VÚB BANKA

PSD2 documentation

API Documentation for Third party providers

Version	1.4
Status	Final draft
Author(s)	developers@vub.sk

The following Terms and Abbreviations have the meaning ascribed to them below, for the purposes of this document:

Abbreviation	Description
API	Application Programming Interface
AISP	Account Information Service Provider.
ASPSP	Account Servicing Payment Service Provider.
Authentication	Identity confirmation.
Authorization	Verification of access to ASPSP resources.
Certificate	A qualified certificate in the sense of e-IDAS. After the SCA RTS has been applied, it will be the only acceptable certificate for authentication of TPP
Directive / PSD2	PSD2 Directive. Directive of the European Parliament and of the Council (EU) No. 2015/2366 In Slovak republic, Directive was implemented into the Slovak Act No. 492/2009 Coll. on payment services as amended.
EV	Extended Validation certificate
IBAN	International Bank Account Number.
JOSE	JSON Object Signing and Encryption.
OIDC	OpenID Connect
Optional input parameter	TPP can ignore this parameter.
Optional output Parameter	The ASPSP may fill the parameter value.
PIISP	Payment Instrument Issuer Service Provider
PISP	Payment Initiation Service Provider.
PSU	Payment Service User.
Resource	All access points of the ASPSP API for TPP access within PSD2.
RTS	Regulatory technical standards of the Europan Banking Authority
SCA	Strong Customer Authentication. Authentication of a payment service user means authentication based on the use of two or more elements that are categorized as knowledge (something the user knows only), ownership (something that only the user has), and inherence (something, the user is) and are independent in the sense that the violation of one element does not impair the reliability of the other elements, while being created in such a way as to protect the confidentiality of the authentication data.

Abbreviation	Description
TPP	Third Party Provider, i.e., a third party that is a payment service provider providing payment service users with a payment initiation or account information service or a payment service provider issuing card based payment facilities.
VUB	Všeobecná úverová banka, a.s., Mlynské nivy 1, 829 90 Bratislava 25, registered with Companies Registry of the District Court Bratislava I, Section: Sa, Insertion No.: 341/B, BIC: SUBASKBX, Identification No.: 31 320 155

Contents

List of Figures

List of Tables

1 DISCLAIMER

The presented documentation describes the Application Programming Interface (API) provided by VUB to PSD2 licensed Third Parties Providers (TPPs) that provides the Payment Initiation Services, Account Information Services and issue the card-based payment instruments. The documentation is provided for purposes of integration of TPPs applications with VUB´s API and describes the prerequisites needed for connection to VUB API, data flows, endpoints and attributes of API.

The documentation may be used only for the purposes described above. Any breach of this obligation may be subject to damage recovery and penalties under Civil Code, Commercial Code, Criminal Law or any other applicable laws and regulations of the Slovak Republic.

2 GOAL OF THE DOCUMENT

Main aim of the document is to describe API provided by VUB to PSD2 licensed TPP.

VUB provides bank interfaces to other licensed TPP. Developers of TPP can integrate their applications with VUB API and gain access to the payment accounts held by VUB, that are available online, upon PSU consent. These services do not replace existing bank services.

This document describes prerequisites needed for connection to VUB API, data flows among all participated parties, endpoints and attributes of API.

Each API is described by OpenAPI specification in the form of Swagger files.

3 BEFORE YOU START

APIs provided by the VUB are intended to be used by *Third Party Providers* (TPP), which are subjects of the license for:

- *Payment Initiation Service Provider* (PISP) - subject has a license for providing payment initiation services to *Payment Service Users* (PSU)
- *Account Information Service Provider* (AISP) - subject has a license for providing account information services to *Payment Service Users* (PSU)
- *Payment Instrument Issuer Service Provider* (PIISP) - subject has a license for providing PIISP services to *Payment Service Users* (PSU)

License can be obtained by relevant national regulator (Slovak National Bank for companies with its seat in Slovak Republic).

Mutual Transport Layer Security (MTLS) is used for secure communication between *Third Party Provider* and *Account Services Payment Service Providers* (ASPSP) – Bank/ VUB. Each TPP, who wants to use VUB API has to provide eIDAS, or EV certificate until RTS SCA has been applied, signed by trusted *Certification Authority* (CA).

To ensure, that only licensed and identified TPP will have an access to VUB API, manual technical TPP enrolment process must be performed in advance, before TPP is able to use the API and before access payment accounts held by VUB.

After successful TPP PSD2 registration by National regulator, the TPP is given by license number and PKI certificate which contains the license number (obtained by trusted Certification Authority). Since this point, the TPP is allowed to perform both communication with an ASPSP and use development portal.
VUB will manage communication with TPP using the IETF RFC 6749 - The OAuth 2.0 Authorization Framework („OAuth framework“, hereafter only). Therefore, in order to access VUB API, TPP must be given by an access token which must be presented when performing a call. The access token usage is defined by the ITEF RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage („**access_token“**, hereafter only). All TPP requests, where technically possible, must be protected by TLS protocol with mutual authentication

Table 1 - VUB provides following set of an APIs.

Accounts information	API provides information and balances related to a payment account.
Accounts transactions	API provides list of transactions in defined range related to a payment account
Balance check	API provides information about sufficient balance with yes/no answer
Payments initiation and submission	API allows to initialize the payment in XML format (pain.001)
Payment status	API provides actual information about initialized payment

3.1 API principles

VUB APIs defines several principles and rules, which needs to be followed by TPP.

a) Each mandatory service operation is related just to one customer's payment account that is accessible online. None of the service operations can provide response for a bulk of accounts.

b) An account identifier, especially IBAN, or any other personal data of the PSU which can be used to carry out fraud shall not appear in an internet address of a service operation. It should be located in the body of a HTTP request, or at least in a HTTP header field.

- This principle ensures that for instance IBAN or any other identification data of PSU item cannot be used neither as a path template parameter nor as a query parameters of a service operation.

c) Naming convention for data elements.

- The names of data elements in service operation parameters and in the data model should be in lower camel case.
- The data element starts with a meaningful word in lower case followed by words with the first capital letters, e.g., accountNumber.
- Non-alphabetic characters should not be used as word delimiters in data elements names, e.g., account_number.

d) When accessing developer portal of VUB and / or payments accounts of PSU held by VUB, all rights and obligations stipulated by Directive implemented in national legislation and relative legal acts shall be followed by every participant.

3.2 Enrolment process

Each TPP, who wants to use VUB API, has to follow enrolment process on VUB Openbanking API portal, located on https://developers.vub.sk, where technical identifiers are assigned to particular TPP. Under Directive, TPP shall identify itself towards the ASPSP of the PSU for each communication session / every time a payment is initiated, and communicate with the ASPSP in a secure way.

The TPP can be seen as a new application that wants to use VUB APIs, and for that reason an enrolment phase will be required.
The enrolment will need to have two phases: the approval of the access to the API (by local authorities and/or the bank) and the technical setup.

For the technical setup, the standard technical method is to enroll the TPP for some kind of credentials that will provide access to the VUB APIs. Client (TPP) side SSL verification will be required (Big IP 5 will validate certificates) in order to avoid MITM attacks if app key is stolen or shared.

Enrolment has the following flow:
- TPP must register in the Portal to get a valid client id and client secret (upon registration TPP identify itself as is required by legislation and security measures).
- once TPP registration is done and identity of TPP is valid , TPP certificate has to be uploaded to API Management solution, link between License number, and certificate is created, after the required check in terms of business approval is verified.

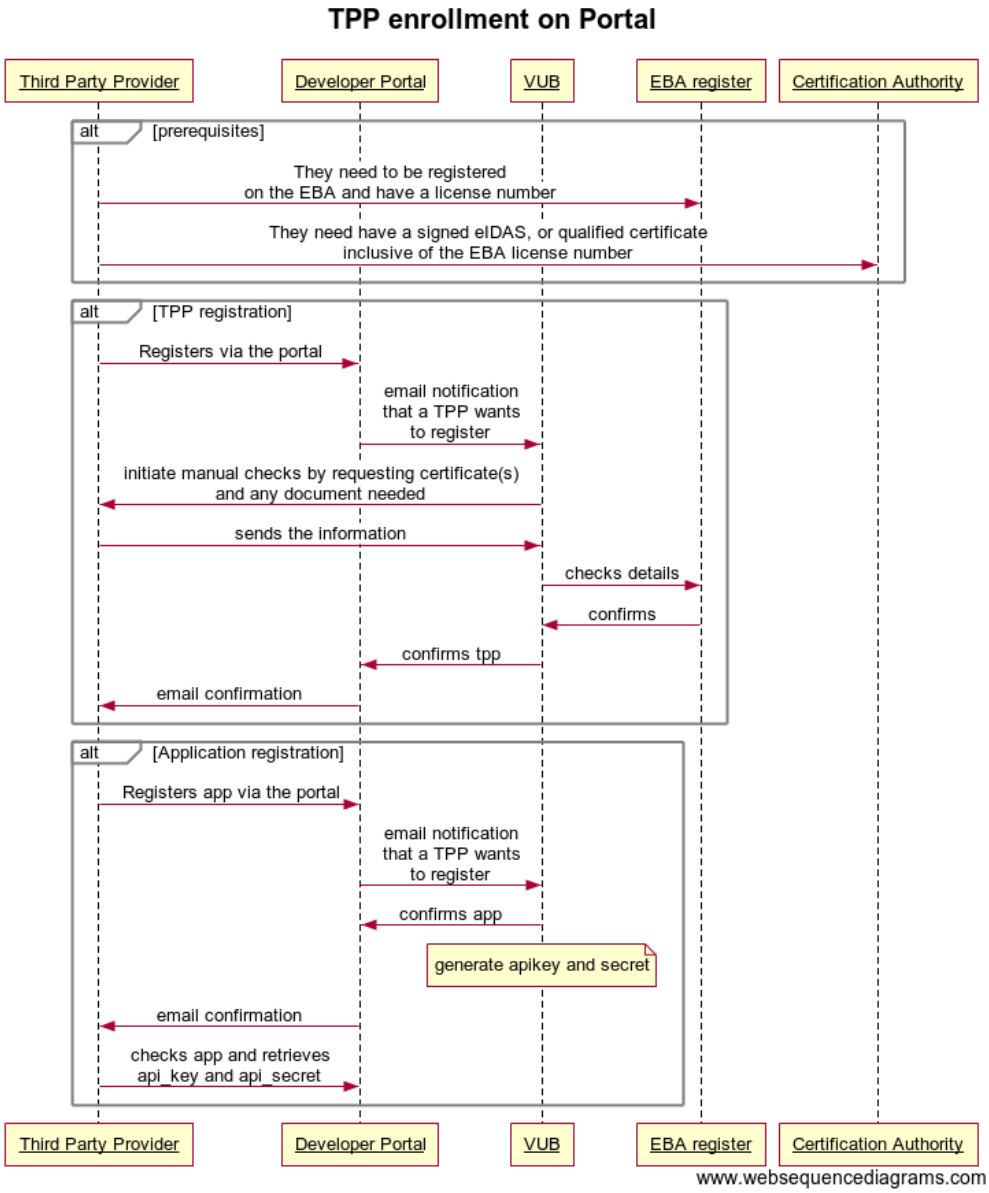

Figure 1 - Developer portal enrolment process

3.2.1 Registering of the organization

To register the Organization, please click on Register button. You have to provide several attributes about your Organization.

- E-mail - primary identifier. Will be used for communication between VUB and TPP, if it is necessary to contact TPP. Will be also used as an Organization admin
- Organization Name - name of the TPP. Name will be used as an identifier, used for validation of TPP in national register of Slovak national Bank or EBA register of TPP. If the validation of TPP will not be successful, you will be asked for documents, which will prove, that you have valid license, or you're in licensing process for particular PSD2 related license (PISP, AISP, PIISP). If TPP is only in licensing process, only access to developer portal can be provided by VUB.

After submitting the form, you will not receive any automatic confirmation e-mail. VUB will start validation process, which consists of two steps:
1. validation in PSD2 National Authority, or EBA registry
2. validation of eIDAS, or EV certificate

After validation, you will receive an approve, or disapprove notification e-mail

3.2.2 Registering of the developers

As an Organizational admin, you can invite other Developers from your Organization, which can have an access to VUB API documentation and sandbox by clicking on *Users* tab in menu.

3.2.3 Registering of the applications

Before your first API call, and also for testing purposes, you need to register an Application. By registering an Application, you need to choose which APIs you want to be subscribed to.

Application registration process is a subject of an approval process by the VUB, as different APIs may require different license.

4 COMMON SECURE COMMUNICATION

A TLS version 1.2+ is required to secure the communication layer. In order to reduce the vulnerability of block ciphers, only AEAD (Authenticated Encryption with Additional Data) is allowed, specifically:

Table 2 - AES_GCM (128,256)

NIST	OpenSSL equivalent
TLS_RSA_WITH_AES_128_GCM_SHA256	AES128-GCM-SHA256
TLS-RSA-WITH-AES-256-GCM-SHA384	AES256-GCM-SHA384
TLS-RSA-WITH-CAMELLIA-128-GCM-SHA256	NA
TLS-RSA-WITH-CAMELLIA-256-GCM-SHA384	NA
TLS-DHE-RSA-WITH-AES-128-GCM-SHA256	DHE-RSA-AES128-GCM-SHA256
TLS-DHE-RSA-WITH-AES-256-GCM-SHA384	DHE-RSA-AES256-GCM-SHA384
TLS-DHE-RSA-WITH-CAMELLIA-128-GCM-SHA256	NA
TLS-DHE-RSA-WITH-CAMELLIA-256-GCM-SHA256	NA
TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256	ECDHE-RSA-AES128-GCM-SHA256
TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384	ECDHE-RSA-AES256-GCM-SHA384
TLS-ECDH-RSA-WITH-AES-128-GCM-SHA256	ECDH-RSA-AES128-GCM-SHA256
TLS-ECDH-RSA-WITH-AES-256-GCM-SHA384	ECDH-RSA-AES256-GCM-SHA384
TLS-ECDH-RSA-WITH-CAMELLIA-128-GCM-SHA256	NA
TLS-ECDH-RSA-WITH-CAMELLIA-256-GCM-SHA384	NA
TLS-ECDHE-ECDSA-WITH-AES-128-GCM-SHA256	ECDHE-ECDSA-AES128-GCM-SHA256
TLS-ECDHE-ECDSA-WITH-AES-256-GCM-SHA384	ECDHE-ECDSA-AES256-GCM-SHA384
TLS-ECDHE-ECDSA-WITH-CAMELLIA-128-GCM-SHA256	NA
TLS-ECDHE-ECDSA-WITH-CAMELLIA-256-GCM-SHA384	NA
TLS-ECDH-ECDSA-WITH-AES-128-GCM-SHA256	ECDH-ECDSA-AES128-GCM-SHA256
TLS-ECDH-ECDSA-WITH-AES-256-GCM-SHA384	ECDH-ECDSA-AES256-GCM-SHA384
TLS-ECDH-ECDSA-WITH-CAMELLIA-128-GCM-SHA256	NA
TLS-ECDH-ECDSA-WITH-CAMELLIA-256-GCM-SHA384	NA

Table 3 - AES_CCM (128,256)

NIST	OpenSSL equivalent

TLS-RSA-WITH-AES-128-CCM	AES128-CCM
TLS-RSA-WITH-AES-256-CCM	AES256-CCM
TLS-RSA-WITH-AES-128-CCM-8	AES128-CCM8
TLS-RSA-WITH-AES-256-CCM-8	AES256-CCM8
TLS-DHE-RSA-WITH-AES-128-CCM	DHE-RSA-AES128-CCM
TLS-DHE-RSA-WITH-AES-256-CCM	DHE-RSA-AES256-CCM
TLS-DHE-RSA-WITH-AES-128-CCM-8	DHE-RSA-AES128-CCM8
TLS-DHE-RSA-WITH-AES-256-CCM-8	DHE-RSA-AES256-CCM8
TLS-ECDHE-ECDSA-WITH-AES-128-CCM	ECDHE-ECDSA-AES128-CCM
TLS-ECDHE-ECDSA-WITH-AES-256-CCM	ECDHE-ECDSA-AES256-CCM
TLS-ECDHE-ECDSA-WITH-AES-128-CCM-8	ECDHE-ECDSA-AES128-CCM8
TLS-ECDHE-ECDSA-WITH-AES-256-CCM-8	ECDHE-ECDSA-AES256-CCM8

Table 4 - CHACHA20_POLY1305

NIST	OpenSSL equivalent
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ECDHE-RSA-CHACHA20-POLY1305
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ECDHE-ECDSA-CHACHA20-POLY1305
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	DHE-RSA-CHACHA20-POLY1305

4.1 TPP and VUB authentication

For the authentication of the ASPSP as a resource provider, the eIDAS-based site authentication certificate will be used or EV certificate. For the authentication of the TPP as a client, the eIDAS-based site authentication certificate will be used or EV certificate.

Technical parameters of the certificate RSA2048 + / SHA256 or ECC certificate [prime256v1, secp256r1, NIST P-256, secp384r1, NIST P-384]

All TPP requests, where technically possible, must be protected by TLS protocol with mutual authentication where PKI certificates are used.

VUB uses Authorization code grant flow according to RFC 6749, Section 4.1 of OAuth framework for APIs, where PSU authentication is necessary, and Client credentials grant flow according to RFC 6749, Section 4.4. of OAuth framework, where PSU participation is not needed.

Table 5 - used OAUTH2 grants flows

	OAUTH2 grant	OAUTH2 scope
Account information	Authorization Code	AISP
Accounts transactions	Authorization Code	AISP
Balance check	Client Credentials	PIISP, PISP
Payment initiation and submission	Authorization Code	PISP
Payment status	Client Credentials	PISP

For APIs, where Authorization code grant flow is used for a TPP access, the TPP client must obtain Authorization code through SCA process made by PSU.

TPP Client must provide Authorization code with Client credentials, given by a Client registration/enrolment process. The Client credentials consists of **client_id** and **client_secret** and is used for automated communication with the VUB to obtain valid **access_token** and **refresh_token**.

For APIs, where Client credentials grant flow is used instead of Authorization code grant flow, TPP client can obtain **access_token** by using Client credentials that consists of **client_id** and **client_secret**.

5 AUTHORIZATION/OAUTH2 CALLS

The request is an OAUTH 2.0. Authorization Code Grant, or Client Credentials grant with PKCE extension (requesting for Code)

5.1 Authorization Code grant: Authorize

TPP does not call /authorize service, this is the last step carried out by VUB SCA process. For TPP, only response is relevant, because response attributes are returned as http query strings to specified TPP redirect URL.

5.1.1 Response (only new fields are listed)

Table 6 - Authorize response attributes

Attribute	Optionality	Type	Description
Code			Authorization code
State			Attribute state from TPP request

5.1.2 Error codes

Error codes are defined according to RFC 6749, Section 4.1.2.1

Sample response

```
HTTP/1.1 302 Found
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=UTF-8
Location: https://www.mymultipay.sk/start?
        code=ba484c02-a85d-4cf1-8253-4b971d2bd726&
        state=notset
```

The PSU is now redirected to the TPP URL with Authorization code and state parameters in URL.

5.2. Authorization Code grant: Token

The TPP will now possess the Authorization code parameter from the VUB. TPP will proceed to obtain an Access Token and Refresh token from the ASPSP. TPP will present its Authorization Code together with **client_id** and **client_secret** in request message.

The Access Token is required by the TPP in order to access PSU Account information. The TPP scope is already associated with the Authorization Code generated in the previous step.

Endpoint: POST https://api.vub.sk/**token**

5.2.1 Request

Table 7 - Token request attributes

Attribute	Optionality	Type	Description
grant_type	Mandatory		Under the existing OAuth2 definition, this value will be the **authorization_code** if the TPP requested **refresh_token**.
client_id	Mandatory		Client ID (Api Key) generated for TPP Application during enrolment process.
client_secret	Mandatory		Client (Shared) secret obtained by the enrolment process.
scope	Mandatory		Permissions for which the token is issued. AISP or PISP.
redirect_uri	Mandatory		The redirect URL matches the URL passed in the authentication request and URL registered for TPP application in enrolment process.
code	Mandatory		Authorization code returned from the code grant.

5.2.2 Response

Table 8 - Token response attributes

Attribute	Optionality	Type	Description
access_token	Mandatory		Short-term (e.g. 60 seconds, one-time) token. This token serves to authorize TPP request on ASPSP API.
token_type	Mandatory		Type of token „Bearer“
expires_in	Mandatory		The remaining time to expiration of **access_token** - in seconds.

refresh_token	Optional		Long-term token (e.g. 90days) issued as a replacement for **authorization_code**.
scope	Optional		List of permissions separated by the space for which the token is issued.

5.2.3 Error codes

Error codes are defined according to RFC 6749, Section 5.2

Sample request

```
POST https://api.vub.sk/token HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded;charset=iso-8859-1
Content-Length: 214
Host: api.vub.sk
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.2 (Java/1.8.0_181)

grant_type=authorization_code&
client_id=l75880dbrt529f4ebab9ece06a652c4ac1&
client_secret=7df903368b7f444c8740b63a6f59p8l8&
scope=AISP&
redirect_uri=https://www.mymultipay.sk/start&
code=59ec296a-d000-43ce-ae0f-75b15a0b07ee
```

Sample response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
"access_token":"ea92fb9c-737a-4375-a905-aaa87fb9c083",
"token_type":"Bearer",
"expires_in": 60,
"refresh_token":"af8fe9b5-8161-4122-864b-d1a286aeb9fc",
"scope":"AISP"
}
```

5.3. Client credentials grant: Token

For calls, where PSU is not required to participate, one-time **access_token** according to 5.2.1 or according to RFC 6749, section 4.4, (Client Credentials Grant) is used with valid **client_id** and **client_secret**.

5.3.1 Request

Table 9 - Token request attributes

Attribute	Optionality	Type	Description
grant_type	Mandatory		Client Credentials exclusively to assign one-time **access_token**
client_id	Mandatory		Client ID (Api Key) generated for TPP Application during enrolment process.
client_secret	Mandatory		Client (Shared) secret obtained by the enrolment process.
scope	Mandatory		Permissions for which the token is issued. "PIISP","PISP"

5.3.2 Response

Table 10 - Token response attributes

Attribute	Optionality	Type	Description
access_token			Short-term (e.g. 60 seconds, one-time) token. This token serves to authorize TPP request on ASPSP API.
token_type			Type of token „Bearer“
expires_in			The remaining time to expiration of **access_token** - in seconds.
scope			List of permissions separated by the space for which the token is issued. "PIISP","PISP"

5.3.3 Error codes

Error codes are defined according to RFC 6749, Section 5.2

Sample request

```
POST https://api.vub.sk/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded;charset=iso-8859-1
Content-Length: 132
Host: api.vub.sk
Connection: Keep-Alive
```

```
User-Agent: Apache-HttpClient/4.5.2 (Java/1.8.0_181)

grant_type=client_credentials&
client_id= l75880dbrt529f4ebab9ece06a652c4ac1&
client_secret=7df903368b7f444c8740b63a6f59p8l8&
scope=PIISP
```

Sample response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
"access_token":"dd2e242e-e212-4b87-8906-435f725966e2",
"token_type":"Bearer",
"expires_in": 60,
"scope":"PIISP"
}
```

5.4. Refresh token grant: Token

When TPP wants to renew **access_token**, the saved **refresh_token** can be used. Token service can be called with valid **refresh_token**, and TPP will obtain new **access_token** and also new **refresh_token** in response. Note that old **refresh_token** is invalidated through this process, but the new **refresh_token** at the response has all the properties of old one (e.g. remaining time and count validity).

The **access_token** can be used within 60 seconds since obtaining and is bounded to the same PSU, IBAN and scope as **refresh_token**, from which it was renewed.

Please note, that security rules allow the use of refresh_token grant for the AISP scope only. It is possible to use this service 4 times a day to get new **access_token** within validity of **refresh_token**. After 4th time of renewal within the same day, PSU has to make new SCA in order to obtain new **refresh_token**. Otherwise new **access_token** can be obtained from the same **refresh_token** day after.

This limit is applied only case when PSU is not present – e.g. TPP is regularly collecting client's data.

When the PSU is present and API calls are initiated upon his interaction, http request header **PSU-Presence** can be filled with value "true" and limit for 4 calls per day is turned off. In this case, PSU-IP-Address, PSU-Device-OS and PSU-User-Agent must be filled as well to match with user model in bank for security reasons.

When using **PSU-Presence** = true for development, please refer to us at developers@vub.sk with list of IBAN accounts, that will be used for testing, so we can add them to whitelist, as we are monitoring usage of this header for fraud attempts. We are obliged to report any attempt of fraud misuse of this header to supervising authority.

Endpoint: POST https://api.vub.sk**/token**

5.4.1 Request

Table 11 - Token request attributes

Attribute	Optionality	Type	Description
grant_type	Mandatory	String	According to OAuth2 definition, this value will be "**refresh_token**".
refresh_token	Mandatory	String	Valid **refresh_token** for which the exchange takes place.
scope	Mandatory	String	The scope of the access request.
PSU-Presence	Optional	Boolean	Flag, whether PSU is present during the API call or not. Default value "false".
PSU–IP-Address	Optional*	String	Identifier of a customer's IP address from which he/she is connected to the TPP infrastructure. It might be in the format of IPv4 o IPv6 address. ASPSP shall indicate which values are acceptable. Must be filled in when PSU-Presence = true
PSU-Device-OS	Optional*	String	A customer's device and/or operating system identification from which he/she is connected

			to the TPP infrastructure. Must be filled in when PSU-Presence = true
PSU-User-Agent	Optional*	String	A customer's web browser of other client device identification from which he/she is connected to the TPP infrastructure. Agent header field of the http request between PSU and TPP.) Must be filled in when PSU-Presence = true

5.4.2 Response

Table 12 - Token response attributes

Attribute	Optionality	Type	Description
access_token	Mandatory	String	Short-term (e.g. 60 seconds, one-time) token. This token serves to authorize TPP request on ASPSP API.
token_type	Mandatory	String	Type of token „Bearer“
expires_in	Mandatory	Number	The remaining time to expiration of **access_token** - in seconds.
refresh_token	Optional	String	New **refresh_token** that replaces the old one from request.

5.4.3 Error codes

Error codes are defined according to RFC 6749, Section 5.2

Sample request (PSU-Presence = true)

```
POST https://api.vub.sk/token HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded;charset=iso-8859-1
PSU-Presence: true
PSU-Device-OS: d56a1c30-a012-4985-8785-a0dcd1fc9d1e
License_number: 31320155
PSU-IP-Address: 10.1.148.131
PSU-User-Agent: 0bff3ed2-b3c7-443e-80b9-1c3fc4480d07
Content-Length: 167
Host: api.vub.sk
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.2 (Java/1.8.0_181)

grant_type=Refresh_token&
client_id=l7590fc340303d4217938ce62ba7e13916&
client_secret=22cab56be73f45a1a26a40983d1a4c88&
refresh_token=ed5e86b1-4f11-442b-b0d8-c18c80c2770f
```

Sample request (PSU-Presence = false)

```
POST https://api.vub.sk/token HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded;charset=iso-8859-1
PSU-Presence: false
License_number: 31320155
Content-Length: 167
Host: api.vub.sk
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.2 (Java/1.8.0_181)

grant_type=Refresh_token&
client_id=l7590fc340303d4217938ce62ba7e13916&
client_secret=22cab56be73f45a1a26a40983d1a4c88&
refresh_token=40eee74f-74c9-4e64-84e1-a4e3fee959b8
```

Sample response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
"access_token":"ea92fb9c-737a-4375-a905-aaa87fb9c083",
"token_type":"Bearer",
"expires_in": 60,
"refresh_token":"af8fe9b5-8161-4122-864b-d1a286aeb9fc",
"scope":"AISP"
}
```

6 ACCOUNT INFORMATION SERVICE PROVIDER

Chapter defines list of methods and alternative of flows provided for AISPs.

Prerequisites:

a) The TPP is registered for the AISP role and valid AISP scope
b) The TPP has been successfully checked and authenticated
c) The TPP has presented its “OAuth2 Authorization Code Grant” access token which allows the ASPSP to identify the relevant PSU

Endpoint	Method	Description
/api/v1/**accounts/information**	POST	**Account information** – service provide information and balances related to an account
/api/v1/**accounts/transactions**	POST	**Account transactions** – service provide list of transactions in defined date and page range related to an account

6.1 Header definition

6.1.1 Request header definition

Table 13 - Account information request header attributes

Attribute	Optionality	Type	Description
Content-Type	Mandatory	String	application/json or application/xml
Authorization	Mandatory	String	Authorization is defined in RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage
Request-ID	Mandatory	String	An unique identifier of a particular request message. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Correlation-ID	Optional	String	An unique correlation identifier correlates the request and the response messages as a pair especially useful for audit logs. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Process-ID	Optional	String	Identifier of a business or technical process to what the set of requests and response pairs are organized (e.g. paging of transaction history should have same Process-ID). Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
PSU–IP-Address	Mandatory	String	Identifier of a customer's IP address from which he/she is connected to the TPP infrastructure. It might be in the format of IPv4 o IPv6 address. ASPSP shall indicate which values are acceptable.
PSU-Device-OS	Mandatory	String	A customer's device and/or operating system identification from which he/she is connected to the TPP infrastructure.
PSU-User-Agent	Mandatory	String	A customer's web browser of other client device identification from which he/she is connected to the TPP infrastructure. Agent header field of the http request between PSU and TPP.)
PSU-Geo-Location	Optional	String	The GPS coordinates of the current customer's location in the moment of connection to the TPP infrastructure. (Required GPS format: Latitude, Longitude)
PSU-Last-Logged-Time	Optional	DateTime	Last date and time when user was logged to TPP app (RFC3339 format)
License_number	Optional	String	License number, assigned by local authority, which can be used for the lookup in local authority register (must be same as the one provided within EV (Extended validation) client certificate, which is used for mutual TLS)

State	Optional	String	To test the services in MOCK mode, this parameter should be set to "PSD2_TEST"

6.1.2 Response header definition

Table 14 - Account attributes response header attributes

Attribute	Optionality	Type	Description
Content-Type	Mandatory	String	application/json or application/xml
Response-ID	Mandatory	String	An unique identifier of a particular request message. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Correlation-ID	Optional	String	An unique correlation identifier correlates the request and the response messages as a pair especially useful for audit logs. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Process-ID	Optional	String	Identifier of a business or technical process to what the set of requests and response pairs are organized (e.g. paging of transaction history should have same Process-ID). Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).

HTTP AISP Request header example:
Header

```
Content-Type: application/json
Authorization: Bearer IDWJJBCHQ5DZJWEMO7ZWM4DLYWOFWKXX
Request-ID: c2c48fc8-1f79-4934-a47b-56d61a28f351
Correlation-ID: 292163f5-4eee-4447-9292-5672fdf0013b
Process-ID: 4b88bf95-e129-42b8-a17d-1d2379810fbe
PSU-Last-Logged-Time: 2017-07-31T14:54:32+01:00
PSU–IP-Address: 192.168.0.100
PSU-Device-OS: iOS 11
PSU-User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36
PSU-Geo-Location: 48.145745, 17.116062
License_number: 12345678
```

HTTP AISP Response header example:
Header

```
Content-Type: application/json;charset=UTF-8
Response-ID: ac30869e-29e2-40f7-83fb-ed1c6bdde216
Correlation-ID: 292163f5-4eee-4447-9292-5672fdf0013b
Process-ID: 4b88bf95-e129-42b8-a17d-1d2379810fbe
```

6.2 AISP: Account information

The operation provides the relevant data about PSU account identified by IBAN and two types of account balances: Interim booked and interim available balance. Only AISP is allowed to use this endpoint.

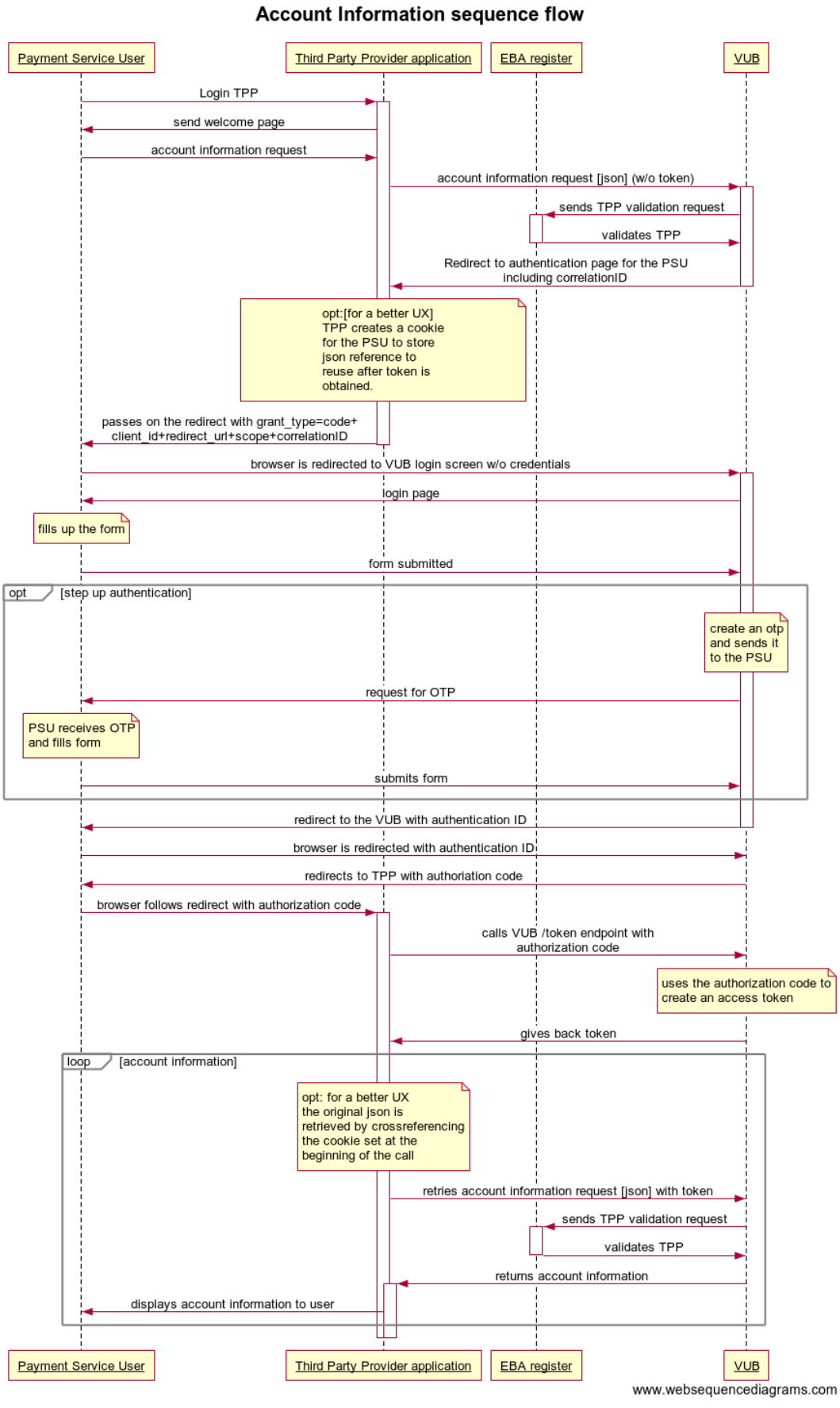

Figure 2 - Account information flow

TPP -> VUB: MTLS account information request [json] (w/o token)	
Description	The TPP invoke the services without access token. The Gateway start the validation of the request.
Endpoint	POST /api/v1/accounts/information
Query parameters	client_id redirect_uri

6.2.1 Request without token

Request format: JSON

Attributes structure Level 1	Optionality	Type	Description
iban	Mandatory	String [34]	International Bank Account Number (IBAN)

Request example:

```
{
  "iban": "SK6807200002891987426353"
}
```

6.2.2 Response without token

Response format: JSON

Attribute	LEVEL1	Optionality	Type	Description
authentication_url	Response	Mandatory	String (64)	**URL** corresponding to the VUB Identity Provider for SCA Process
correlation_id	Response	Mandatory	String (35)	Gateway Correlation ID for association with subsequent flows

Response example:

```
{
“authentication_url”:”https://idp.vub.sk/login”,
“correlation_id”:”1111111111111111111”
}
```

TPP -> Gateway: MTLS account information request [json] (with token)	
Description	The TPP invoke the services with access token.
Endpoint	POST /api/v1/accounts/information

Request header: Authorization: Bearer <access_token>

6.2.3 Request with token

Table 15 - Account information request attributes

Attributes structure Level 1	Optionality	Type	Description
iban	Mandatory	String [34]	International Bank Account Number (IBAN)

Sample request

```
{
  "iban": "SK6807200002891987426353"
}
```

6.2.4 Response with token

Table 16 - Account information response attributes

Attributes structure Level 1	Level 2	Level 3	Optionality	Type	Description
account	*name*		Mandatory	String [70]	**Account name** - usually client name
account	*productName*		Optional	String [70]	**Product name** - commercial product designation
account	*type*		Optional	Enum	**Account type** is enumeration: ISO 20022 - Cash Account Type Code e.g. (CACC - Current account)
account	*baseCurrency*		Mandatory	String [3]	**Account currency** (currency code according to ISO 4217 - 3 capital letters)
balances	*typeCodeOrProprietary*		Mandatory	Enum	**Balance type** is enumeration: ISO 20022 - Balance Type Code. Following balances mandatory are published: - ITBD (Interim booked balance) - ITAV (Interim available balance)
balances	*amount*	*value*	Mandatory	Number Float [12.2]	**Balance amount.** Numeric value of the amount as a fractional number. The fractional part has a maximum of two digits
balances	*amount*	*Currency*	Mandatory	String [3]	**Balance currency** (currency code according to ISO 4217 - 3 capital letters)
balances	*creditDebitIndicator*		Mandatory	Enum	**Credit/Debit indicator** is enumeration: - CRDT (Credit) - DBIT (Debit)

balances	*dateTime*		Mandatory	DateTime	**Timestamp of balances** (official local date and time of Slovak republic in RFC 3339 format)

Links to ISO 20022 enumerations:

- Account types: https://www.iso20022.org/standardsrepository/public/wqt/Description/mx/dico/codesets/_a3ed5tp-Ed-ak6NoX_4Aeg_-1826678245
- Balance type: https://www.iso20022.org/standardsrepository/public/wqt/Description/mx/dico/codesets/_bbFhQNp-Ed-ak6NoX_4Aeg_142948041

Sample response

```
{
  "account": {
    "name": "John Doe",
    "productName": "BestAccount",
    "type": "CACC",
    "baseCurrency": "EUR"
  },
  "balances": [
    {
      "typeCodeOrProprietary": "ITBD",
      "amount": {
        "value": 1234.56,
        "currency": "EUR"
      },
      "creditDebitIndicator": "CRDT",
      "dateTime": "2017-07-31T14:54:32+01:00"
    }
  ]
}
```

6.2.5 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

6.3 AISP: Account transactions

This operation provides the list of financial transactions performed on a customer's bank account within a date period. Transaction history will only include transactions that affect the balance (reserved and booked transaction). Transactions will be ordered from the most recent to the oldest. The range of attributes provided for transactions is based on ISO 20 022 - CAMT. Only AISP is allowed to use this endpoint.

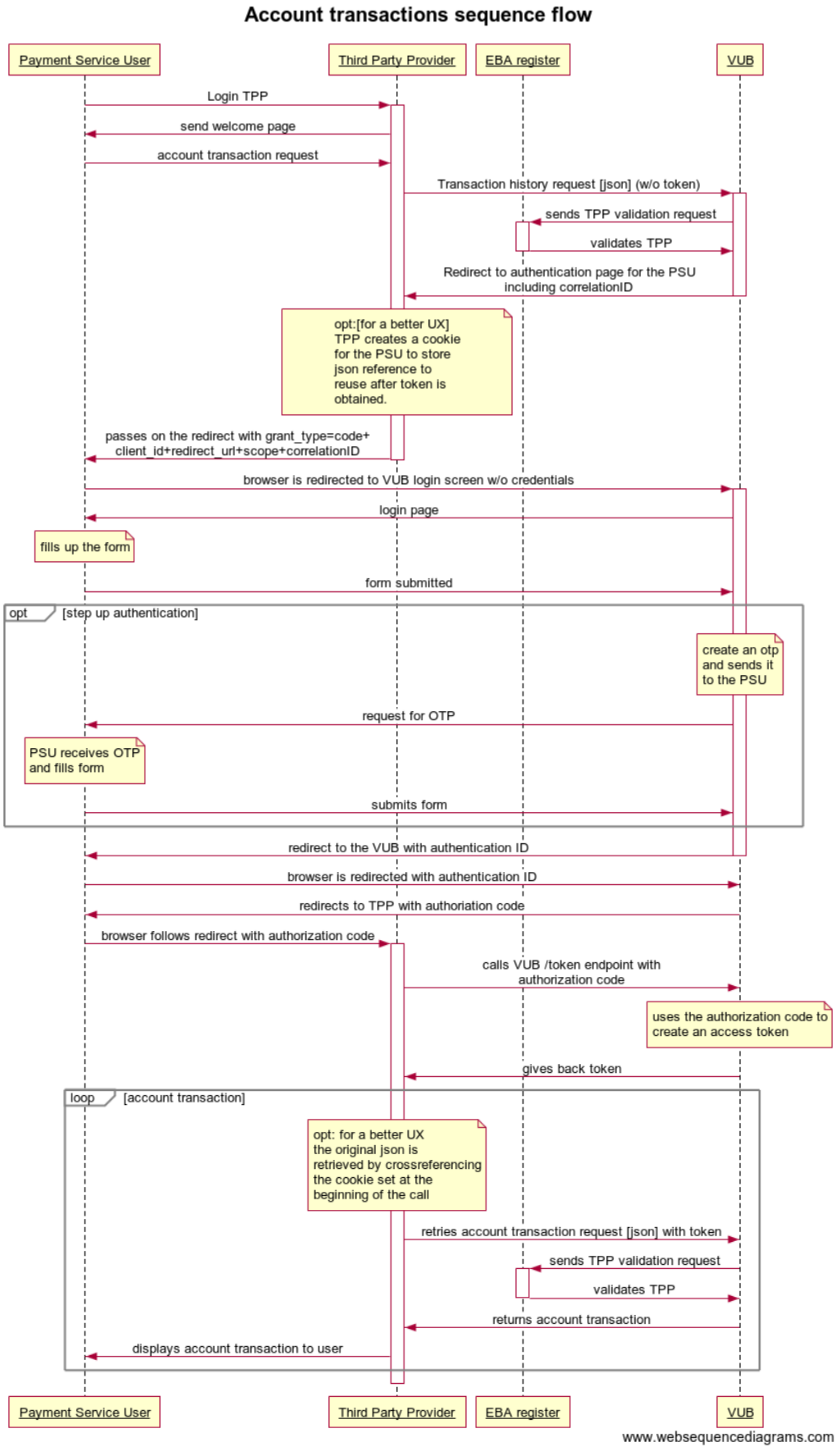

Figure 3 - Transaction history flow

TPP -> Gateway: MTLS account transaction information request [json] (w/o token)	
Description	The TPP invoke the services without access token to get redirection for VUB Identity provider, where PSU can make SCA. The Gateway start the validation of the request.
Endpoint	POST /api/v1/accounts/transactions
Query parameters	client_id redirect_uri

6.3.1 Request without token

Request format: JSON

Table 17 - Account transactions request attributes

Attributes structure Level 1	Optionality	Type	Description
iban	Mandatory	String [34]	International Bank Account Number (IBAN)
dateFrom	Optional	Date	The starting date of a date period for transaction history. Default value is actual day.
dateTo	Optional	Date	The end date of a date period for transaction history. ASPSPs provide transaction's history for at least 13 months. Default value is actual day.
pageSize	Optional	Integer	The number of records included in one page for displaying. Default value is 50 records. ASPSP has to supports at least 100 records on page.
page	Optional	Integer	The sequence number of a page in regards to page size for a record set. Because it starts at number 0, it should be considered as an offset from the beginning from a page set. Default value is 0.
Status	Optional	Enum	Transaction status indicator is enumeration: - BOOK (booked transactions) - INFO (settled transactions) - ALL (all transactions) Default value is ALL

6.3.2 Response without token

Response format: JSON

Table 18 - Account transactions response attributes

Attribute	LEVEL1	Optionality	Type	Description
authentication_url	Response	Mandatory	String (64)	**URL** corresponding to the VUB Identity Provider
correlation_id	Response	Mandatory	String (35)	Gateway Correlation ID for association with subsequent flows

Response example:

```
{
“authentication_url”:”https://idp.vub.sk/login”,
“correlation_id”:”11111111111111111111”
}
```

TPP -> Gateway: MTLS account information request [json] (with token)	
Description	The TPP invoke the services with access token.
Endpoint	POST /api/v1/accounts/transactions

Request header: Authorization: Bearer <access_token>

6.3.3 Request with token

Table 19 - transaction history request attributes

Attributes structure Level 1	Optionality	Type	Description
iban	Mandatory	String [34]	International Bank Account Number (IBAN)
dateFrom	Optional	Date	The starting date of a date period for transaction history. Default value is actual day.
dateTo	Optional	Date	The end date of a date period for transaction history. ASPSPs provide transaction's history for at least 13 months. Default value is actual day.
pageSize	Optional	Integer	The number of records included in one page for displaying. Default value is 50 records. ASPSP has to supports at least 100 records on page.
page	Optional	Integer	The sequence number of a page in regards to page size for a record set. Because it starts at number 0, it should be considered as an offset from the beginning from a page set. Default value is 0.
status	Optional	Enum	Transaction status indicator is enumeration: - BOOK (booked transactions) - INFO (settled transactions) - ALL (all transactions) Default value is ALL

Sample request

```
{
  "dateFrom": "2017-07-31",
  "dateTo": "2017-08-01",
  "pageSize": 50,
  "page": 0,
  "iban": "SK6807200002891987426353",
  "status": "BOOKED"
}
```

6.3.4 Response with token

Collection of information sets about customer's financial transactions executed at their bank account.

Table 20 - - Account transactions response attributes

Attributes structure					Optionality	Type	Description
Level0	Level 1	Level 2	Level 3	Level 4			
pageCount					Optional	Number	Number of pages in the selected range
transactions	amount	value			*Mandatory*	*Number Float [12.2]*	**Transaction amount value** in account currency. Numeric value of the amount as a fractional number.
	amount	*currency*			Mandatory	String [3]	**Transaction amount currency**. Formated in Alphabetic codes from ISO 4712.
	creditDebitIndicator				Mandatory	Enum	**Credit/Debit indicator is enumeration:** - CRDT (Credit) - DBIT (Debit)
	reversalIndicator				Optional	boolean	The flag determining that it is the **reversal transaction** for some previous one.
	status				Mandatory	Enum	**The status of a transaction**, related to the query parameter 'transactionStatus'. Transaction status indicator is enumeration: - BOOK (booked transactions) - INFO (settled transactions
	bookingDate				Mandatory for booked tnx.	Date	**Transaction booking date.** The date of the execustion of the transaction.
	valueDate				Mandatory	Date	**Transaction value date.** The requested date by a bank customer to execute the transaction.

	bankTransactionCode				Optional	String [11]	**The category code** of the transaction type from the SBA's code list.
	transactionDetails	*references*	*accountServicerReference*		Optional	String [35]	**The unique identifier of the transaction generated by a ASPSP** that it should be considered as a ASPSP reference.
	transactionDetails	*references*	*instructionIdentification*		Optional	String [35]	Technical identification of the payment generated by a client.
	transactionDetails	*references*	*endToEndIdentification*		Mandatory in case this attribute is provided by client	String [35]	Unique identification defined by a requestor.
	transactionDetails	*references*	*transactionIdentification*		Optional	String [35]	The payment reference for related fees.
	transactionDetails	*references*	*mandateIdentification*		Mandatory for Direct debit tnx.	String [35]	The **mandate referece** as its reference number.
	transactionDetails	*references*	*chequeNumber*		Optional	String [35]	**For card transactions**, this is the card number in format **** **** **** 1111
	transactionDetails	*counterValueAmount*	*amount*	*value*	Optional	Number Float [12.2]	**Transaction amount value** in account currency.
	transactionDetails	*counterValueAmount*	*amount*	*currency*	Optional	String [3]	**Transaction amount currency**. Formated in Alphabetic codes from ISO 4712.

	transactionDetails	*counterValueAmount*	*currencyExchange*	*exchangeRate*	Optional	Number Float [12.2]	The used **exchange rate for conversion** from the instructed currency to the target account currency.
	transactionDetails	*relatedParties*	*debtor*	*name*	Optional	String [140]	**Name of the debtor**
	transactionDetails	*relatedParties*	*debtorAccount*	*identification*	Optional	String [34]	Unique identification of the **debtor account**, usually IBAN.
	transactionDetails	*relatedParties*	*creditor*	*name*	Optional	String [140]	**Name of the creditor**
	transactionDetails	*relatedParties*	*creditor*	*identification*	Optional	String [35]	The **creditor identifier** (CID) in the direct debit transaction.
	transactionDetails	*relatedParties*	*creditorAccount*	*identification*	Optional	String [34]	Unique identification of the **creditor account**, usually IBAN.
	transactionDetails	*relatedParties*	*tradingParty*	*name*	Optional	String [140]	Name of a third party. For card transaction, this is the name of merchant.
	transactionDetails	*relatedParties*	*tradingParty*	*identification*	Optional	String [35]	Unique identification of a third party. For card transaction, this is ID of merchant.
	transactionDetails	*relatedParties*	*tradingParty*	*merchantCode*	Optional	String [4]	A **Merchant Category Code** (MCC) coordinated by MasterCard and Visa.
	transactionDetails	*relatedAgents*	*debtorAgent*	*financialInstitutionIdentification*	Optional	String [11]	Corresponding identification of a **debtor bank** managing the account, usually Bank Identification Code (BIC).
	transactionDetails	*relatedAgents*	*creditorAgent*	*financialInstitutionIdentification*	Optional	String [11]	Corresponding identification of a **creditor bank** managing the account, usually Bank Identification Code (BIC).
	transactionDetails	*remittanceInformation*			Mandatory in case this attribute is provided by client	String [140]	The text aimed as the information for a receiver of the transaction.

	transactionDetails	*relatedDates*	*acceptanceDateTime*		Optional	Date	**Transaction entry date**. The date of receiving the transaction in a bank.
	transactionDetails	*additionalTransactionInformation*			Optional	String [140]	Bank transaction description.

Links to enumerations:

- The category code of the transaction type from the SBA's code list can be find in document „*xmlstatement_sk_v2.4_2016.docx*“ in section „*4.3.3 Transaction Codes*“: http://www.sbaonline.sk/files/subory/KPS/verejne/xmlstatement_sk_v2.4_2016.docx

Sample response


```
{
  "pageCount": 100,
  "transactions": [
    {
      "amount": {
        "value": 1234.56,
        "currency": "EUR"
      },
      "creditDebitIndicator": "CRDT",
      "reversalIndicator": false,
      "status": "BOOKED",
      "bookingDate": "2017-08-11",
      "valueDate": "2017-08-11",
      "bankTransactionCode": "CO11",
      "transactionDetails": {
        "references": {
          "accountServicerReference": "0e24e604dbf144d5ae64a19b1baedd27",
          "instructionIdentification": "9b76608457de48b2be531bd2804ae0b7",
          "endToEndIdentification": "/VS123/SS456/KS0308",
          "transactionIdentification": "9579749f59a74af5ab1778ac26e42f3b",
          "mandateIdentification": "343ed098e0c74e15b3813a04cf529a1f",
          "chequeNumber": "**** **** **** 1111"
        },
        "counterValueAmount": {
          "amount": {
            "value": 1234.56,
            "currency": "EUR"
          },
          "currencyExchange": {
            "exchangeRate": 1234.56
          }
        },
        "relatedParties": {
          "debtor": {
            "name": "Joe Doe"
```

```
        },
        "debtorAccount": {
          "identification": "SK6807200002891987426353"
        },
        "creditor": {
          "name": "Jane Doe",
          "identification": "c28a41eaee9e4bf78e1b654df1672bc1"
        },
        "creditorAccount": {
          "identification": "SK6807200002891987426353"
        },
        "tradingParty": {
          "name": "Merchant ID",
          "identification": "AAA-GG-SSSS",
          "merchantCode": "3370"
        }
      },
      "relatedAgents": {
        "debtorAgent": {
          "financialInstitutionIdentification": "GIBASKBX"
        },
        "creditorAgent": {
          "financialInstitutionIdentification": "GIBASKBX"
        }
      },
      "remittanceInformation": "Message for the receiver.",
      "relatedDates": {
        "acceptanceDateTime": "2017-08-11"
      },
      "additionalTransactionInformation": "Payment order (EB Sporopay)"
    }
  }
 ]
}
```


6.3.5 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

7 PIISP, PISP: BALANCE CHECK

Following sections describe the technical definition of provided endpoints for PIISPs.

Endpoint	Method	Description
/api/v1/**accounts/balanceCheck**	POST	**Balance check** – service provide information about sufficient balance with the yes/no answer

7.1. Header definition

Table 21 - Balance check request header definition

Attribute	Optionality	Type	Description
Content-Type	Mandatory	String	application/json or application/xml
Authorization	Mandatory	String	Authorization is defined in RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage
Request-ID	Mandatory	String	An unique identifier of a particular request message. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Correlation-ID	Optional	String	An unique correlation identifier correlates the request and the response messages as a pair especially useful for audit logs. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Process-ID	Optional	String	Identifier of a business or technical process to what the set of requests and response pairs are organized (e.g. paging of transaction history should have same Process-ID). Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
PSU–IP-Address	Mandatory	String	Identifier of a customer's IP address from which he/she is connected to the TPP infrastructure. It might be in the format of IPv4 o IPv6 address. ASPSP shall indicate which values are acceptable.
PSU-Device-OS	Mandatory	String	A customer's device and/or operating system identification from which he/she is connected to the TPP infrastructure.
PSU-User-Agent	Mandatory	String	A customer's web browser of other client device identification from which he/she is connected to the TPP infrastructure. Agent header field of the http request between PSU and TPP.)
PSU-Geo-Location	Optional	String	The GPS coordinates of the current customer's location in the moment of connection to the TPP infrastructure. (Required GPS format: Latitude, Longitude)
PSU-Last-Logged-Time	Optional	DateTime	Last date and time when user was logged to TPP app (RFC3339 format)
License_number	Optional	String	License number, assigned by local authority, which can be used for the lookup in local authority register (must be same as the one provided within EV (Extended validation) client certificate, which is used for mutual TLS)

State	Optional	String	If you want to test the services, this parameter should be set to "PSD2_TEST"

Table 22 - Balance check response header definition

Attribute	Optionality	Type	Description
Content-Type	Mandatory	String	application/json or application/xml
Response-ID	Mandatory	String	An unique identifier of a particular request message. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Correlation-ID	Optional	String	An unique correlation identifier correlates the request and the response messages as a pair especially useful for audit logs. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Process-ID	Optional	String	Identifier of a business or technical process to what the set of requests and response pairs are organized (e.g. paging of transaction history should have same Process-ID). Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).

Request header definition

Sample request header

```
Content-Type: application/json
Authorization: Bearer IDWJJBCHQ5DZJWEMO7ZWM4DLYWOFWKXX
Request-ID: c2c48fc8-1f79-4934-a47b-56d61a28f351
Correlation-ID: 292163f5-4eee-4447-9292-5672fdf0013b
Process-ID: 4b88bf95-e129-42b8-a17d-1d2379810fbe
PSU-Last-Logged-Time: 2017-07-31T14:54:32+01:00
PSU–IP-Address: 192.168.0.100
PSU-Device-OS: iOS 11
PSU-User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36
PSU-Geo-Location: 48.145745, 17.116062
License_number: 12345678
```

Sample response header

```
Content-Type: application/json
Response-ID: ac30869e-29e2-40f7-83fb-ed1c6bdde216
Correlation-ID: 292163f5-4eee-4447-9292-5672fdf0013b
Process-ID: 4b88bf95-e129-42b8-a17d-1d2379810fbe
```

The operation provides the resolution whether the balance of a bank customer's account identified by IBAN is sufficient for asked amount or not.

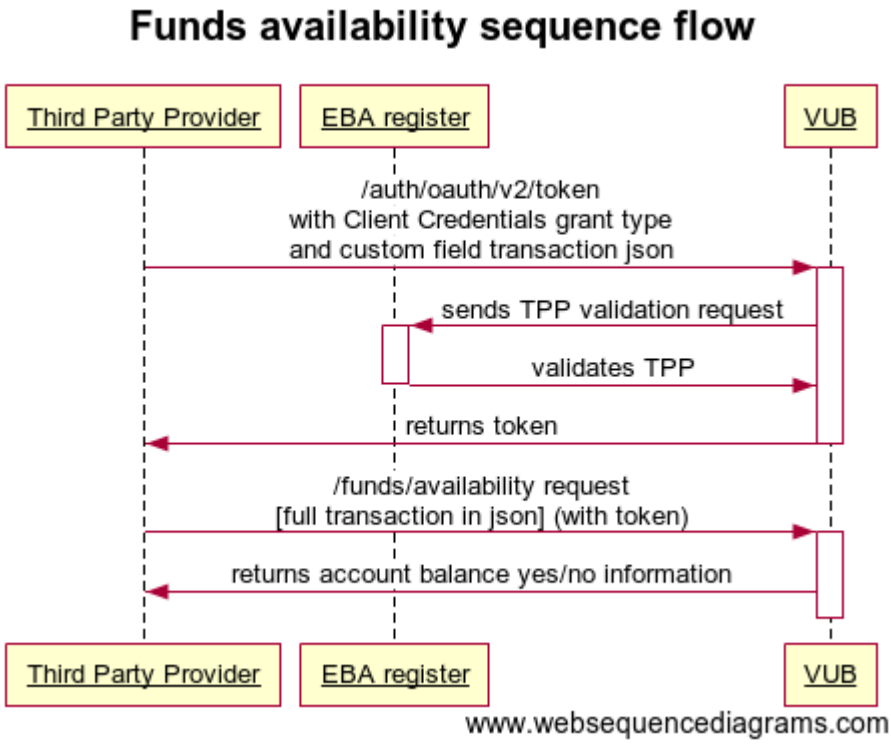

Figure 4 - Balance check flow

Endpoint: POST /api/v1/accounts/balanceCheck

7.1.1 Request

Attributes structure Level 1	Level 2	Level 3	Optionality	Type	Description
instructionIdentification			Mandatory	String	**Technical identification of payment**, generated by the PIISP
creationDateTime			Optional	DateTime	**The date and time** in RFC3339 format at which a particular action has been requested or executed.
iban			Mandatory	String [34]	**International Bank Account Number** (IBAN)
amount	*value*		Mandatory	Number Float [12.2]	**Transaction amount value**. Numeric value of the amount as a fractional number.
amount	*currency*		Mandatory	String [3]	**Transaction amount currency**. Formated in Alphabetic codes from ISO 4712.
relatedParties	*tradingParty*	*identification*	Optional	String [35]	Unique identification of a third party. For card transaction, this is ID of merchant.

relatedParties	*tradingParty*	*name*	Optional	String [140]	Name of a third party. For card transaction, this is the name of merchant.
relatedParties	*tradingParty*	*address*	Optional	String [70]	**Merchant cummulative address** identification usually containing concatenation of street name, street number, etc.
relatedParties	*tradingParty*	*countryCode*	Optional	String [2]	The two letter **merchant country code** adopted from ISO3166.
relatedParties	*tradingParty*	*merchantCode*	Optional	String [4]	A **Merchant Category Code** (MCC) coordinated by MasterCard and Visa.
references	*chequeNumber*		Optional	String [35]	**For card transactions**, this is the card number in format **** **** **** 1111
references	*holderName*		Optional	String[35]	**Card holder name**

Sample request


```
{
  "instructionIdentification": "9b76608457de48b2be531bd2804ae0b7",
  "creationDateTime": "2017-07-31T14:54:32+01:00",
  "iban": "SK6807200002891987426353",
  "amount": {
    "value": 1234.56,
    "currency": "EUR"
  },
  "relatedParties": {
    "tradingParty": {
      "identification": "AAA-GG-SSSS",
      "name": "Merchant ID",
      "address": "My street 123, MyLand",
      "countryCode": "SK",
      "merchantCode": "3370"
    }
  },
  "references": {
    "chequeNumber": "**** **** **** 1111",
    "holderName": "Jane Doe"
  }
}
```


7.1.2 Response (if no error)

Attributes structure Level 1	Optionality	Type	Description
response	Mandatory	Enum	**response** is enumeration: - APPR (sufficient funds on the account) - DECL (insufficient funds in the account)
dateTime	Mandatory	DateTime	**The date and time** in RFC3339 format at which a particular action has been requested or executed.

Sample response

```
{
  "response": "APPR",
  "dateTime": "2017-07-31T14:54:32+01:00"
}
```

7.1.3 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

8 PAYMENT INITIATION SERVICE PROVIDER

Chapter defines list of services and alternative of flows provided for PISP.

Restriction:

a) PISP can initialize and authorize only single payment order. No bulk/batch payments are allowed.

In following sections, technical definition of provided endpoints for PISP is described.

Endpoints	Method	Description
/api/v1/**payments/standard/iso**	POST	**Standard payment initialization** – service allows to initialize payment in XML format (PAIN.001)
/api/v1/**payments/submission**	POST	**Standard payment submission** – service allows to authorization of initialized payment
/api/v1/**payments/transactionid/{orderId}/status**	GET	**Payment order status** – service provide actual information about initialized payment

8.1. Header definition

Recommended set of request and response headers for PISP endpoints

Table 23 - Payment initiation request header definition

Attribute	Optionality	Type	Description
Content-Type	Mandatory	String	application/json or application/xml
Authorization	Mandatory	String	Authorization is defined in RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage
Request-ID	Mandatory	String	An unique identifier of a particular request message. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Correlation-ID	Optional	String	An unique correlation identifier correlates the request and the response messages as a pair especially useful for audit logs. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Process-ID	Optional	String	Identifier of a business or technical process to what the set of requests and response pairs are organized (e.g. paging of transaction history should have same Process-ID). Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
PSU–IP-Address	Mandatory	String	Identifier of a customer's IP address from which he/she is connected to the TPP infrastructure. It might be in the format of IPv4 o IPv6 address. ASPSP shall indicate which values are acceptable.
PSU-Device-OS	Mandatory	String	A customer's device and/or operating system identification from which he/she is connected to the TPP infrastructure.
PSU-User-Agent	Mandatory	String	A customer's web browser of other client device identification from which he/she is connected to the TPP infrastructure. Agent header field of the http request between PSU and TPP.)
PSU-Geo-Location	Optional	String	The GPS coordinates of the current customer's location in the moment of connection to the TPP infrastructure. (Required GPS format: Latitude, Longitude)
PSU-Last-Logged-Time	Optional	DateTime	Last date and time when user was logged to TPP app (RFC3339 format)
License_number	Optional	String	License number, assigned by local authority, which can be used for the lookup in local authority register (must be same as the one provided within EV (Extended validation) client certificate, which is used for mutual TLS)

State	Optional	String	If you want to test the services, this parameter should be set to "PSD2_TEST"

Table 24 - Payment initiation response header definition

Attribute	Optionality	Type	Description
Content-Type	Mandatory	String	application/json or application/xml
Response-ID	Mandatory	String	An unique identifier of a particular request message. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Correlation-ID	Optional	String	An unique correlation identifier correlates the request and the response messages as a pair especially useful for audit logs. Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).
Process-ID	Optional	String	Identifier of a business or technical process to what the set of requests and response pairs are organized (e.g. paging of transaction history should have same Process-ID). Although it may be arbitrary string, it is strongly recommended to use a Universally Unique Ifentifier (UUID) version 4 form (RFC4122).

Response header

Sample request Header

```
Content-Type: application/json
Authorization: Bearer IDWJJBCHQ5DZJWEMO7ZWM4DLYWOFWKXX
Request-ID: c2c48fc8-1f79-4934-a47b-56d61a28f351
Correlation-ID: 292163f5-4eee-4447-9292-5672fdf0013b
Process-ID: 4b88bf95-e129-42b8-a17d-1d2379810fbe
PSU-Last-Logged-Time: 2017-07-31T14:54:32+01:00
PSU–IP-Address: 192.168.0.100
PSU-Device-OS: iOS 11
PSU-User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36
PSU-Geo-Location: 48.145745, 17.116062
License_number: 12345678
```

Sample response Header

```
Content-Type: application/json
Response-ID: ac30869e-29e2-40f7-83fb-ed1c6bdde216
Correlation-ID: 292163f5-4eee-4447-9292-5672fdf0013b
Process-ID: 4b88bf95-e129-42b8-a17d-1d2379810fbe
```

8.2. PISP: Single Payment - initiation

The operation allows initializing the payment in XML format (PAIN.001). The PISP sends a ISO20022 pain.001 based structure, that specifies the payment activation request that is related to a commercial transaction between a PSU and the merchant.

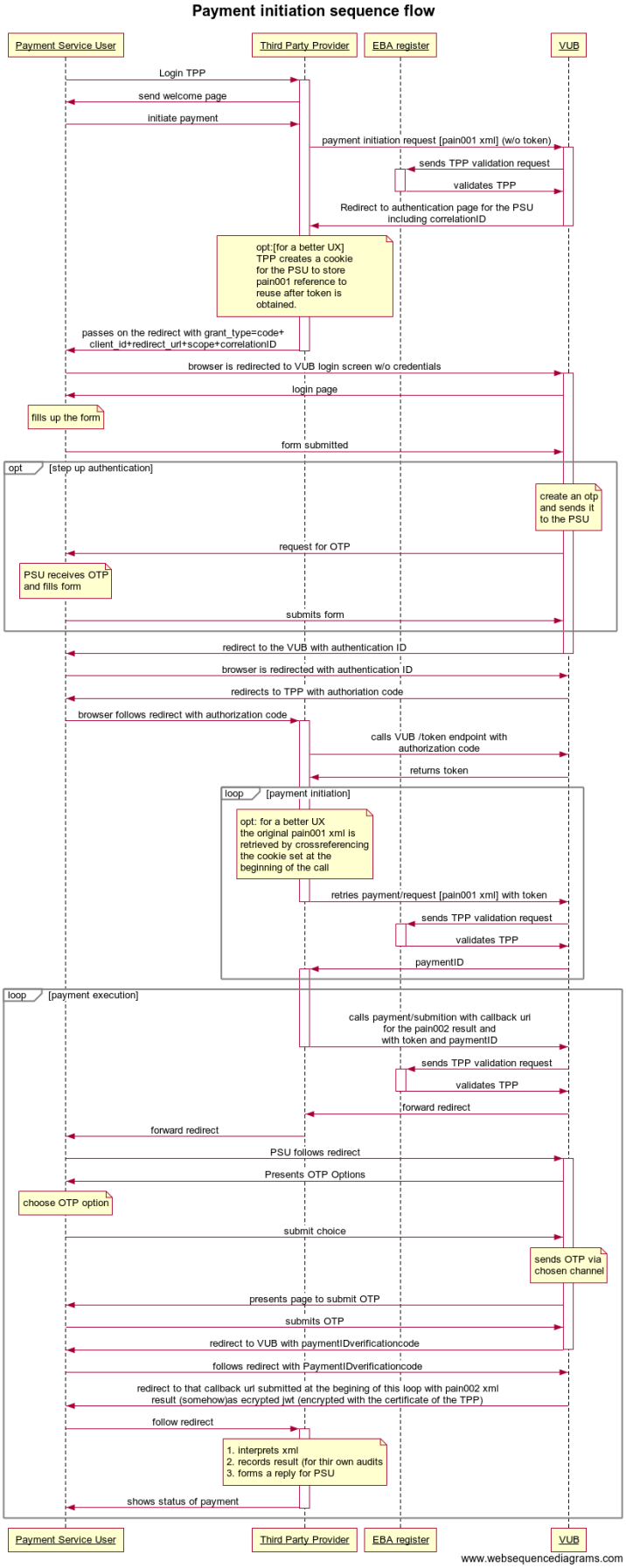

Figure 5 - Payment initiation flow

Endpoint for standard payment:

Endpoint: POST /api/v1/payments/standard/iso

Endpoint for e-commerce payment:

Endpoint: POST /api/v1/payments/ecomm/sba

8.2.1 Request without token

Message contains xml: pain.001.001.03

- Link to message definition: https://www.iso20022.org/documents/general/Payments_Maintenance_2009.zip
- Link to message examples: https://www.iso20022.org/documents/messages/pain/instances/pain.001.001.03.zip

Sample request

```
<?xml version="1.0"?>
 <Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
  <CstmrCdtTrfInitn>
   <GrpHdr>
    <MsgId>ABC/123456/XXX000</MsgId>
    <CreDtTm>2018-01-11T08:30:00</CreDtTm>
    <NbOfTxs>1</NbOfTxs>
    <CtrlSum>1.11</CtrlSum>
    <InitgPty>
     <Nm>Test PSD2</Nm>
      <Id>
       <OrgId>
        <Othr>
         <Id>ffdc2f2d-1288-4212-be381</Id>
        </Othr>
       </OrgId>
      </Id>
    </InitgPty>
   </GrpHdr>
   <PmtInf>
    <PmtInfId>A2018-01-08</PmtInfId>
    <PmtMtd>TRF</PmtMtd>
    <NbOfTxs>1</NbOfTxs>
    <CtrlSum>1.11</CtrlSum>
    <PmtTpInf>
     <InstrPrty>NORM</InstrPrty>
     <SvcLvl>
      <Cd>NURG</Cd>
     </SvcLvl>
    <CtgyPurp>
     <Cd>SEPA</Cd>
    </CtgyPurp>
   </PmtTpInf>
   <ReqdExctnDt>2018-01-11</ReqdExctnDt>
    <Dbtr>
     <Nm>Test PSD2</Nm>
    </Dbtr>
    <DbtrAcct>
     <Id>
      <IBAN>SK3102000000002624762547</IBAN>
     </Id>
    </DbtrAcct>
```

```
        <DbtrAgt>
          <FinInstnId>
            <BIC>SUBASKBX</BIC>
          </FinInstnId>
        </DbtrAgt>
        <CdtTrfTxInf>
          <PmtId>
            <InstrId>A/2018-01-11</InstrId>
            <EndToEndId>/VS1234567890/KS0000/SS1234567890</EndToEndId>
          </PmtId>
          <Amt>
            <InstdAmt Ccy="EUR">1.11</InstdAmt>
          </Amt>
          <ChrgBr>SHAR</ChrgBr>
          <UltmtDbtr>
            <Nm>Test PSD2</Nm>
            <Id>
              <OrgId>
                <Othr>
                  <Id>12345</Id>
                </Othr>
              </OrgId>
            </Id>
          </UltmtDbtr>
          <CdtrAgt>
            <FinInstnId>
              <BIC>TATRSKBX</BIC>
            </FinInstnId>
          </CdtrAgt>
          <Cdtr>
            <Nm>Test PSD2</Nm>
          </Cdtr>
          <CdtrAcct>
            <Id>
              <IBAN>SK3511000000002624762547</IBAN>
            </Id>
          </CdtrAcct>
          <Purp>
            <Cd>ACCT</Cd>
          </Purp>
          <RmtInf>
            <Ustrd>The Second Payment Service Directive</Ustrd>
          </RmtInf>
        </CdtTrfTxInf>
      </PmtInf>
    </CstmrCdtTrfInitn>
  </Document>
```

8.2.2 Response without token

Response format: JSON

Attribute	LEVEL1	Optionality	Type	Description
authentication_url	Response	Mandatory	String (64)	**URL** corresponding to the VUB Identity Provider for SCA Process
correlation_id	Response	Mandatory	String (35)	Gateway Correlation ID for association with subsequent flows

Response example:

```
{
“authentication_url”:”https://idp.vub.sk/login”,
“correlation_id”:”1111111111111111111”
}
```

8.2.3 Request with token

Message contains xml: pain.001.001.03

- Link to message definition: https://www.iso20022.org/documents/general/Payments_Maintenance_2009.zip
- Link to message examples: https://www.iso20022.org/documents/messages/pain/instances/pain.001.001.03.zip

Sample request

```
<?xml version="1.0"?>
 <Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
  <CstmrCdtTrfInitn>
   <GrpHdr>
    <MsgId>ABC/123456/XXX000</MsgId>
    <CreDtTm>2018-01-11T08:30:00</CreDtTm>
    <NbOfTxs>1</NbOfTxs>
    <CtrlSum>1.11</CtrlSum>
    <InitgPty>
     <Nm>Test PSD2</Nm>
      <Id>
       <OrgId>
        <Othr>
         <Id>ffdc2f2d-1288-4212-be381</Id>
        </Othr>
       </OrgId>
      </Id>
    </InitgPty>
   </GrpHdr>
   <PmtInf>
    <PmtInfId>A2018-01-08</PmtInfId>
    <PmtMtd>TRF</PmtMtd>
    <NbOfTxs>1</NbOfTxs>
    <CtrlSum>1.11</CtrlSum>
    <PmtTpInf>
     <InstrPrty>NORM</InstrPrty>
     <SvcLvl>
      <Cd>NURG</Cd>
     </SvcLvl>
    <CtgyPurp>
     <Cd>SEPA</Cd>
    </CtgyPurp>
   </PmtTpInf>
   <ReqdExctnDt>2018-01-11</ReqdExctnDt>
    <Dbtr>
     <Nm>Test PSD2</Nm>
    </Dbtr>
    <DbtrAcct>
     <Id>
      <IBAN>SK3102000000002624762547</IBAN>
     </Id>
    </DbtrAcct>
    <DbtrAgt>
```

```
          <FinInstnId>
            <BIC>SUBASKBX</BIC>
          </FinInstnId>
        </DbtrAgt>
        <CdtTrfTxInf>
          <PmtId>
            <InstrId>A/2018-01-11</InstrId>
            <EndToEndId>/VS1234567890/KS0000/SS1234567890</EndToEndId>
          </PmtId>
          <Amt>
            <InstdAmt Ccy="EUR">1.11</InstdAmt>
          </Amt>
          <ChrgBr>SHAR</ChrgBr>
          <UltmtDbtr>
            <Nm>Test PSD2</Nm>
            <Id>
              <OrgId>
                <Othr>
                  <Id>12345</Id>
                </Othr>
              </OrgId>
            </Id>
          </UltmtDbtr>
          <CdtrAgt>
            <FinInstnId>
              <BIC>TATRSKBX</BIC>
            </FinInstnId>
          </CdtrAgt>
          <Cdtr>
            <Nm>Test PSD2</Nm>
          </Cdtr>
          <CdtrAcct>
            <Id>
              <IBAN>SK3511000000002624762547</IBAN>
            </Id>
          </CdtrAcct>
          <Purp>
            <Cd>ACCT</Cd>
          </Purp>
          <RmtInf>
            <Ustrd>The Second Payment Service Directive</Ustrd>
          </RmtInf>
        </CdtTrfTxInf>
      </PmtInf>
    </CstmrCdtTrfInitn>
  </Document>
```

8.2.4 Response with token

Message contains xml: pain.002.001.03

Table 25 - Payment initiation response attributes

Attribute	XML structure	Optionality	Type	Description

orderId	TxInfAndSts/AcctSvcrRef	Mandatory	String (35)	**OrderId is** Unique reference, as assigned by the account servicing institution, to unambiguously identify the instruction.
status	TxInfAndSts/TxSts	Mandatory	Enum	**Transaction status indicator** is enumeration: - ACTC (AcceptedTechnicalValidation) - ACWC (AcceptedWithChange) - RJCT (Rejected)
reasonCode	TxInfAndSts/StsRsnInf/Rsn	Optional	Enum	ISO 20022 **Rejected Status Reason Code**
statusDateTime	GrpHdr/CreDtTm	Mandatory	Enum	**Transaction entry date**. The date of receiving the transaction in a bank.

- Link to definitions: https://www.iso20022.org/documents/general/Payments_Maintenance_2009.zip
- Link to message examples: https://www.iso20022.org/documents/messages/pain/instances/pain.002.001.03.zip
- Links to enumerations:
  Status Reason Code
  https://www.iso20022.org/sites/default/files/documents/External_code_lists/ExternalCodeSets_4Q2017_05Mar2018_v1.xls, (sheets: 16-StatusReason, 60-ReceivedReason, 61-AcceptedReason, 62-PendingProcessingReason, 63-RejectedReason

Sample response

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.03">
  <CstmrPmtStsRpt>
    <GrpHdr>
      <CreDtTm>2019-05-30T19:13:05</CreDtTm>
    </GrpHdr>
    <OrgnlPmtInfAndSts>
      <TxInfAndSts>
        <AcctSvcrRef>f75ee4db4ccf02568abf834984426b60</AcctSvcrRef>
        <TxSts>ACTC</TxSts>

      </TxInfAndSts>
    </OrgnlPmtInfAndSts>
  </CstmrPmtStsRpt>
</Document>
```

8.2.5 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing

400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

8.3. PISP: Single Payment - submission

The operation provides authorization of initialized payment.

Endpoint: POST /api/v1/payments/submission

8.3.1 Request

The authorization header contains the same **access_token**, that was granted for Payment initiation.

Sample request

```
{
"orderId":"ffdc2f2d-1288-4212-be38-a011838ee051"
}
```

8.3.2 Response (if no error)

Payment submission returns “location” attribute as response http header. This attribute contains VUB identity provider URL for PSU to authorize the transaction.
Upon successful payment authorization, API Gateway sends confirmation document to redirect URL encoded as JSON Web token.

For example
https://www.mymultipay.sk?jwt=eyJ0eXAiOiJKV1QiLCJhb

In specific cases (small amount, trusted IP Address, ...), the payment can be executed without the need of PSU authorization. Security assessments and rules in this case are the same as through other bank channels.

8.3.3 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

8.4. PISP: Payment Status

The operation provides information about processing status of a received payment instruction based on payment orderId identification.

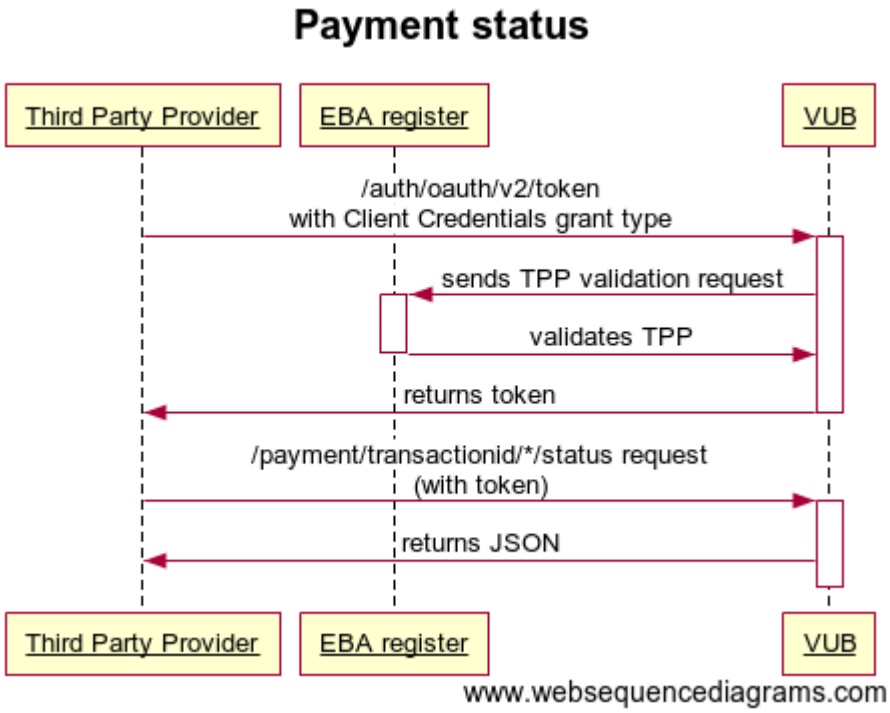

Figure 6 - Payment status flow

Endpoint: GET /api/v1/payments/transactionid/{orderId}/status

8.4.1 Request

Payload is empty.

8.4.2 Response (if no error)

Table 26 - Payment status response attributes

Attributes structure Level 1	Optionality	Type	Description
orderId	Mandatory	String	**OrderId is** Unique reference, as assigned by the account servicing institution, to unambiguously identify the instruction.

status	Mandatory	Enum	**Transaction status indicator** is enumeration: - ACTC (AcceptedTechnicalValidation) - ACWC (AcceptedWithChange) - RJCT (Rejected) - PDNG (Pending) - ACSP (AcceptedSettlementInProcess) - ACSC (AcceptedSettlementCompleted)
reasonCode	Optional	Enum	ISO 20022 **Rejected Status Reason Code**
statusDateTime	Mandatory	DateTime	**The date and time** in RFC3339 format at which a particular action has been requested or executed.

- Links to enumerations:
  Status Reason Code
  https://www.iso20022.org/sites/default/files/documents/External_code_lists/ExternalCodeSets_4Q2017_05Mar2018_v1.xls, (sheets: 16-StatusReason, 60-ReceivedReason, 61-AcceptedReason, 62-PendingProcessingReason, 63-RejectedReason

Sample response

```
{
  "orderId": "248a00f8cdaa465da98455f4df5ad417",
  "status": "ACTC",
  "reasonCode": "MONY",
  "statusDateTime": "2018-08-30T18:42:41.394Z"
}
```

8.4.3 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

Expected flow of payment's states:

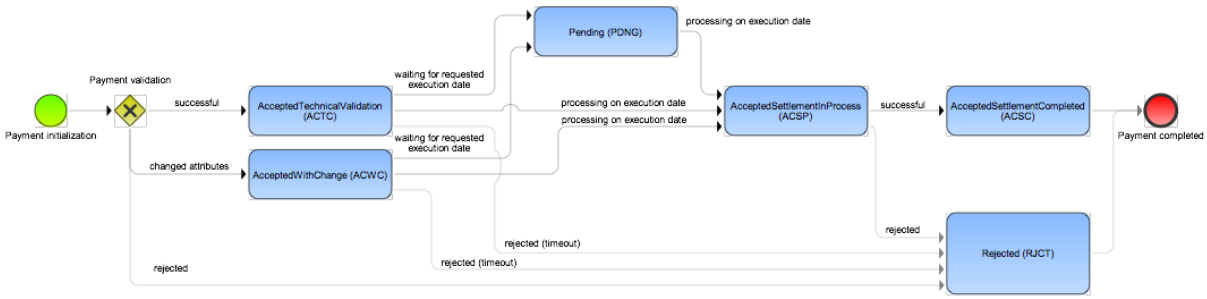

Figure 7 - Flow of payment states

This operation provides following payment status codes:

Table 27 - Payment status codes

Attribute	Description
ACTC	**AcceptedTechnicalValidation** - Authentication and syntactical and semantical validation are successful.
ACWC	**AcceptedWithChange** - Instruction is accepted but a change will be made, such as date or remittance not change
PDNG	**Pending** – payment initiation or individual transaction included in the payment initiation is pending. Further checks and status update will be perform.
ACSP	**AcceptedSettlementInProcess** - All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution.
ACSC	**AcceptedSettlementCompleted** – Settlement on the debtor's account has been completed. Usage: this can by used by the first agent to reports to the debtor that the transaction has been completed. Warning: this status is provided for transaction status reasons, not for financial information. It can only be used after bilateral agreement.
RJCT	**Rejected** - Payment initiation or individual transaction included in the payment initiation has been rejected

- Link to definition: https://www.iso20022.org/standardsrepository/public/wqt/Description/mx/dico/codesets/_Z7RUV9p-Ed-ak6NoX_4Aeg_-481257913

8.5. PISP: Standing Order - initiation

The operation allows initializing the standing order in JSON format. The PISP sends a structure that specifies the standing order activation request.

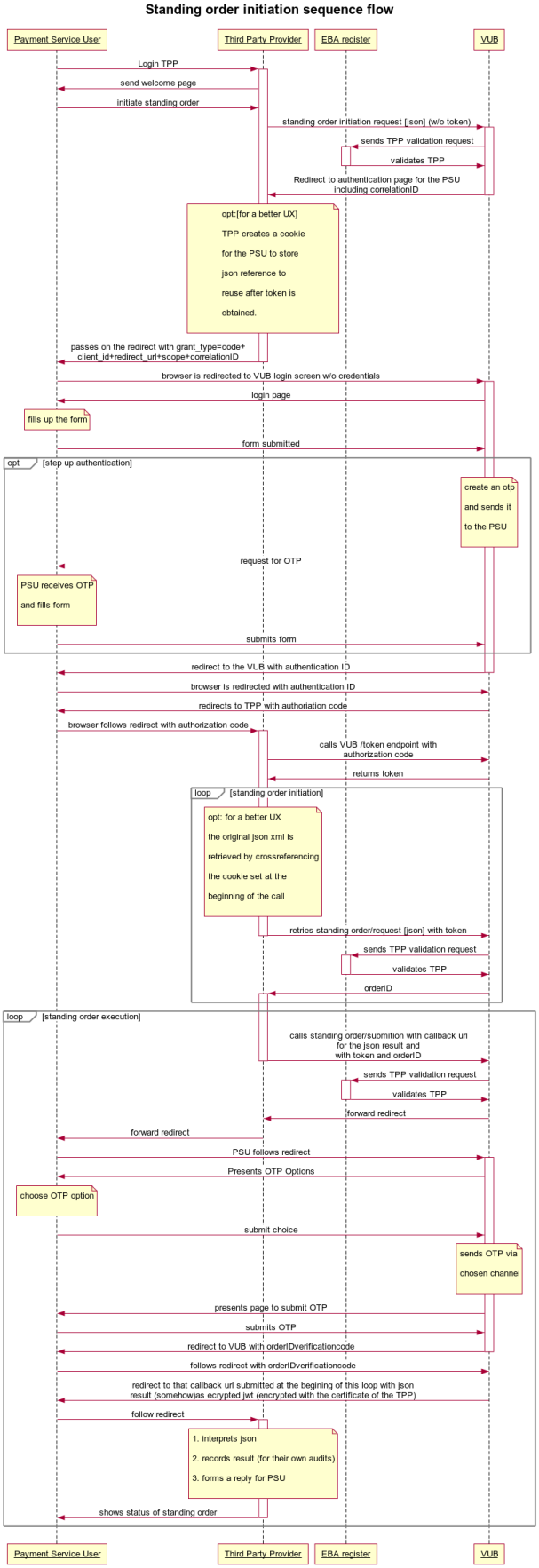

Figure 8 – Standing order initiation flow

Endpoint for standing order initiate:

Endpoint: POST /api/v1/payments/standingOrder

8.5.1 Request without token

Sample request

```
{
"debtorAccount": {
    "debtorIBAN": "SK7502000000002354125541"
},
"creditorAccount": {
    "creditorIBAN": "SK8802000000002215412214",
    "swiftCode": "SUBASK"
},
"creditorName": "Jozef Mrkvicka",
"amount": 1200.00,
"currency": "EUR",
"frequency": "MONTHLY",
"dayInMonthNumber": 15,
"paymentValidity": {
    "dateFrom": "2019-09-14",
    "dateTo": "2022-09-13"
},
"endToEndInformation": "VS11111111/KS0308/SS1565",
"remittanceInformation": "prevod penazi",
"variableSymbol": "11111111",
"constantSymbol": "0308",
"specificSymbol": "1565",
"isLastDayInMonthPayment": false,
"isRealizationDateMoovedSooner": true,
"ultimateDebtorName": "Ivan Mrkvicka",
"ultimateCreditorName": "Jozef Mrkvicka"
}
```

8.5.2 Response without token

Response format: JSON

Attribute	LEVEL1	Optionality	Type	Description
authentication_url	Response	Mandatory	String (64)	**URL** corresponding to the VUB Identity Provider for SCA Process
correlation_id	Response	Mandatory	String (35)	Gateway Correlation ID for association with subsequent flows

Response example:

```
{
“authentication_url”:”https://idp.vub.sk/login”,
“correlation_id”:”11111111111111111111”
}
```

8.5.3 Request with token

This request is the same as /wo token, but it contains http request header *Authorization* with valid OAuth 2.0 access token.

Sample request


```
{
"debtorAccount": {
    "debtorIBAN": "SK7502000000002354125541"
},
"creditorAccount": {
    "creditorIBAN": "SK8802000000002215412214",
    "swiftCode": "SUBASK"
},
"creditorName": "Jozef Mrkvicka",
"amount": 1200.00,
"currency": "EUR",
"frequency": "MONTHLY",
"dayInMonthNumber": 15,
"paymentValidity": {
    "dateFrom": "2019-09-14",
    "dateTo": "2022-09-13"
},
"endToEndInformation": "VS11111111/KS0308/SS1565",
"remittanceInformation": "prevod penazi",
"variableSymbol": "11111111",
"constantSymbol": "0308",
"specificSymbol": "1565",
"isLastDayInMonthPayment": false,
"isRealizationDateMoovedSooner": true,
"ultimateDebtorName": "Ivan Mrkvicka",
"ultimateCreditorName": "Jozef Mrkvicka"
}
```


8.5.4 Response with token

Message contains JSON response with response attributes

Table 28 – Standing order initiate response attributes

Attribute	Optionality	Type	Description
orderId	Optional	String (35)	**OrderId is** Unique reference, as assigned by the account servicing institution, to unambiguously identify the instruction.
status	Mandatory	Enum	**Transaction status indicator** is enumeration: - ACTC (AcceptedTechnicalValidation) - ACWC (AcceptedWithChange) - RJCT (Rejected)
reasonCode	Optional	Enum	ISO 20022 **Rejected Status Reason Code**
statusDateTime	Mandatory	Enum	**Transaction entry date**. The date of receiving the transaction in a bank.

Sample response

```
{
    "orderId": "016878392cae4d568c574a2a6feed4a9",
    "status": "ACTC",
    "reasonCode": "",
    "statusDateTime": "2019-09-14T00:00:00.000+02:00"
}
```


8.5.5 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

8.6. PISP: Standing Order - submission

The operation provides authorization of initialized standing order. Standing order submission is same as submission for standard payment described in 8.3.

The service will decide by the OrderId what kind of transaction is being processed.

Endpoint: POST /api/v1/payments/submission

8.6.1 Request

The authorization header contains the same **access_token**, which was granted for Standing order initiate.

Sample request

```
{
"orderId":"ffdc2f2d-1288-4212-be38-a011838ee051"
}
```

8.6.2 Response (if no error)

Standing order submission returns “location” attribute as response http header. This attribute contains VUB identity provider URL for PSU to authorize the transaction.
Upon successful payment authorization, API Gateway sends confirmation document to redirect URL encoded as JSON Web token.

For example
https://www.mymultipay.sk?jwt=eyJ0eXAiOiJKV1QiLCJhb

8.6.3 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

8.7. PISP: Bulk payment - initiation

The operation allows initializing multiple payments in XML format (PAIN.001). The PISP sends a ISO20022 pain.001 based structure, that specifies the payment activation request that is related to a commercial transaction between a PSU and the merchant.

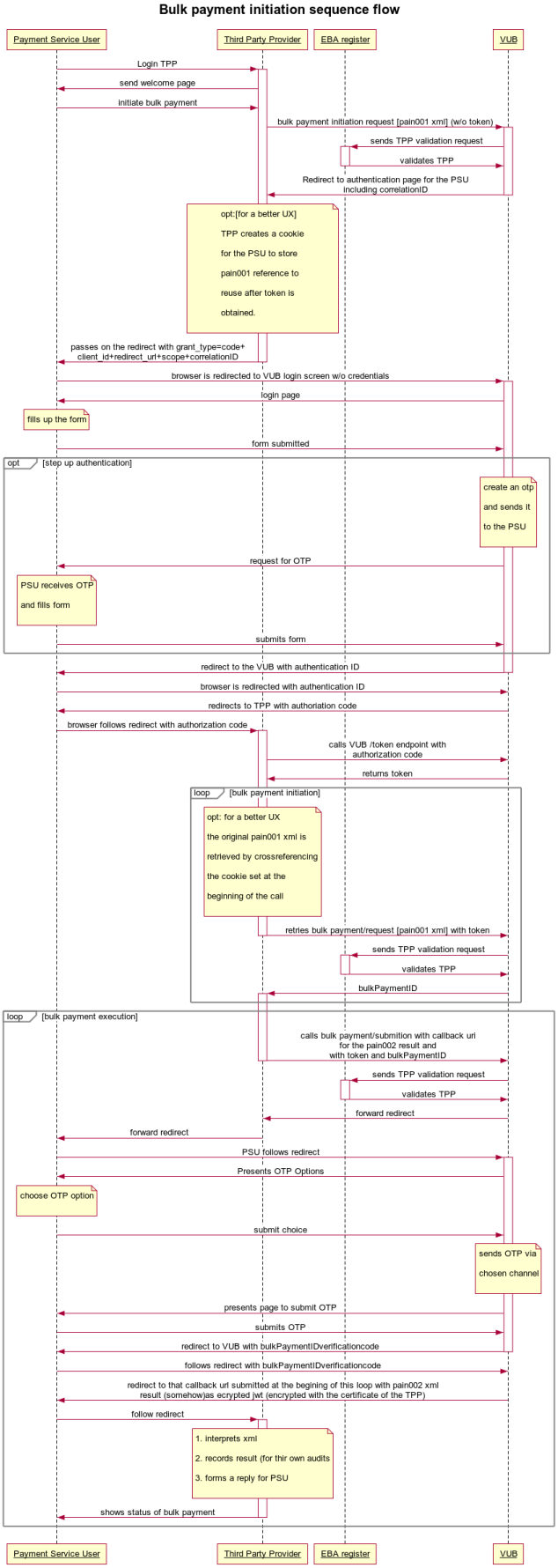

Figure 9 – Bulk payment initiation flow

Endpoint for bulk payment initiate:

Endpoint: POST /api/v1/payments/bulk/standard/iso

8.7.1 Request without token

Message contains xml: pain.001.001.03

- Link to message definition: https://www.iso20022.org/documents/general/Payments_Maintenance_2009.zip
- Link to message examples: https://www.iso20022.org/documents/messages/pain/instances/pain.001.001.03.zip

Sample request

```
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
   <CstmrCdtTrfInitn>
      <GrpHdr>
         <MsgId>bulk6ff5351b352587d803be1413ab8c</MsgId>
         <CreDtTm>2019-11-01T12:08:13.40+01:00</CreDtTm>
         <NbOfTxs>2</NbOfTxs>
         <CtrlSum>3</CtrlSum>
         <InitgPty>
            <Nm>APG</Nm>
         </InitgPty>
      </GrpHdr>
      <PmtInf>
         <PmtInfId>TPP1BULK300000006_1</PmtInfId>
         <PmtMtd>TRF</PmtMtd>
         <NbOfTxs>2</NbOfTxs>
         <CtrlSum>3</CtrlSum>
         <ReqdExctnDt>2019-11-01</ReqdExctnDt>
         <Dbtr/>
         <DbtrAcct>
            <Id>
               <IBAN>SK1302000000190029015556</IBAN>
            </Id>
         </DbtrAcct>
         <DbtrAgt>
            <FinInstnId>
               <BIC>SUBASKBX</BIC>
            </FinInstnId>
         </DbtrAgt>
         <CdtTrfTxInf>
            <PmtId>
               <InstrId>1907300005TPP</InstrId>
               <EndToEndId>NOTPROVIDED</EndToEndId>
            </PmtId>
            <PmtTpInf>
               <InstrPrty>HIGH</InstrPrty>
            </PmtTpInf>
            <Amt>
               <InstdAmt Ccy="EUR">1</InstdAmt>
            </Amt>
            <CdtrAgt>
               <FinInstnId>
                  <BIC>ARTTCZPP</BIC>
               </FinInstnId>
            </CdtrAgt>
```

```
            <Cdtr>
               <Nm>PRIJEMCA</Nm>
               <PstlAdr>
                  <Ctry>SK</Ctry>
                  <AdrLine>ulica</AdrLine>
               </PstlAdr>
            </Cdtr>
            <CdtrAcct>
               <Id>
                  <IBAN>CZ4922000000001016180012</IBAN>
               </Id>
            </CdtrAcct>
            <InstrForDbtrAgt>Instrukcia pre agenta 1</InstrForDbtrAgt>
            <RmtInf>
               <Ustrd>aaaa</Ustrd>
            </RmtInf>
         </CdtTrfTxInf>
         <CdtTrfTxInf>
            <PmtId>
               <InstrId>1207300005TPP</InstrId>
               <EndToEndId>NOTPROVIDED</EndToEndId>
            </PmtId>
            <PmtTpInf>
               <InstrPrty>HIGH</InstrPrty>
            </PmtTpInf>
            <Amt>
               <InstdAmt Ccy="EUR">2</InstdAmt>
            </Amt>
            <CdtrAgt>
               <FinInstnId>
                  <BIC>ARTTCZPP</BIC>
               </FinInstnId>
            </CdtrAgt>
            <Cdtr>
               <Nm>PRIJEMCA</Nm>
               <PstlAdr>
                  <Ctry>SK</Ctry>
                  <AdrLine>ulica</AdrLine>
               </PstlAdr>
            </Cdtr>
            <CdtrAcct>
               <Id>
                  <IBAN>CZ4922000000001016180012</IBAN>
               </Id>
            </CdtrAcct>
            <InstrForDbtrAgt>Instrukcia pre agenta 2</InstrForDbtrAgt>
         </CdtTrfTxInf>
      </PmtInf>
   </CstmrCdtTrfInitn>
</Document>
```

8.7.2 Response without token

Response format: JSON

Attribute	LEVEL1	Optionality	Type	Description
authentication_url	Response	Mandatory	String (64)	**URL** corresponding to the VUB Identity Provider for SCA Process
correlation_id	Response	Mandatory	String (35)	Gateway Correlation ID for association with subsequent flows

Response example:

```
{
“authentication_url”:”https://idp.vub.sk/login”,
“correlation_id”:”1111111111111111111”
}
```

8.7.3 Request with token

This request is the same as /wo token, but it contains http request header *Authorization* with valid OAuth 2.0 access token.

Message contains xml: pain.001.001.03

- Link to message definition: https://www.iso20022.org/documents/general/Payments_Maintenance_2009.zip
- Link to message examples: https://www.iso20022.org/documents/messages/pain/instances/pain.001.001.03.zip

Sample request

```
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
   <CstmrCdtTrfInitn>
      <GrpHdr>
         <MsgId>bulk6ff5351b352587d803be1413ab8c</MsgId>
         <CreDtTm>2019-11-01T12:08:13.40+01:00</CreDtTm>
         <NbOfTxs>2</NbOfTxs>
         <CtrlSum>3</CtrlSum>
         <InitgPty>
            <Nm>APG</Nm>
         </InitgPty>
      </GrpHdr>
      <PmtInf>
         <PmtInfId>TPP1BULK300000006_1</PmtInfId>
         <PmtMtd>TRF</PmtMtd>
         <NbOfTxs>2</NbOfTxs>
         <CtrlSum>3</CtrlSum>
         <ReqdExctnDt>2019-11-01</ReqdExctnDt>
         <Dbtr/>
         <DbtrAcct>
            <Id>
               <IBAN>SK1302000000190029015556</IBAN>
            </Id>
         </DbtrAcct>
         <DbtrAgt>
            <FinInstnId>
               <BIC>SUBASKBX</BIC>
            </FinInstnId>
```

```
            </DbtrAgt>
            <CdtTrfTxInf>
               <PmtId>
                  <InstrId>1907300005TPP</InstrId>
                  <EndToEndId>NOTPROVIDED</EndToEndId>
               </PmtId>
               <PmtTpInf>
                  <InstrPrty>HIGH</InstrPrty>
               </PmtTpInf>
               <Amt>
                  <InstdAmt Ccy="EUR">1</InstdAmt>
               </Amt>
               <CdtrAgt>
                  <FinInstnId>
                     <BIC>ARTTCZPP</BIC>
                  </FinInstnId>
               </CdtrAgt>
               <Cdtr>
                  <Nm>PRIJEMCA</Nm>
                  <PstlAdr>
                     <Ctry>SK</Ctry>
                     <AdrLine>ulica</AdrLine>
                  </PstlAdr>
               </Cdtr>
               <CdtrAcct>
                  <Id>
                     <IBAN>CZ4922000000001016180012</IBAN>
                  </Id>
               </CdtrAcct>
               <InstrForDbtrAgt>Instrukcia pre agenta 1</InstrForDbtrAgt>
               <RmtInf>
                  <Ustrd>aaaa</Ustrd>
               </RmtInf>
            </CdtTrfTxInf>
            <CdtTrfTxInf>
               <PmtId>
                  <InstrId>1207300005TPP</InstrId>
                  <EndToEndId>NOTPROVIDED</EndToEndId>
               </PmtId>
               <PmtTpInf>
                  <InstrPrty>HIGH</InstrPrty>
               </PmtTpInf>
               <Amt>
                  <InstdAmt Ccy="EUR">2</InstdAmt>
               </Amt>
               <CdtrAgt>
                  <FinInstnId>
                     <BIC>ARTTCZPP</BIC>
                  </FinInstnId>
               </CdtrAgt>
               <Cdtr>
                  <Nm>PRIJEMCA</Nm>
                  <PstlAdr>
                     <Ctry>SK</Ctry>
                     <AdrLine>ulica</AdrLine>
                  </PstlAdr>
               </Cdtr>
               <CdtrAcct>
                  <Id>
                     <IBAN>CZ4922000000001016180012</IBAN>
```

```
            </Id>
          </CdtrAcct>
          <InstrForDbtrAgt>Instrukcia pre agenta 2</InstrForDbtrAgt>
        </CdtTrfTxInf>
      </PmtInf>
    </CstmrCdtTrfInitn>
  </Document>
```

8.7.4 Response with token

Message contains xml with bulkOrderId and multiple OrderId's for each individual transaction inside the bulk.

Table 29 – Bulk payment initiate response attributes

Attribute	Optionality	Type	Description
bulkPayment/ bulkOrderId	Optional	String (35)	**bulkOrderId is** Unique reference, as assigned by the account servicing institution, to unambiguously identify the bulk instruction. This ID is used for submission.
bulkPayment/ status	Mandatory	Enum	**Transaction status indicator** is enumeration: - ACTC (AcceptedTechnicalValidation) - ACWC (AcceptedWithChange) - RJCT (Rejected)
bulkPayment/ reasonCode	Optional	Enum	ISO 20022 **Rejected Status Reason Code**
bulkPayment/ statusDateTime	Mandatory	Enum	**Transaction entry date**. The date of receiving the transaction in a bank.
partialPayment/ orderId	Optional	String (35)	**OrderId is** Unique reference, as assigned by the account servicing institution, to unambiguously identify the instruction.

Sample response

```
<Document>
  <bulkPayment>
    <bulkOrderId>587916caa3614806ab2c52ba29d14f9b</bulkOrderId>
    <status>ACTC</status>
    <statusDateTime>2019-11-01T12:00:58.546+01:00</statusDateTime>
  </bulkPayment>
  <partialPayment>
    <orderId>4f744f954c1f4187bd7570f5bf8cfb09</orderId>
    <orderId>cecda0f1999e4af68fa9e2d4a9b99662</orderId>
  </partialPayment>
</Document>
```

Sample response - error

```
<Document>
  <bulkPayment>
```

```
        <status>RJCT</status>
        <reasonCode>DINV</reasonCode>
        <statusDateTime>2019-11-01T12:00:21.742+01:00</statusDateTime>
      </bulkPayment>
    </Document>
```

8.7.5 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

8.8. PISP: Bulk payment - submission

The operation provides authorization of initialized bulk payment. Bulk payment submission is same as submission for standard payment described in 8.3.

The service will decide by the OrderId what kind of transaction is being processed.

Endpoint: POST /api/v1/payments/submission

8.8.1 Request

The authorization header contains the same **access_token**, which was granted for Bulk payment initiate.

Sample request

```
{
"orderId":"ffdc2f2d-1288-4212-be38-a011838ee051"
}
```

8.8.2 Response (if no error)

Standing order submission returns “location” attribute as response http header. This attribute contains VUB identity provider URL for PSU to authorize the transaction.
Upon successful payment authorization, API Gateway sends confirmation document to redirect URL encoded as JSON Web token.

For example
https://www.mymultipay.sk?jwt=eyJ0eXAiOiJKV1QiLCJhb

8.8.3 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

8.9. PISP: Request to cancel Payment - initiation

The operation allows cancellation of standard payment for TPP who created it.

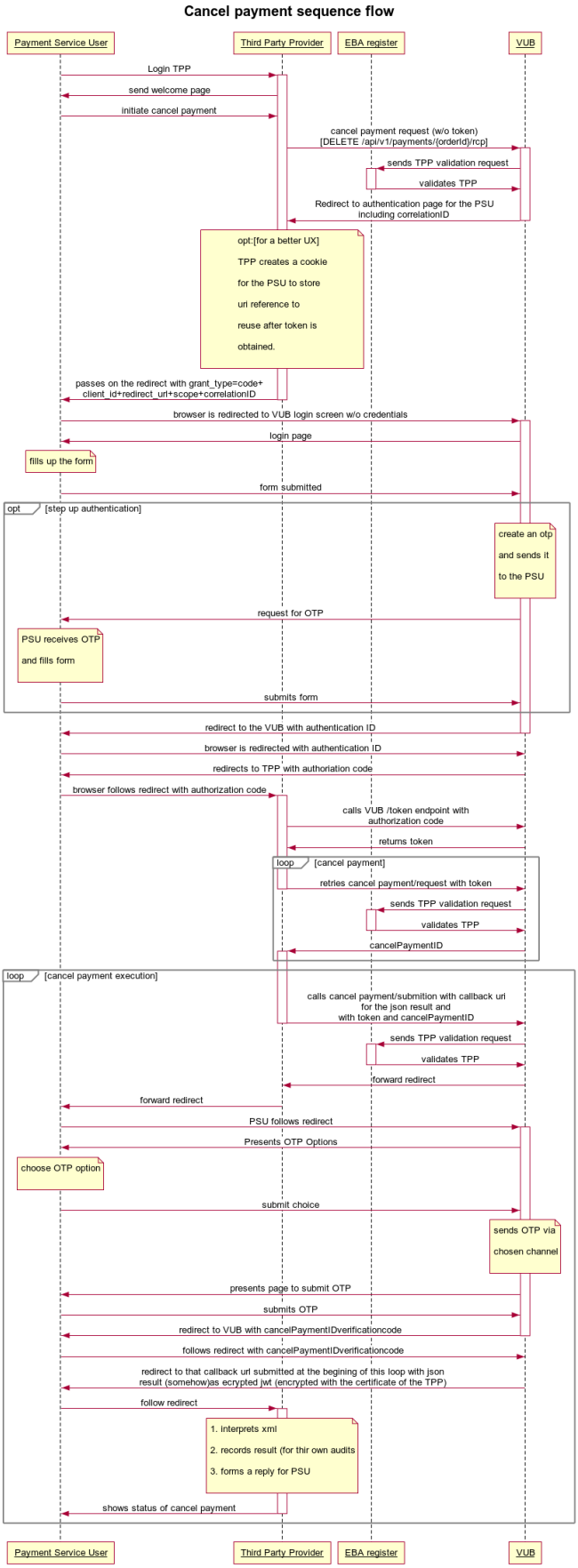

Figure 10 – Request to cancel payment flow

Endpoint for standing order initiate:

Endpoint: DELETE /api/v1/payments/{OrderId}/rcp

8.9.1 Request without token

Sample request

```
DELETE api.vub.sk/api/v1/payments/016878392cae4d568c574a2a6feed4a9/rcp
```

Payload is empty.

Response format: JSON

Attribute	LEVEL1	Optionality	Type	Description
authentication_url	Response	Mandatory	String (64)	**URL** corresponding to the VUB Identity Provider for SCA Process
correlation_id	Response	Mandatory	String (35)	Gateway Correlation ID for association with subsequent flows

Response example:

```
{
“authentication_url”:”https://idp.vub.sk/esa/authorize-payment-cancel”,
“correlation_id”:”2f043731-e663-499e-9537-2829c4be3241”
}
```

8.9.2 Request with token

This request is the same as /wo token, but it contains http request header *Authorization* with valid OAuth 2.0 access token.
Payload is empty.

8.9.3 Response with token

Message contains JSON response with response attributes

Table 30 – Standard payment cancellation response attributes

Attribute	Optionality	Type	Description
orderId	Mandatory	String (35)	**OrderId is** Unique reference, as assigned by the account servicing institution, to unambiguously identify the instruction.

Sample response

```
{
    "orderId": "016878392cae4d568c574a2a6feed4a9"
}
```

8.9.4 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

8.10. PISP: Request to cancel Payment - submission

The operation provides authorization of initialized payment. Submission of payment cancellation is same as submission for standard payment described in 8.3.

The service will decide by the OrderId what kind of transaction is being processed.

Endpoint: POST /api/v1/payments/submission

8.10.1 Request

The authorization header contains the same **access_token**, which was granted for Standard payment cancellation.

Sample request

```
{
"orderId":"ffdc2f2d-1288-4212-be38-a011838ee051"
}
```

8.10.2 Response (if no error)

Standing order submission returns “location” attribute as response http header. This attribute contains VUB identity provider URL for PSU to authorize the transaction.
Upon successful payment authorization, API Gateway sends confirmation document to redirect URL encoded as JSON Web token.

For example
https://www.mymultipay.sk?jwt=eyJ0eXAiOiJKV1QiLCJhb

8.10.3 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

8.11. PISP: Request to cancel Standing Order - initiation

The operation allows cancellation of standing order for TPP who created it.

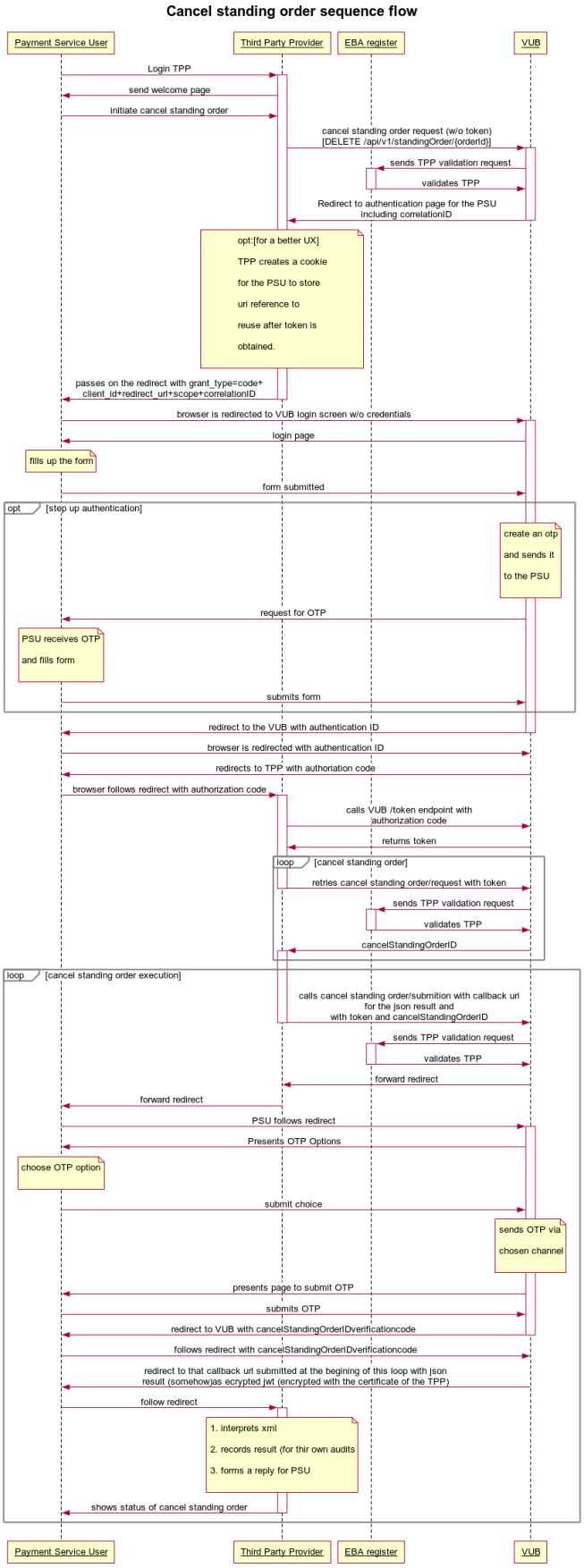

Figure 11 – Request to cancel Standing Order flow

Endpoint for standing order cancellation:

Endpoint: DELETE /api/v1/payments/standingOrder/{OrderId}

8.11.1 Request without token

Sample request

```
DELETE api.vub.sk/api/v1/payments/standingOrder/016878392cae4d568c574a2a6feed4a9
```

Payload is empty.

8.11.2 Response without token

Response format: JSON

Attribute	LEVEL1	Optionality	Type	Description
authentication_url	Response	Mandatory	String (64)	**URL** corresponding to the VUB Identity Provider for SCA Process
correlation_id	Response	Mandatory	String (35)	Gateway Correlation ID for association with subsequent flows

Response example:

```
{
“authentication_url”:”https://idp.vub.sk/esa/authorize-payment-cancel”,
“correlation_id”:”2f043731-e663-499e-9537-2829c4be3241”
}
```

8.11.3 Request with token

This request is the same as /wo token, but it contains http request header *Authorization* with valid OAuth 2.0 access token.
Payload is empty.

8.11.4 Response with token

Message contains JSON response with response attributes

Table 31 – Standing order cancellation response attributes

Attribute	Optionality	Type	Description
orderId	Mandatory	String (35)	**OrderId is** Unique reference, as assigned by the account servicing institution, to unambiguously identify the instruction.

Sample response

```
{
    "orderId": "016878392cae4d568c574a2a6feed4a9"
}
```

8.11.5 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		

8.12. PISP: Request to cancel Standing Order - submission

The operation provides authorization of Standing Order cancellation. Submission for Standing order cancellation is the same as submission for standard payment described in 8.3.

The service will decide by the OrderId what kind of transaction is being processed.

Endpoint: POST /api/v1/payments/submission

8.12.1 Request

The authorization header contains the same **access_token**, which was granted for Standing order cancellation.

Sample request

```
{
"orderId":"ffdc2f2d-1288-4212-be38-a011838ee051"
}
```

8.12.2 Response (if no error)

Submission of Standing order cancellation returns “location” attribute as response http header. This attribute contains VUB identity provider URL for PSU to authorize the transaction.
Upon successful payment authorization, API Gateway sends confirmation document to redirect URL encoded as JSON Web token.

For example
https://www.mymultipay.sk?jwt=eyJ0eXAiOiJKV1QiLCJhb

8.12.3 Error codes

Recommended set of HTTP Status codes and corresponding custom error codes:

HTTP Status	Error code	Description
400	**parameter_missing**	Mandatory parameter is missing
400	**parameter_invalid**	Value of input parameter is not valid
500, 503	**server_error**	Authorization server error.
Rest of HTTP Status codes and error codes are defined according to RFC 6749, Section 5.2		