign-In" button at the top of the page.
You will now be asked for your login data and will then be taken to the system.

2.2.3 Creating application

Afterwards you have to select the application you want to use in order to "consume" it. This application practically serves as a storage location for the later API.
At the top left of the window you will find the menu. Under "API CONCEPTS" it also contains a more detailed description of the API.

Please click on the text "APPLICATIONS".

Figure 3 - Create application

Now another area opens in which you already see an entry "DefaultApplication". This application has already been automatically added for you during the registration process.

Please add your first own application via the button "ADD APPLICATION".

You will now be guided through the installation with the following dialog. First of all create a name for the application a meaningful name such as "BankingApp". Enter this name in the upper field "Name". The "Add" button completes the process and the application is ready. You can now open the newly created "BankingApp".

Figure 4 - Add application

2.2.4 Generate key

In order to communicate with the API, the system needs keys, here called "Sandbox Keys". These keys have to be generated. Please open your created "BankingApp" and switch to the page "Sandbox Keys".

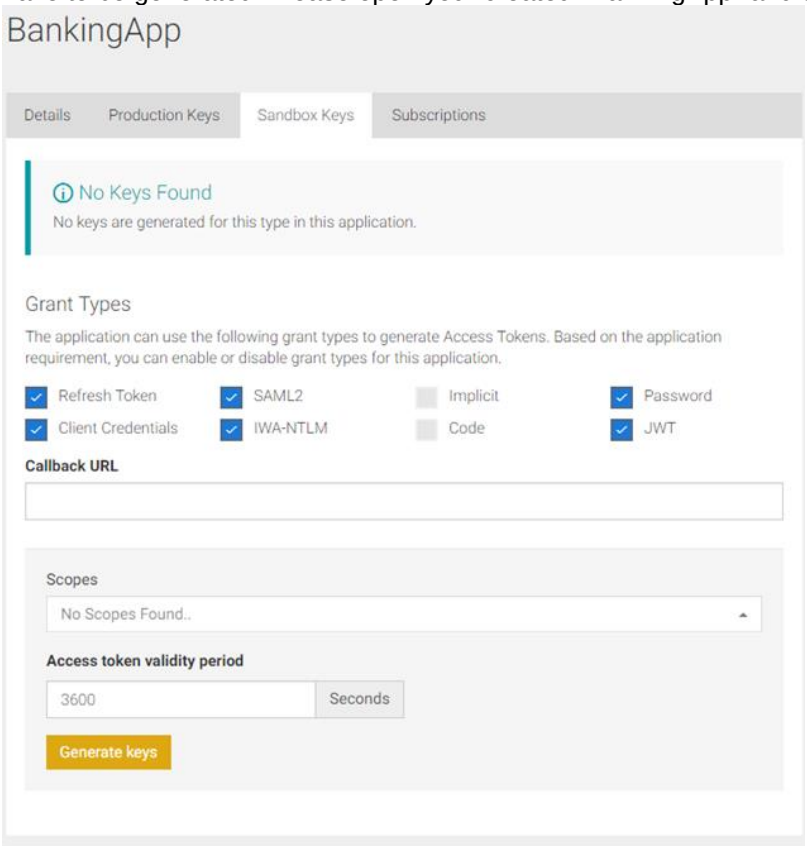

Figure 5 - Generate key

The button "Generate Keys" initiates the generation. Please click on the button now. You do not need to make any changes to the settings above.

You can now view the newly created keys on the page.
They are:

- Consumer Key
- consumer secret
- access token

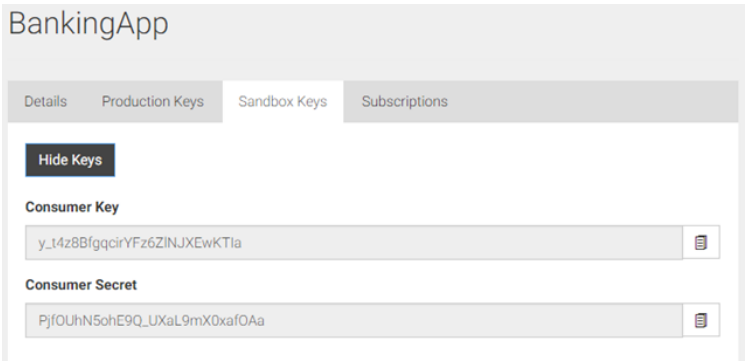

Figure 6 - Keys

You need the Access Token to test the API now.

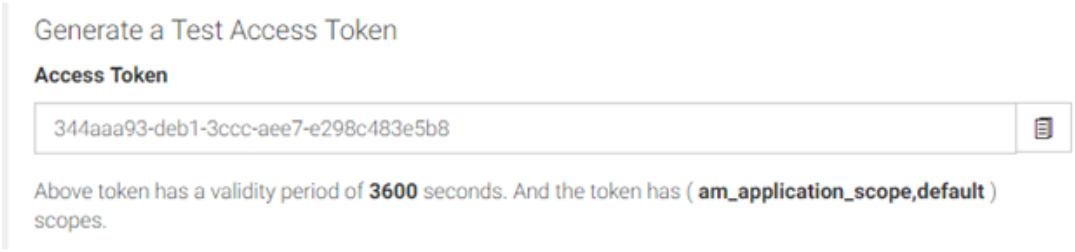

Figure 7 - Access token

2.2.5 Subscribe to API

You must subscribe to the API before you can consume it. This process also only takes place one time. Please click on "APIs" in the left side menu.

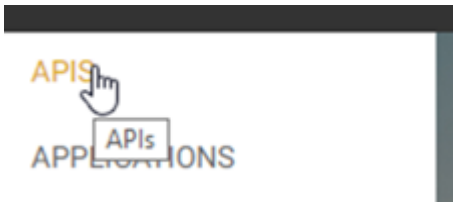

Figure 8 - API

Select the XS2A-NextGenPSD2BerlinGroup API by clicking on the blue link below the colored icon.

Figure 9 - Click API

Now you have to select the storage location or the application "BankingApp" and confirm the process with the button "Subscribe".

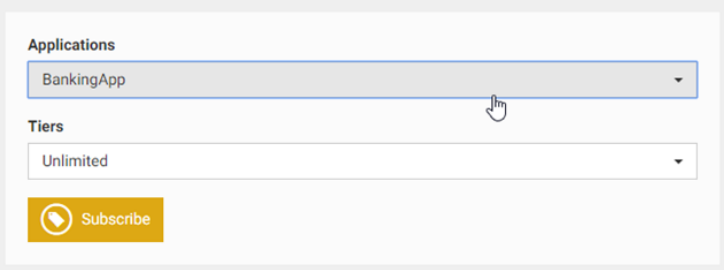

Figure 10 – Subscribe

Please close the success message by clicking on the button "Stay on this page".

2.3 Trying it out

Now you can start testing the API. Switch to the "API Console" tab and select the "Sandbox". This is the name of our test environment.

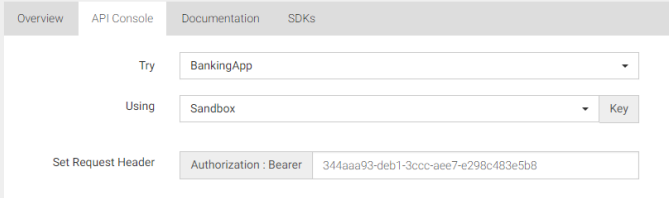

Figure 11 - API console

There you will also find the necessary Access Token.

Scroll down the page, for example to the point "/v1/accounts".

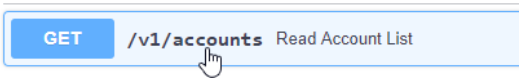

Figure 12 - Get accounts 1/2

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.

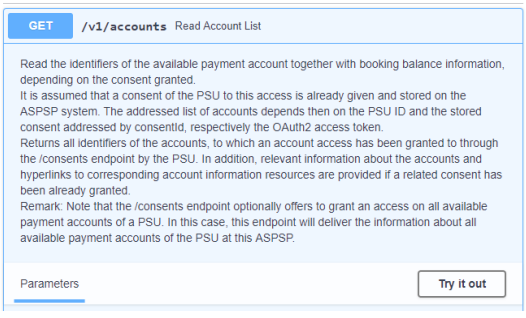

Figure 13 - Get accounts 2/2

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".

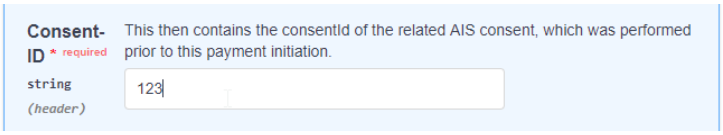

Figure 14 - Consent-ID

If all 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.

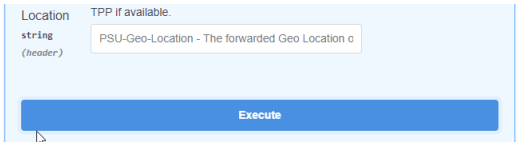

Figure 15 - Execute

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

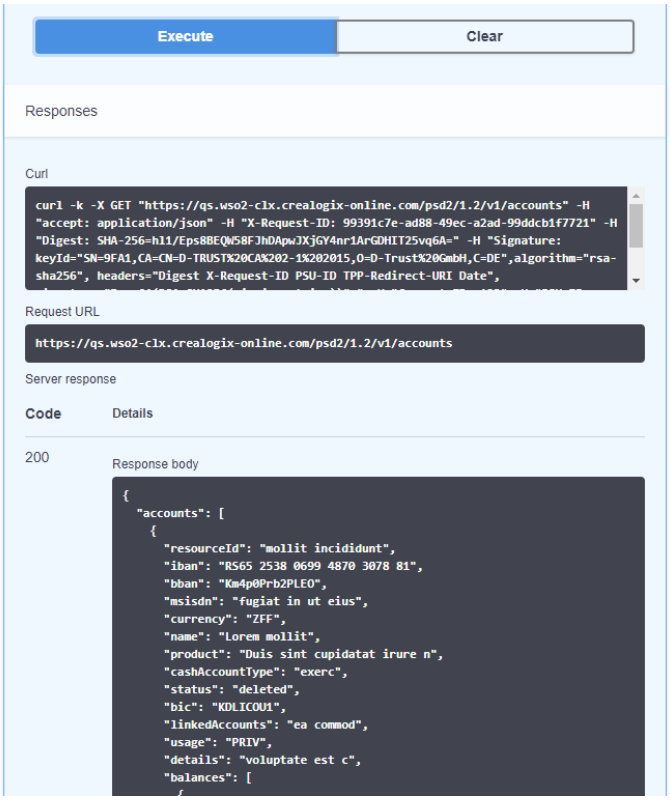

Figure 16 - Response

3 Pre-Authentication

We will follow the pre-authentication approach of Berlin Group.

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")
- Password Credential Flow ("API")

All flows can be used in parallel. For example, the TPP can therefore first try the Password Credential Flow without PSU activity and, if necessary, revert to the Authorization Code Flow. If the TPP logins the first time the TPP will need the Authorization code flow. **Please refer to the ASPSP specific part of this documentation which processes are supported.**

The login can be performed in three ways:

- The PSU provides the TPP his credentials and TAN and the TPP can login via Embedded pre-authentication flow (like embedded approach).
- Or 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 Aareal Bank. Thus, the relevant data traffic takes place between PSU and OAuth without having to supply the TPP with confidential information.
- The PSU provides the TPP his credentials and the TPP can login via standard OAuth procedure Password credentials flow (like embedded redirect). TAN cannot be provided here!

The Embedded pre-authentication flow and conditionally the Password credentials flow makes it possible for the TPP to request AIS or PIIS requests without PSU involvement.

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”.

3.1 OAuth methods (independent from Berlin Group)

3.1.1 Embedded pre-authentication

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

POST auth/token → *login request with credentials*

POST challenge (conditional) → *send chosen SCA methods and device for*

PUT challenge → *send TAN, response with PSD2 token after successful PSU login*

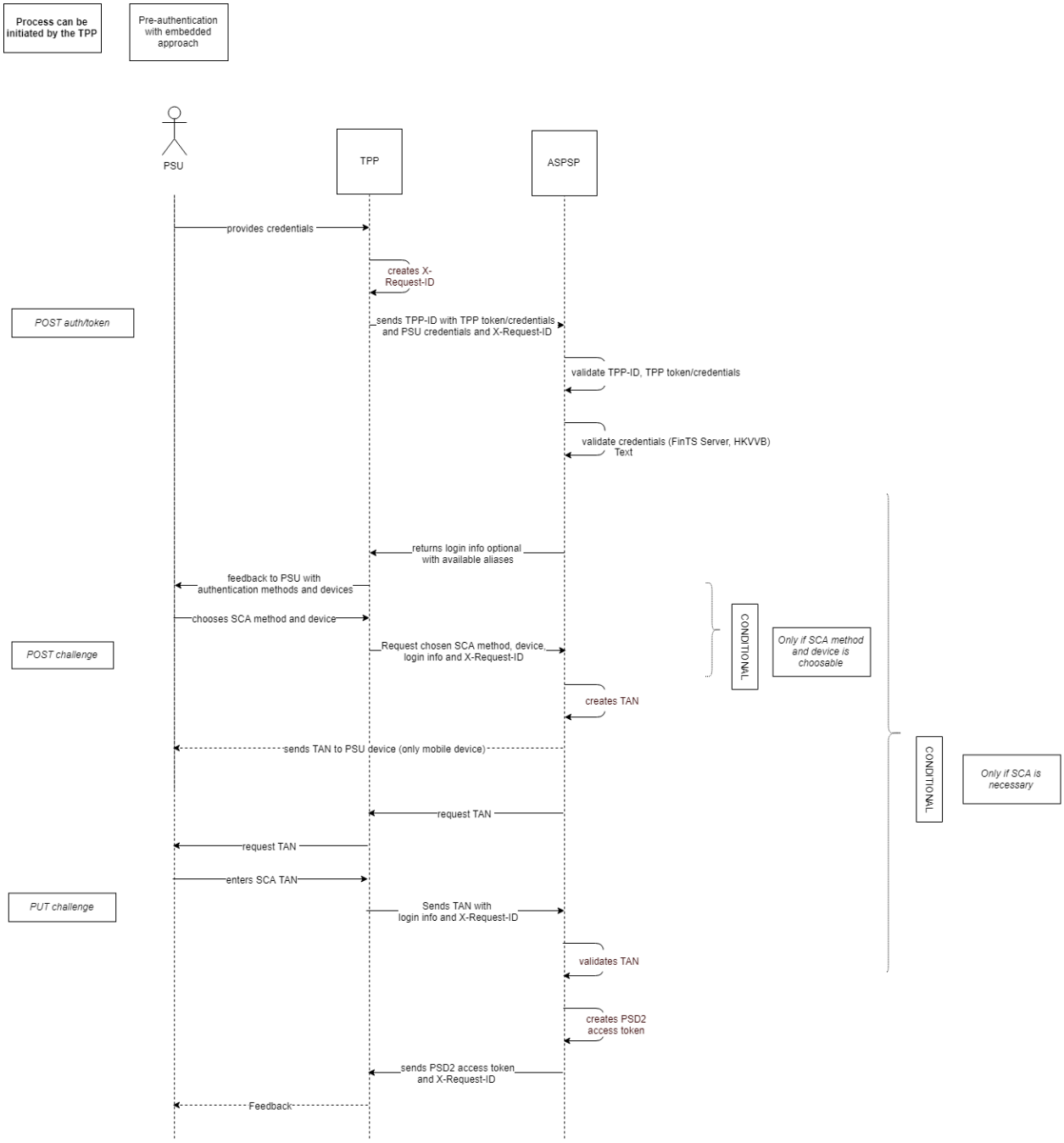

Figure 17 - Embedded pre-authentication

3.1.2 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.

3.1.2.1 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.

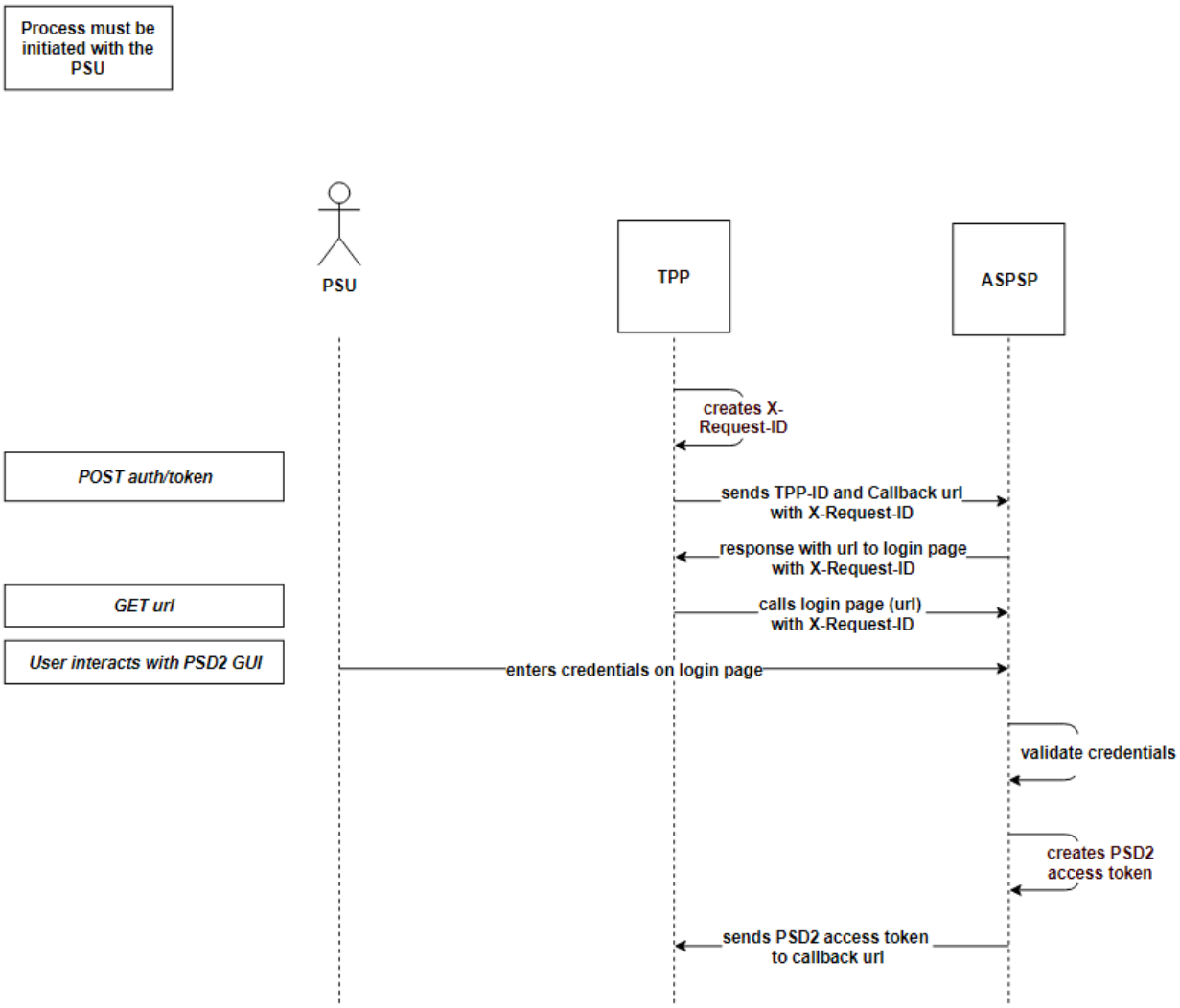

Figure 18 - Pre-auth Authorization code flow (SCA not necessary)

3.1.2.2 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*

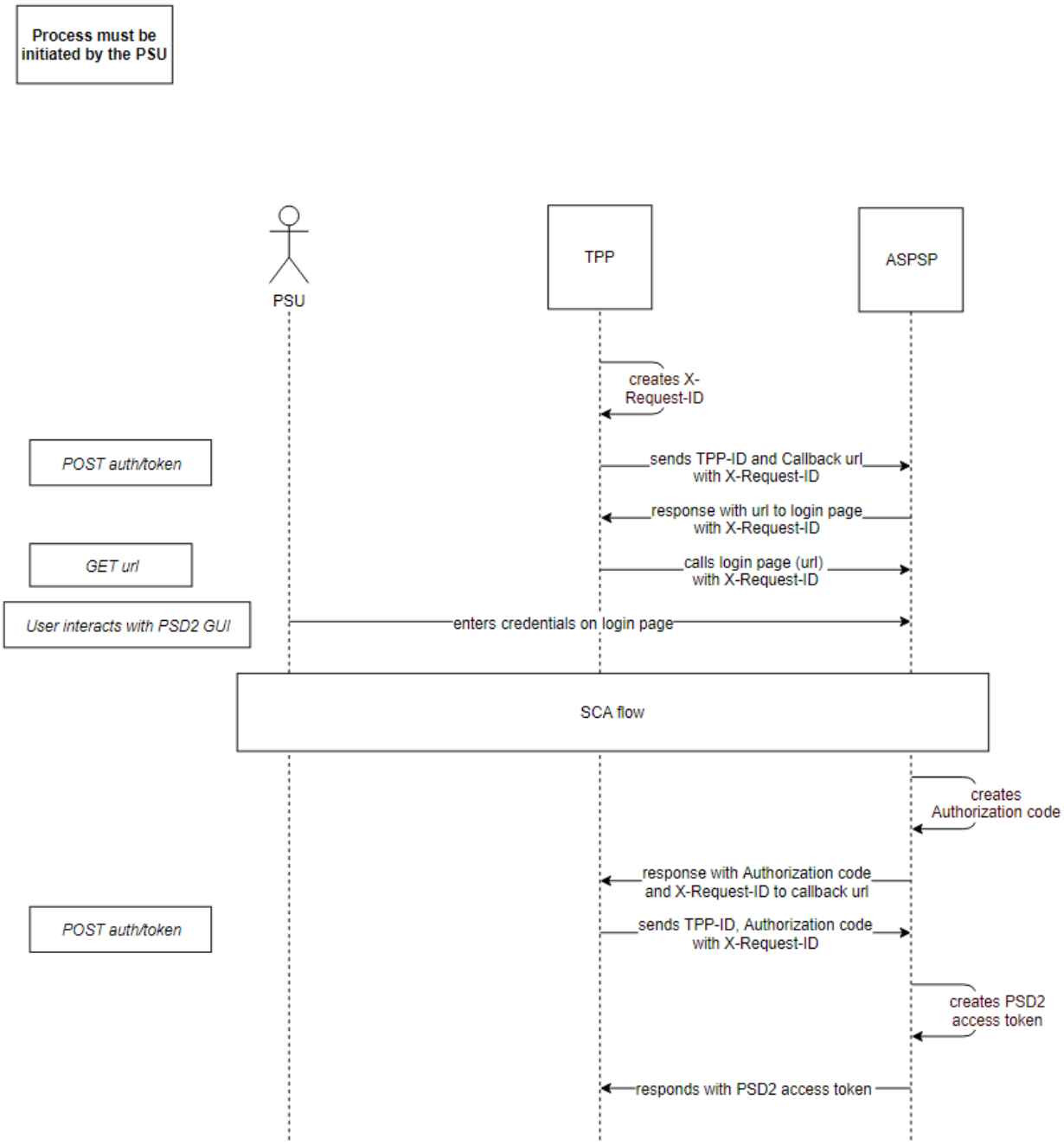

Figure 19 - Pre-auth Authorization code flow (SCA necessary)

The blackbox “SCA flow” is described in chapter *3.1.3*.

3.1.3 SCA flow

This flow shows the execution of the Strong customer authentication.

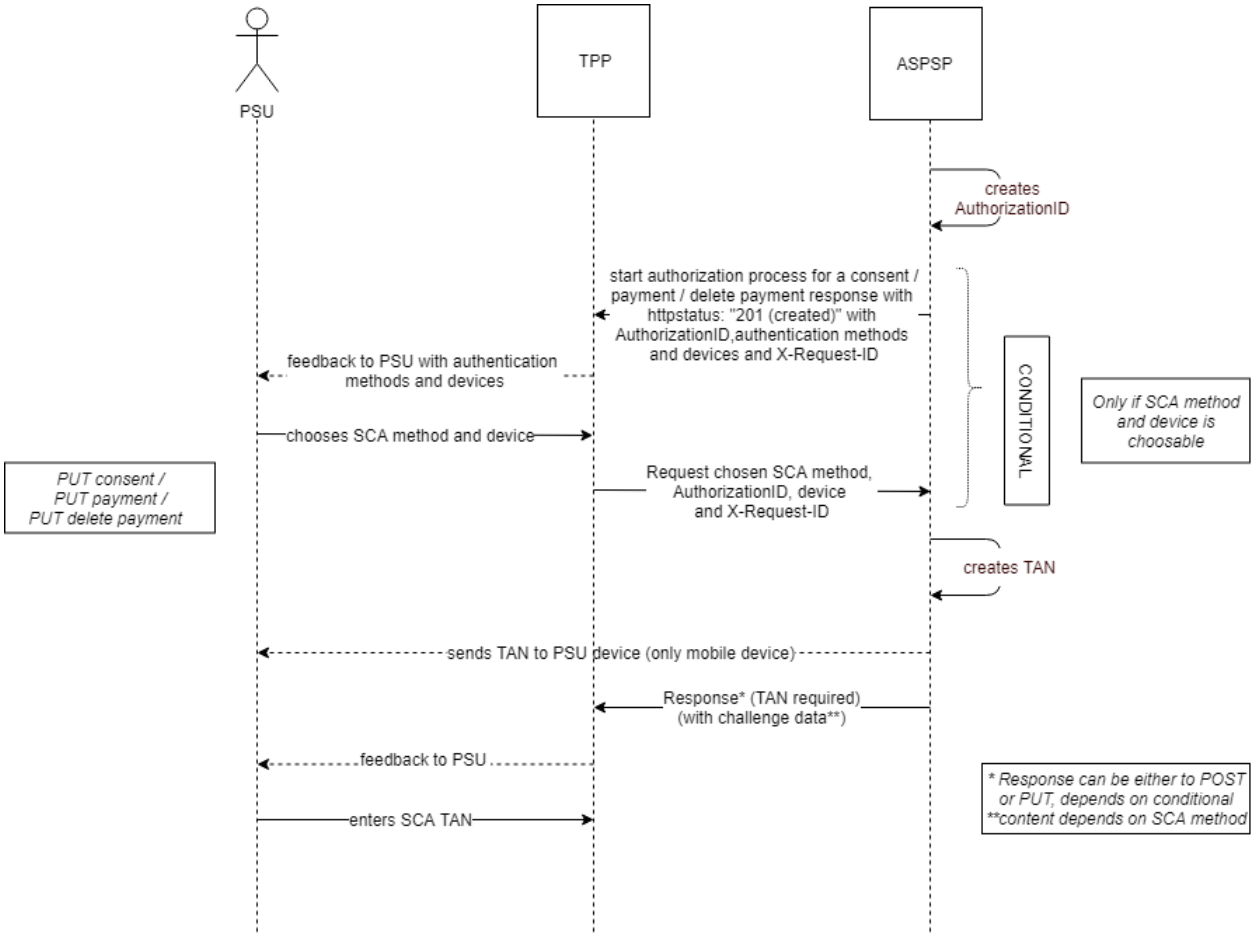

Figure 20 - SCA flow

3.1.4 Password credentials flow (SCA not possible)

Here the TPP stores the credentials of the PSU and transfers them to the OAuth server (standard procedure) when logging in. The process can be performed without a PSU but fails if a SCA is required. This flow is only possible within the SCA exemption period.

POST auth/token → request with credentials, response with PSD2 token

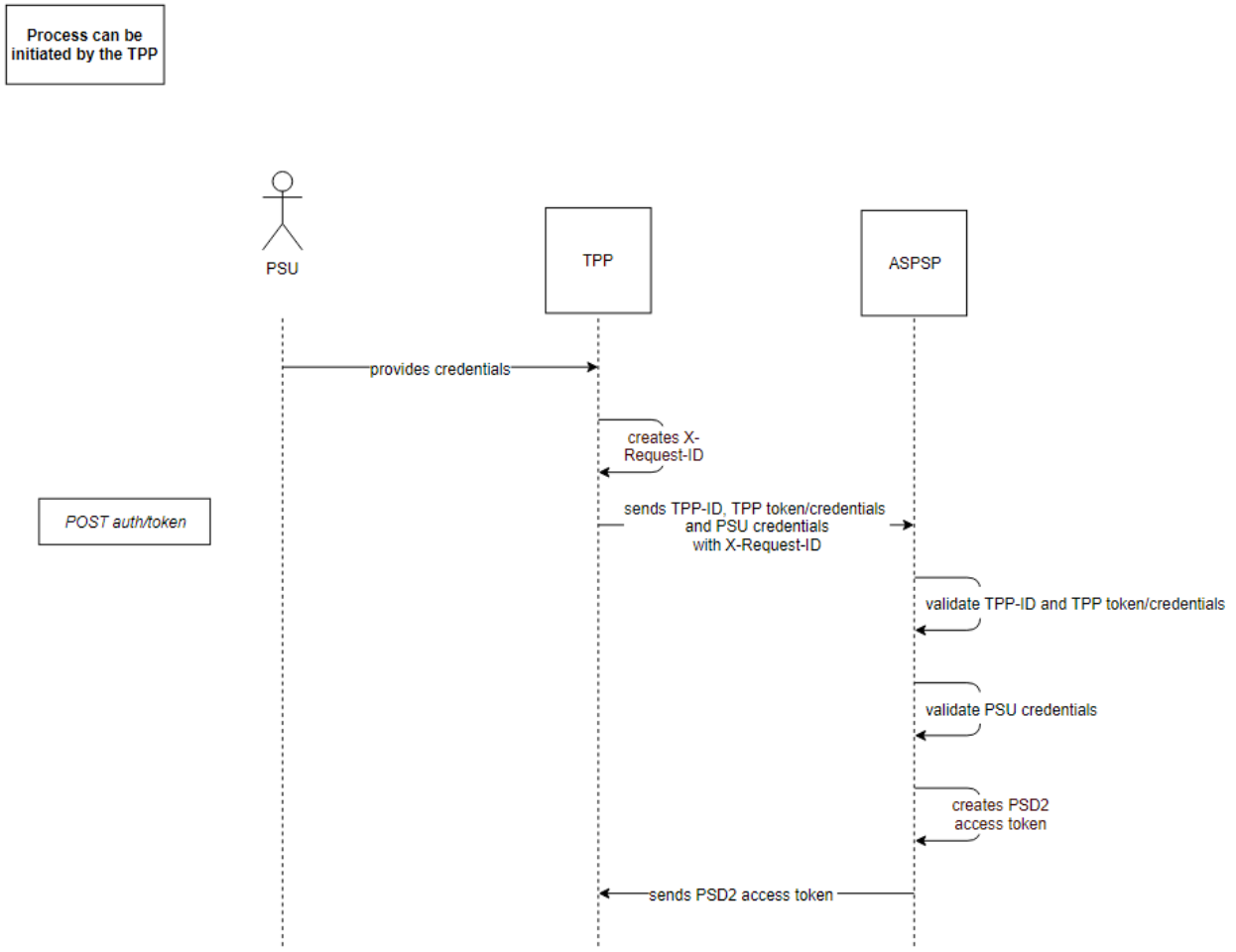

Figure 21 - Password credentials flow (SCA not necessary)

3.2 Possible error codes with Pre-Authentication

HTTP Code	error (Enumeration)	error_description	Beschreibung
400	unauthorized	Bad Credentials	ClientID or ClientSecret wrong
400	unsupported_grant_type	Unsupported grant type: <recived grant type>	
400	invalid_client	Given client ID does not match authenticated client	
400	invalid_grant	Bad Credentials	Username or Password wrong
400	user_locked	User is locked	
400	unsupported_authorization_method		If Password Credentials Flow depends on two-factor authentication

4 Architecture and Workflows

The TPP will register via User Interface integrated in API Management. The API Management transfers TPP information and the certificate to the TPP Management Module. The TPP Management module stores the information in the database. In addition, the TPP Management module validates the issuer and then client certificate against the Trust Center. Only QWAC certificates with ETSI Standard TS 119 495 can be connected.

4.1 Registration of TPP

4.1.1 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.

4.1.2 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.

4.1.3 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.

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.

4.2 Onboarding

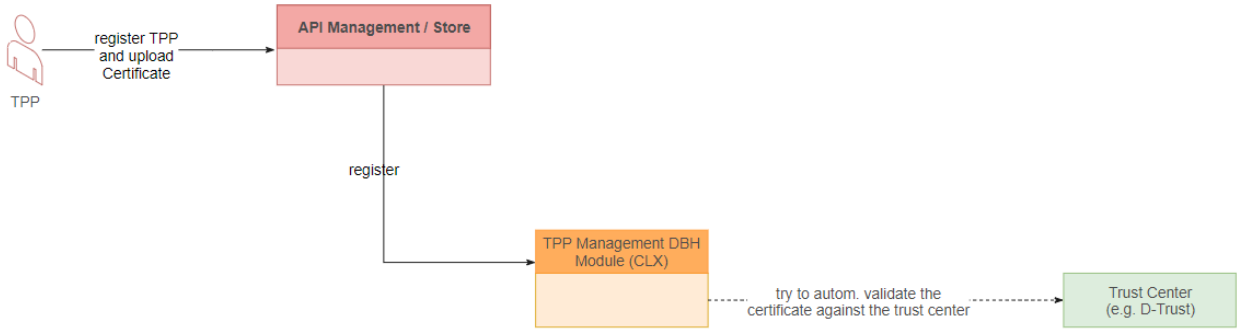

Figure 22 - Onboarding

5 TPP Management Module

With this module (TPP MM) the ASPSP can manage the TPP, e.g. establish a self-service onboarding/registration process for the TPP. Please see the User interface draft below. Amongst other data, the TPP will provide the TPP certificate.

The OAuth2 Server can also connect to the module to perform validity checks of the TPP certificate issuer and the certificate itself any time after the onboarding process. It will store registration info in a database. The TPP MM will be used for updating TPP information or renew the certificate.

The TPP MM will automatically make certificate validation checks against the QTSP. This is necessary as the certificate of a TPP can become invalid any time e.g. if the TPP lost his role(s) at the NA. With the OCSP protocol a real-time request against the QTSP is possible.

5.1 TPP registration

After TPP was registered via API Management for the first time the TPP will now import his data here via Import button.

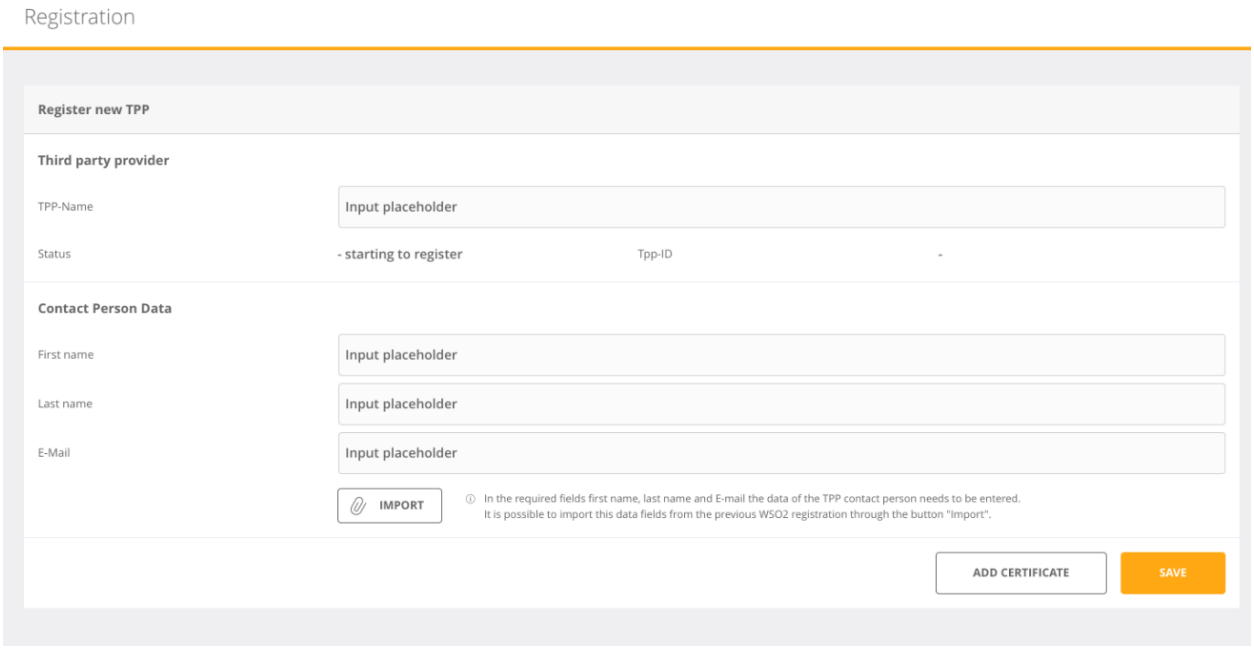

Figure 23 - TPP Registration

5.2 Add / Update certificate

Add or update certificates (by TPP).

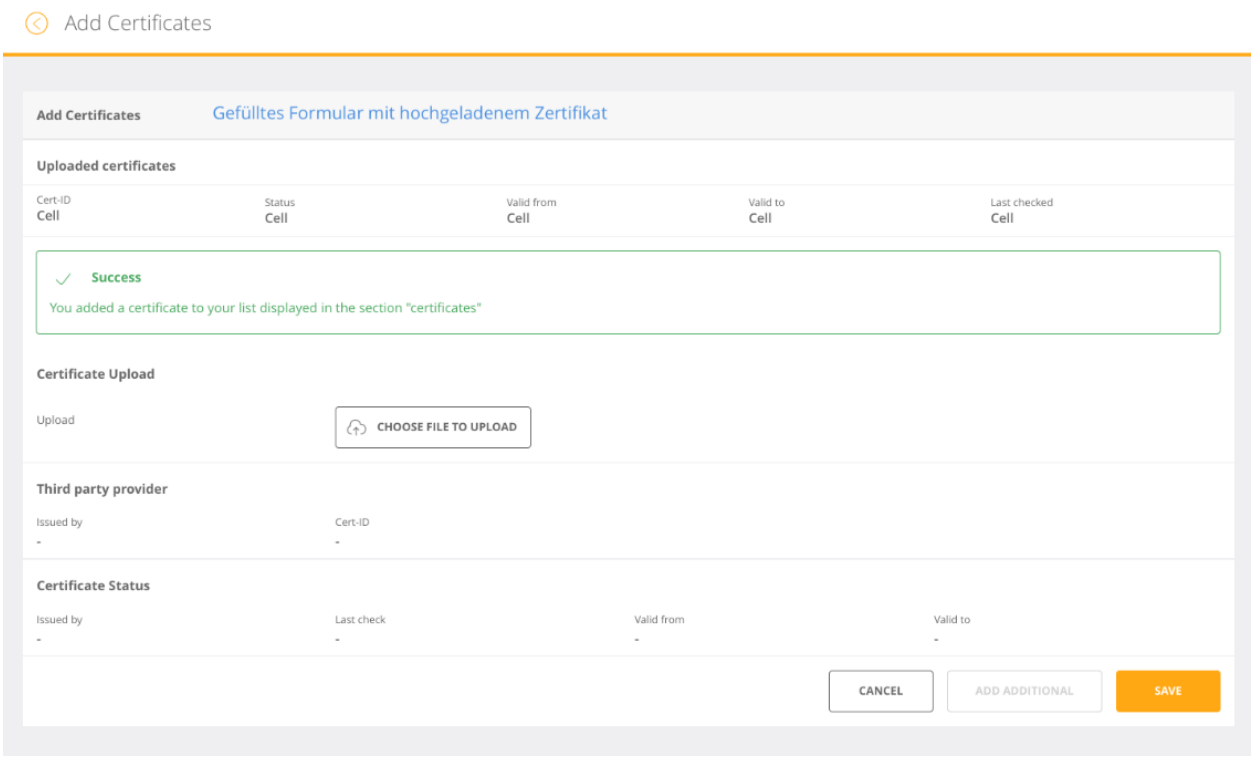

Figure 24 - Add certificate

6 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 of all, the PSU provides the account data (e.g. IBANs) to the TPP. 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 particular 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 inform the CMM to make the consent invalid. 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.

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

6.1 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.

6.2 Consent status (Berlin group)

Code	Description
received	The consent data have been received and are technically correct. The data is not authorised yet.
rejected	The consent data have been rejected e.g. since no successful authorisation 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.

7 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.

7.1 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}

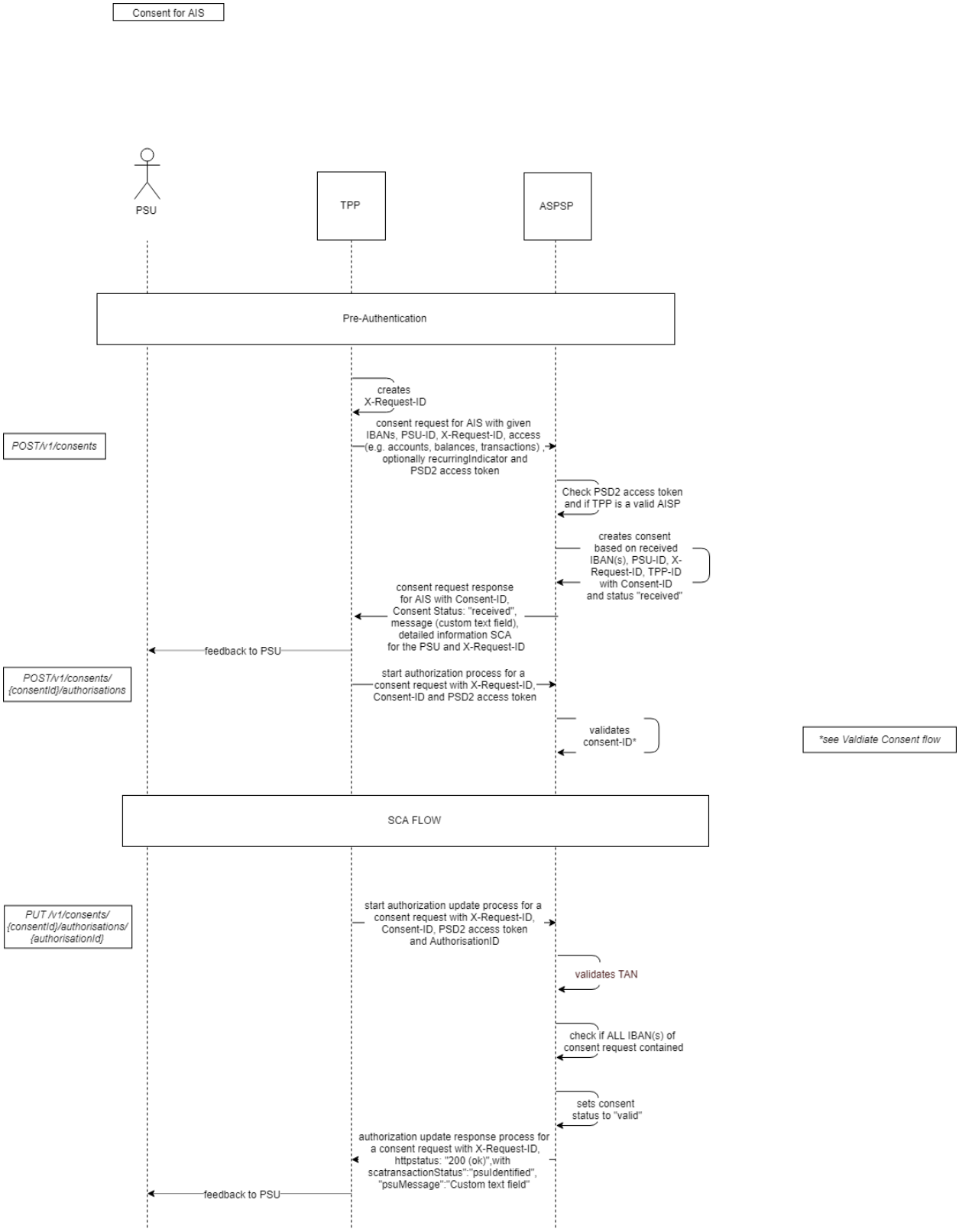

Figure 25 - Consent for AIS

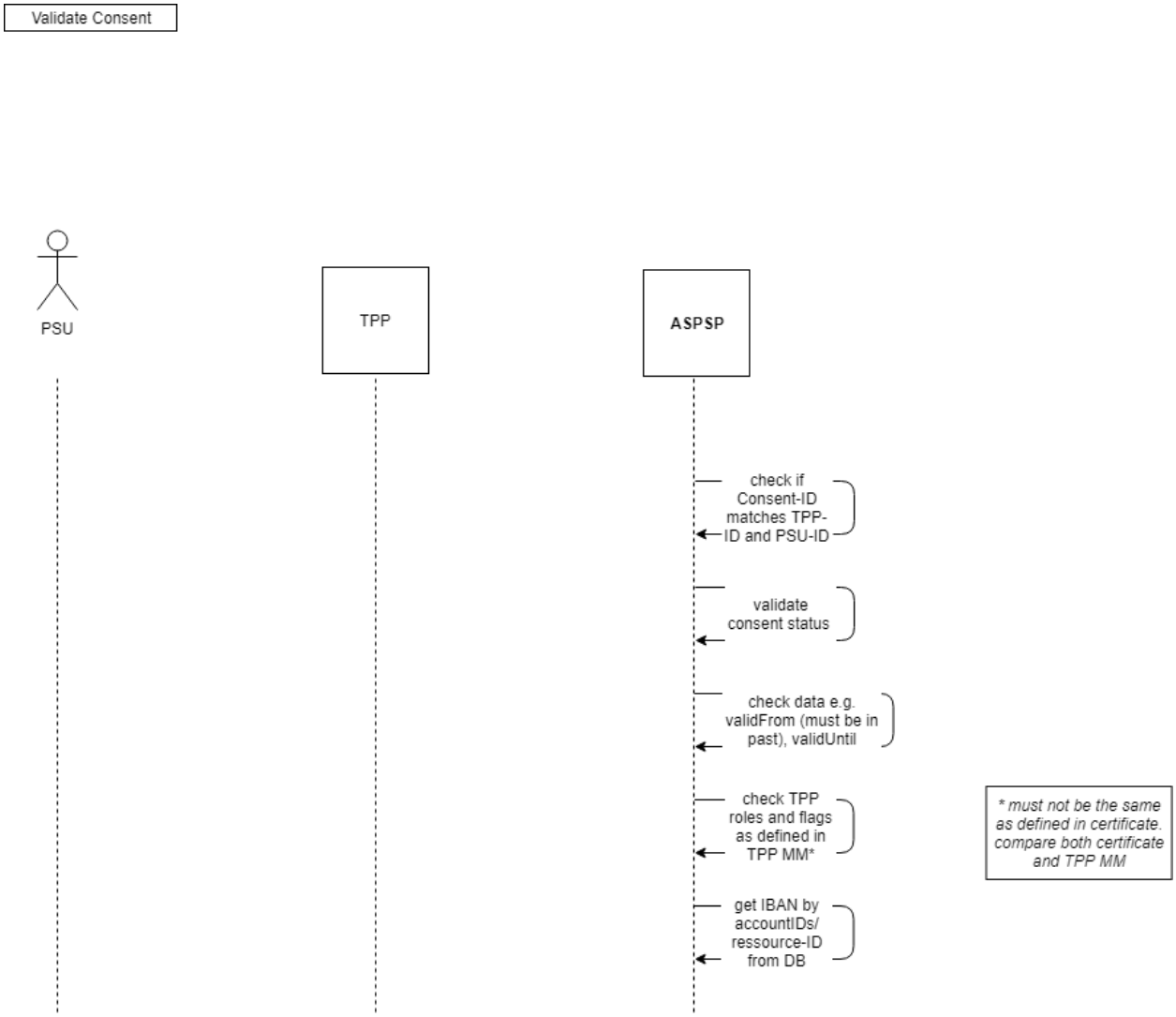

Figure 26 - Validate consent

7.2 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/

7.2.1 Get accounts

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

GET/v1/accounts

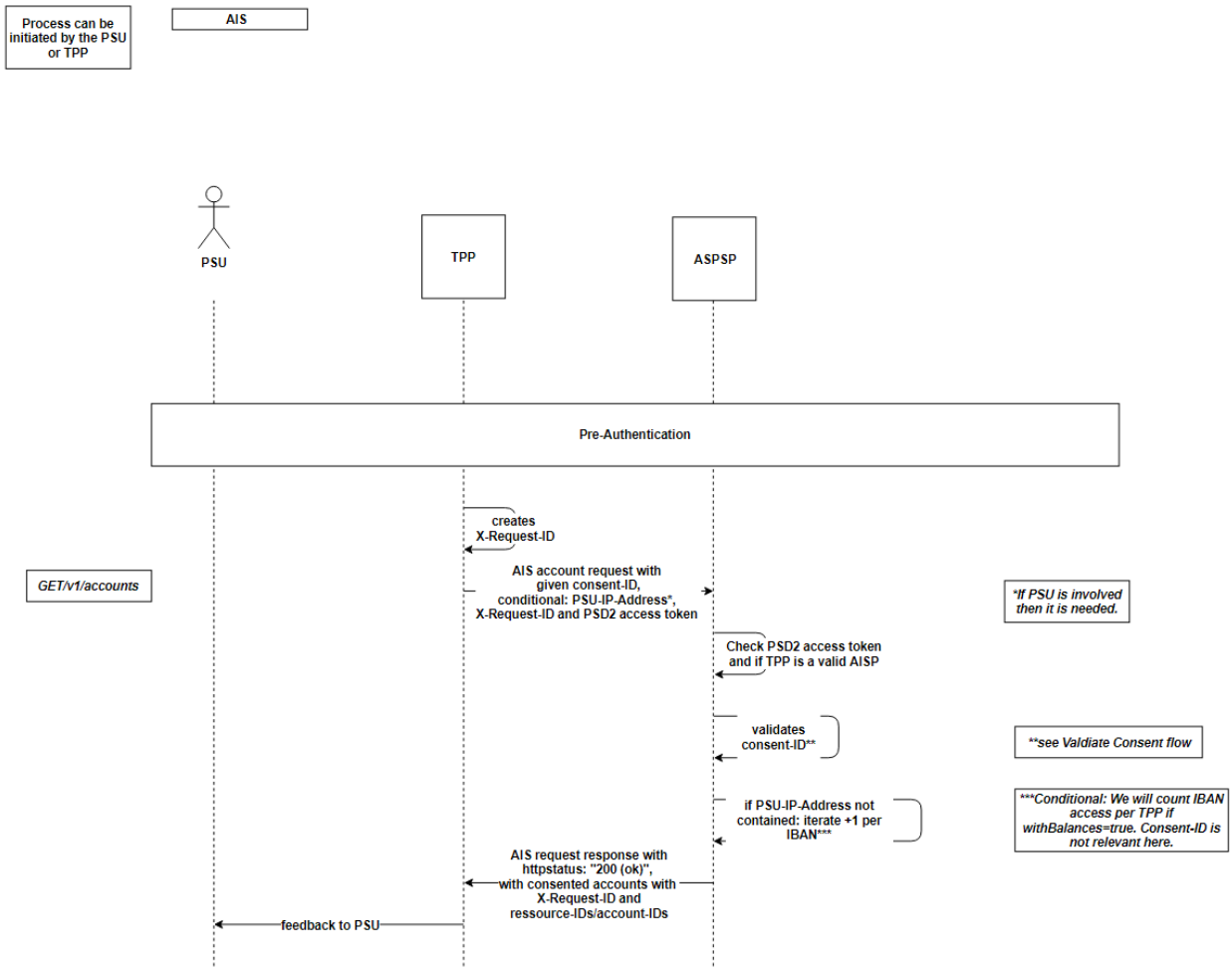

Figure 27 - Get accounts

7.2.2 Get Balances

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

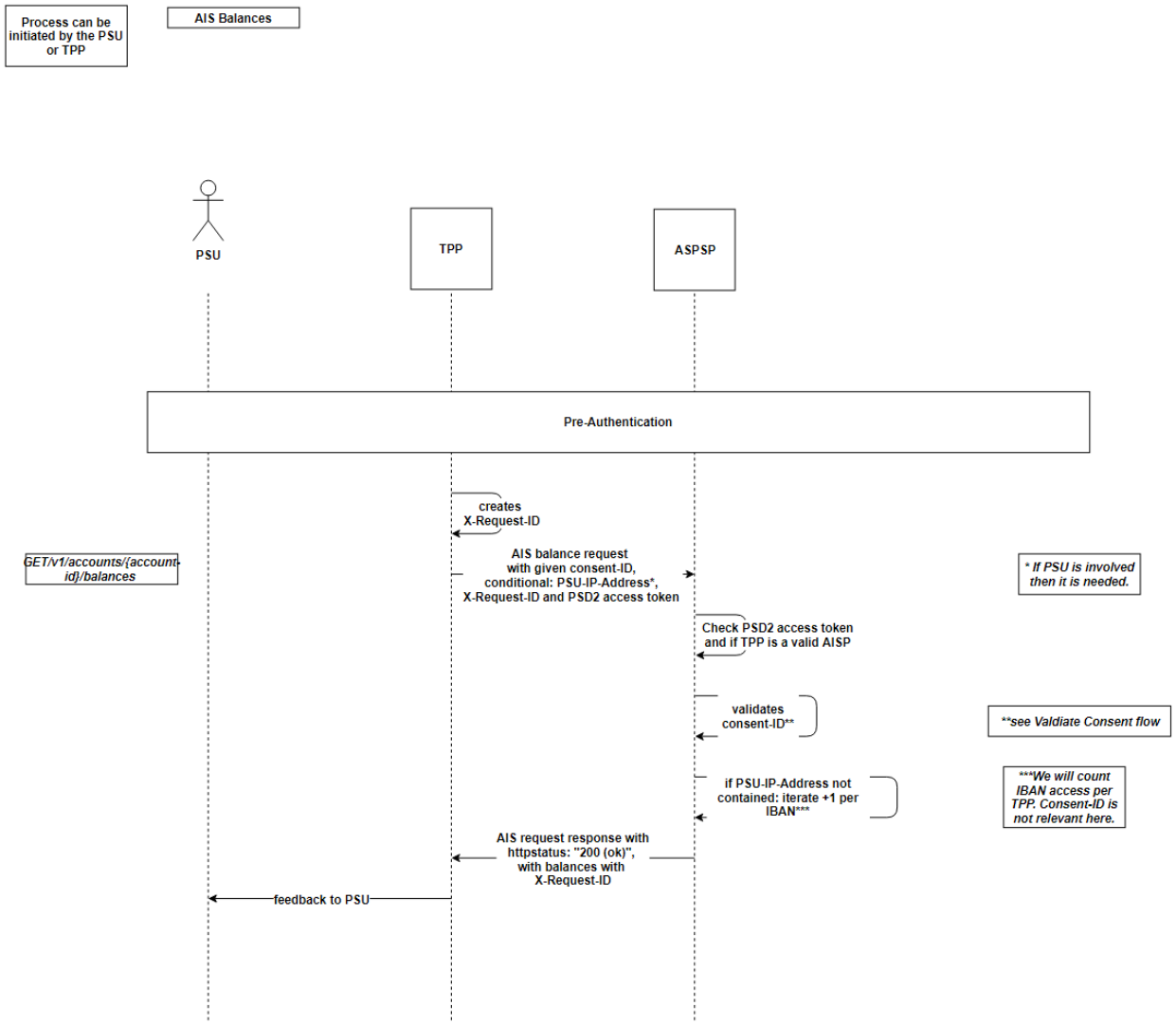

Figure 28 - Get Balances

7.2.3 Get Transactions

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

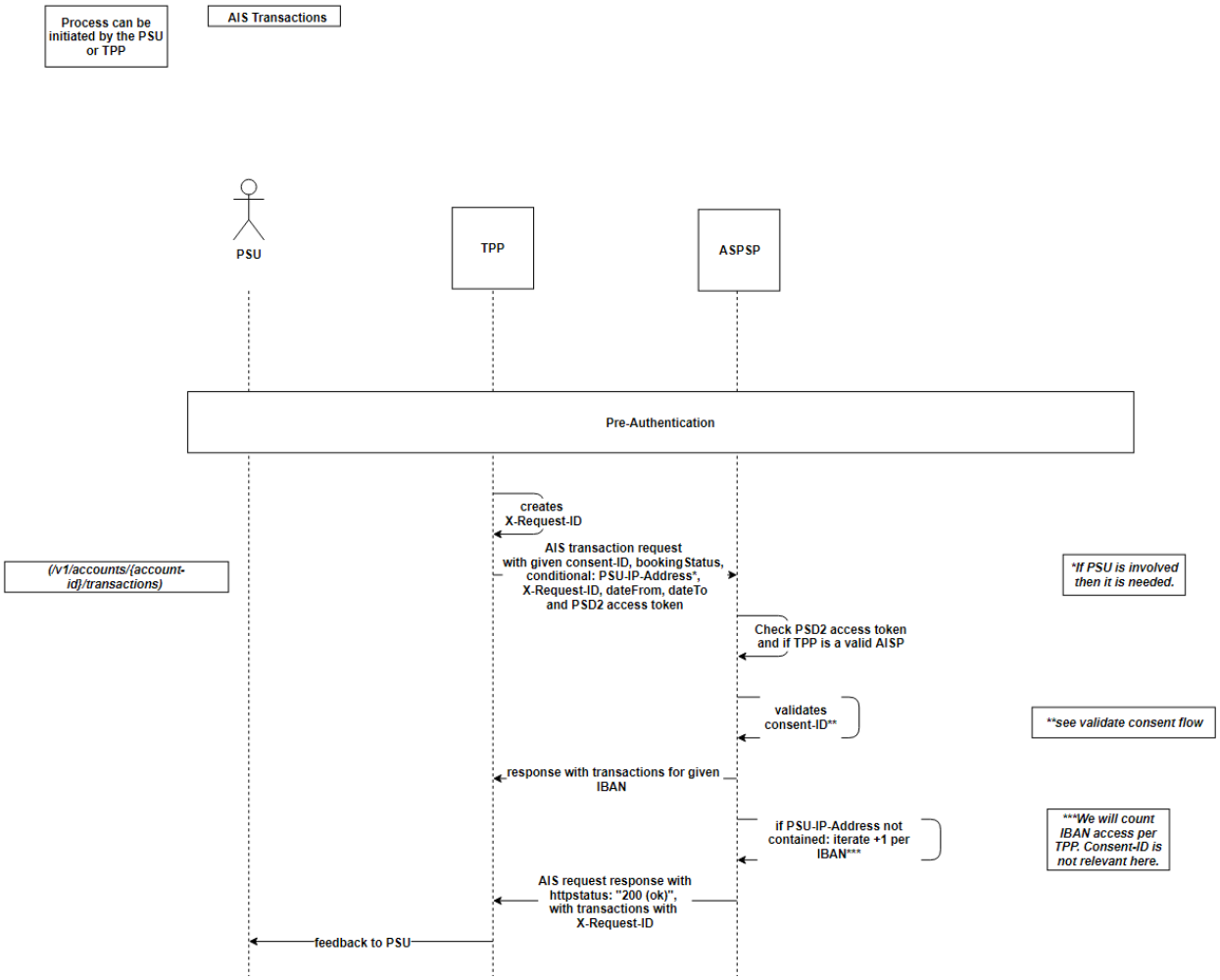

Figure 29 - Get transactions

7.2.4 Get Transactions older 90 days

SCA is necessary if transactions are older than 90 days. This will only be necessary if Login does not require SCA.

If SCA is required, the API returns http 403 response with the links to the authorization API. The link is similar to the consent authentication but uses a transactionID instead of a ConsentID.

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

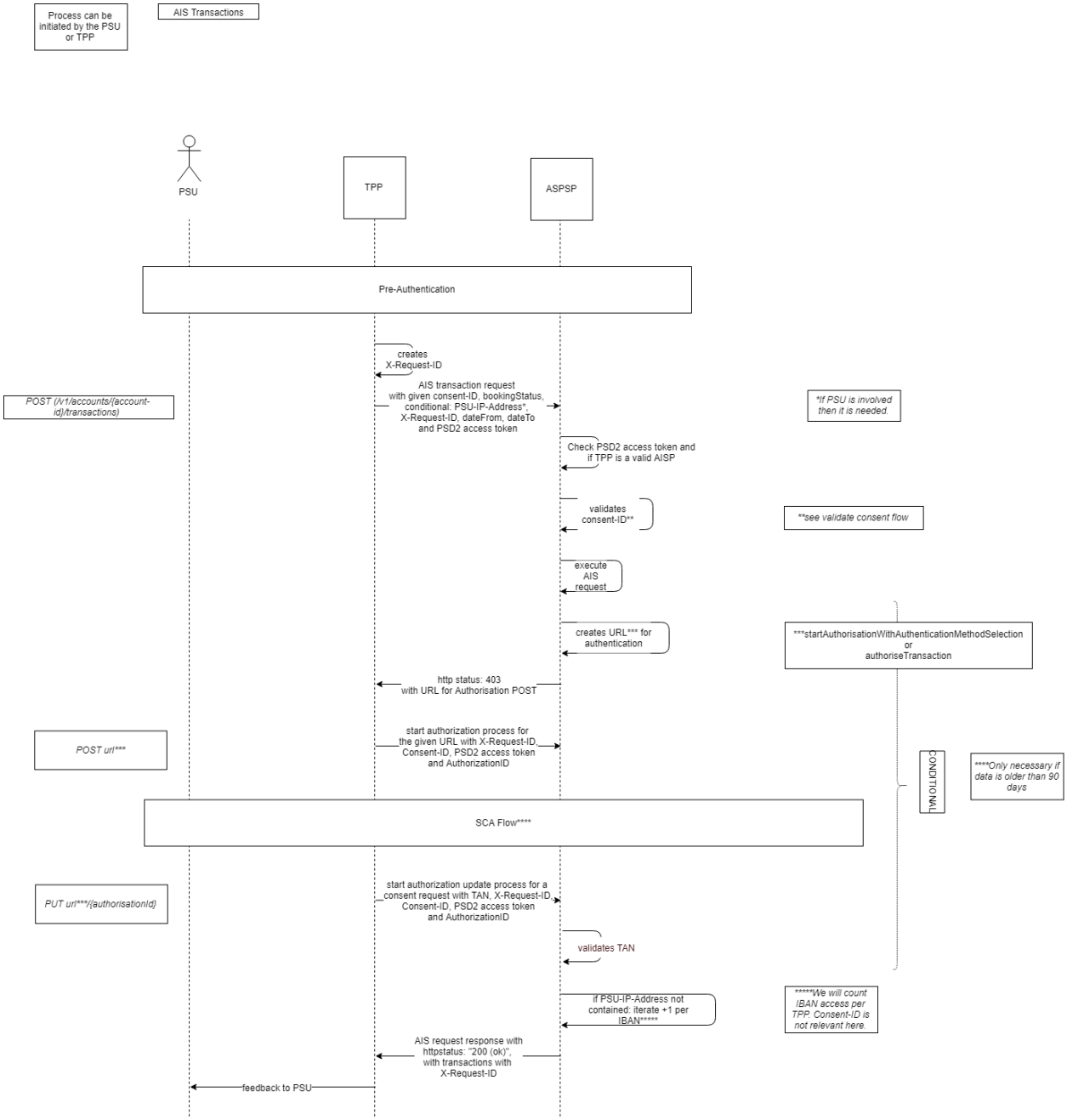

Figure 30 - Get transactions (older than 90 days)

7.3 Payments

There will always be a SCA necessary for payments.

7.3.1 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

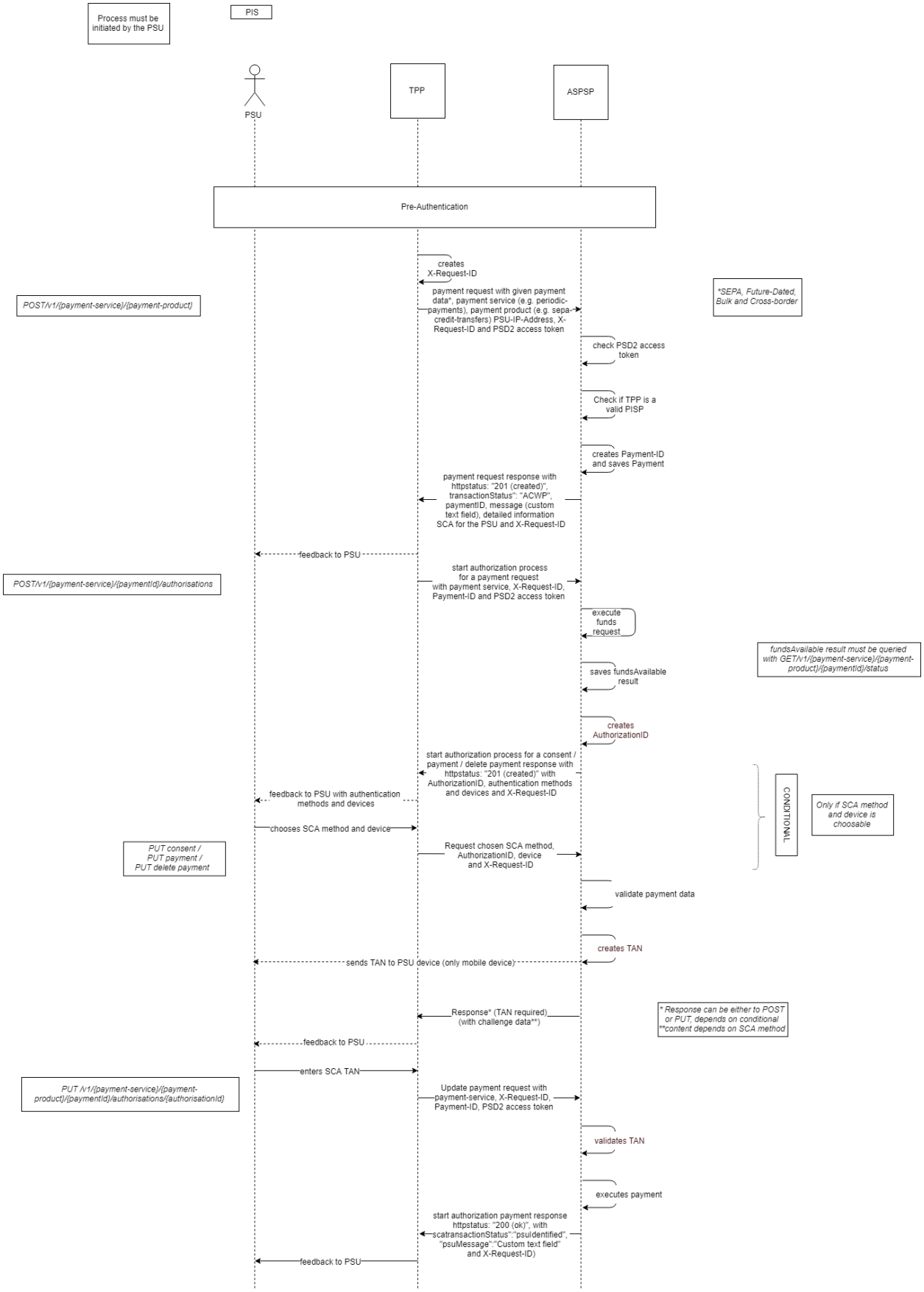

Figure 31 - Payment initiation

7.3.2 Get Payment

Returns the content of a payment object.

First step: Pre-Step Authentication

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

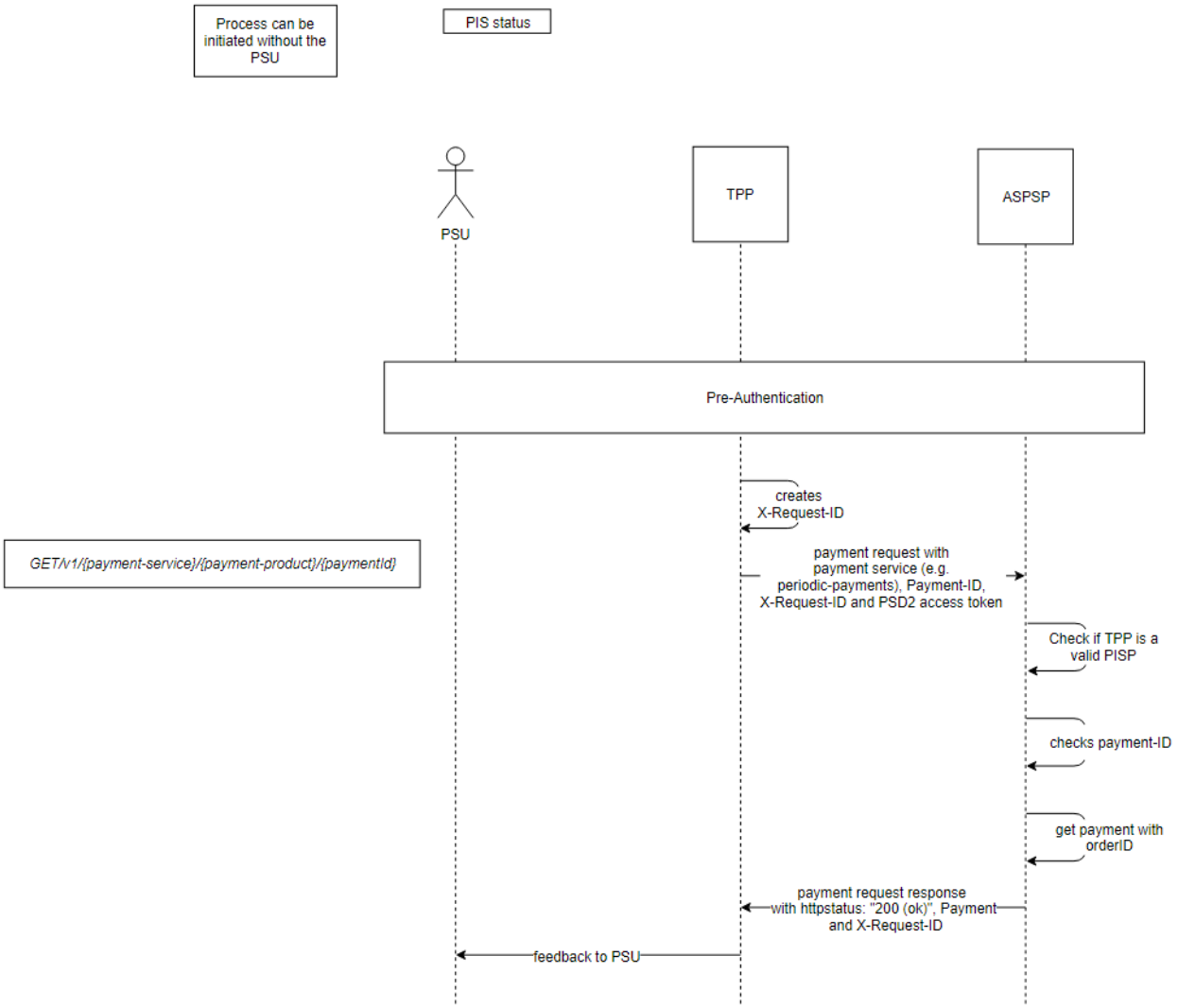

Figure 32 - Get Payment

7.3.3 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.

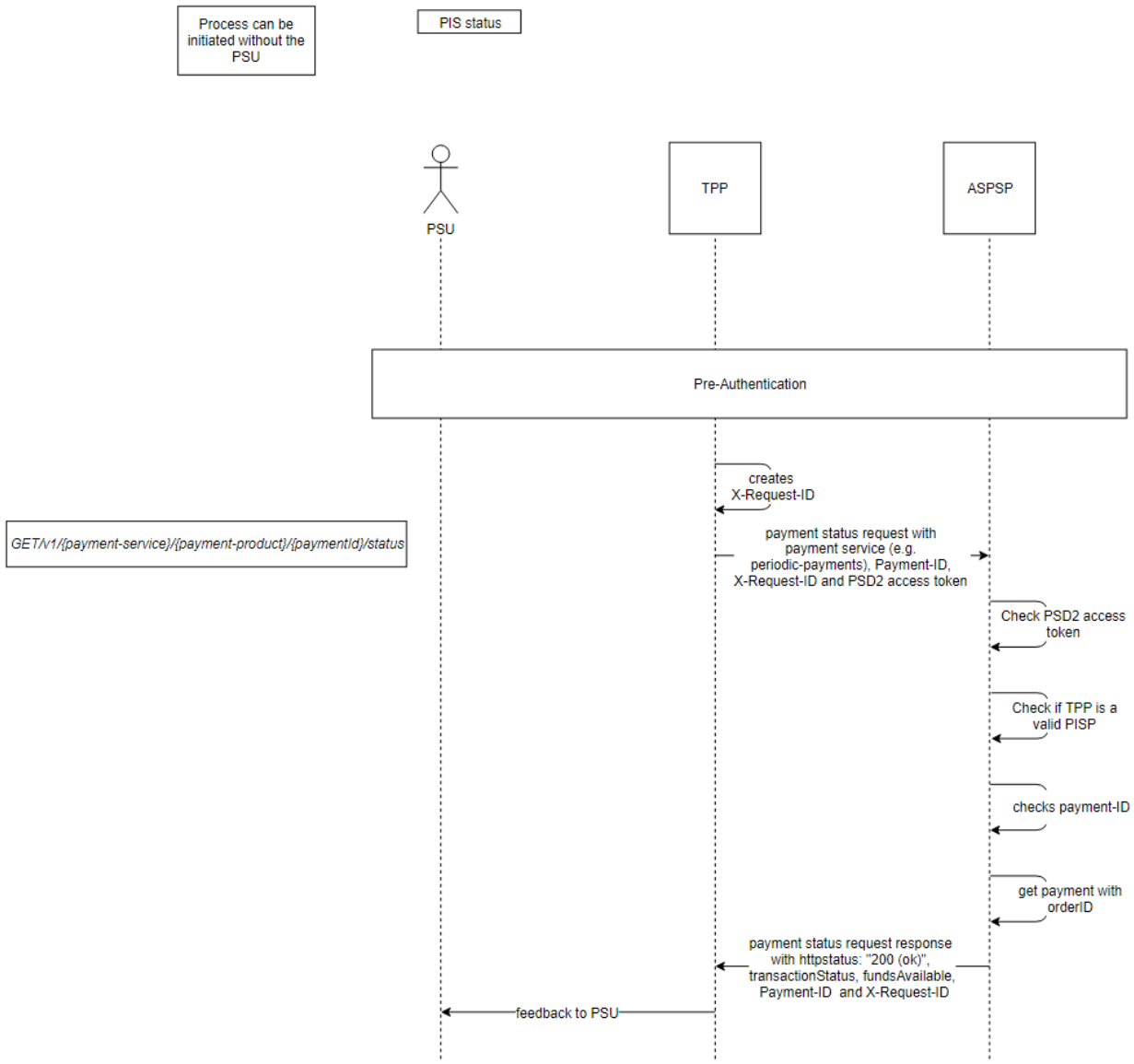

Figure 33 - Get Payment status

7.4 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 transported to TPP. Because for the funds request the TPP can NOT enter a consent-ID.

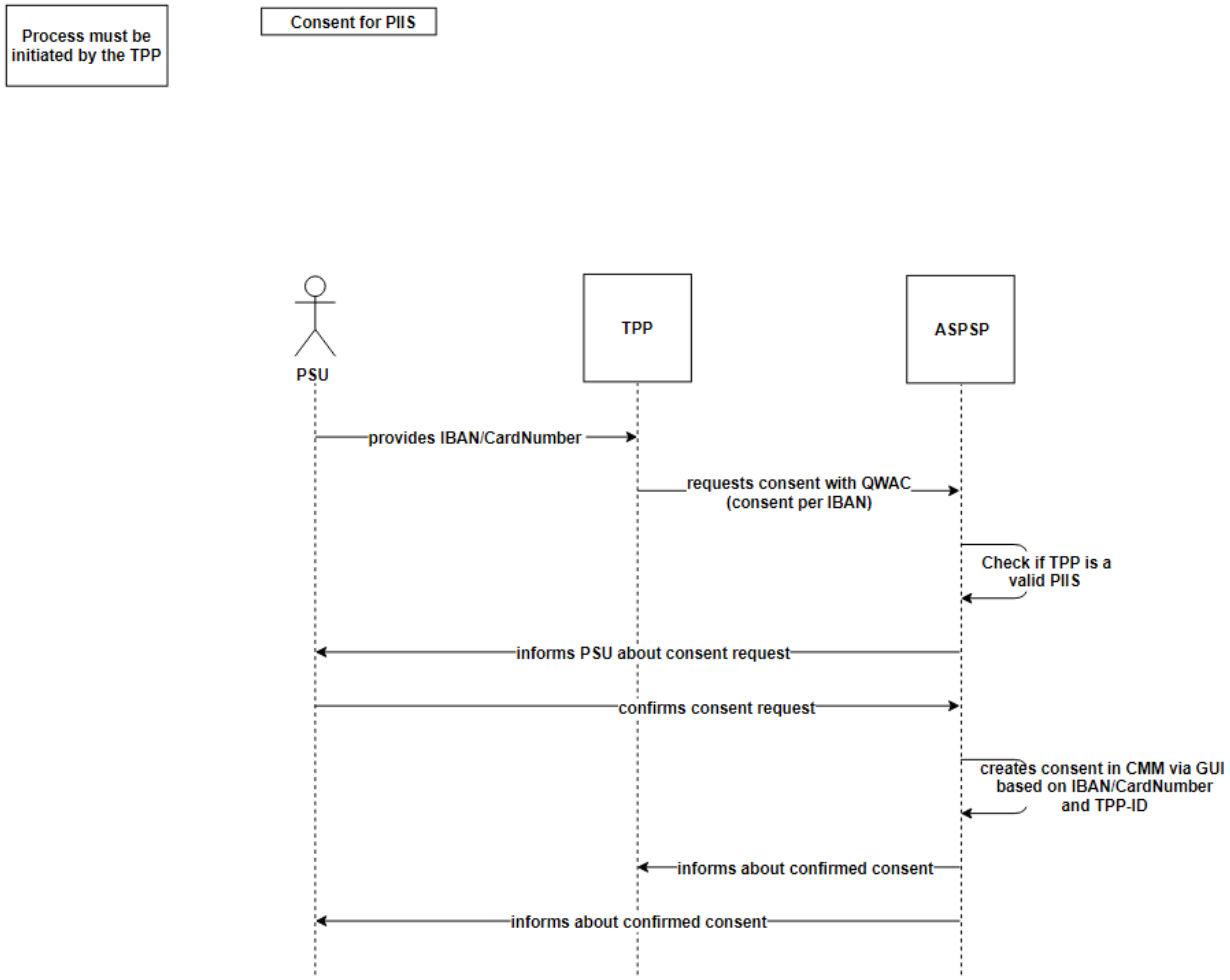

Figure 34 - Consent for PIIS

First step: Pre-Step Authentication

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

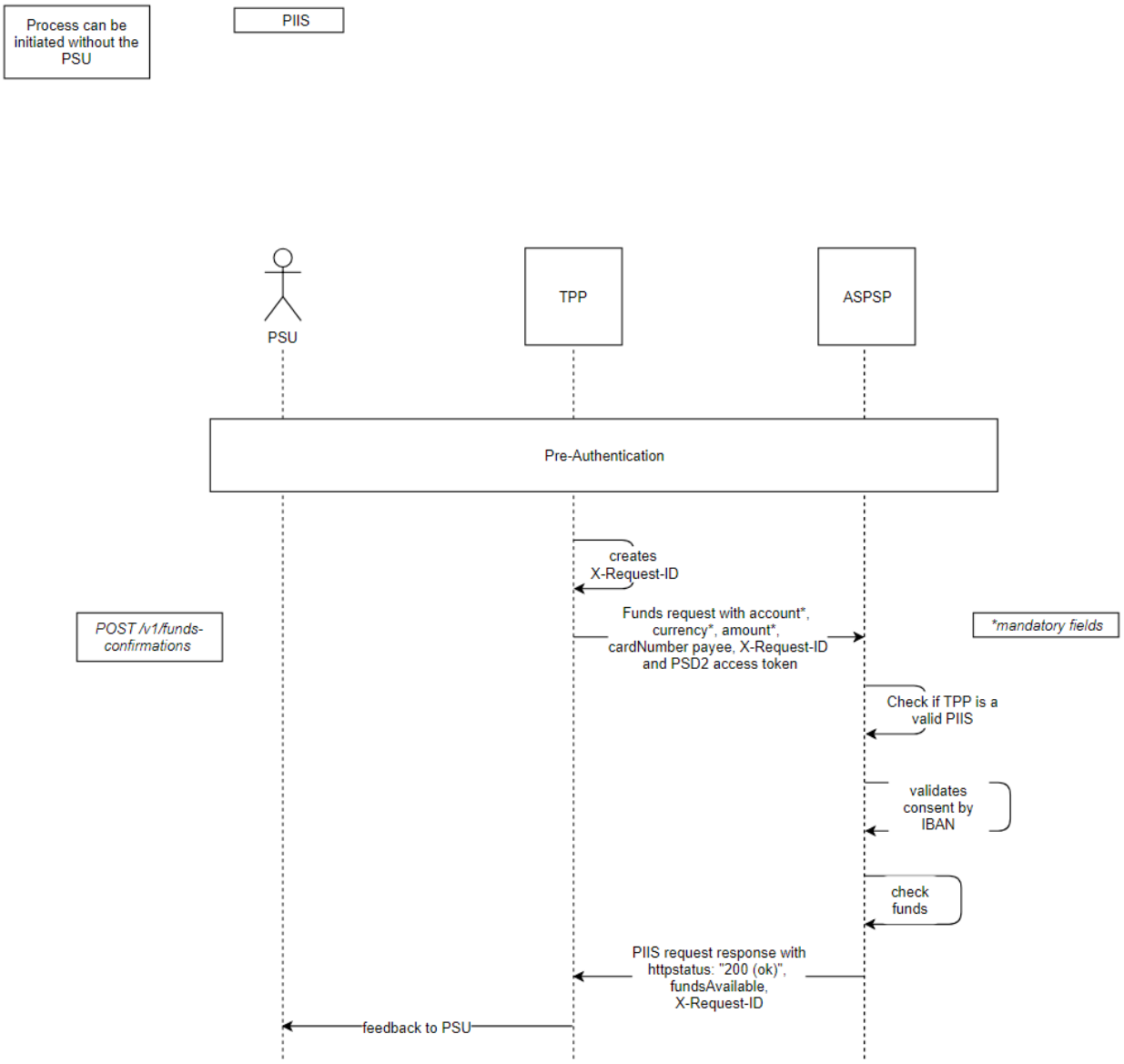

Figure 35 - Funds confirmation

Differences to Consent for AIS

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

8 Comply only (MVP)

For the PSD2 comply only solution that is defined as Minimum Viable Product at the Aareal Bank 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. Aareal Bank can only offer functions that are contained in their specific online banking.

8.1 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 36 - Not included (Berlin Group)

8.2 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 37 - 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.

8.3 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 38 - Not included in general

9 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
WSO2 API Manager	Description: https://wso2.com/api-management/ Documentation: https://docs.wso2.com/display/AM250/WSO2+API+Manager+Documentation
WSO2 Analytics	https://docs.wso2.com/display/AM250/Analytics
WSO2 Admin Guide	https://docs.wso2.com/display/AM250/Product+Administration

10 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	