STET PSD2 API

Documentation

Author: Robache Hervé

Date: 2017-11-15

Version: 1.2.3 (English)

Table of content

1. Introduction

1.1. Context

The revised Payment Service Directive (PSD2) points out some new roles providing services to a Payment Service User (PSU):

- Third Party Providers (TPP) which can be subdivided into three categories
  - Account Information Service Providers (AISP)
  - Payment Initiation Service Providers (PISP)
  - Payment Issuer Instrument Service Providers (PIISP)
- Account Servicing Payment Service Providers (ASPSP).

Each country has to transpose the PSD2, within its own national law.

The PSD2 is completed by a set of documents provided by the European Banking Authority (EBA). Among these documents the Regulatory Technical Standards (RTS) will detail some requirements, for instance on security principles: traceability, strong customer authentication…

As those RTS are not yet finalised, this API and its documentation may be subject to update.

1.2. Mission

STET has been mandated by its shareholders in order to design and provide an open API (Aka STET PSD2 API) that would specify the different interactions between TPPs and ASPSPs for carrying out the different use cases of PSD2 and could be extended to other (non-PSD2) use cases as well.

The STET PSD2 API does not cover:

- Interactions between PSUs and TPP
- Interactions between PSUs and ASPSP
- Registration information management

The technical characteristics of this API are provided within a SWAGGER 2.0 file. The present document purpose is to provide extra-information on this API and to give some interaction samples.

1.3. Licence

This specification is published under the following licence

> “Creative Commons – Attribution 3.0 France (CC BY 3.0 FR)”

This work has been coordinated by STET with the following contributors:

- BNP Paribas
- BPCE
- Le Groupe Crédit Agricole
- Crédit Mutuel – CIC
- La Banque Postale
- Société Générale

2. Business Model

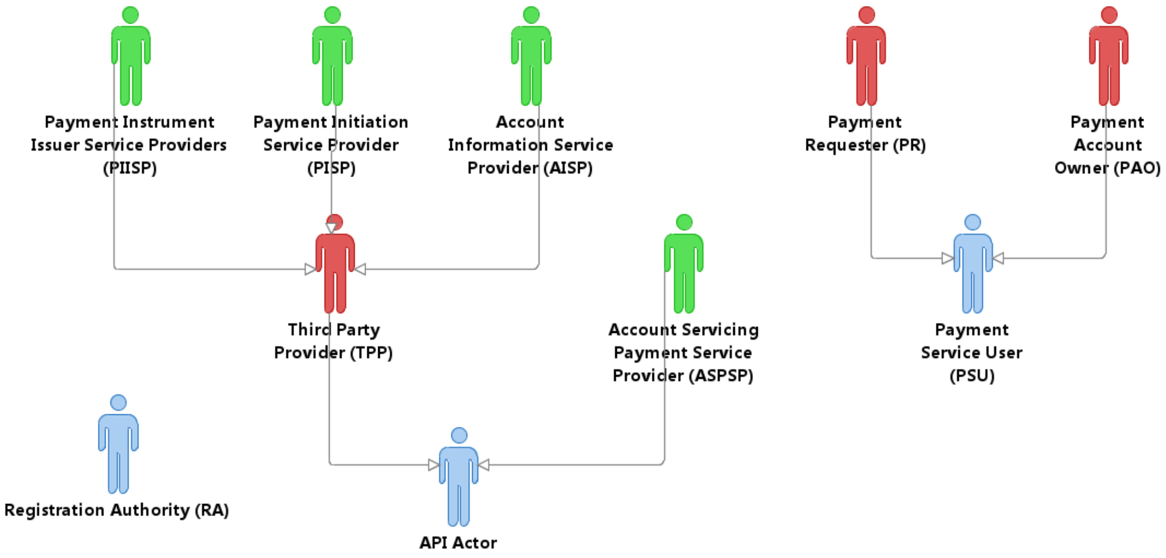
2.1. Actors and Roles

A PSD2 actor is either an entity or a physical person which can endorse one or several roles.

Most of the roles are defined in PSD2. However some extra-roles have been specified for the purpose of the STET PSD2 API during the analysis phase of the project.

Within the following diagram:

- Actors are cyan-coloured
- Pure PSD2 roles are green-coloured
- Specific STET PSD2 API roles are red-coloured


Payment Instrument Issuer Service Providers (PIISP)
Payment Initiation Service Provider (PISP)
Account Information Service Provider (AISP)
Payment Requester (PR)
Payment Account Owner (PAO)
Third Party Provider (TPP)
Account Servicing Payment Service Provider (ASPSP)
Payment Service User (PSU)
Registration Authority (RA)
API Actor

2.1.1. Payment Service User (PSU)

PSUs are the end-users of the services provided by TPPs and ASPSPs.

They are either physical persons or entities (organisations, companies, administrations…).

They do not interact directly with the STET PSD2 API.

A given PSU endorses at least one of the following roles:

- Payment Account Owner (PAO) for one or several accounts held by one or several ASPSPs.
- Payment Requester (PR) asking either for a payment or a coverage check.

2.1.2. API actors

2.1.2.1. Account Servicing Payment Service Provider (ASPSP)

These are Payment Service Providers (PSPs) which are in charge of holding bank accounts for their customers (PSU).

2.1.2.2. Third Party Provider (TPP)

These actors can intermediate between PSUs and ASPSPs, acting on behalf of a PAO or a PR.

On one hand, a given PAO may contract with a TPP in order to use the services provided by this TPP:

- Account Information Services (AISP role) will allow the PAO to get information, through a single interface, about all of its accounts, whatever the ASPSP holding this account.
- Payment Instrument Issuer Service (PIISP role) that will check the coverage of a given payment amount by the PSU’s account.

On the other hand, a PR may also contract with a TPP that will provide the following services:

- Payment Initiation Services for requesting a payment request approval by the PSU and requesting the subsequent execution through a Credit Transfer (PISP role).

2.1.3. Registration Authorities (RA)

RAs are in charge of registering and overviewing the PSD2 actors.

The registration information is the foundation on which each actor can rely in order to know:

- Who is a given actor?
    - Identity
    - Contacts (business, legal, operational…)
    - Insurance coverage
    - Authentication media
        - X.509 certificates
        - Certification chain and services (revocation list, OCSP)
- For which roles this actor has been registered
    - AISP
    - PISP
    - PIISP
    - ASPSP
- Technical characteristics
    - APIs that are provided
    - URLs that are to be used, for test or live processing.

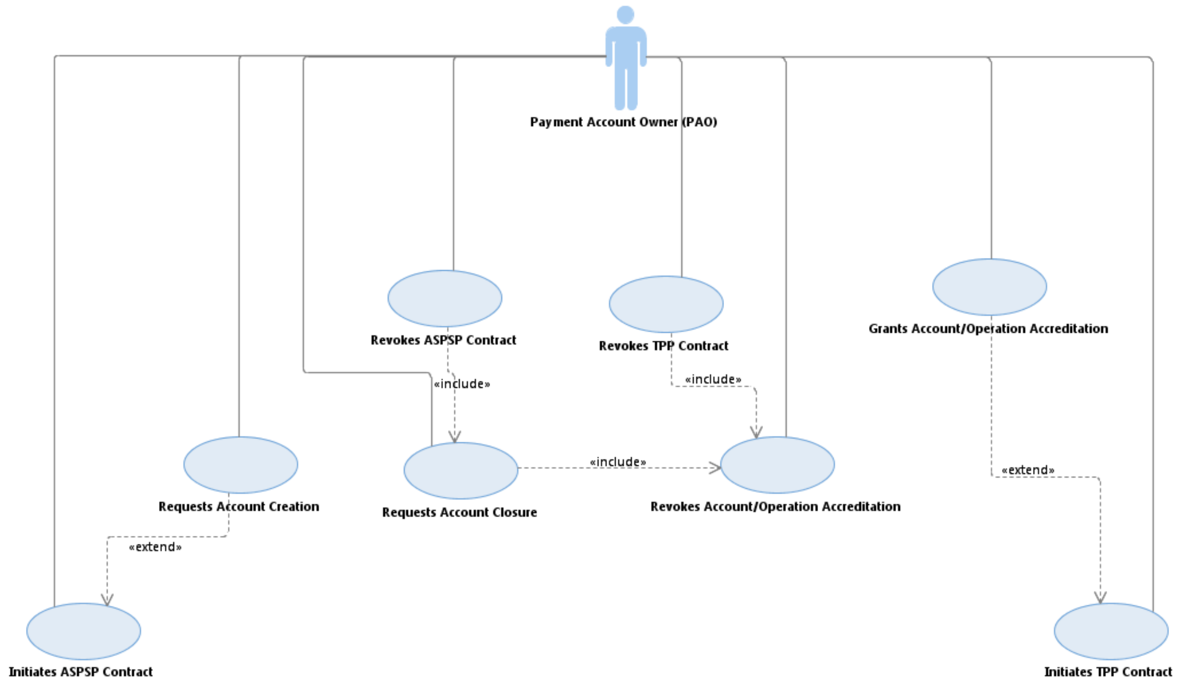
Registration Authorities must keep track of changes for each actor in order to recover the full history of the actor.

2.2. Use cases

Some of the use cases that are listed may be directly implemented by the STET PSD2 API, for they rely on interactions between TPPs and ASPSPs.

Other uses cases are tagged as “NON-API” and are only described for global understanding purpose.

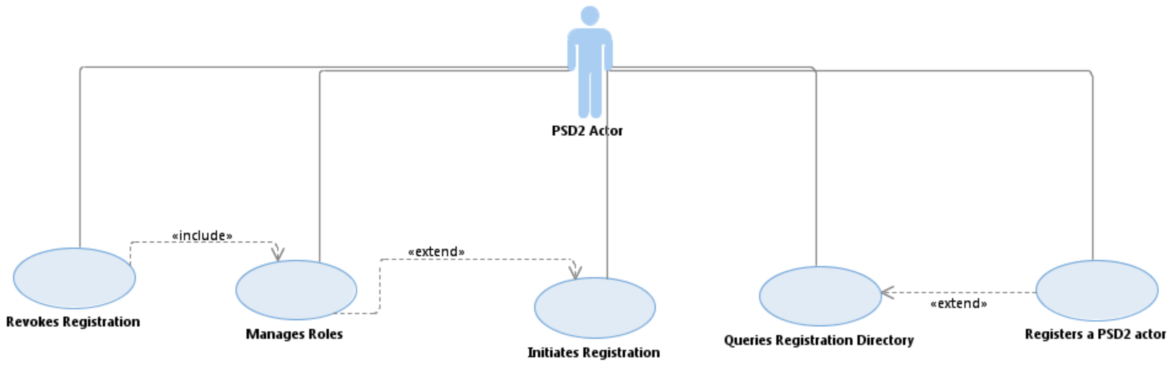
2.2.1. PAO uses cases (NON-API)



USE CASE (PAO)	DESCRIPTION	INTERACTIONS
Initiates ASPSP Contract	The user contracts with an ASPSP in order to use its services. This use case is likely extended by one or more occurrences of the “Requests Account Creation” use case	ASPSP
Requests Account Creation	The user asks the ASPSP to open a new payment account Requires a contract between the PAO and the ASPSP	ASPSP

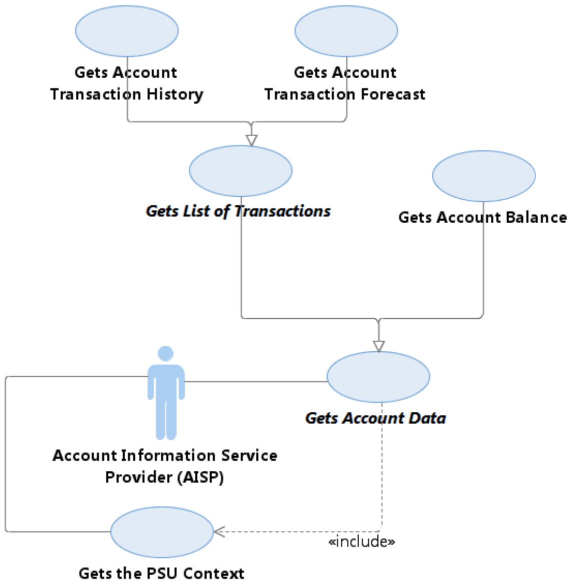
USE CASE (PAO)	DESCRIPTION	INTERACTIONS
Requests Account Closure	The user asks the ASPSP to close an existing payment account This use case includes the “revokes Account/Operation Accreditation” use case for all operations on this account and for all granted TPP.	ASPSP TPP (indirectly)
Revokes ASPSP Contract	The user revokes the contract with the ASPSP This use case includes the “Requests Account Closure” use case for each account that is held by the ASPSP. This use case includes the “Revokes Account/Operation Accreditation” use case for all operations on each of these accounts and for all granted TPP.	ASPSP TPP (indirectly)
Initiates TPP Contract	The user contracts with a TPP having AISP and/or PIISP roles in order to use its service This use case is likely extended by one or more occurrences of the “Grants Account/Operation Accreditation” use case	TPP
Grants Account/Operation accreditation	The user allows the ASPSP to give access to the TPP for a given set of operations on a given PAO account Requires a contract between the PAO and the ASPSP, a contract between the PAO and the TPP and the registration of this PAO-TPP relationship by the ASPSP	ASPSP TPP (indirectly)
Revokes Account/Operation accreditation	The user asks the ASPSP to revoke the TPP access for a given set of operations on a given PAO account	ASPSP TPP (indirectly)
Revokes TPP Contract	The user revokes the contract with the TPP. This use case includes the “Revokes Account/Operation Accreditation” for all grants given to the TPP, whatever the ASPSP. Since this cannot be automated, it is the PAO’s duty to initiate all the relevant revocations with each ASPSP.	TPP ASPSP

2.2.2. Registration use cases (NON-API)



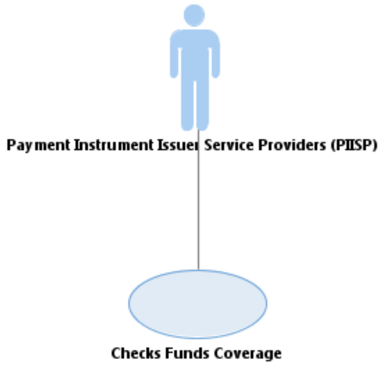
USE CASE (PSD2 ACTOR)	DESCRIPTION	INTERACTIONS
Initiates Registration	The user asks the RA for registration. This use case is likely extended by one or more occurrences of the “Manages Roles” use cases	RA other actors (indirectly)
Manages Roles	The user asks the RA to be referenced for a given set of roles. This use case can be replayed in order to reference or dereference any role.	RA other actors (indirectly)
Revokes registration	The user informs the RA that its registration is to be cancelled	RA other actors (indirectly)
Queries Registration Directory	The user queries the RA directory in order to get data on other PSD2 actors: roles, certificates…	RA other actors (indirectly)
Registers a PSD2 actor	The user registers a given PSD2 actor into its own Directory	None

2.2.3. AISP use cases


Gets Account Transaction History
Gets Account Transaction Forecast
Gets List of Transactions
Gets Account Balance
Gets Account Data
Account Information Service Provider (AISP)
«include»
Gets the PSU Context

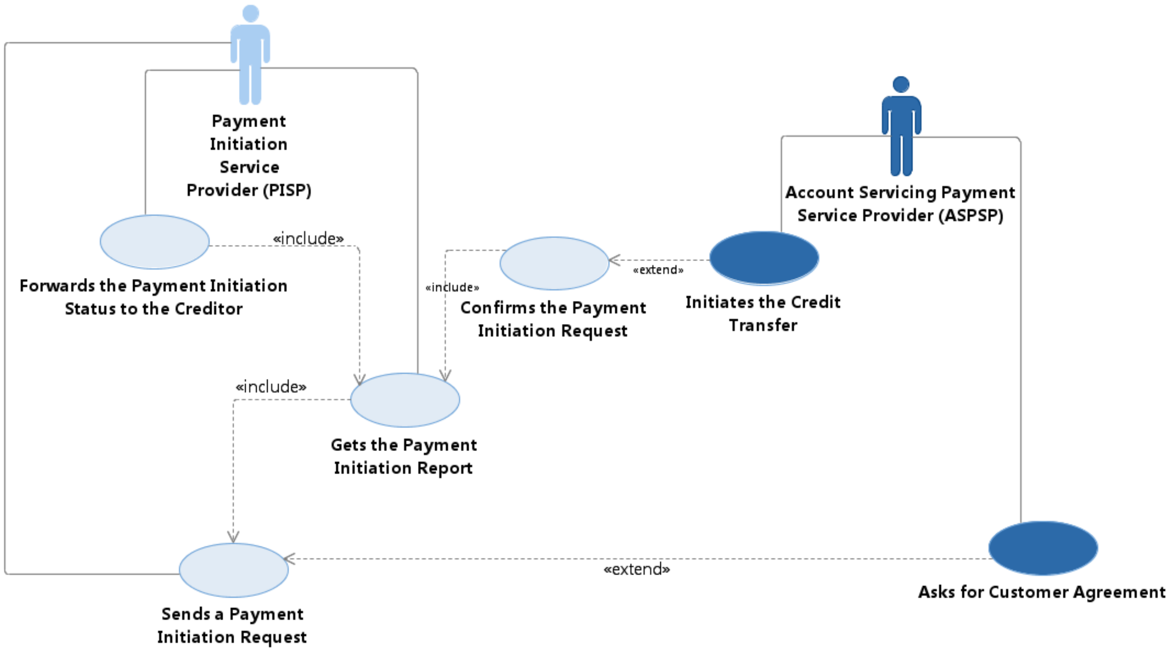
USE CASE (AISP)	DESCRIPTION	INTERACTIONS
Gets the PSU Context	The user queries the ASPSP in order to get - the PSU accounts it is allowed to access - the operations it is allowed to process on each PSU account	ASPSP
Gets Account Data	*This use case is abstract. Its purpose is to stress that the "Gets the PSU Context" is a prerequisite for all other use cases on a given account*	*none*
Gets Account Balance	The user queries the ASPSP in order to get the balance on one given account. The ASPSP can provide several balance computing's (Instant Balance, Accounting Balance…), each balance type being specified with an explicit label.	ASPSP
Gets List of Transactions	This use case is abstract and can be seen as the common interface for the two following uses-cases.	ASPSP
Gets Account Transaction History	The user queries the ASPSP in order to get all the transactions that have been committed to one given PSU account within a given range of value dates.	ASPSP
Gets Account Transaction Forecast	The user queries the ASPSP in order to get all the transactions that are known by the ASPSP to be committed to a given PSU account	ASPSP

2.2.4. PIISP use cases



USE CASE (PIISP)	DESCRIPTION	INTERACTIONS
Checks Funds Coverage	The user queries the ASPSP in order to check if a given transaction amount can be covered by one given PSU account	ASPSP

2.2.5. PISP uses cases


Payment Initiation Service Provider (PISP)
Account Servicing Payment Service Provider (ASPSP)
«include»
Forwards the Payment Initiation Status to the Creditor
«include»
«extend»
Confirms the Payment Initiation Request
Initiates the Credit Transfer
«include»
Gets the Payment Initiation Report
«extend»
Sends a Payment Initiation Request
Asks for Customer Agreement

USE CASE (PISP)	DESCRIPTION	INTERACTIONS
Sends a Payment Initiation Request	The user sends to the ASPSP all the information needed to initiate a Payment from one PAO account (debtor) to one PR account (creditor)	ASPSP
Gets the Payment Initiation Report	The user gets the status of the Payment Initiation Request from the ASPSP.	ASPSP
Confirms the Payment Initiation Request	The user confirms the Payment Initiation Request to the ASPSP so that the latest can initiate the subsequent Credit Transfer	ASPSP
Forwards the Payment Initiation status to the Creditor (Non-API)	The user informs the PR of the status of the Payment Request	PR (Creditor)

USE CASE (ASPSP)	DESCRIPTION	INTERACTIONS
Asks for Customer Agreement (Non-API)	Provided the Payment Request is valid, the user asks the PAO in order to get the consent to the payment request	PSU
Initiates the Credit Transfer (Non-API)	Provided the PAO has given its consent, the ASPSP initiates the relevant Credit Transfer.	PR's ASPSP (Creditor Agent)

3. Prerequisites and technical details

3.1. Actors registration

PSD2 actors must be registered by a registration authority. The information that has been collected must be accessible to other actors in order to provide trust and interoperability.

A non-registered actor cannot interact with another actor.

Each actor must be provided with at least one X.509 certificate, for TLS 1.2 purpose, delivered by a registered Qualified Certification Service Providers (QTSP).

3.2. Cross-Authentication and Data Encryption

The STET PSD2 API relies on TLS 1.2 protocol in order to get cross-authentication between actors. Moreover, this protocol also ensures data confidentiality during their transport on the network.

Whenever a TPP connects as a client to an ASPSP API service, it will check the ASPSP certificate and present its own certificate. In case of authentication failure, on one side or the other, the connection must be closed.

No additional encrypting or authenticating feature is required.

3.3. Authorization

3.3.1. Levels of authorization

The following levels of authorization may be checked and combined in order to compute the effective rights granted to the TPP:

AUTHORIZATION LEVEL	DESCRIPTION
Authorization by TPP role	Once the TPP has been registered for a given role, it can call any of the PSD2 features provided by an ASPSP through the STET PSD2 API.
Authorization by TPP-ASPSP agreement	The TPP can call any of the additional (non PSD2) features provided by an ASPSP through the STET PSD2 API, provided there is a bilateral agreement to use these features.

AUTHORIZATION LEVEL	DESCRIPTION
Authorization by TPP-PSU agreement	If the PSU has contracted with a TPP, it must - Give a list of the ASPSPs that it allows the TPP to access - Inform each of those relevant ASPSPs that will further allow the TPP to access the PSU data. The PSU can either decide or not to include some of its ASPSPs to the list given to the TPP.
Authorization by PSU context	The PSU is able to specify, with each of its ASPSPs, a PSU context detailing, for each of its relevant accounts: - If this account will be visible or not by the TPP - Which features can be used by the TPP The PSU can modify at any time a previous PSU context.

3.3.2. AISP and PIISP authorization levels

Since a TPP is acting on behalf of a PSU being a PAO, the PSD2 use cases that are linked with AISP and PIISP roles require the following authorization levels:

- Authorization by Role
- Authorization by TPP-PSU agreement
- Authorization by PSU context

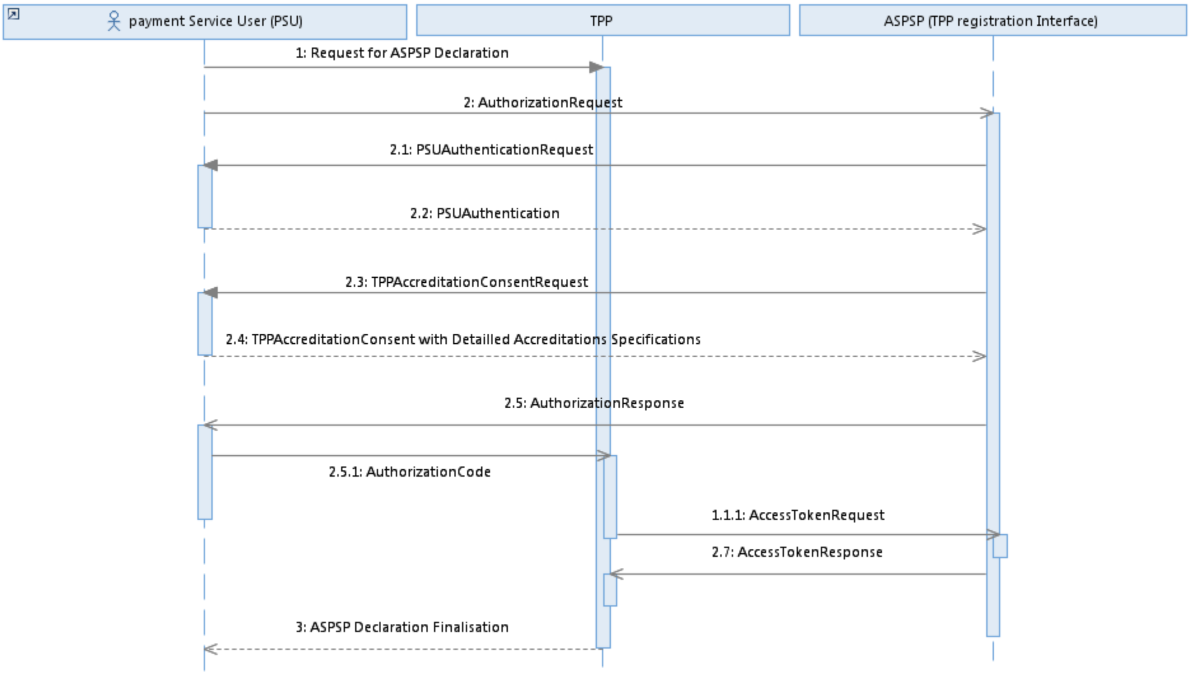
3.3.2.1. List of the relevant ASPSPs

When contracting with a TPP, the PSU will provide a list of the ASPSPs that it allows the TPP to access. This list may not be exhaustive and so may not include some of the PSU's ASPSPs.

3.3.2.2. Registration of the TPP-PSU agreement by each ASPSP

This registration is due to enable the further access of the TPP to the PSU's data that is hosted by a given ASPSP.

The registration process relies on an OAUTH2 sequence for obtaining an Authorization Code Grant (cf. https://tools.ietf.org/html/rfc6749#section-4.1) and can be summarized through the following steps.


payment Service User (PSU)
TPP
ASPSP (TPP registration Interface)
1: Request for ASPSP Declaration
2: AuthorizationRequest
2.1: PSUAuthenticationRequest
2.2: PSUAuthentication
2.3: TPPAccreditationConsentRequest
2.4: TPPAccreditationConsent with Detailled Accreditations Specifications
2.5: AuthorizationResponse
2.5.1: AuthorizationCode
1.1.1: AccessTokenRequest
2.7: AccessTokenResponse
3: ASPSP Declaration Finalisation

- The PSU specifies, to the TPP, the identity of one of its ASPSPs
- The TPP initiates the OAUTH2 sequence by redirecting the PSU to the relevant ASPSP's authorization infrastructure, through the following URL pattern and parameters

```
GET /authorize?response_type=code&client_id={clientId}&redirect_uri={redirectUrl}&scope={scope}[&state={state}]
```

NAME		DATA	TYPE AND CONSTRAINS
response_type	[1..1]	Expected type of token	String[10] Must be valued with "code"
client_id	[0..1]	TPP identification	String[34]
redirect_uri	[0..1]	Call-back URL of the TPP	String[140]
scope	[0..1]	Specifies the generic accreditations that both the PSU and the TPP agreed on: AISP or PIISP.	String[140] Space delimited roles list. Default value is "AISP"
state	[0..1]	Internal state that can be used by the TPP for context management.	String[34]

- The ASPSP
  - Authenticates the PSU
  - Computes the relevant TPP checks (roles, validity, non-revocation…)
  - Asks the PSU for detailed accreditations in order to build the PSU context (see below). The PSU is also able to anticipate this step (i.e. step 2.3 and 2.4) by providing his/her banks with his/her accreditation consent prior to the OAUTH2 process.

- Afterwards, the ASPSP redirects the PSU to the TPP, using the previously given call-back URL (redirect_url) and the following parameters:

NAME		DATA	TYPE AND CONSTRAINS
code	[1..1]	Short-time code to use in order to get the access token	String[34]
state	[0..1]	Internal state if provided by the TPP	String[34]

- In order to get the access token, the TPP is now able to call, through a POST request, the ASPSP's authorization infrastructure with the following parameters.

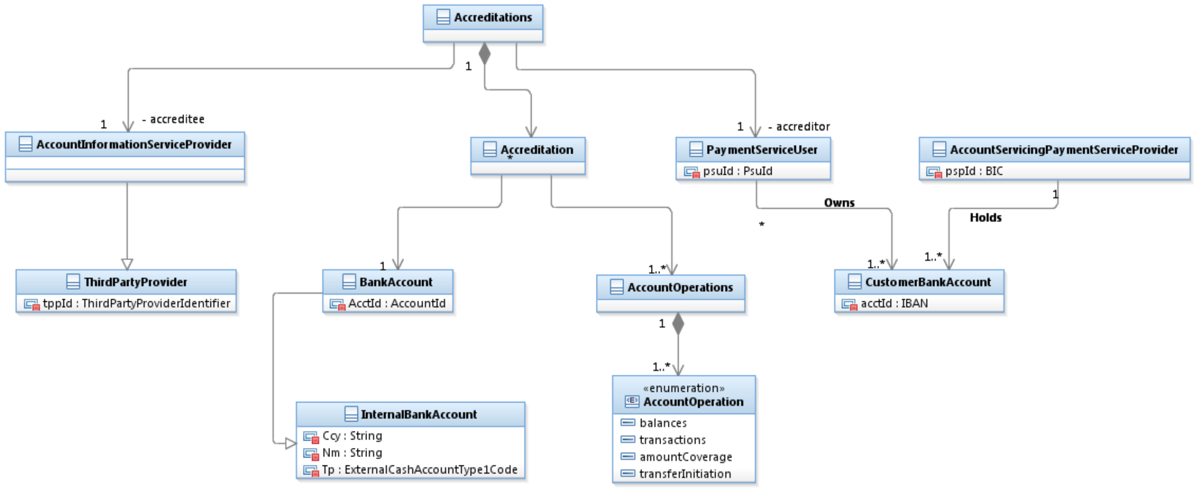
NAME		DATA	TYPE AND CONSTRAINS
grant_type	[1..1]	Requested authorization type	String[34] Must be valued with "authorization_code"
code	[1..1]	Short-time code previously provided by the ASPSP	String[34]
redirect_url	[0..1]	Call-back URL of the TPP	String[140] Must be equal to the one provided during the authorization code request
client_id	[1..1]	TPP identification. In fact, the identification born by the certificate is more relevant at this stage.	String[34]

- The ASPSP
  - Identifies and authenticates the TPP through the presented X.509 certificate
  - Computes the relevant TPP checks (roles, validity, non-revocation…)
- The ASPSP answers through a HTTP200 (OK) response that embeds the following data.

NAME		DATA	TYPE AND CONSTRAINS
access_token	[1..1]	Access token provided by the ASPSP to the TPP.	String[140]
token_type	[1..1]	Type of the provided access token ("bearer" or "MAC")	String[10] Must be values with "bearer"
expires_in	[0..1]	Token lifetime, in seconds. The token can be used several times as far as it is not expired.	Numeric
refresh_token	[0..1]	Refresh token that can be used for a future token renewal request.	String[140]

3.3.2.3. PSU context model

The PSU context can be seen as a collection of individual accreditations.



This collection is specific to a given PSU, a given TPP and a given ASPSP.

Each single accreditation relies on a specific account that is owned by the PSU and is held by the ASPSP. It specifies which operations the TPP is allowed to carry out on this account.

3.3.3. PISP authorization levels

3.3.3.1. General rules

In PISP use cases, the TPP acts on behalf of a Payment Requester (PR) and not on behalf of a PAO.

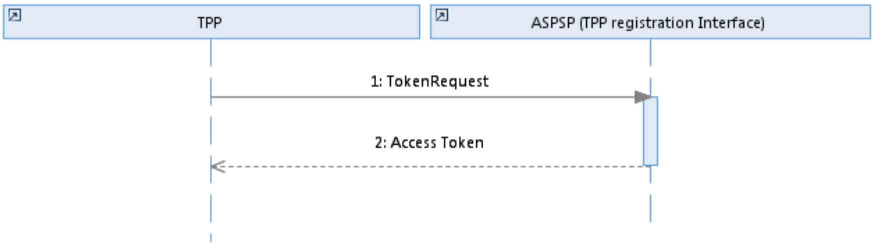
While there is obviously a contract between the PR and the TPP, there is no contract between the PAO and the TPP and so no *a priori* authorization given by the PAO. In fact, the PAO will give its authorization later through the acceptance of a payment request.

That for, the PSD2 use cases that are linked with the PISP role only require an “Authorization by Role” authorization level for accessing the ASPSP API services.

It must be noticed that a PAO may ask to be placed under an OPT-OUT statement by its ASPSPs, avoiding any incoming payment request to be processed on its accounts.

3.3.3.2. Registration of the TPP access

The registration of the TPP by the ASPSP relies on an OAUTH2 sequence for obtaining a Client Credential (cf. https://tools.ietf.org/html/rfc6749#section-4.1) and can be summarized through the following steps.



- The TPP sends directly, through a POST request, its access token request to the ASPSP authorization infrastructure with the following URL pattern and parameters

```
GET /token?grant_type=client_credentials&scope={scope}
```

NAME		DATA	TYPE AND CONSTRAINS
grant_type	[1..1]	Requested authorization type	String[34] Must be valued with "client_credentials"
scope	[0..1]	Specifies the generic accreditations that both the PSU and the TPP agreed on: PISP.	String[140] Space delimited roles list. Default value is "PISP"

- The ASPSP
  - Identifies and authenticates the TPP through the presented X.509 certificate
  - Computes the relevant TPP checks (roles, validity, non-revocation…)
- The ASPSP answers through a HTTP200 (OK) response that embeds the following data.

NAME		DATA	TYPE AND CONSTRAINS
access_token	[1..1]	Access token provided by the ASPSP to the TPP.	String[140]
token_type	[1..1]	Type of the provided access token ("bearer" or "MAC")	String[10] Must be values with "bearer"
expires_in	[0..1]	Token lifetime, in seconds. The token can be used several times as far as it is not expired.	Numeric

3.4. Applicative authentication

Some requests sent by the TPP have to be signed using http-signature mechanism which is specified by the following IETF draft-paper:

- https://datatracker.ietf.org/doc/draft-cavage-http-signatures/

The way it should be implemented is the following

- Computing a SHA256 digest of the HTTP body and adding this digest as an extra HTTP header.
- Using a specific Qualified Certificate in order to apply a RSA-SHA256 signature on
    - all headers that are present in the HTTP request, including the previously computed digest
    - on the specific “(request-target)” field which if specified by the IETF draft-paper
- Adding this signature within an extra HTTP header embedding
    - The key identifier which must specify the way to get the relevant qualified certificate
    - The algorithm that has been used
    - The list of headers that have been signed
    - The signature itself.

3.5. Fraud detection oriented information

Whenever the TPP is able to provide the information relating to its connection with the PSU, the following extra HTTP-headers must be set within the HTTP request in order to allow the ASPSP to integrate this information into its own fraud detection process.

DATA	COMMENT	EXTRA HTTP HEADER
IP Address of the PSU terminal when connecting to the TPP	In regards with GDPR rules, this must be subject to PSU's consent	Psu-Ip-Address
IP Port of the PSU terminal when connecting to the TPP		Psu-Ip-Port
HTTP Method used for the most relevant PSU's terminal request to the TTP		Psu-Http-Method
Timestamp of the most relevant PSU's terminal request to the TTP		Psu-TimeStamp
"User-Agent" header field sent by the PSU terminal when connecting to the TPP		Psu-User-Agent
Referer" header field sent by the PSU terminal when connecting to the TPP		Psu-Referer
Accept" header field sent by the PSU terminal when connecting to the TPP		Psu-Accept
Accept-Charset" header field sent by the PSU terminal when connecting to the TPP		Psu-Accept-Charset
Accept-Encoding" header field sent by the PSU terminal when connecting to the TPP		Psu-Accept-Encoding
Accept-Language" header field sent by the PSU terminal when connecting to the TPP		Psu-Accept-Language

3.6. STET PSD2 API technical summary

TOPIC	CHOICE	COMMENT
Access network	Internet	
Network protocol	HTTP 1.1 (Minimum)	
Data encryption **Cross-authentication**	TLS 1.2	Could be enforced through STS and/or TFS
Authorization protocol	OAUTH2	One of the following token modes - Authorization Code Grant (AISP, PIISP) - Client credential (PISP)
Applicative protocol	REST	In respect of the Richardson Maturity Model, on level three in order to provide HYPERMEDIA links.
Applicative authentication	http-signature	Notice this is actually an IETF draft, waiting for approval and so subject to some midifications. https://datatracker.ietf.org/doc/draft-cavage-http-signatures/
PSU Strong Customer Authentication models	REDIRECT or DECOUPLED	
Data format	JSON/UTF8	With use of ISO20022 based data structures
Technical documentation	SWAGGER 2.0	

4. Functional model

The functional model focuses on the business and functional processes.

Further details are specified within the applicative model which is provided through a SWAGGER 2.0 file and some log examples that illustrate relevant use cases (cf. § 5 and further) on these topics:

- Technical data formats
- Error cases
- HYPERMEDIA links

4.1. Retrieval of the PSU accounts (AISP)

4.1.1. Prerequisites

- The TPP has been registered by the Registration Authority for the AISP role.
- The TPP and the PSU have a contract that has been enrolled by the ASPSP
  - At this step, the ASPSP has delivered an “OAUTH2 Authorization Code” access token to the TPP (cf. § 3.3.2).
- The TPP and the ASPSP have successfully processed a mutual check and authentication
- The TPP has presented its “OAUTH2 Authorization Code” access token which allows the ASPSP to identify the relevant PSU and retrieve the linked PSU context

4.1.2. Business flow

The TPP sends a request to the ASPSP for retrieving the list of the PSU accounts.

The ASPSP retrieves the relevant PSU accounts and builds the answer as an accounts list. The result may be subject to pagination in order to avoid an excessive result set.

Each account will be provided with its characteristics, a balance report and the list of functionalities that have been granted by the PSU to the TPP.

4.1.3. Request content

The API entry point is `GET /accounts`

No applicative authentication is needed for this request.

The only information provided by the TPP through its request is the “OAUTH2 Authorization Code” access token.

4.1.4. Response content (if no error)

For each account, the ASPSP provides the following data:

MULT	NAME		DATA
[1..1]	id		Id of the account as defined by the ASPSP
[1..1]	name		Label of the PSU account. In case of a set of pending card transactions, the name shall specify the holder’s name and the imputation date
[0..1]	details		Specifications that might be provided by the ASPSP - characteristics of the account - characteristics of the relevant card
[0..1]	linkedAccount		Case of a set of pending card transactions, the APSP will provide the id of the relevant cash account the card is set up on.
[0..1]	usage		Specifies the usage of the account. **Value** **Significance** **ORGA** professional account **PRIV** private personal account
[1..1]	type		Specifies the type of the account (based on “External Code Sets” ISO20022). **Value** **Significance** **CACC** Checking Account
[1..1]	ccy		Currency used for the account.
[0..*]	balances		Balance report on the account.
	[1..1]	name	Label of the balance
	[1..1]	Amt	Amount of the balance
	[0..1]	ccy	Currency used for the account.
	[1..1]	Sts	Type of balance **Value** **Significance** **CLBD** Accounting Balance **XPCD** Instant Balance **VALU** Value-date balance **OTHR** Other Balance
	[0..1]	lastCommitedTransaction	Identification of the last commited transaction. This is actually useful for instant balance.
[0..1]	psuStatus		Relationship between the PSU and the account - Account Holder - Co-account Holder - Attorney

For each account, the ASPSP might also provide some hyperlinks in order to specify which further actions can be performed on each account if allowed by the PSU:

- balance (getting a balance-report)

- transactions (getting the transactions).

4.2. Retrieval of an account balance-report (AISP)

4.2.1. Prerequisites

- The TPP has been registered by the Registration Authority for the AISP role
- The TPP and the PSU have a contract that has been enrolled by the ASPSP
    - At this step, the ASPSP has delivered an “OAUTH2 Authorization Code” access token to the TPP (cf. § 3.3.2).
- The TPP and the ASPSP have successfully processed a mutual check and authentication
- The TPP has presented its “OAUTH2 Authorization Code” access token which allows the ASPSP to identify the relevant PSU and retrieve the linked PSU context (cf. § 3.3.2)
- The TPP has previously retrieved the list of available accounts for the PSU

4.2.2. Business flow

The AISP requests the ASPSP on one of the PSU’s accounts.

The ASPSP answers by providing a balance-report on this account. The balance-report is a list of balances that shall at least include the accounting balance.

4.2.3. Request content

The API entry point is GET /accounts/{id}/balances-report

No applicative authentication is needed for this request.

The AISP provides through its request:

- The “OAUTH2 Authorization Code” access token.
- The Id of the relevant account, as retrieved from the list of the PSU’s accounts (cf. § 4.1).

4.2.4. Response content (if no error)

The balance-report embeds for each balance the following data.

<table>
<tr><th>MULT.</th><th colspan="2">NAME</th><th>DATA</th></tr>
<tr><td><b>[1..1]</b></td><td colspan="2">id</td><td>Alias of the account.</td></tr>
<tr><td><b>[0..1]</b></td><td colspan="2">timeStampOfValueRef</td><td>Reference timestamp</td></tr>
<tr><td><b>[1..*]</b></td><td colspan="2">balances</td><td>Balance report on the account.</td></tr>
<tr><td></td><td>[1..1]</td><td>name</td><td>Label of the balance</td></tr>
<tr><td></td><td>[1..1]</td><td>Amt</td><td>Amount of the balance</td></tr>
<tr><td></td><td>[0..1]</td><td>ccy</td><td>Currency used for the account.</td></tr>
<tr><td></td><td>[1..1]</td><td>Sts</td><td>Type of balance
<table>
<tr><th>Value</th><th>Signifiance</th></tr>
<tr><td><b>CLBD</b></td><td>Accounting Balance</td></tr>
<tr><td><b>XPCD</b></td><td>Instant Balance</td></tr>
<tr><td><b>VALU</b></td><td>Value-date balance</td></tr>
<tr><td><b>OTHR</b></td><td>Other Balance</td></tr>
</table></td></tr>
<tr><td></td><td>[0..1]</td><td>lastCommitedTransaction</td><td>Identification of the last commited transaction. This is actually useful for instant balance.</td></tr>
</table>

4.3. Retrieval of an account transaction set (AISP)

4.3.1. Prerequisites

- The TPP has been registered by the Registration Authority for the AISP role
- The TPP and the PSU have a contract that has been enrolled by the ASPSP
  - At this step, the ASPSP has delivered an “OAUTH2 Authorization Code” access token to the TPP (cf. § 3.3.2).
- The TPP and the ASPSP have successfully processed a mutual check and authentication
- The TPP has presented its “OAUTH2 Authorization Code” access token which allows the ASPSP to identify the relevant PSU and retrieve the linked PSU context (cf. § 3.3.2)
- The TPP has previously retrieved the list of available accounts for the PSU

4.3.2. Business flow

The AISP requests the ASPSP on one of the PSU’s accounts. It may specify some selection criteria.

The ASPSP answers by a set of transactions that matches the query. The result may be subject to pagination in order to avoid an excessive result set.

4.3.3. Request content

The API entry point is GET /accounts/{id}/transactions

No applicative authentication is needed for this request.

The AISP provides through its request:

- The “OAUTH2 Authorization Code” access token.
- The Id of the relevant account, as retrieved from the list of the PSU’s accounts (cf. § 4.1)
- the following optional selection criteria:

MULT.	NAME	DATA
[0..1]	fromImputationDate	Minimal imputation date of transaction. This date is inclusive: all transactions imputed to this date will be part of the result.
[0..1]	toImputationDate	Maximal imputation date of transaction. This date is exclusive: all transactions imputed to this date will not be part of the result.
[0..1]	afterTransactionId	Last known transaction. Only transactions having a transaction id greater than this value will be part of the result

4.3.4. Response content (if no error)

The transaction set embeds for each transaction the following data.

<table>
<tr><th>MULT.</th><th colspan="2">NAME</th><th>DATA</th></tr>
<tr><td><b>[0..1]</b></td><td colspan="2">NtryRef</td><td>Technical incremental identification of the transaction.</td></tr>
<tr><td><b>[1..1]</b></td><td colspan="2">Amt</td><td>Amount of the transaction.</td></tr>
<tr><td><b>[0..1]</b></td><td colspan="2">ccy</td><td>Currency used for the transaction.</td></tr>
<tr><td><b>[1..1]</b></td><td colspan="2">CdtDbtInd</td><td>Accounting flow of the transaction:<table><tr><th>Value</th><th>Signifiance</th></tr><tr><td><b>CRDT</b></td><td>Credit</td></tr><tr><td><b>DBIT</b></td><td>Debit</td></tr></table></td></tr>
<tr><td><b>[1..1]</b></td><td colspan="2">Sts</td><td>Type of Transaction<table><tr><th>Value</th><th>Signifiance</th></tr><tr><td><b>BOOK</b></td><td>Booked transaction</td></tr><tr><td><b>PDNG</b></td><td>Pending transaction</td></tr><tr><td><b>OTHR</b></td><td>Other</td></tr></table></td></tr>
<tr><td><b>[1..1]</b></td><td colspan="2">BookgDt</td><td>Booking date of the transaction on the account.</td></tr>
<tr><td><b>[1..1]</b></td><td colspan="2">RmtInf</td><td>Remittance information</td></tr>
<tr><td></td><td>[1..1]</td><td>Ustrd</td><td>Free text up to 140 characters</td></tr>
</table>

4.4. Request for payment coverage check (PIISP)

4.4.1. Prerequisites

- The TPP has been registered by the Registration Authority for the PIISP role
- The TPP and the PSU have a contract that has been registered by the ASPSP
    - At this step, the ASPSP has delivered an “OAUTH2 Authorization Code” access token to the TPP (cf. § 3.3.2).
- The TPP and the ASPSP have successfully processed a mutual check and authentication
- The TPP has presented its “OAUTH2 Authorization Code” access token which allows the ASPSP to identify the relevant PSU and retrieve the linked PSU context (cf. § 3.3.2)

4.4.2. Business flow

The PIISP requests the ASPSP for a payment coverage check against either a bank account or a card primary identifier.

4.4.3. Request content

The API entry point is POST /accounts/coverage-control

No applicative authentication is needed for this request.

The PIISP provides the following data to the ASPSP:

- The “OAUTH2 Authorization Code” access token.
- The following additional parameters.

	NAME		DATA
[1..1]	amount		
	[1..1]	InstdAmt	Amount whose coverage is requested
	[1..1]	Ccy	Used currency
[1..1]	accountId		Account Id to be checked
[1..1]	accountIdType		Type of account Id. This must be set to one of the following values: - “cardPan” - “iban” - “other”

4.4.4. Response content (no error)

The result is given by the ASPSP through a structure embedding:

- The processed request
- The result of the processing, being “true” if the coverage has been successfully checked or false if not.

4.5. Payment initiation on behalf of a merchant (PISP)

4.5.1. Prerequisites

- The TPP has been registered by the Registration Authority for the PISP role
- The TPP was provided with an “OAUTH2 Client Credential” access token by the ASPSP (cf. § 3.3.3).
- The TPP and the ASPSP have successfully processed a mutual check and authentication
- The TPP has presented its “OAUTH2 Client Credential” access token

4.5.2. Business flow

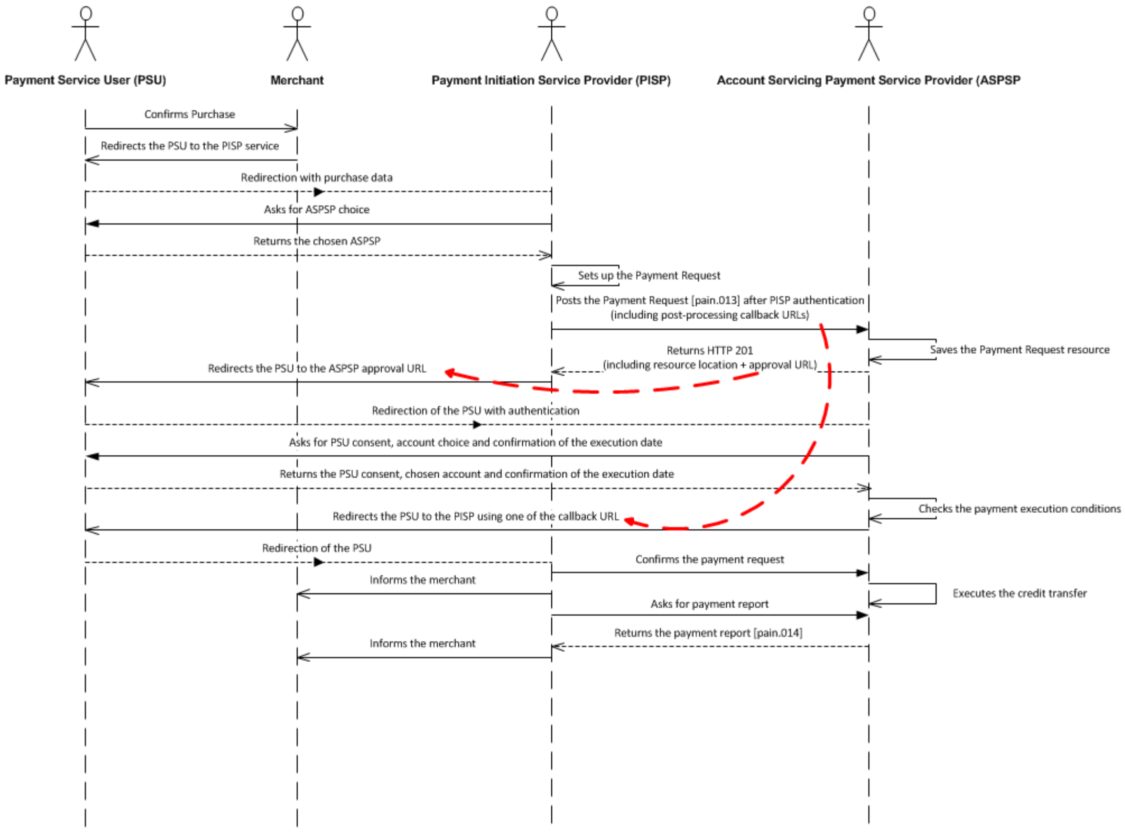
The PSU buys some goods or services on an e-commerce website held by a merchant. Among other payment method, the merchant suggests the use of a PISP service. As there is obviously a contract between the merchant and the PISP, there is no need of such a contract between the PSU and this PISP to initiate the process.

Case of the PSU that chooses to use the PISP service:

- The merchant forwards the requested payment characteristics to the PISP and redirects the PSU to the PISP portal.
- The PISP requests from the PSU which ASPSP will be used.
- The PISP prepares the payment request and sends this request to the ASPSP.
- The ASPSP saves the payment request and answers to the TPP through a location link of the saved payment request.
- The ASPSP also provides the URL to be used for redirecting the PSU in order to perform a REDIRECT or a DECOUPLED strong customer authentication (SCA).

Afterwards

- The TPP redirects the PSU to the ASPSP which authenticates the PSU (REDIRECT or DECOUPLED model)
- The ASPSP asks the PSU to give (or deny) its consent to the payment request
- The PSU chooses which of its accounts shall be used by the ASPSP for the future Credit Transfer.
- The ASPSP redirects the PSU to the PISP


Payment Service User (PSU)
Merchant
Payment Initiation Service Provider (PISP)
Account Servicing Payment Service Provider (ASPSP
Confirms Purchase
Redirects the PSU to the PISP service
Redirection with purchase data
Asks for ASPSP choice
Returns the chosen ASPSP
Sets up the Payment Request
Posts the Payment Request [pain.013] after PISP authentication
(including post-processing callback URLs)
Saves the Payment Request resource
Returns HTTP 201
(including resource location + approval URL)
Redirects the PSU to the ASPSP approval URL
Redirection of the PSU with authentication
Asks for PSU consent, account choice and confirmation of the execution date
Returns the PSU consent, chosen account and confirmation of the execution date
Checks the payment execution conditions
Redirects the PSU to the PISP using one of the callback URL
Redirection of the PSU
Confirms the payment request
Informs the merchant
Executes the credit transfer
Asks for payment report
Returns the payment report [pain.014]
Informs the merchant

4.5.3. Request content

The API entry point is POST /payment-requests

Applicative authentication is needed for this request, using http-signature (cf. §3.4)

The TPP provides through its request:

- The "OAUTH2 Client Credential" token
- The payment request itself through an ISO20022 "pain.013" message-based structure (CreditorPaymentActivationRequest).

This structure embeds only one payment instruction.

For further processing of the payment request, le TPP must also provide some call-back URLs:

- The first URL is mandatory and will be used by the ASPSP in order to inform the TPP of a successful processing of the payment request, meaning that all checks have been passed and the PSU has given his consent to the execution of the subsequent Credit Transfer.
- The second URL is optional and will be used by the ASPSP in order to inform the TPP of an unsuccessful processing of the payment request.

The way those URL's are used by the ASPSP is a redirection of the PSU to the TPP.

	NAME						DATA
[1..1]	GrpHdr						Header of the payment request message
	[1..1]	MsgId					Id of the message. Due to ISO20022 respect, this field is present but has no API usage.
	[1..1]	CredDtTm					Creation timestamp of the message
	[1..1]	NbOfTxs					Count of payment request within the message. Must be valued with “1”.
	[1..1]	InitgPty					TPP identification structure
		[1..1]	Nm				Name of the TPP
		[1..1]	PstlAdr				Postal Address of the TPP
			[1..1]	Ctry			Country code
			[1..2]	AdrLine			Two lines of address
		[1..1]	Id				
			[1..1]	OrgId			
				[1..1]	Othr		
					[1..1]	Id	Id that has been given to the TPP by the registration authority
					[1..1]	Issr	Name of the relevant registration authority
[1..1]	PmtInf						Payment request structure Only one occurrence is allowed
	[1..1]	PmtInfId					Payment request Id
	[1..1]	PmtMtd					Payment method to be used. Must always be valued with “TRF” (Credit Transfer)
	[1..1]	PmtTpInf					Payment characteristics
		[1..1]	InstrPrty				Requested priority: either “HIGH” or “NORM”. High priority will be used to request an instant payment.
		[1..1]	SvcLvl				Requested service level
			[1..1]	Cd			Code value must be one of the following - “SEPA” for SEPA Credit Transfer - “NURG” otherwise
		[0..1]	CtgyPurp				Payment category purpose
			[1..1]	Cd			Must be valued with “DVPM” (delivery against payment)
	[1..1]	ReqdExctnDt					Requested execution date. Due to ISO20022 respect, this field is present but has no API usage.
	[0..1]	Dbtr					PSU identification
		[1..1]	Nm				Name of the PSU
		[1..1]	PstlAdr				Postal Address of the PSU
			[1..1]	Ctry			Country code
			[1..2]	AdrLine			Two lines of address
	[0..1]	DbtrAcct					PSU’s bank account that has to be debited. This structure can be used only if the PSU has accepted to disclose his/her account id to the merchant or the PISP.
		[1..1]	Id				Either Iban or Other must be set
			[0..1]	Iban			IBAN of the PSU’s bank account
			[0..1]	Othr			Other Id for the PSU’s bank account
				[1..1]	Id		Id
				[1..1]	Issr		Issuer of the Id

	NAME							DATA
	[0..1]	DbtrAgt						Identification of the merchant's ASPSP. Mandatory for non SEPA payments requests.
		[1..1]	FinInstnId					
			[1..1]	Bicfi				
	[0..1]	ChrgBr						Specifies which party/parties will bear the charges associated with the processing of the payment transaction. Set to "SLEV" (Charges are to be applied following the rules agreed in the service level and/or scheme) or omitted.
	[1..1]	CdtTrfTx						Payment instruction. Only one occurrence is allowed
		[1..1]	PmtId					Payment instruction Id
			[1..1]	InstrId				Instruction Id that has been set by the TPP
			[1..1]	EndToEndId				Id that has been set by the merchant for giving a reference to the PSU
		[1..1]	Amt					
			[1..1]	InstdAmt				Payment instruction amount
			[1..1]	Ccy				Used currency
		[0..1]	CdtrAgt					Identification of the merchant's ASPSP. Mandatory for non SEPA payments requests.
			[1..1]	FinInstnId				
				[1..1]	Bicfi			
		[1..1]	Cdtr					Merchant identification
			[1..1]	Nm				Name of the merchant
			[1..1]	PstlAdr				Postal address of the merchant
				[1..1]	Ctry			Country code
				[1..2]	AdrLine			Two lines of address
			[0..1]	Id				Optional organizational Id of the Merchant
				[1..1]	OrgId			
					[1..1]	Othr		
						[1..1]	Id	Id
						[1..1]	Issr	Issuer of the Id
		[1..1]	CdtrAcct					Merchant's bank account that has to be credited
			[1..1]	Id				Either Iban or Other must be set
				[0..1]	Iban			IBAN of the merchant's bank account
				[0..1]	Othr			Other Id for the merchant's bank account
					[1..1]	Id		Id
					[1..1]	Issr		Issuer of the Id
		[0..1]	UltmtCdtr					
			[1..1]	Nm				Name of the real merchant in case of a hosted market place
		[0..1]	Purp					Purpose of the payment instruction
			[1..1]	Cd				Must be valued with "COMC" (Commerce) , "CPKC" (Carpark) or "TRPT" (Transport)
		[1..1]	RmtInf					Remittance information
			[1..1]	Ustrd				Free text up to 140 characters
[1..1]	SplmtryData							Technical supplementary data given by the TPP
	[1..1]	Envlp						

	NAME			DATA
		[1..1]	successfulReportUrl	TPP call-back URL to be used in case of a successful processing of the payment request
		[0..1]	unsuccessfulReportUrl	TPP call-back URL to be used in case of a failure during processing of the payment request

4.5.4. Response content (if no error)

The ASPSP answers with two headers

- A “location” link of the saved payment request. This link refers to the REST Id of the saved payment to be used afterwards.
- A “consent_approval_url” link that is to be used by the PISP in order to start the ASPSP’s authentication and consent management process. The ASPSP has to build this URL in a way it will allow the identification of the saved payment request, for instance by referring its REST id.

These two links are not subject to the standardisation, so each ASPSP can build them regarding its own IT constraints and rules.

4.6. Retrieval of a previously posted payment request (PISP)

4.6.1. Prerequisites

- The TPP has been registered by the Registration Authority for the PISP role
- The TPP was provided with an “OAUTH2 Client Credential” access token by the ASPSP (cf. § 3.3.3).
- The TPP has previously posted a Payment Request which has been saved by the ASPSP (cf. § 4.5.3)
  - The ASPSP has answered with a location link to the saved Payment Request (cf. § 4.5.4)
- The TPP and the ASPSP have successfully processed a mutual check and authentication
- The TPP has presented its “OAUTH2 Client Credential” access token

4.6.2. Business flow

The PISP asks to retrieve the payment request that has been saved by the ASPSP. The PISP uses the location link provided by the ASPSP in response of the posting of this request.

The ASPSP returns the previously posted payment request.

4.6.3. Request content

The API entry point is GET /payment-requests/{paymentInformationId}

No applicative authentication is needed for this request.

The PISP provides through its request:

- The “OAUTH2 Client Credential” token
- The REST Id of the saved payment request

4.6.4. Response content (if no error)

The response given by the ASPSP includes the previously posted Payment Request which has been enriched with the REST Id of the resources that has been created by the ASPSP

- Payment request
- Payment instructions

These REST Ids are the ones to be used when asking for a given resource through the API.

	NAME						DATA
[1..1]	GrpHdr						Header of the payment request message
	[1..1]	MsgId					Id of the message. Due to ISO20022 respect, this field is present but has no API usage.
	[1..1]	CredDtTm					Creation timestamp of the message
	[1..1]	NbOfTxs					Count of payment instructions within the payment request.
	[1..1]	InitgPty					TPP identification structure
		[1..1]	Nm				Name of the TPP
		[1..1]	PstlAdr				Postal Address of the TPP
			[1..1]	Ctry			Country code
			[1..2]	AdrLine			Two lines of address
		[1..1]	Id				
			[1..1]	OrgId			
				[1..1]	Othr		
					[1..1]	Id	Id that has been given to the TPP by the registration authority
					[1..1]	Issr	Name of the relevant registration authority
[1..1]	PmtInf						Payment request structure Only one occurrence is allowed
	[1..1]	PmtInfId					Payment request Id
	[1..1]	PmtInfRscId					REST Resource Id of the payment request
	[1..1]	PmtMtd					Payment method to be used. Must always be valued with “TRF” (Credit Transfer)
	[1..1]	PmtTpInf					Payment characteristics
		[1..1]	InstrPrty				Requested priority: either “HIGH” or “NORM”. High priority will be used to request an instant payment.
		[1..1]	SvcLvl				Requested service level
			[1..1]	Cd			Code value must be one of the following - “SEPA” for SEPA Credit Transfer - “NURG” otherwise
		[1..1]	CtgyPurp				Payment category purpose
			[1..1]	Cd			Must be valued with “DVPM” (delivery against payment)
	[1..1]	ReqdExctnDt					Requested execution date. Due to ISO20022 respect, this field is present but has no API usage.
	[0..1]	Dbtr					PSU identification
		[1..1]	Nm				Name of the PSU as provided by the PISP
		[1..1]	PstlAdr				Postal Address of the PSU as provided by the PISP
			[1..1]	Ctry			Country code
			[1..2]	AdrLine			Two lines of address
	[0..1]	DbtrAcct					PSU’s bank account that has to be debited. This structure can be used only if the PSU has accepted to disclose his/her account id to the merchant or the PISP.
		[1..1]	Id				Either Iban or Other must be set
			[0..1]	Iban			IBAN of the PSU’s bank account
			[0..1]	Othr			Other Id for the PSU’s bank account
				[1..1]	Id		Id

	NAME							DATA
				[1..1]	Issr			Issuer of the Id
	[0..1]	DbtrAgt						Identification of the merchant's ASPSP. Mandatory for non SEPA payments requests.
		[1..1]	FinInstnId					
			[1..1]	Bicfi				
	[0..1]	ChrgBr						Specifies which party/parties will bear the charges associated with the processing of the payment transaction. Set to "SLEV" (Charges are to be applied following the rules agreed in the service level and/or scheme) or omitted.
	[1..1]	CdtTrfTx						Payment instruction. Only one occurrence is allowed
		[1..1]	PmtId					Payment instruction Id
			[1..1]	InstrId				Instruction Id that has been set by the TPP
			[1..1]	InstrResId				REST Resource Id of the payment instruction
			[1..1]	EndToEndId				Id that has been set by the merchant for giving a reference to the PSU
		[1..1]	Amt					
			[1..1]	InstdAmt				Payment instruction amount
			[1..1]	Ccy				Used currency
		[0..1]	CdtrAgt					Identification of the merchant's ASPSP. Mandatory for non SEPA payments requests.
			[1..1]	FinInstnId				
				[1..1]	Bicfi			
		[1..1]	Cdtr					Merchant identification
			[1..1]	Nm				Name of the merchant
			[1..1]	PstlAdr				Postal address of the merchant
				[1..1]	Ctry			Country code
				[1..2]	AdrLine			Two lines of address
			[0..1]	Id				Optional organizational Id of the Merchant
				[1..1]	OrgId			
					[1..1]	Othr		
						[1..1]	Id	Id
						[1..1]	Issr	Issuer of the Id
		[1..1]	CdtrAcct					Merchant's bank account that has to be credited
			[1..1]	Id				As provided by the PISP
				[0..1]	Iban			IBAN of the merchant's bank account
				[0..1]	Othr			Other Id for the merchant's bank account
						[1..1]	Id	Id
						[1..1]	Issr	Issuer of the Id
		[0..1]	UltmtCdtr					
			[1..1]	Nm				Name of the real merchant in case of a hosted market place
		[1..1]	Purp					Purpose of the payment instruction
			[1..1]	Cd				Must be valued with "COMC" (Commerce) , "CPKC" (Carpark) or "TRPT" (Transport)
		[1..1]	RmtInf					Remittance information
			[1..1]	Ustrd				Free text up to 140 characters

	NAME			DATA
[1..1]	SplmtryData			Technical supplementary data given by the TPP
	[1..1]	Envlp		
		[1..1]	successfulReportUrl	TPP callback URL to be used in case of a successful processing of the payment request
		[0..1]	unsuccessfulReportUrl	TPP callback URL to be used in case of a failure during processing of the payment request

4.7. Retrieval of a payment request status (PISP)

4.7.1. Prerequisites

- The TPP has been registered by the Registration Authority for the PISP role
- The TPP was provided with an “OAUTH2 Client Credential” access token by the ASPSP (cf. § 3.3.3).
- The TPP has previously posted a Payment Request which has been saved by the ASPSP (cf. § 4.5.3)
    - The ASPSP has answered with a location link to the saved Payment Request (cf. § 4.5.4)
- The TPP and the ASPSP have successfully processed a mutual check and authentication
- The TPP has presented its “OAUTH2 Client Credential” access token

4.7.2. Business flow

Once the payment request has been validated, the PSU is requested by the ASPSP to give its consent.

After the PSU’s decision, the PISP asks the ASPSP to get a global status of a payment request, in order to get information about the PSU’s decision and the further processing of this request.

4.7.3. Request content

The API entry point is GET /payment-requests/{paymentInformationId}/report

No applicative authentication is needed for this request.

The PISP provides through its request:

- The “OAUTH2 Client Credential” token
- The REST Id of the payment request

4.7.4. Response content (if no error)

The ASPSP answers with an ISO20022 “pain.014” message based structure.

	NAME					DATA
[1..1]	OrgnlPmtInfId					Original payment request Id
[1..1]	OrgnlNbOfTxs					Count of payment instructions within the original payment request
[1..1]	PmtInfSts					Payment request status. Must be set with on the following values: - RJCT (Rejected): either the payment request was invalid or it has been refused by the PSU. - ACSP (AcceptedSettlementInProcess): The payment request has been validated and accepted by the PSU.
[0..1]	StsRsnInf					In case of reject, this must be set with one of the following : - FF01: The payment request is not valid and cannot be processed. - CUST: The PSU has refused the payment request
[1..1]	TxInfAndSts					Payment instruction of the original payment request
	[1..1]	OrgnlInstrId				Original instruction Id that has been set by the TPP
	[1..1]	OrgnlEndToEndId				Id that has been set by the merchant in the original payment instruction for giving a reference to the PSU
	[1..1]	TxSts				Payment instruction status. Must be set with one of the following values : - RJCT (final status): instruction could not be executed, either for the payment request has been globally rejected by the PSU, or for execution context issue (e.g. lack of funds on the debtor account) - PDNG: instruction is pending - ACSP: instruction is currently being processed - ACSC (final status): instruction has been successfully cleared and settled
	[1..1]	StsRsnInf				In case of instruction reject, this must be set with one of the following values: - FF01: reject occurred on the global payment request level. - CUST: reject occurred due to execution context issue.
	[1..1]	OrgnlTxRef				Original payment instruction reference. The ASPSP may add some relevant data.
		[1..1]	Amt			
			[1..1]	InstdAmt		Original instruction amount.
			[1..1]	Ccy		Original instruction currency.
		[1..1]	ReqdExctnDt			Requested execution date. Due to ISO20022 respect, this field is present but has no API usage.
		[1..1]	PmtTpInf			Original payment instruction characteristics.
			[1..1]	InstrPrty		Original payment instruction priority
			[1..1]	SvcLvl		Original payment instruction service level
				[1..1]	Cd	
			[1..1]	CtgyPurp		Original payment instruction category purpose
				[1..1]	Cd	
		[1..1]	PmtMtd			Original payment instruction payment method

	NAME							DATA
		[1..1]	RmtInf					Original payment instruction remittance information
			[1..1]	Ustrd				
		[0..1]	Dbtr					PSU identification
			[1..1]	Nm				Name of the PSU as provided by the PISP
			[1..1]	PstlAdr				Postal address of the PSU as provided by the PISP
				[1..1]	Ctry			Country code
				[1..2]	AdrLine			Two lines of address
		[1..1]	DbtrAgt					Identification of the PSU's ASPSP
			[1..1]	FinInstnId				
				[1..1]	Bicfi			The ASPSP set up this value with its BIC
		[1..1]	CdtrAgt					Identification of the merchant's ASPSP
			[1..1]	FinInstnId				
				[1..1]	Bicfi			The ASPSP set up this value with the BIC computed from the merchant's IBAN
		[1..1]	Cdtr					Identification of the merchant
			[1..1]	Nm				Name of the merchant
			[1..1]	PstlAdr				Postal address of the merchant
				[1..1]	Ctry			Country code
				[1..2]	AdrLine			Two lines of address
			[0..1]	Id				Optional organizational Id of the Merchant
				[1..1]	OrgId			
					[1..1]	Othr		
						[1..1]	Id	Id
						[1..1]	Issr	Issuer of the Id
		[1..1]	CdtrAcct					Merchant's bank account
			[1..1]	Id				As provided by the PISP
				[0..1]	Iban			IBAN of the merchant's bank account
				[0..1]	Othr			Other Id for the merchant's bank account
					[1..1]	Id		Id
					[1..1]	Issr		Issuer of the Id
		[0..1]	UltmtCdtr					Real merchant in case of a hosted market place
			[1..1]	Nm				Name of the real merchant

4.8. Confirmation of a payment initiation request (PISP)

4.8.1. Prerequisites

- The TPP has been registered by the Registration Authority for the PISP role
- The TPP was provided with an “OAUTH2 Client Credential” access token by the ASPSP (cf. § 3.3.3).
- The TPP has previously posted a Payment Request which has been saved by the ASPSP (cf. § 4.5.3)
    - The ASPSP has answered with a location link to the saved Payment Request (cf. § 4.5.4)
    - The TPP has retrieved the saved Payment request in order to get the relevant resource Ids (cf. § 4.6).
- The TPP and the ASPSP have successfully processed a mutual check and authentication
- The TPP has presented its “OAUTH2 Client Credential” access token

4.8.2. Business flow

Once the payment request has been validated and accepted by the PSU, it is the due to the PISP to confirm this payment request to the ASPSP in order for the latest to execute the individual payment instruction that is embedded within the payment request.

4.8.3. Request content

The API entry point is POST /payment-requests/{paymentInformationId}/confirmation

The PISP provides through its request:

- The “OAUTH2 Client Credential” token
- The REST Id of the saved payment request

4.8.4. Response content (if no error)

The ASPSP answers with an ISO20022 “pain.014” message-based structure in order to give an update of the payment request to the PISP in a same way as § 4.7.4.

5. AISP Use cases

5.1. PSU Context Retrieval

5.1.1. Request

```
GET http://localhost:8080/v1/accounts
```

5.1.1.1. Headers

```
Date: 2017-07-12T15:43:09.573+02:00
Psu-User-Agent: Mozilla
Psu-Referer: http://en.wikipedia.org/wiki/Main_Page
Accept: application/hal+json
Psu-Accept-Charset: en-US
Authorization: Bearer 1234567890AZERTYUIOP
Psu-Accept-Language: gzip, deflate
Psu-Http-Method: POST
Psu-Ip-Port: 12345
Psu-Ip-Address: 10.10.10.10
Psu-Accept: text/plain
Psu-TimeStamp: 2017-06-08T09:33:55.954+02:00
Psu-Accept-Encoding: utf-8
Content-Type: application/json
User-Agent: Swagger-Codegen/1.0.0/java
```

5.1.1.2. Body

No body data

5.1.2. Response

```
Status code: 200
```

5.1.2.1. Headers

```
Server: Apache-Coyote/1.1
Content-Type: application/hal+json;charset=UTF-8
```

```
Transfer-Encoding: chunked
Date: Wed, 12 Jul 2017 13:43:10 GMT
```

5.1.2.2. Body

```
{
   "_embedded" : {
      "accounts" : [ {
         "id" : "Alias1",
         "name" : "Compte de Mr et Mme Dupont",
         "usage" : "PRIV",
         "type" : "CACC",
         "Ccy" : "EUR",
         "psuStatus" : "Co-account Holder",
         "_links" : {
            "balances" : {
               "href" : "v1/accounts/Alias1/balances"
            },
            "transactions" : {
               "href" : "v1/accounts/Alias1/transactions"
            }
         }
      }, {
         "id" : "Alias2",
         "name" : "Compte de Mme Dupont",
         "usage" : "PRIV",
         "type" : "CACC",
         "Ccy" : "EUR",
         "psuStatus" : "Account Holder",
         "_links" : {
            "balances" : {
               "href" : "v1/accounts/Alias2/balances"
            },
            "transactions" : {
               "href" : "v1/accounts/Alias2/transactions"
            }
      }
    } ]
  },
  "_links" : {
    "self" : {
      "href" : "v1/accounts"
    }
  }
}
```

5.2. Account Balances Retrieval

5.2.1. Request

```
GET http://localhost:8080/v1/accounts/Alias1/balances-report
```

5.2.1.1. Headers

```
Date: 2017-07-12T15:43:10.504+02:00
Psu-User-Agent: Mozilla
Psu-Referer: http://en.wikipedia.org/wiki/Main_Page
Accept: application/hal+json
Psu-Accept-Charset: en-US
Authorization: Bearer 1234567890AZERTYUIOP
Psu-Accept-Language: gzip, deflate
Psu-Http-Method: POST
Psu-Ip-Port: 12345
Psu-Ip-Address: 10.10.10.10
Psu-Accept: text/plain
Psu-TimeStamp: 2017-06-08T09:33:55.954+02:00
Psu-Accept-Encoding: utf-8
Content-Type: application/json
User-Agent: Swagger-Codegen/1.0.0/java
```

5.2.1.2. Body

No body data

5.2.2. Response

```
Status code: 200
```

5.2.2.1. Headers

```
Server: Apache-Coyote/1.1
Content-Type: application/hal+json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 12 Jul 2017 13:43:10 GMT
```

5.2.2.2. Body

```
{
  "id" : "Alias1",
  "timeStampOfValueRef" : "2017-01-12T20:13:00.000Z",
  "balances" : [ {
    "name" : "Solde comptable au 12/01/2017",
    "Amt" : "123.45",
    "Ccy" : "EUR",
    "Sts" : "CLBD",
    "lastCommitedTransaction" : "A452CH"
  }, {
    "name" : "Solde instantané au 12/01/2017 20:13",
    "Amt" : "105.65",
    "Ccy" : "EUR",
    "Sts" : "XPCD",
    "lastCommitedTransaction" : "A452D0"
  } ],
  "_links" : {
    "self" : {
      "href" : "v1/accounts/Alias1/balances"
    },
    "transactions" : {
      "href" : "v1/accounts/Alias1/transactions"
    }
  }
}
```

5.3. Account Transactions Retrieval

5.3.1. Request

```
GET http://localhost:8080/v1/accounts/Alias1/transactions
```

5.3.1.1. Headers

```
Date: 2017-07-12T15:43:10.657+02:00
Psu-User-Agent: Mozilla
Psu-Referer: http://en.wikipedia.org/wiki/Main_Page
Accept: application/hal+json
Psu-Accept-Charset: en-US
Authorization: Bearer 1234567890AZERTYUIOP
Psu-Accept-Language: gzip, deflate
Psu-Http-Method: POST
Psu-Ip-Port: 12345
Psu-Ip-Address: 10.10.10.10
Psu-Accept: text/plain
Psu-TimeStamp: 2017-06-08T09:33:55.954+02:00
Psu-Accept-Encoding: utf-8
Content-Type: application/json
User-Agent: Swagger-Codegen/1.0.0/java
```

5.3.1.2. Body

No body data

5.3.2. Response

```
Status code: 200
```

5.3.2.1. Headers

```
Server: Apache-Coyote/1.1
```

```
Content-Type: application/hal+json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 12 Jul 2017 13:43:10 GMT
```

5.3.2.2. Body

```
{
  "_embedded" : {
    "transactions" : [ {
      "NtryRef" : "AF5T2",
      "Amt" : "12,25",
      "Ccy" : "EUR",
      "CdtDbtInd" : "DBIT",
      "Sts" : "BOOK",
      "BookgDt" : "2017-01-12",
      "RmtInf" : {
        "Ustrd" : [ "Chèque n°XXXXXXX" ]
      }
    }, {
      "NtryRef" : "AF5T3",
      "Amt" : "66,38",
      "Ccy" : "EUR",
      "CdtDbtInd" : "DBIT",
      "Sts" : "BOOK",
      "BookgDt" : "2017-01-12",
      "RmtInf" : {
        "Ustrd" : [ "Prélèvement ICS XXXXXXX" ]
      }
    }, {
      "NtryRef" : "AF5T4",
      "Amt" : "60,00",
      "Ccy" : "EUR",
      "CdtDbtInd" : "DBIT",
      "Sts" : "BOOK",
      "BookgDt" : "2017-01-12",
      "RmtInf" : {
```

```
                "Ustrd" : [ "Retrait Carte" ]
            }
        } ]
    },
    "_links" : {
        "self" : {
            "href" : "v1/accounts//Alias1/transactions/transactions"
        },
        "balances" : {
            "href" : "v1/accounts/Alias1/balances"
        },
        "last" : {
            "href" : "v1/accounts//Alias1/transactions/transactions"
        },
        "next" : {
            "href" : "v1/accounts/Alias1/transactions"
        }
    }
}
```

6. PIISP Use cases

6.1. Account Amount Coverage Check

6.1.1. Request

```
POST http://localhost:8080/v1/accounts/coverage-control
```

6.1.1.1. Headers

```
Date: 2017-07-12T15:43:10.751+02:00
Psu-User-Agent: Mozilla
Psu-Referer: http://en.wikipedia.org/wiki/Main_Page
Accept: application/hal+json
Psu-Accept-Charset: en-US
```

```
Authorization: Bearer 1234567890AZERTYUIOP
Psu-Accept-Language: gzip, deflate
Psu-Http-Method: POST
Psu-Ip-Port: 12345
Psu-Ip-Address: 10.10.10.10
Psu-Accept: text/plain
Psu-TimeStamp: 2017-06-08T09:33:55.954+02:00
Psu-Accept-Encoding: utf-8
Content-Type: application/json
User-Agent: Swagger-Codegen/1.0.0/java
Content-Length: 115
```

6.1.1.2. Body

```
{
  "amount" : {
    "InstdAmt" : "12345",
    "Ccy" : "EUR"
  },
  "accountId" : "YY13RDHN98392489481620896668799742",
  "accountIdType" : "iban"
}
```

6.1.2. Response

```
Status code: 200
```

6.1.2.1. Headers

```
Server: Apache-Coyote/1.1
Content-Type: application/hal+json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 12 Jul 2017 13:43:10 GMT
```

6.1.2.2. Body

```
{
  "request" : {
        "amount" : {
            "InstdAmt" : "12345",
            "Ccy" : "EUR"
        },
        "accountId" : "YY13RDHN98392489481620896668799742",
        "accountIdType" : "iban"
    },
    "result" : true,
    "_links" : {
        "self" : {
            "href" : "v1/accounts/coverage-control"
        }
    }
}
```

7. PISP Use cases

7.1. Payment Initiation Request

7.1.1. Request

```
POST http://localhost:8080/v1/payment-requests
```

7.1.1.1. Headers

```
Date: 2017-07-12T15:43:11.006+02:00
Psu-User-Agent: Mozilla
Psu-Referer: http://en.wikipedia.org/wiki/Main_Page
Accept: application/hal+json
Psu-Accept-Charset: en-US
Authorization: authorization_example
Psu-Accept-Language: gzip, deflate
Psu-Http-Method: POST
Psu-Ip-Port: 12345
Psu-Ip-Address: 10.10.10.10
Psu-Accept: text/plain
```

```
Psu-TimeStamp: 2017-06-08T09:33:55.954+02:00
Psu-Accept-Encoding: utf-8
Content-Type: application/json
User-Agent: Swagger-Codegen/1.0.0/java
Digest: SHA-256=ImMgVN/f7naN5wX+fhH8suw6oIsKz9u+u4oU/KftJf8=
Content-Length: 1124
Signature: keyId="Test",algorithm="rsa-sha256",headers="date psu-user-agent psu-referer
accept psu-accept-charset authorization psu-accept-language psu-http-method psu-ip-port psu-
ip-address psu-accept psu-timestamp psu-accept-encoding content-type user-agent digest
content-length (request-
target)",signature="v7isLVvx8/LqqS2ATg63IRxDdFzMeqxgIwaZfwYNqBveQyP1LOXBcjYMfHK
cnJPiPydRrUYdQqSjl6HrcxXPzCBpFqOFt7H1/+UGrJAvHEaEGK1UnPB4k1WeT2DVdK/Zruu4
yFXGMhem6He+IR9FcPeovgF3SfTclwEPT9skleA="
```

7.1.1.2. Body

```
{
    "GrpHdr" : {
        "MsgId" : "MSG092GH",
        "CredDtTm" : "2017-07-12T15:43:10.970+02:00",
        "NbOfTxs" : 1,
        "InitgPty" : {
            "Nm" : "MyPisp",
            "PstlAdr" : {
                "Ctry" : "FR",
                "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
            },
            "Id" : {
                "OrgId" : {
                    "Othr" : {
                        "Id" : "JDJZH453",
                        "Issr" : "ACPR"
                    }
                }
            }
        }
```

```
},
"PmtInf" : {
   "PmtInfId" : "MyPmtInfId",
   "PmtMtd" : "TRF",
   "PmtTpInf" : {
      "InstrPrty" : "NORM",
      "SvcLvl" : {
         "Cd" : "SEPA"
      },
      "CtgyPurp" : {
         "Cd" : "DVPM"
      }
   },
   "ReqdExctnDt" : "2017-07-14T15:43:10.981+02:00",
   "Dbtr" : {
      "Nm" : "MyCustomer",
      "PstlAdr" : {
         "Ctry" : "FR",
         "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
      }
   },
   "ChrgBr" : "SLEV",
   "CdtTrfTx" : [ {
      "PmtId" : {
         "InstrId" : "MyInstrId",
         "EndToEndId" : "MyEndToEndId"
      },
      "Amt" : {
         "InstdAmt" : "123,45",
         "Ccy" : "EUR"
      },
      "Cdtr" : {
         "Nm" : "myMerchant",
         "PstlAdr" : {
            "Ctry" : "FR",
               "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
            }
         },
         "CdtrAcct" : {
            "Id" : {
               "Iban" : "YY64COJH41059545330222956960771321"
            }
         },
         "UltmtCdtr" : {
            "Nm" : "myUltimateMerchant",
            "PstlAdr" : {
               "Ctry" : "FR",
               "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
            }
         },
         "Purp" : {
            "Cd" : "COMC"
         },
         "RmtInf" : {
            "Ustrd" : [ "MyRemittanceInformation" ]
         }
      } ]
   },
   "SplmtryData" : {
      "Envlp" : {
         "successfulReportUrl" : "http://myPisp/PaymentSuccess",
         "unsuccessfulReportUrl" : "http://myPisp/PaymentFailure"
      }
   }
}
```

7.1.2. Response

```
Status code: 201
```

7.1.2.1. Headers

```
Server: Apache-Coyote/1.1
location: v1/payments/paymentRequest/MyPmtInfRscId
consent_approval_url: v1/payments/authenticate?PmtInfRscId=MyPmtInfRscId
Content-Length: 0
Date: Wed, 12 Jul 2017 13:43:10 GMT
```

7.1.2.2. Body

No body data

7.2. Payment Initiation Request Retrieval

7.2.1. Request

```
GET http://localhost:8080/v1/payment-requests/MyPmtInfRscId
```

7.2.1.1. Headers

```
Date: 2017-07-12T15:43:11.144+02:00
Psu-User-Agent: Mozilla
Psu-Referer: http://en.wikipedia.org/wiki/Main_Page
Accept: application/hal+json
Psu-Accept-Charset: en-US
Authorization: authorization_example
Psu-Accept-Language: gzip, deflate
Psu-Http-Method: POST
Psu-Ip-Port: 12345
Psu-Ip-Address: 10.10.10.10
Psu-Accept: text/plain
Psu-TimeStamp: 2017-06-08T09:33:55.954+02:00
Psu-Accept-Encoding: utf-8
Content-Type: application/json
User-Agent: Swagger-Codegen/1.0.0/java
```

7.2.1.2. Body

No body data

7.2.2. Response

```
Status code: 200
```

7.2.2.1. Headers

```
Server: Apache-Coyote/1.1
Content-Type: application/hal+json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 12 Jul 2017 13:43:10 GMT
```

7.2.2.2. Body

```
{
   "GrpHdr" : {
      "MsgId" : "MSG092GH",
      "CredDtTm" : "2017-07-12T13:43:10.970Z",
      "NbOfTxs" : 1,
      "InitgPty" : {
         "Nm" : "MyPisp",
         "PstlAdr" : {
            "Ctry" : "FR",
            "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
         },
         "Id" : {
            "OrgId" : {
               "Othr" : {
                  "Id" : "JDJZH453",
                  "Issr" : "ACPR"
               }
            }
         }
      }
   },
   "PmtInf" : {
      "PmtInfId" : "MyPmtInfId",
      "PmtInfRscId" : "MyPmtInfRscId",
      "PmtMtd" : "TRF",
      "PmtTpInf" : {
         "InstrPrty" : "NORM",
         "SvcLvl" : {
            "Cd" : "SEPA"
         },
         "CtgyPurp" : {
            "Cd" : "DVPM"
         }
      },
      "ReqdExctnDt" : "2017-07-14T13:43:10.981Z",
      "Dbtr" : {
         "Nm" : "MyCustomer",
         "PstlAdr" : {
            "Ctry" : "FR",
            "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
         }
      },
      "ChrgBr" : "SLEV",
      "CdtTrfTx" : [ {
         "PmtId" : {
            "InstrId" : "MyInstrId",
            "InstrRscId" : "MyInstrRscId",
            "EndToEndId" : "MyEndToEndId"
         },
         "Amt" : {
            "InstdAmt" : "123,45",
            "Ccy" : "EUR"
         },
         "Cdtr" : {
            "Nm" : "myMerchant",
            "PstlAdr" : {
               "Ctry" : "FR",
               "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
            }
      },
      "CdtrAcct" : {
        "Id" : {
          "Iban" : "YY64COJH41059545330222956960771321"
        }
      },
      "UltmtCdtr" : {
        "Nm" : "myUltimateMerchant",
        "PstlAdr" : {
          "Ctry" : "FR",
          "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
        }
      },
      "Purp" : {
        "Cd" : "COMC"
      },
      "RmtInf" : {
        "Ustrd" : [ "MyRemittanceInformation" ]
      }
    } ]
  },
  "SplmtryData" : {
    "Envlp" : {
      "successfulReportUrl" : "http://myPisp/PaymentSuccess",
      "unsuccessfulReportUrl" : "http://myPisp/PaymentFailure"
    }
  },
  "_links" : {
    "status" : {
      "href" : "v1/payments/paymentReport/MyPmtInfRscId"
    }
  }
}
```

7.3. Payment Initiation Status Report Request

7.3.1. Request

```
GET http://localhost:8080/v1/payment-requests/MyPmtInfRscId/report
```

7.3.1.1. Headers

```
Date: 2017-07-12T15:43:11.220+02:00
Psu-User-Agent: Mozilla
Psu-Referer: http://en.wikipedia.org/wiki/Main_Page
Accept: application/hal+json
Psu-Accept-Charset: en-US
Authorization: authorization_example
Psu-Accept-Language: gzip, deflate
Psu-Http-Method: POST
Psu-Ip-Port: 12345
Psu-Ip-Address: 10.10.10.10
Psu-Accept: text/plain
Psu-TimeStamp: 2017-06-08T09:33:55.954+02:00
Psu-Accept-Encoding: utf-8
Content-Type: application/json
User-Agent: Swagger-Codegen/1.0.0/java
```

7.3.1.2. Body

No body data

7.3.2. Response

```
Status code: 200
```

7.3.2.1. Headers

```
Server: Apache-Coyote/1.1
Content-Type: application/hal+json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 12 Jul 2017 13:43:10 GMT
```

7.3.2.2. Body

```
{
  "OrgnlPmtInfId" : "MyPmtInfId",
  "OrgnlNbOfTxs" : 1,
  "PmtInfSts" : "ACSP",
  "TxInfAndSts" : [ {
    "OrgnlInstrId" : "MyInstrId",
    "OrgnlEndToEndId" : "MyEndToEndId",
    "TxSts" : "ACSP",
    "OrgnlTxRef" : {
      "Amt" : {
        "InstdAmt" : "124.35",
        "Ccy" : "EUR"
      },
      "ReqdExctnDt" : "2016-12-30T23:00:00.000Z",
      "PmtTpInf" : {
        "InstrPrty" : "NORM",
        "SvcLvl" : {
          "Cd" : "SEPA"
        },
        "CtgyPurp" : {
          "Cd" : "DVPM"
        }
      },
      "PmtMtd" : "TRF",
      "RmtInf" : {
        "Ustrd" : [ "MyRemittanceInformation" ]
      },
      "Dbtr" : {
        "Nm" : "MyCustomer",
        "PstlAdr" : {
          "Ctry" : "FR",
          "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
        }
      },
      "DbtrAgt" : {
        "FinInstnId" : {
          "Bicfi" : "BNKCEUEUXXX"
        }
      },
      "CdtrAgt" : {
        "FinInstnId" : {
          "Bicfi" : "BNKMEUEUXXX"
        }
      },
      "Cdtr" : {
        "Nm" : "myMerchant",
        "PstlAdr" : {
          "Ctry" : "FR",
          "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
        }
      },
      "CdtrAcct" : {
        "Id" : {
          "Iban" : "YY64COJH41059545330222956960771321"
        }
      },
      "UltmtCdtr" : {
        "Nm" : "myPreferedUltimateMerchant"
      }
    }
  } ],
  "_links" : {
    "self" : {
      "href" : "v1/payments/MyPmtInfRscId"
    },
    "confirmation" : {
      "href" : "v1/payments/MyPmtInfRscId/transactions/{id}/confirmation"
    },
    "paymentReport" : {
      "href" : "v1/payments/MyPmtInfRscId/report"
    }
  }
}
```

7.4. Payment Initiation Request Confirmation

7.4.1. Request

```
POST http://localhost:8080/v1/payment-requests/MyPmtInfRscId/confirmation
```

7.4.1.1. Headers

```
Date: 2017-07-12T15:43:11.413+02:00
Psu-User-Agent: Mozilla
Psu-Referer: http://en.wikipedia.org/wiki/Main_Page
Accept: application/hal+json
Psu-Accept-Charset: en-US
Authorization: authorization_example
Psu-Accept-Language: gzip, deflate
Psu-Http-Method: POST
Psu-Ip-Port: 12345
Psu-Ip-Address: 10.10.10.10
Psu-Accept: text/plain
Psu-TimeStamp: 2017-06-08T09:33:55.954+02:00
Psu-Accept-Encoding: utf-8
Content-Type: application/json
User-Agent: Swagger-Codegen/1.0.0/java
```

7.4.1.2. Body

No body data

7.4.2. Response

```
Status code: 200
```

7.4.2.1. Headers

```
Server: Apache-Coyote/1.1
Content-Type: application/hal+json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 12 Jul 2017 13:43:11 GMT
```

7.4.2.2. Body

```
{
  "OrgnlPmtInfId" : "MyPmtInfId",
  "OrgnlNbOfTxs" : 1,
  "PmtInfSts" : "ACSP",
  "TxInfAndSts" : [ {
    "OrgnlInstrId" : "MyInstrId",
    "OrgnlEndToEndId" : "MyEndToEndId",
    "TxSts" : "ACSP",
    "OrgnlTxRef" : {
      "Amt" : {
        "InstdAmt" : "124.35",
        "Ccy" : "EUR"
      },
      "ReqdExctnDt" : "2016-12-30T23:00:00.000Z",
      "PmtTpInf" : {
        "InstrPrty" : "NORM",
        "SvcLvl" : {
          "Cd" : "SEPA"
        },
        "CtgyPurp" : {
          "Cd" : "DVPM"
        }
      },
      "PmtMtd" : "TRF",
      "RmtInf" : {
        "Ustrd" : [ "MyRemittanceInformation" ]
      },
      "Dbtr" : {
        "Nm" : "MyCustomer",
        "PstlAdr" : {
          "Ctry" : "FR",
          "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
        }
      },
      "DbtrAgt" : {
        "FinInstnId" : {
          "Bicfi" : "BNKCEUEUXXX"
        }
      },
      "CdtrAgt" : {
        "FinInstnId" : {
          "Bicfi" : "BNKMEUEUXXX"
        }
      },
      "Cdtr" : {
        "Nm" : "myMerchant",
        "PstlAdr" : {
          "Ctry" : "FR",
          "AdrLine" : [ "120, rue de La Gare", "75012 Paris" ]
        }
      },
      "CdtrAcct" : {
        "Id" : {
          "Iban" : "YY64COJH41059545330222956960771321"
        }
      },
      "UltmtCdtr" : {
        "Nm" : "myPreferedUltimateMerchant"
      }
    }
  } ],
  "_links" : {
        "self" : {
            "href" : "v1/payments/MyPmtInfRscId"
        },
        "confirmation" : {
            "href" : "v1/payments/MyPmtInfRscId/transactions/{id}/confirmation"
        },
        "paymentReport" : {
            "href" : "v1/payments/MyPmtInfRscId/report"
        }
    }
}
```