2019

PSD2 Gateway Operational Guidelines for AISs and PISs

VERSION 1.0

fabrick

Summary

1. OVERVIEW 4
1.1 Context and purpose of the document 4
1.2 The Fabrick Platform Ecosystem 5
1.3 Legal framework 6
2. Northbound. Public Layer for TPPs 10
2.1 Berlin Group. Fabrick PSD2 Gateway Standard 10
2.2 Environments 12
2.3 TPP Authentication 13
2.3.1 eIDAS Certificates Background 13
2.3.2 Gateway Certificate Tasks and Public Register 16
2.3.3 TPP Onboarding 17
2.3.4 Company data required for Onboarding 21
2.3.5 Real Time checks on TPP (Certificate Validation) 24
2.4 Strong Customer Authentication 24
2.4.1 SCA 24
2.4.2 Redirect Approach 25
2.4.3 SCA using the OAuth2 approach 27
2.4.4 SCA Using Decoupled Approach 27
2.4.5 SCA Using the Embedded Approach 28
2.4.6 General considerations about SCA 30
2.5 PISP API Flows 32
2.5.1 PISP flows 32
2.5.2 Single Payment 35
2.5.3 Recurring Payments 39
2.5.4 Future Date Payments 41
2.5.5 Bulk Payments 41
2.6 AISP API Flows 42
2.6.1 Consent Management 42
2.6.2 Get list of reachable accounts 46
2.6.3 Get account details of a list of accessible accounts 47
2.6.4 Account Transactions 48
2.6.5 Balance 49

2.7 PIISP Api Flow 49
2.7.1 Funds Availability 49
2.8 Fabrick delta fields comparing to BG standard 51
2.9 Public Portal 53
2.9.1 Online Documentation 53
3 Environments 54
3.1 Development environment 54
3.2 Live environment 55
3.3 Table of domains 56
4 Support and Help Desk 57
4.1 Tools used 57
4.2 Credentials 57
4.3 Help desk and troubleshooting processes 58

fabrick

GLOSSARY

Abbreviations	Meaning
AISP	Account Information Service Provider
API	Application Programming Interface
ASPSP	Account Servicing Payment Service Provider
CA	Certification Authority
EBA	European Banking Authority
eIDAS	electronic IDentification, Authentication and trust Services
EU	European Union
Json	Javascript object notation
PIISP	Payment Instrument Issuer Service Providers
PISP	Payment Initiation Service Provider
PSD2	Payment Service Directive 2 (EU 2015/2366)
RTS	Regulatory Technical Standard
SSL	Socket Secure Layer
TPP	Third Party Providers
URN	Unique Reference Number
Xml	eXtensible markup language
XS2A	Access to account interface (Interface to be implemented between TPPs and ASPSPs in order to provide access to accounts)

fabrick

1. OVERVIEW

1.1 Context and purpose of the document

The purpose of this document is to provide guidelines to TPPs (PISs, AISs and PIISs) in order to implement the Fabrick Platform API interface.

The Fabrick PSD2 gateway is a software platform offered to servicing payment service providers (i.e. payment institutions, electronic money institution and credit institution) aimed at creating an API interface dedicated to Third Party Providers in the manner defined by the EU directive 2015/2366 and subsequent EU RTS 2018/389.

This document will not address in detail the issues and the technicalities introduced by this regulatory framework and will refer to the reading of the PSD2 directive for any clarification of the context.

The document will explain the ways in which the API gateway will provide interfaces, modes, functions and processes implemented, as well as the technological architecture and the connection with the ASPSPs on the gateway, to TPPs.

Given the “handbook” nature of this document, the available features on the gateway will be handled in detail, while also providing links to other detailed documents in order to describe technical aspects or to understand specific vocabulary. Furthermore, this document will contain links to documentation regarding widespread and well-known market standards concerning both the “open banking” standards and the most common technological standards.

1.2 The Fabrick Platform Ecosystem

Fabrick will develop the PSD2 Gateway in the context of a pre-existing open banking platform based upon a “web API” technology.

As such, the PSD2 gateway can be considered as a "platform-in-a-platform".

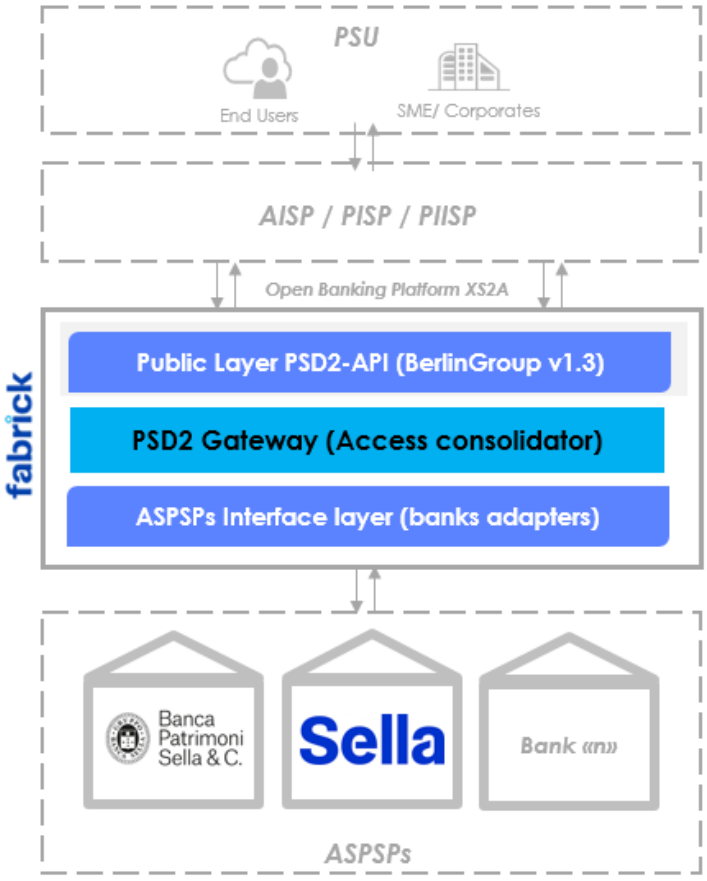
Indeed, the list of "banking" functions available on the Fabrick platform is overall more complete than the perimeter required by the PSD2 directive.

However, the ultimate goal of the gateway is to enable strict compliance with the RTSs and to adopt the most common and widespread European standard. The gateway, in comparison to the platform as a whole, will deliver a narrower set of functions but in a more standardized way, involving, potentially, many banking institutions. This will allow the adoption of a common standard and promote interoperability.

Overall, the business architecture enabled by the gateway is the one provided by the XS2A interface (Access to Accounts) provided by the PSD2 directive.

Indeed, as illustrated in Figure 1 below, the gateway implements the access interface to payment accounts on the one hand, while on the other it allows to connect to ASPSPs (generally banks).

Figure 1



fabrick

1.3 Legal framework

With the PSD2 the European Union has published a new directive on payment services concerning the domestic market. Member States have adopted the directive into their national law by January 13th, 2018.

Of the many different aspects covered, the PSD2 contains regulations relative to new services operated by so-called Third Party Payment Service Providers (or TPPs) on behalf of a Payment Service User (PSU). These new services can be:

- Payment Initiation Service (PIS) to be operated by a Payment Initiation Service Provider (PISP) TPP as defined by article 66 of PSD2,
- Account Information Service (AIS) to be operated by an Account Information Service Provider (AISP) TPP as defined by article 67 of PSD2, and
- Confirmation of the Availability of Funds service to be used by Payment Instrument Issuing Service Provider (PIISP) TPP as defined by article 65 of PSD2.

In order for said new services to operate, the TPP will need to access the account of the PSU, which is usually managed by another PSP, known as the Account Servicing Payment Service Provider (ASPSP). As shown in the following figure 1, an ASPSP has to provide an interface (called "PSD2 compliant Access to Account Interface" or short "XS2A Interface") to its systems to be used by a TPP for necessary accesses regulated by the PSD2.

Regulatory Technical Standards (RTS) from the European Banking Authority (EBA), and published in the Official Journal of the European Commission define further how to comply with the usability requirements.

There are other important documents for a deeper understanding of the evolving legal framework in which the Fabrick platform operates, and we of here a non-exhaustive selection for reference:

Regulatory Technical Standards on strong customer authentication and secure communication under PSD2

The proposed Regulatory Technical Standards on strong customer authentication and secure communication are key to achieving the objective of the PSD2 of enhancing consumer protection, promoting innovation and improving the security of payment services across the European Union.
Status: Final draft adopted by the EBA and submitted to the European Commission

Regulatory Technical Standards on passporting under PSD2

These Regulatory Technical Standards (RTS) on the framework for cooperation and exchange of information between competent authorities for passporting will ensure that information about those payment institutions that carry out business in one or more EU Member States is exchanged consistently between the national authorities of the home and host Member States. They distinguish the notifications related to branch establishments, agent's engagement and free provision of services.
Status: Adopted and published in the Official Journal

Guidelines on authorisation and registration under PSD2

These Guidelines are in support of the objective of PSD2 of strengthening an integrated payments market across the European Union, ensuring a consistent application of the legislative framework, and promoting equal conditions for competition.

Status: Final and translated into the EU official languages

Guidelines on major incidents reporting under PSD2

These Guidelines are in support of the objectives of the PSD2 of strengthening the integrated payments market across the European Union, ensuring a consistent application of the legislative framework, promoting equal conditions for competition, providing a secure framework on the payments environment and protecting consumers.
Status: Final and translated into the EU official languages

Guidelines on procedures for complaints of alleged infringements of the PSD2

These draft Guidelines are part of the EBA's work to support the core objectives of the PSD2 of strengthening the integrated payments market across the European Union, ensuring a consistent application of the legislative framework and promoting transparency.
Status: Final and translated into the EU official languages

Guidelines on security measures for operational and security risks under the PSD2

The Guidelines have been developed in close cooperation with the European Central Bank (ECB), and are in support of the objectives of PSD2, such as strengthening the integrated payments market in the EU, mitigating the increased security risks arising from electronic payments, and promoting equal conditions for competition.
Status: Final and translated into the EU official languages

Regulatory Technical Standards on central contact points under PSD2

These draft RTS specify the criteria for determining when the appointment of a central contact point under the PSD2 is appropriate and the functions that these contact points should have.
Status: Final draft adopted by the EBA and submitted to the European Commission

Guidelines on the criteria on how to stipulate the minimum monetary amount of the professional indemnity insurance under PSD2

These Guidelines define the criteria Competent Authorities should consider when stipulating the minimum monetary amount of the professional indemnity insurance (PII) or comparable guarantee for payment initiation and account information service providers under the revised Payment Service Directive (PSD2)
Status: Final and translated into the EU official languages

Technical Standards on the EBA Register under PSD2

These draft regulatory technical standards (RTS) and implementing technical standards (ITS) on the electronic central register under the Payment Services Directive (Directive (EU) 2015/2366) (PSD2) respectively set requirements on the development, operation and maintenance of the register and the information to be contained in it.
Status: Final draft adopted by the EBA and submitted to the European Commission

Guidelines on fraud reporting under PSD2

The Guidelines, which are addressed to payment service providers and competent authorities, are aimed at contributing to the objective of PSD2 to increase the security of retail payments in the EU.
Status: Final and translated into the EU official languages

Guidelines on the conditions to be met to benefit from an exemption from contingency measures under Article 33(6) of Regulation (EU) 2018/389 (RTS on SCA & CSC)

These Guidelines clarify a number of issues identified by market participants and competent authorities in relation to the four conditions to be met to benefit from an exemption from the fallback option envisaged under Article 33(6) of Regulation (EU) 2018/389 (RTS) on strong customer authentication and common and secure communication (SCA and CSC).
Status: Under development

(325.1k)

Communications underpin the new security requirements under PSD2 and regulate access by Account Information Service Providers (AISPs), Payment Initiation Service Providers (PISPs). The Board of Supervisors has adopted said opinion on the use of eIDAS certificates for PSD2 purposes and based on feedback from the industry, and the EBA.

Furthermore, the EBA opinion paper on SCA and CSC is of particular importance:

- Opinion on the implementation of the RTS on SCA and CSC (EBA-2018-Op-04) [PDF, 192KB]

The document presents a table that clearly summarizes what are the relevant points and features to be enabled for the API initiatives:

Requirement	**Article**
Enabling CBPIIs, AISPs and PISPs to access the necessary data from payment accounts accessible online	Articles 65, 66 and 67 PSD2 Article 30 RTS
Conforming to (widely used) standard(s) of communication issued by international or European standardization organisations	Article 30(3) RTS
Allowing the payment service user (PSU) to authorise and consent to a payment transaction via a PISP	Article 64(2) PSD2 Article 30(1)(c) RTS
Enabling PISPs and AISPs to ensure that, when they transmit the personalised security credentials issued by the ASPSP, they do so through safe and efficient channels	Articles 66(3)(b) and 67(2)(b) PSD2
Enabling the identification of the AISP/PISP/CBPII and supporting eIDAS certificates	Articles 65(2)(c), 66(2)(d) and 67(2)(c) PSD2 Articles 30(1)(a) and 34 RTS
Allowing 90-day reauthentication for AISPs	Article 10(2)(b) RTS
Enabling the ASPSPs and AISPs to count the number of access requests during a given period	Article 36(5) RTS
Allowing a change control process	Article 30(4) RTS
Allowing the possibility of cancelling an initiated transaction in accordance with PSD2, including recurring transactions	Articles 64(2), 80(2) and 80(4) PSD2
Allowing error messages explaining the reason for the unexpected event or error	Article 36(2) RTS
Supporting access via technology service providers on behalf of authorised actors	Article 19(6) PSD2
Allowing AISPs and PISPs to rely on all authentication procedures issued by the ASPSP to its customers	Article 97(5) PSD2 Article 30(2) RTS
Enabling the AISP to access the same information as is accessible to the individual consumer and corporates in relation to their designated payment accounts and associated payment transactions	Article 67(2)(d) PSD2 Articles 30(1)(b) and 36(1)(a) RTS
Enabling the ASPSP to send, upon request, an immediate yes/no confirmation to the PSP (PISP and CBPII) on whether or not there are funds available	Article 36(1)(c) RTS
Enabling dynamic linking to a specific amount and payee, including batch payments	Article 97(2) PSD2 Article 5 RTS
Enabling the ASPSP to apply the same exemptions from SCA for transactions initiated by PISPs as when the PSU interacts directly with the ASPSP	Articles 18(2)(c)(v) and (vi), 18(3), 30(2) and 32(3) RTS

Enabling SCA composed of two different elements	Article 4 RTS
Enabling a secure data exchange between the ASPSP and the PISP, AISP and CBPII, mitigating the risk of any misdirection of communication to other parties	Articles 28 and 35 RTS
Ensuring security at transport and application levels	Article 97(3) PSD2 Articles 30(2)(c) and 35 RTS
Supporting the needs to mitigate the risk of fraud, have reliable and auditable exchanges and enable providers to monitor payment transactions	Article 97(3) PSD2 Articles 3, 22 and 35 RTS
Allowing traceability	Article 29 RTS
Allowing the ASPSP's dedicated interface to provide at least the same availability and performance as the user interface	Article 32 RTS

fabrick

2. Northbound. Public Layer for TPPs

2.1 Berlin Group. Fabrick PSD2 Gateway Standard

The Fabrick PSD2 Gateway has decided to adopt the NextGenPSD2 standard of the Berlin Group initiative.

The Berlin Group is a Joint Initiative on a PSD2 Compliant XS2A Interface operational at the pan-European level.

The reasons for this choice are as follows:

- Fabrick understands and encourages the statement of a European standard for API interoperability in the electronic payment industry in order to avoid the risks and costs of a “too fragmented” market.
- Because of this determination, Fabrick has carried out an analysis of the standardization initiatives under way in Europe, identifying only two having a degree of maturity and reliability to be a valid option. One is “NextGenPSD2 of the Berlin Group” and the other is “Open Banking UK” standard currently in force at the banks participating in the English Open Banking initiative. However, we felt that the Berlin Group's standard is better suited to the RTS dictates, which is in force in the rest of the European Union because it was designed "from the scratch" precisely to satisfy those requirements.
- The emerging scenario would seem to point to the fact that most initiatives for banks and PSD2 gateways, in Italy as in the rest of the continent, are adopting this specific standard.

The Berlin Group has today released three main versions of the XS2A interface standard. Fabrick is in line with the most recent version as of today's date (v.1.3 - October 19, 2018).

The entire northbound interface of the Fabrick PSD2 Gateway implements this standard and the following paragraphs will seek to illustrate practical use case examples. Nonetheless, Fabrick, or the banks and other ASPSPs that are its customers, may adopt implementation choices that are dependent on the peculiarities of each.

Furthermore, it is necessary to underline that this analysis will not report all the basic choices of the protocol (such as charset, the organization of resources and endpoints, nor the details of individual messages).

For these details, refer to the following documents:[1]

- 01. NextGenPSD2 Access to Account Interoperability Framework - General Introduction Paper V1_20180208.pdf
- 02. NextGenPSD2 Access to Account Interoperability Framework - Operational Rules V1_20180208.pdf
- 03. NextGenPSD2 Access to Account Interoperability Framework - Implementation Guidelines V1.3_20181019.pdf
- 04. NextGenPSD2 Access to Account Interoperability Framework - ChangeLog V12 V13 20181019.pdf

[1] Most recent BG documents are available at: https://www.berlin-group.org/nextgenpsd2-downloads/

It is also important to underline that the standard allows some basic choices for those who are going to implement it, such as:

1) NextGenPSD2 supports two types of possible message payloads: json or xml in ISO 20022 format. This choice was made in order to support backward compatibility to systems that currently implement the ISO20022, which is very common in electronic payment systems in Europe today. Furthermore, the method of choice for serializing the REST APIs by the global API industry is the JSON format. Fabrick endorses the distribution of this method. For this reason, the ISO20022 XML standard will be used in the Fabrick XS2A interface only where it is otherwise impossible to do without it. Most messages will require a JSON type serialization. In all the cases where both type of serialization are allowed JSON will be used.
2) The RTS and consequently the standard provide for the possibility of digitally signing individual messages via seal certificates if the ASPSP requires it.
   Given the implicitly secure nature of the chosen transport channel (ssl) and the use of eIDAS digital certificates for the mutual authentication of peers in the connection, we do not believe that the signature of individual messages entails a substantial reduction in the risk of repudiation of transactions. Therefore, to date, Fabrick does not support this option also because none of its client banks has ever explicitly requested this additional level of security.
3) To date, the Berlin Group standard covers a list of payment features that represents the common functional subset across Europe. As far as payments are concerned, the functionalities concentrate on the “wire transfer” functions in place in the most common circuits between banks (for example SEPA, SWIFT, TARGET2 etc., etc.). It is possible that the development of the functional perimeter following the evolution of coverage requirements will grow in time depending on what will be established between the API Evaluation Group and future EBA guidelines, involving the national schemes and payment systems in use in each country of the European Union.
   Should such a scenario occur while awaiting a definition of such payment schemes on the Berlin Group interface, Fabrick will design APIs that match the underlying philosophy promulgated by the NextGENPSD2, seeking to respect the methods of interaction between PISPs and ASPSPs.
   At the present date, the Fabrick PSD2 gateway covers exclusively the payment functionalities provided by the NextGENPSD2 XS2A interface.
4) As of today, Fabrick has chosen not to support the "basket of requests" option that can be managed with a single SCA. This flow of calls is controversial from the point of view of regulatory obligations and in all probability, the ASPSPs could not allow the execution of this type of call by virtue of the need (emphasized by the RTS) to carry out the SCA for each individual payment. Moreover, for "massive" payments from a single payment account, the standard provides for a special "Bulk Payments" function supported by Fabrick[2].

[2] See chapter 2.5.4 for details about Bulk Payments

5) As of today, Fabrick has chosen not to support the option to initialize a payment that uses the “id” of a previously provided consent. This use case could allow a TPP that performs both the roles of AIS and PIS (mixed flow), to perform the SCA without having to identify the user again in the creation of the authorization resource. However, we believe this approach is "confused" and controversial compared to what established by the RTS regarding the need to carry out the Strong Customer Authentication for each payment.
6) As of today, Fabrick allows you to query the account balance in the "balance" flow only and does not support the reading of the balances in the "Accounts" and "Transactions" functions.
7) Payment with multiple authorization (Multilevel SCA Approach) and the details of a single payment transaction are functions that, as of today's date, no participating ASPSP needs to offer. These flows will be enabled only in the presence of ASPSP that offer these functions in their online services.

We emphasize that all the choices made are included among those admitted within the protocol itself. All mandatory fields and flows have been implemented.

2.2 Environments

The northbound interface of the Fabrick PSD2 Gateway will have two (2) available environments for TPPs: a DEVELOPMENT environment; and a LIVE environment. This distinction serves the ability to separate production operations and development activities for security and functional purposes. The DEVELOPMENT environment includes a TEST environment and a SANDBOX environment. These environments will respond to the needs of TPPs to develop, test and operate in a live environment with their own solutions.

These environments will respond to the needs of TPPs to develop, test and operate in a live environment with their own solutions.

Note that environments are multi-tenant in the SANDBOX (and LIVE) environments, while in the TEST environment, by not closing the end-to-end test with the staging environments of the banks, the structure is "mono- tenant”.

These environments and their characteristics are summarized in the following table:

Environment		Access	TPP Identification	ASPSP	Domain
DEVELOPMENT	TEST	TPP Authorized and TPP waiting for authorization	Any valid client certificate	In TEST environment the ASPSP is a bank simulation software that returns mocked-up informations. To access this environment, a	test-psd2gateway.fabrick.com

				X509 certificate will suffice.	
	SANDBOX	TPP Authorized and TPP waiting for authorization	Any valid client certificate	In SANDBOX environment there's a set of endpoints for each single ASPSP on the https connection. A domain for each single ASPSP will be provided. This environment will be connected to each ASPSP of development environment.	sandbox-psdgw-{tennantname}.fabrick.com Example:
LIVE		TPP Authorized	eIDAS qualified certificate	A certificate for each single ASPSP on the https connection. A domain for each single ASPSP will be provided. To operate in this environment a qualified certificate needs to be presented.	psdgw-{tennantname}.fabrick.com Example:

2.3 TPP Authentication

2.3.1 eIDAS Certificates Background

As required by the RTS in web-API connections, the parties will perform the mutual authentication through special eIDAS digital identity certificates[3].

Furthermore, the RTS provides two types of certificates:

QWAC- Certificates used for mutual authentication of peers that are connected through an SSL session

QSEAL - Seal certificates that have the function of "signing" the individual transactions.

The former are used at the "transport" level and therefore in the management of the https sessions of communication between client and server API, while the latter are used for encryption and signing for the application and protocol level each single call to the gateway.

[3] RTS 2018/239 EU. Art. 34. certificates

We emphasize that, based on the "Opinion on the use of eIDAS certificates under the RTS on SCA and CSC" issued by EBA in December 2018, strengthened by the outcome of the work of the API Evaluation Group, in which the subject was substantially in agreement, the obligations in terms of using eIDAS certificates are configured in the following pattern:

1. The use of QSEAL certificates is optional. **By default, the Fabrick PSD2 Gateway does not oblige TPPs to sign transactions unless a single participating ASPSP decides to impose this choice**.

2. With regard to the QWAC certificates, it is now clear that only for the TPP the usage is mandatory. The qualified certificate, in fact, is decisive in order to allow the dedicated interface to recognize the PSD2 role of the connected TPP. We also believe that the use of eIDAS certificates is not mandatory on the part of the ASPSP especially when they use "access consolidation" infrastructures such as the Fabrick PSD2 platform.

   As a consequence, the PSD2 gateway chooses by default to expose the single ASPSP APIs set to a dedicated “second level” domain, using a single X509 certificate common to all participants. If one or more ASPSP members of the platform wish to have a "site" with dedicated endpoints, by submitting their own certificate, they can request it and Fabrick will manage the endpoints with dedicated top-level domains on their systems. To obtain this result, Fabrick will have to install the single ASPSP certificate (eIDAS or less) on its systems and the public DNS must be appropriately configured by the ASPSP to point to the systems supplied by Fabrick.

Qualified Trusted Service Provider should issue these certificates “as referred to in Article 3(30) of Regulation (EU) No 910/2014 or for website authentication as referred to in Article 3(39) of that Regulation”.

In order to agree on a standard for certificates, the ETSI (European Telecommunications Standards Institute) issued a specific specification document: ETSI TS 119 495 V1.1.2 (2018-07). The document is available at the link:

https://www.etsi.org/deliver/etsi_ts/119400_119499/119495/01.01.02_60/ts_119495v010102p.pdf

This TSP Policy requirement is mandatory to be fully compliant with the aforementioned RTS.

These qualified certificates, in addition to guaranteeing the identity of peers in an ssl connection, will contain some specific attributes aimed at giving some additional information on the connected counterpart:

- The “Authorization Number”. This is a global URN related to the single peer (TPP or ASPSP) that is required to be available in the public registry of the National Competent Authority. This number should be expressed in a specific format:

GEN-5.2.1-3: The organizationIdentifier attribute shall contain information using the following structure in the presented order:

• "PSD" as 3 character legal person identity type reference;

• 2 character ISO 3166 [7] country code representing the NCA country; • hyphen-minus "-" (0x2D (ASCII), U+002D (UTF-8)); and

• 2-8 character NCA identifier (A-Z uppercase only, no separator);

• hyphen-minus "-" (0x2D (ASCII), U+002D (UTF-8)); and

• PSP identifier (authorization number as specified by the NCA).

EXAMPLE: The organizationIdentifier "PSDES-BDE-3DFD21" means a certificate issued to a PSP where the authorization number is 3DFD21, authorization was granted by the Spanish NCA Banco de España (identifier after second hyphen-minus is decided by Spanish numbering system).

Any separator in NCA identifier shall be removed

- The “Roles of payment service provider”. This attribute shall contain one or more roles. The roles shall be declared by an NCA via their public registry for PSP subjects. The role name shall be:

Abbreviation	PSD Role
PSP_AP	APSPS Account Servicing Payment Service Provider
PSP_PI	PISP Payment Initiator Service Provider
PSP_AI	AISP Account Information Service Provider
PSP_IC	PIISP Payment Instrument Issuer Service Providers

- The “Name and identifier of the competent authority”. Expressed in this format:

The NCAId shall contain information using the following structure in the presented order:

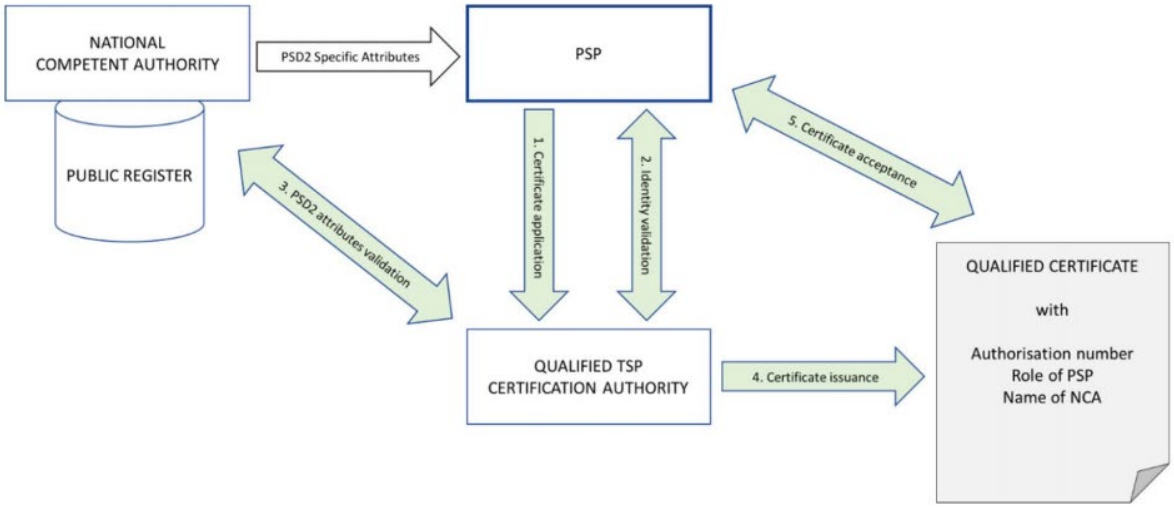
• 2 character ISO 3166 [7] country code representing the NCA country;

• hyphen-minus "-" (0x2D (ASCII), U+002D (UTF-8)); and

• 2-8 character NCA identifier (A-Z uppercase only, no separator).

The certificate issuance will follow this flow:

Figure 2



Before the issuance process can start, the PSP needs to be registered by an NCA and all relevant information needs to be available in the public registry:

1) The PSP submits the certificate application and provides all necessary documentation containing PSD2 specific attributes to the Trust Service Provider (TSP) with granted qualified status according to eIDAS.

2) The TSP performs identity validation as required by its certificate policy.

3) The TSP validates PSD2 specific attributes using information provided by the NCA (e.g. public register, authenticated letter).

4) The TSP issues the qualified certificate in compliance with the profile requirements given in the present document.

5) The PSP accepts the certificate.

2.3.2 Gateway Certificate Tasks and Public Register

In this context, the Fabrick PSD2 Gateway has two tasks in order to enable the aforementioned mechanism of mutual authentication in a manner compliant with the law:

A. Exposing, if requested by the single ASPSP, their own qualified certificates on the client bank endpoints.

Regarding this first point, we underline how the legislation does not have a specific role for API initiatives for access consolidation. This determines that as a supplier of ASPSPs the gateway will have to expose differentiated endpoints for each customer bank in order to use for the mutual authentication eIDAS certificates purchased by themselves.

The global uri structure is:

https://**{provider}**/v**{version}**/**{service-endpoint}**

where the provider can be:

Environment	Provider	Certificate
Test. Mock up for TPP developing purpose (see Enviroments chapter)	test-psd2gateway.fabrick.com	NotQualified OV – Organization Validated
Sandbox on each single bank	sandbox-psdgw-{tennantname}.fabrick.com	NotQualified OV – Organization Validated
Production on each single bank	psdgw-{tennantname}.fabrick.com	Whenever requested qualified purchased by bank... otherwise by default NotQualified OV – Organization Validated

B. Authenticate and authorize API invocation with respect to certificates used by TPPs.

The other significant task concerns the validation and verification of the certificates used by Third Party Providers.

We must consider that:

- Given the lack of passporting information within the certificate, the Fabrick Platform will build its own database of all PSPs and credit institutions that can operate as TPPs. This registry will be based on commercial registries and on the central EBA registry and will serve to validate the self-initiation of the TPP "operation by operation".

- Fabrick will adopt a digital process for TPP onboarding. The purpose of this process is to validate TPPs and to determine the registration of the TPPs that will have the right to invoke the APIs.

- Fabrick will perform real-time checks for each single call through its "TPP registry" in order to determine if the authorization is still valid and if the information on the role present in the certificate matches the information in the registry.

2.3.3 TPP Onboarding

The PSD2 directive determines that the ASPSPs are obliged to provide to authorized Third Parties the service of access to accounts and initialization of payments even in the absence of any further legal contract between the entities[4].

As a result, the Fabrick PSD2 Gateway will adopt a digital onboarding process with the aim of:

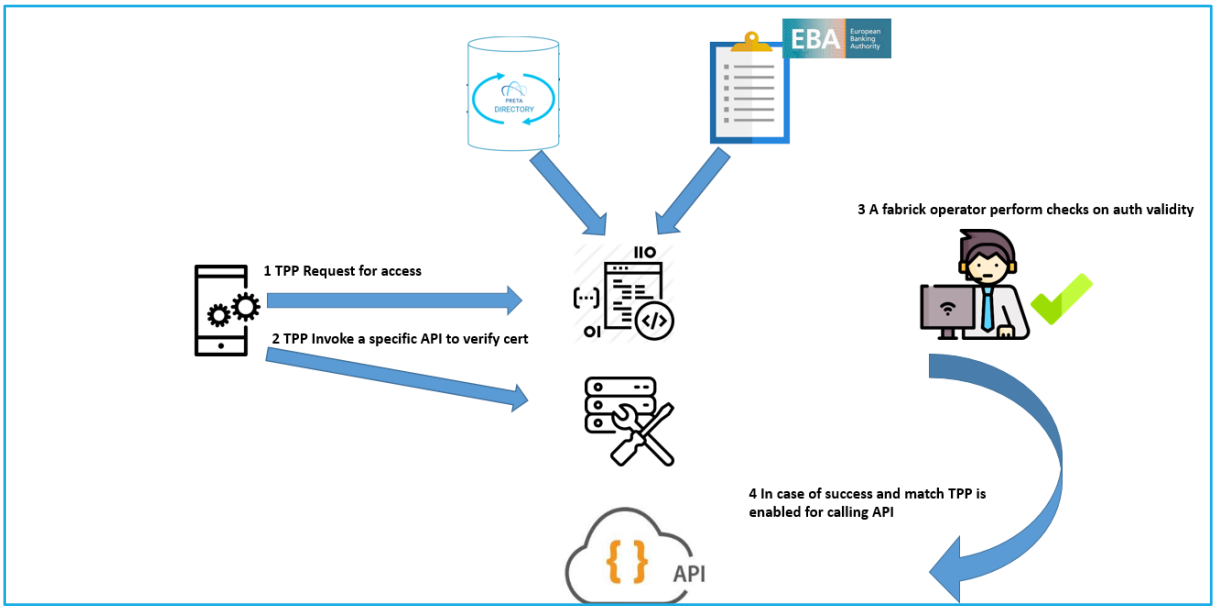
[4] Art. 65-66-67 EU 2015/2366

- Validating the identity and authorization of the TPP;
- Enabling the TPP to invoke the API according to its role (AISP, PISP, and PIISP);
- Collecting and verifying all the information necessary for the management of the relationship between Fabrick and the TPP;
- Releasing the credentials to access the platform's Web features and take advantage of services such as statistics, help desk, additional features etc. to the TPP;
- Collecting a series of optional information in order to encourage future agreements between ASPSP and TPPs to develop business cases that go beyond the perimeter of the PSD2 in a streamlined fashion.

Moreover, Fabrick will foresee a simplified onboarding process for access to test environments, at least in the official test phase scheduled between March and September 2019, in order to allow subjects awaiting authorization from their NCA to start development and integration activities.

The digital process can be summarized as follows:

Figure 3



Here we start from the consideration that the Fabrick PSD2 Gateway will have an internal registry of all the TPPs enabled in Europe and all the banks.

This registry will validate the access of the TPP and suspend or cancel the authorizations, promptly if necessary.

The phases of TPP onboarding are as follows in chronological order:

Phase 1. The TPP requires access by filling out an online form in which it will insert a series of information about its business and its Authorization Number in the context of the EBA registry. Furthermore, personal information about contact and key persons will be required.

Phase 2. Fabrick sends an email to the TPP containing a GUIDE that allows the validation of the certificate to be executed by the TPP in phase 3. This validation occurs in two distinct processes: one for the DEVELOPMENT environment and one for the LIVE environment. This is because the TPP will be allowed to access the DEVELOPMENT environment with a temporary non-qualified certificate, in order to develop and perform test calls and receive mock responses. To operate in the LIVE environment, a specific API call with a qualified certificate is required.

Fabrick will then validate each single personal account connected to the TPP on the platform through an "email approach".

WARNING: This phase will be different and segregated from the DEVELOPMENT environment access.

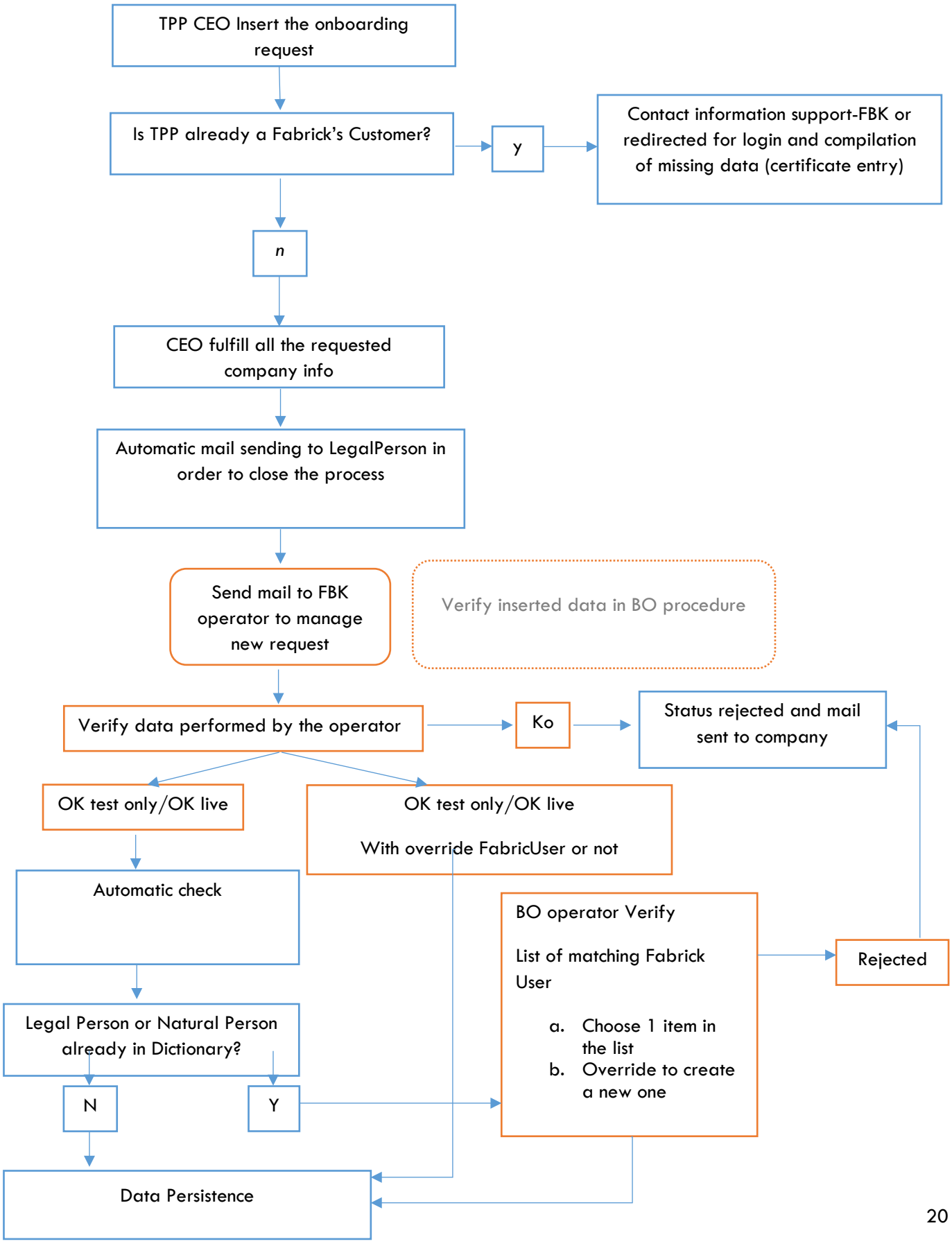
Phase 3. The TPP invokes a specific API function using its own qualified certificate or a temporary one according to which environment the TPP is going to access. This step is required in order to get a first validation and acquisition of the certificate's basic information.

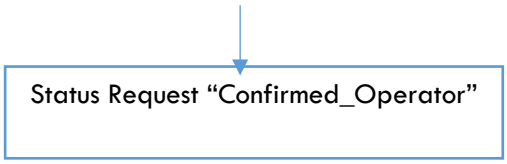
Phase 4. A Fabrick operator from the Fabrick Platform Back Office environment performs additional checks, verifying:

- The data provided by the TPP;
- The data stored in the certificate;
- The data stored in the TPP registry.

If this data is coherent, the operator will enable the TPP. As a result, Fabrick users should be created for key person to access the portal and their platform dashboard and a welcome procedure is activated.

Flow:


TPP CEO Insert the onboarding request
Is TPP already a Fabrick's Customer?
y
Contact information support-FBK or redirected for login and compilation of missing data (certificate entry)
n
CEO fulfill all the requested company info
Automatic mail sending to LegalPerson in order to close the process
Send mail to FBK operator to manage new request
Verify inserted data in BO procedure
Verify data performed by the operator
Ko
Status rejected and mail sent to company
OK test only/OK live
OK test only/OK live
With override FabricUser or not
Automatic check
BO operator Verify
List of matching Fabrick User
a. Choose 1 item in the list
b. Override to create a new one
Rejected
Legal Person or Natural Person already in Dictionary?
N
Y
Data Persistence



2.3.4 Company data required for Onboarding

The Fabrick PSD2 Gateway will have a registration request form addressed to authorize TPPs.

This module will have a first step that will allow the entering of company data and subsequent steps in which it will be possible to associate personal information of key persons within the company who have contact roles. Each of these persons will have a Fabrick account with access to the platform dashboard.

However, only the CEO or Director (company’s legal representative) will have the grant to make some choices such as disabling the company from the API gateway.

The information requested to the TPP are summarized in this table:

Field	**Type, Constraints, Structure**	**Mandatory**	**Notes**
Global URN Authorization Number	String	Conditional. Y if Sandbox and live environment has been checked	Enter a note that specifies the expected format of the URN as specified in chapter 2.2.1 with a specific example. Evaluate, if the value entered matches a company in the internal registry of the TPPs, to pre-fill the other available values of the form in a dynamic way.
Company Business Name	Long String	Y	
HEADQUARTER (This field set is mandatory)			
Nation	Choice From a list with flag and name. (List of the 31 countries where PSD2 is in force + a single voice “other” that enables a free text box)	Y	
City	String	Y	
Post Code	String	N	

State / Province	String	N	
Address	String at least 2 lines	Y	
OPERATIONAL HEADQUARTER (This field set can be the same of Headquarters). Required to be filled only if different. Otherwise, the same data will be reported in the request entity.			
State	String	N	
Post Code	String	N	
State / Province	String	N	
Address	String	N	
OTHER			
Company Email	Email Format	Y	
Company Phone Number	Phone Format (International Prefix chosen by a dropdown menu with nation + number text box for the remaining part of the number)	N	
OTHER PSD2 SPECIFIC FIELDS			
PSD Role	Multiple Choice Not Exclusive PIS AIS PIIS	At least one is mandatory	Explain that the role will be checked against the central EBA TPP Registry including passporting in the countries where ASPSPs operate.
Competent CA	Dropdown list with 31 national authorities	Y	
KEY PERSON TO CONNECT. AT LEAST CEO OR DIRECTOR IS MANDATORY.			
Role	Dropdown List with Free Text "Other" Enabled. CEO. Warning: During onboarding, only the CEO can be added. In a second phase on the dashboard they can add: • Executive Director • Director • CIO • CTO • COO • Administrative Contact • Legal Contact	Y	

	• Other…		
First Name	String	Y	
Last Name	String	Y	
Email	Email	Y	
Landline Phone Number	Phone Format (International Prefix chosen by a dropdown menu with nation + number text box for the remaining part of the number)	N	
Mobile Phone Number	Phone Format (International Prefix chosen by a dropdown menu with nation + number text box for the remaining part of the number)	N	

Other assumptions:

- At the end of the whole process, accounts for key persons will be created by requiring their email as an identifier for access. This email should be validated sending a "validate account" specific mail. The email contains also the link to the API resources required for validating the client certificate.
- The onboarding process form must contain a captcha field to avoid "machine" access.
- Due to the possible lack of qualified certificates, the access to the TEST environment may be granted also with an unqualified certificate. For this reason, the onboarding process is mandatory during the development period of a TPP due to the approach chosen by Fabrick to allow access to the development environment for TPPs that are still requesting the authorization and may not have obtained it yet. This approach is harmonized with points 45-46 of "OPINION PAPER ON CONDITIONS TO BE MET UNDER ART 33(6) OF RTS ON SCA&CSC"

2.3.5 Real Time checks on TPP (Certificate Validation)

The Fabrick PSD2 gateway will need to identify, authenticate and authorize the TPPs in each single API call.

The verification of the certificate will be performed in two separate phases:

As shown in Figure 4, the https call first goes through the layer of the hardware balancer (the overall and detailed IT architecture of the platform will be addressed in chapter 6).

In this first layer the client authentication at “level 6” of the OSI stack will be performed.

In practice, access to the resources available behind the Fabrick endpoints will not be granted if the TPP certificates are not released by a QTSP by checking the root certificate marked as “trusted” in a specific list that Fabrick will create on F5 Load Balancer.

At the same time the balancer, which in the dedicated architecture also takes care of the SSL offloading operation (SSL traffic decryption), will insert in the request flow towards the API gateways in a specific http header (named x-certificate) the whole serialized client certificate.

2.4 Strong Customer Authentication

2.4.1 SCA

Some of the following transactions (described in the flows documented in the coming chapters) require strong customer authentication (SCA) of the PSU at the XS2A interface as part of the transaction:

- Payment initiation transactions.
- Establish account information consent transactions.

Requirements for the application of SCA and possible exemptions are defined in article 97 of [PSD2] and chapter III of [EBA-RTS]. For each individual transaction, the ASPSP has to decide if the SCA has to be executed.

This decision has to be compliant with the requirements defined by PSD2 and EBA-RTS. If a SCA is necessary, then it has to be decided based on:

- The procedure; and
- The personalized security credentials to be used by the PSU.

If several SCA procedures are available for the PSU, then the ASPSP shall offer these procedures to the TPP/PSU to choose between these procedures. The specifications of the Berlin Group Joint Initiative distinguish the following four approaches to SCA as part of a transaction at the XS2A interface of an ASPSP:

- Redirect approach;
- OAuth2 approach;

- Decoupled approach; and
- Embedded approach.

The ASPSP decides which of these approaches to SCA are supported by its XS2A interface implementation. The TPP can indicate in its first message of the Payment Initiation or Establish Consent information request whether the TPP prefers a redirect-based SCA approach or not. In this context, the OAuth2 approach is also seen as a technically redirect-based SCA approach. The ASPSP shall consider this indication when deciding on the SCA approach to be performed. The XS2A documentation of the ASPSP will provide the TPP with the necessary information. If SCA is necessary as part of a transaction, the TPP has to use one of the approaches supported by the ASPSP.

As such, every single bank that will expose its APIs on the PSD2 gateway will expose the necessary functions (Southbound) to allow strong authentication by providing all the challenges it already offers the PSU on the other channels. At the same time, choose at least one of the four authentication procedures described above.

According to the Berlin Group initiative, we stress that:

- Currently, only the SCA of one PSU can be handled directly as part of the transaction at the XS2A interface. SCA for more than one PSU must be handled outside the XS2A interface. Future releases of this specification may support this approach (distributed approach) for SCA at the XS2A interface.
- Currently, only the SCA as performed by the ASPSP is directly supported as part of the transaction at the XS2A interface. The SCA method offered by PIS or AIS providers are not yet explicitly supported by the XS2A interface. Future releases may support this approach (delegated approach) for SCA at the XS2A interface.

Fabrick will publish and keep updated on its website a table that summarizes the choices made by the ASPSPs that adopt the gateway on the methods of SCA allowed. In the "TEST" environment, Fabrick will expose APIs for the embedded and redirect approaches with a "plain vanilla" authentication challenge.

2.4.2 Redirect Approach

For the redirect approach, the individual steps of the SCA are not executed at the XS2A interface, but directly between the PSU and the ASPSP. In this case, the PSU is redirected to a web interface of the ASPSP for authentication. Depending on the device used, the PSU may also be redirected to a special authentication app of the ASPSP.

Furthermore, the FABRICK PSD2 Gateway allows banks to use the redirect option in two different ways:

- Using a bank own web page to manage the SCA.

- Entrusting the authentication page to the Fabrick PSD2 Gateway that will authenticate the redirected PSU by connecting to the bank in machine2machine mode.

In the second option, once the PSU has been redirected to the Fabrick web interface the SCA of the PSU is executed step by step and directly between the ASPSP and the PSU. After completion of the SCA the PSU is redirected back to the TPP. The following figure shows the (much-simplified) top-level information flow for a payment initiation transaction with SCA based on the redirect approach:

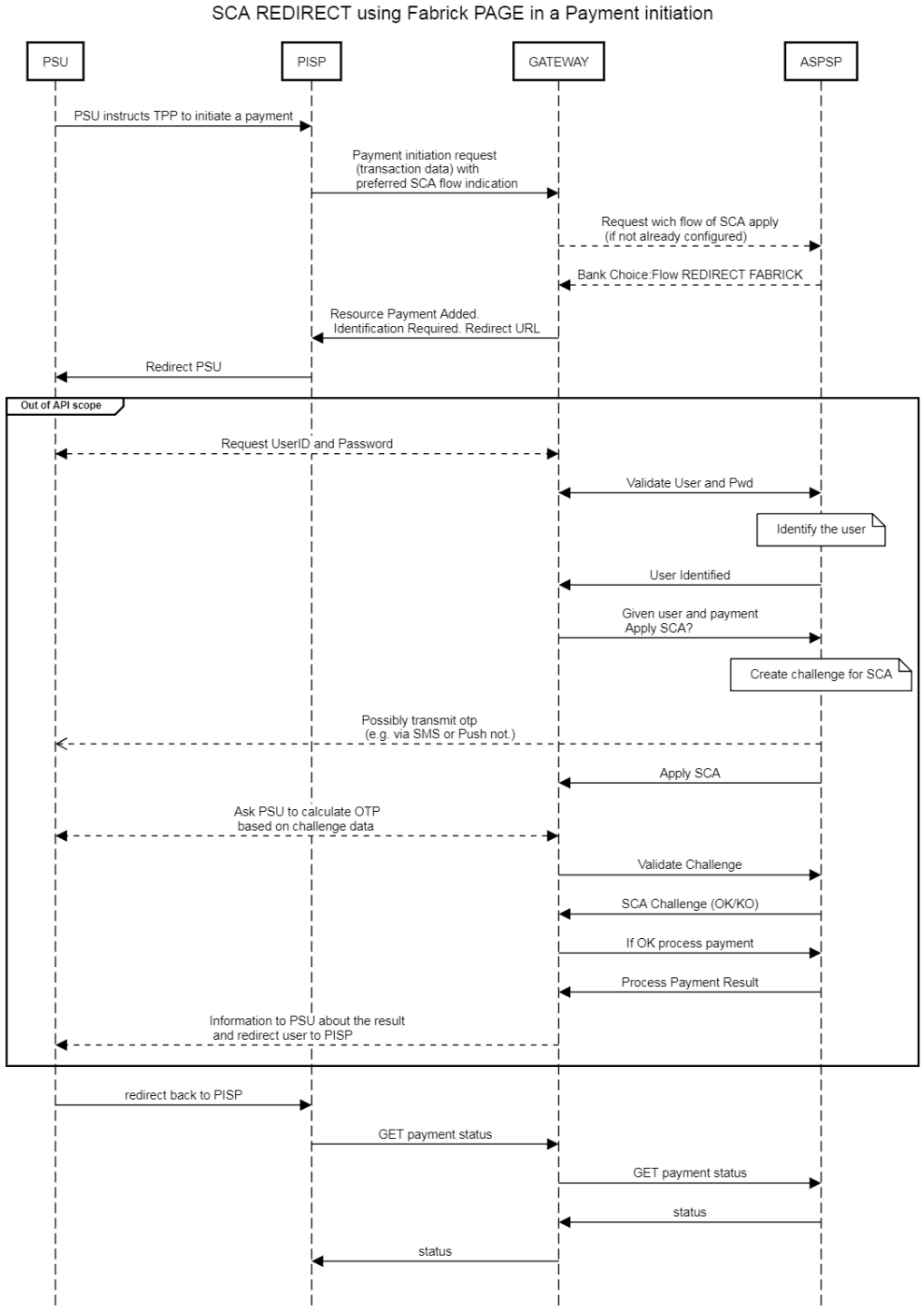


Figure 5

2.4.3 SCA using the OAuth2 approach

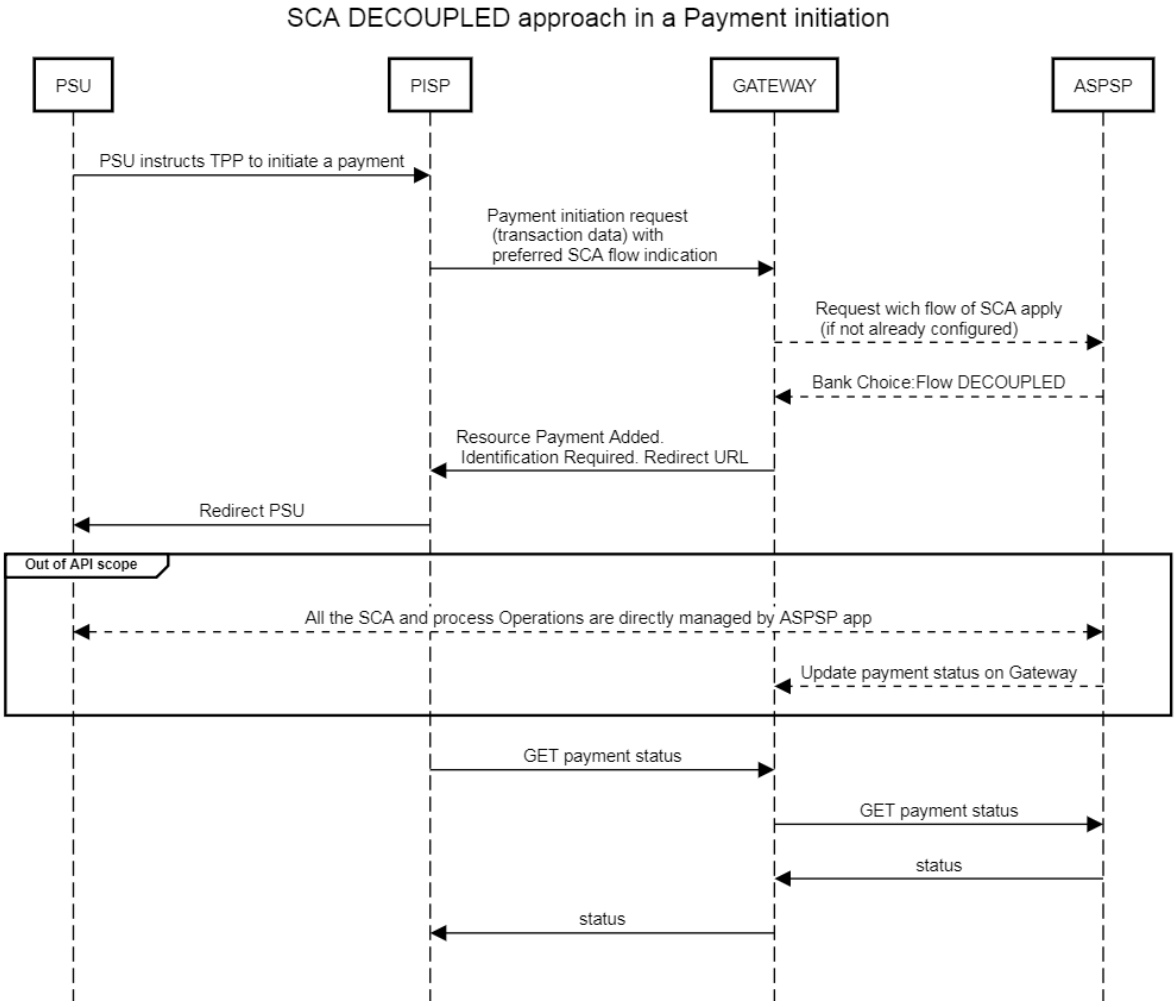
The transaction flow of the OAuth2 approach to SCA is similar to the redirect approach; the difference being that the redirection to the authentication server of the ASPSP is embedded into the OAuth2 protocol, where the scope attribute of the OAuth authorization request is linked to the created payment initiation or consent resource. The access to the interface to retrieve the actual account data is then performed using the access token delivered by the ASPSP's authentication server.

2.4.4 SCA Using Decoupled Approach

The transaction flow of the decoupled approach to SCA is similar to that of the redirect approach. The difference is that the ASPSP asks the PSU to authenticate e.g. by sending a push notification with payment transaction details to a dedicated mobile app or via any other application or device, which is independent of the online banking front-end. As opposed to the redirection flow, there is no impact on the PSU/TPP interface during the technical processing.

The following figure shows the (much-simplified) top-level information flow for a payment initiation transaction with SCA based on the decoupled approach:

Figure 5

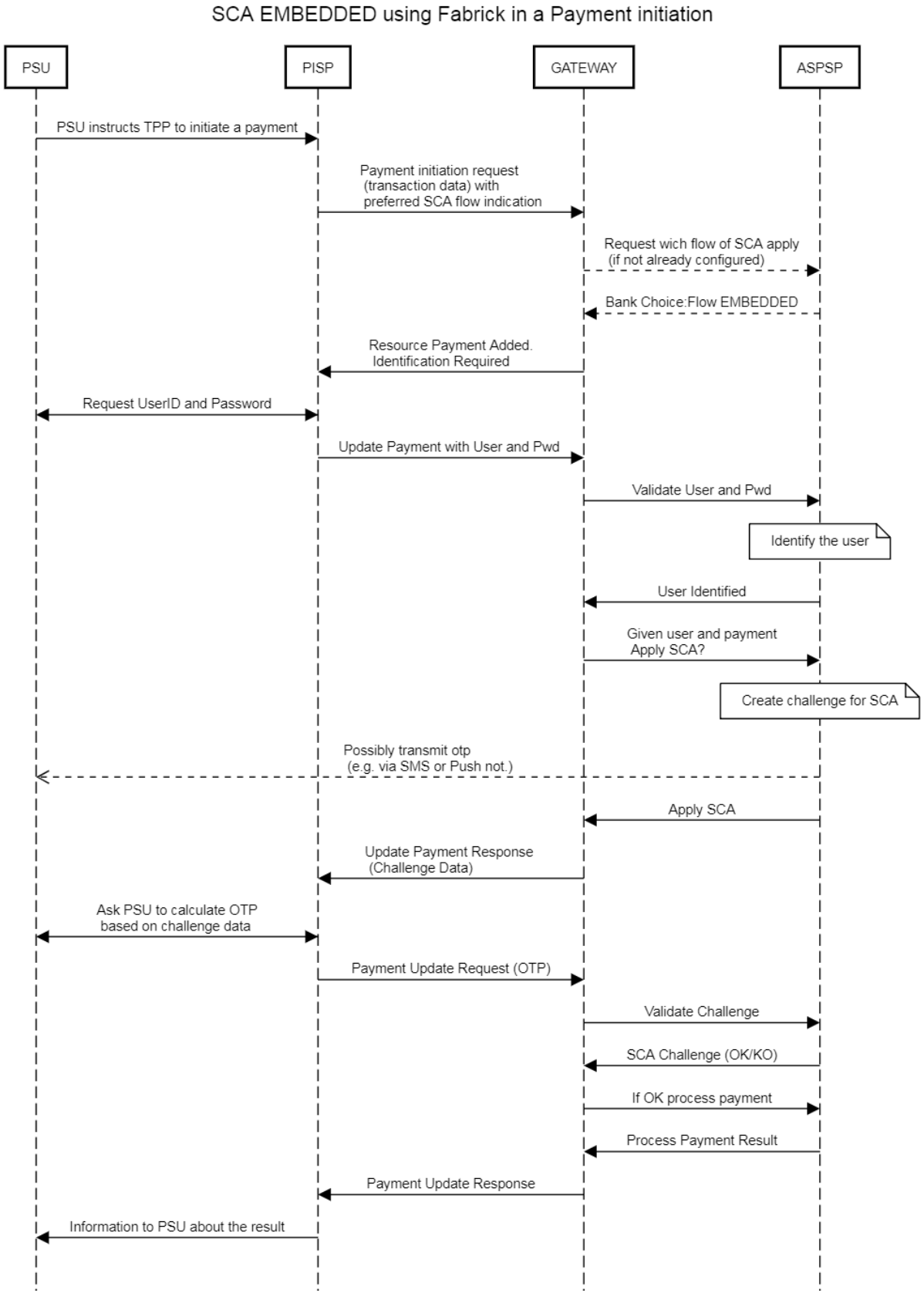


2.4.5 SCA Using the Embedded Approach

When applying the embedded approach the SCA of the PSU is executed entirely as part of the transaction at the XS2A interface. Clearly, in this procedure the user transmits the credentials via API through the TPP.

The following figure shows the (much-simplified) top-level information flow for a payment initiation transaction with SCA based on the embedded approach:

Figure 6


SCA EMBEDDED using Fabrick in a Payment initiation
PSU
PISP
GATEWAY
ASPSP
PSU instructs TPP to initiate a payment
Payment initiation request (transaction data) with preferred SCA flow indication
Request wich flow of SCA apply (if not already configured)
Bank Choice:Flow EMBEDDED
Resource Payment Added. Identification Required
Request UserID and Password
Update Payment with User and Pwd
Validate User and Pwd
Identify the user
User Identified
Given user and payment Apply SCA?
Create challenge for SCA
Possibly transmit otp (e.g. via SMS or Push not.)
Apply SCA
Update Payment Response (Challenge Data)
Ask PSU to calculate OTP based on challenge data
Payment Update Request (OTP)
Validate Challenge
SCA Challenge (OK/KO)
If OK process payment
Process Payment Result
Payment Update Response
Information to PSU about the result

2.4.6 General considerations about SCA

Although the choice of authentication procedure rests mainly with the ASPSP, among the choices presented by the market, the method that more than others would foster the growth of the ecosystem and the technological progress of the payments system, is the Embedded Approach.

Albeit this method may be considered more risky due to the passing of credentials through the information systems of the TPPs, the risk is nonetheless mitigated by the regulatory oversight to which the TPPs will be subject, as well as by the introduction of the second factor linked to the transaction.

It is in any case necessary for there to be an adequate trade-off between the perception of risk and the business developing consistent with the spirit of the PSD2.

Indeed, it would appear definitive that the “redirect” and “decoupled” approaches presented in the previous paragraphs, coming out of the "pure API" scenario, do indeed hinder the development of technological proposals such as voice banking, IOT etc. since not all banks will be able to develop appropriate authentication procedures.

Furthermore, it is worthy to note how the most open approach possible through the embedded procedure is in the best interests not only of the TPPs but also of the banks themselves, since the procedure will allow the evolution of open banking models capable of keeping the digital offering at the levels required by the market.

To allow banks an easy assessment of the characteristics in terms of risk and opportunities of each choice, here is a table that seeks to help and guide decision:

SCA Method	**Supported by Berlin Group Joint Initiative**	**Risks**	**Mitigations**	**Opportunities**	**Implementation Complexity for Bank**
Embedded	Yes	Compared to other procedures, there is an additional operational risk due to the forwarding of the PSU’s credentials through the application of the TPP.	The TPPs are supervised subjects.	It’s the only method that truly allows the development of open banking on technologies that go beyond those managed by the bank (e.g. voice, IoT, smartwatches, smart products etc.).	LOW. The bank has only to validate credentials in specific functions and create OTPs where necessary. No complexity UI<->API integration needed. The PSU UX is enhanced.
Redirect (with auth-page on Fabrick Platform)	Yes	The credentials are not forwarded through the TPP’s UI but are sent in a specific UI provided by Fabrick for each	Fabrick is a supervised subject capable of providing certified and	Fabrick, as a consolidator of accesses, is able to develop homogeneous	MEDIUM – LOW. The bank has only to validate credentials in

		bank. From the point of view of the bank, the possible risk concerns the forwarding of user credentials through the Fabrick platform.	guaranteed procedures and processes regarding non-tracking, as well as absolute security in the processing of data.	technological solutions for the “redirect” approach for users of multiple banks.	specific functions and create OTPs where necessary. Reduced complexity UI<->API integration needed.
Redirect with page on Bank Website	Yes	The integration of the solution between API and web interface is complex and can be an obstacle to UX.	N /A	No specific business opportunities compared to other solutions.	HIGH. An integration project required for each bank.
Decoupled	Yes	The integration of the solution between API and app is complex and can be an obstacle to UX.	N /A	No specific business opportunities compared to other solutions.	HIGH. An integration project required for each bank.
Single Sign ON	NO	The SCA is applied directly by the TPP (always in specific cases e.g. the consent renewal). In this case, the bank credentials are not used at all. The risk that an employee of the TPP can steal the credentials is none, however the TPPs have access to all payment accounts as the authentication of the client is entrusted to them.	No mitigation is possible	An excellent Customer Experience.	MEDIUM-LOW “Identification” is required, authentication is not required. The model can be easily scaled

2.5 PISP API Flows

2.5.1 PISP flows

The PIS within the XS2A interface can perform the following operations:

- Initialize payments by managing SCA authentication when necessary. These payments may be:
    - Single;
    - Bulk; or
    - Recurring.
- Read the status of the initialized payments at any time without the need for SCA and even in unattended mode.
- Make a reading of the details of the initialized payment at any time without the need for SCA and even in unattended mode.

To perform these operations, PISs are enabled to access the following endpoints on the Northbound interface.

Endpoints/Resources	Method	Condition	Description	Engaging Southbound Immediately
payments/{paymentproduct}	POST	Mandatory according to the BG. The product will be dependent upon the offer of each single bank.	Create a payment initiation resource addressable under {paymentId} with all data relevant for the corresponding payment product. This is the first step in the API to initiate the related payment.	No
payments/{paymentproduct}/{paymentId}	GET		Read the details of an initiated payment. SCA never required.	No
payments/{paymentproduct}/{paymentId} /status	GET	Mandatory according to the BG	Read the transaction status of the payment	Yes

bulk-payments/{paymentproduct}	POST	Optional according to the BG. The product will be dependent upon the offer of each single bank.	Create a bulk payment initiation resource addressable under {paymentId} with all data relevant for the corresponding payment product. This is the first step in the API to initiate the related bulk payment.	No
bulk-payments/{paymentproduct}/{paymentId}	GET	Optional according to the BG. The product will be dependent upon the offer of each single bank.	Read the details of an initiated bulk payment.	No
bulk-payments/{paymentproduct}/{paymentId}/status	GET	Optional according to the BG. The product will be dependent upon the offer of each single bank.	Read the transaction status of the bulk payment	Yes
periodicpayments/{paymentproduct}	POST	Optional according to the BG. The product will be dependent upon the offer of each single bank.	Create a standing order initiation resource for recurrent i.e. periodic payments addressable under {paymentId} with all data relevant for the corresponding payment product and the execution of the standing order. This is the first step in the API to initiate the related recurring/periodic payment.	No
periodicpayments/{paymentproduct}/{paymentId}	GET	Optional according to the BG. The product will be dependent upon the offer of each single bank.	Read the details of an initiated standing order for recurring/periodic payments.	No
periodicpayments/{paymentproduct}/{paymentId}/status	GET	Optional according to the BG. The product will be dependent upon the offer of each single bank.	Read the transaction status of the standing order for recurring/periodic payments.	Yes
{paymentservice}/{paymentproduct}/{paymentId}/authorisations	POST	Mandatory according to the BG	Create an authorization sub-resource and start the authorization process. Can also transmit authentication and authorization related data. This method is iterated *n* times for *n* SCA authorizations in a corporate context, each creating a single authorization sub-endpoint for the corresponding PSU authorizing the transaction.	No

			The ASPSP might render the use of this access method unnecessary in case of only one SCA process needed, since the related authorization resource might be automatically created by the ASPSP after the submission of the payment data with the first POST payments/{payment-product} call.	
{paymentservice}/{paymentproduct}/{paymentId}/authorisations	GET	Mandatory according to the BG	Read a list of all authorization sub resources IDs which have been created.	No
{paymentservice}/{paymentproduct}/{paymentId}/authorisations/{authorisationId}	PUT	Mandatory according to the BG	Update data on the authorization resource if needed. It may authorize a payment within the Embedded SCA Approach where needed. Independently from the SCA Approach, it supports e.g. the selection of the authentication method and a non-SCA PSU authentication.	Yes
{paymentservice}/{paymentproduct}/{paymentId}/authorisations/{authorisationId}	GET	Mandatory according to the BG	Read the SCA status of the authorization.	No
{paymentservice}/{paymentproduct}/{paymentId}	DELETE	Optional according to the BG. Implemented by the Fabrick Gateway	Cancels the addressed payment with resource identification paymentId if applicable to the payment-service, payment-product and received in product related timelines (e.g. before end of business day for scheduled payments of the last business day before the scheduled execution day). The response to this DELETE command will tell the TPP whether: • The access method was rejected; • The access method was successful; or • The access method is generally applicable, but further authorization processes required.	Yes, in some cases.

{paymentservice}/{paymentproduct}/{paymentId}/cancellation-authorisations	POST	Optional according to the BG. Implemented by the Fabrick Gateway	Starts the authorization of the cancellation of the addressed payment with resource identification paymentId if mandated by the ASPSP (i.e. the DELETE access method is not sufficient) and if applicable to the payment-service, and received in product-related timelines (e.g. before end of business day for scheduled payments of the last business day before the scheduled execution day).	No
{payment-service}{paymentproduct}/{paymentId}/cancellation-authorisations	GET	Optional according to the BG. Implemented by the Fabrick Gateway	Retrieve a list of all created cancellation authorization sub-resources. If the POST command on this endpoint is supported, then this GET method also needs to be supported.	No
{paymentservice}/{paymentproduct}/{paymentId}/cancellationauthorisations/{cancellationId}	PUT	Mandatory for Embedded SCA Approach, optional for other approaches.	Update data on the cancellation authorization resource if needed. It may authorize a cancellation of the payment within the Embedded SCA Approach where needed. Independently from the SCA Approach, it supports e.g. the selection of the authentication method and a non-SCA PSU authentication.	Yes
{paymentservice}/{paymentproduct}/{paymentId}/cancelationauthorisations/{cancellationId}	GET	Mandatory according to the BG.	Read the SCA status of the cancellation authorization.	No

2.5.2 Single Payment

Transactions according to this use case can be used to initiate a single payment in the form of a credit transfer from an account of the PSU to an account of the payee. Debit payments are not supported in the first version of the specification but may be specified in a later stage as an extended service.

While the transaction at the XS2A interface is initiated by the TPP, it must first be initiated by the PSU at the PSU–TPP interface. The PSU–TPP interface is not within the scope of this document.

The technical form of the payment scheme in the context of processing the ASPSP is a bank transfer.

The payload data of the Payment Initiation Request consist of all payment related data of the payment initiation. This data varies for different payment products. In [XS2A-ImplG] data, there are definitions for:

- SCT;
- SCT INST;
- SEPA Fast Payment (Target 2);
- Cross border credit transfer;
- Some Domestic Credit Transfer Services in non-euro currency.

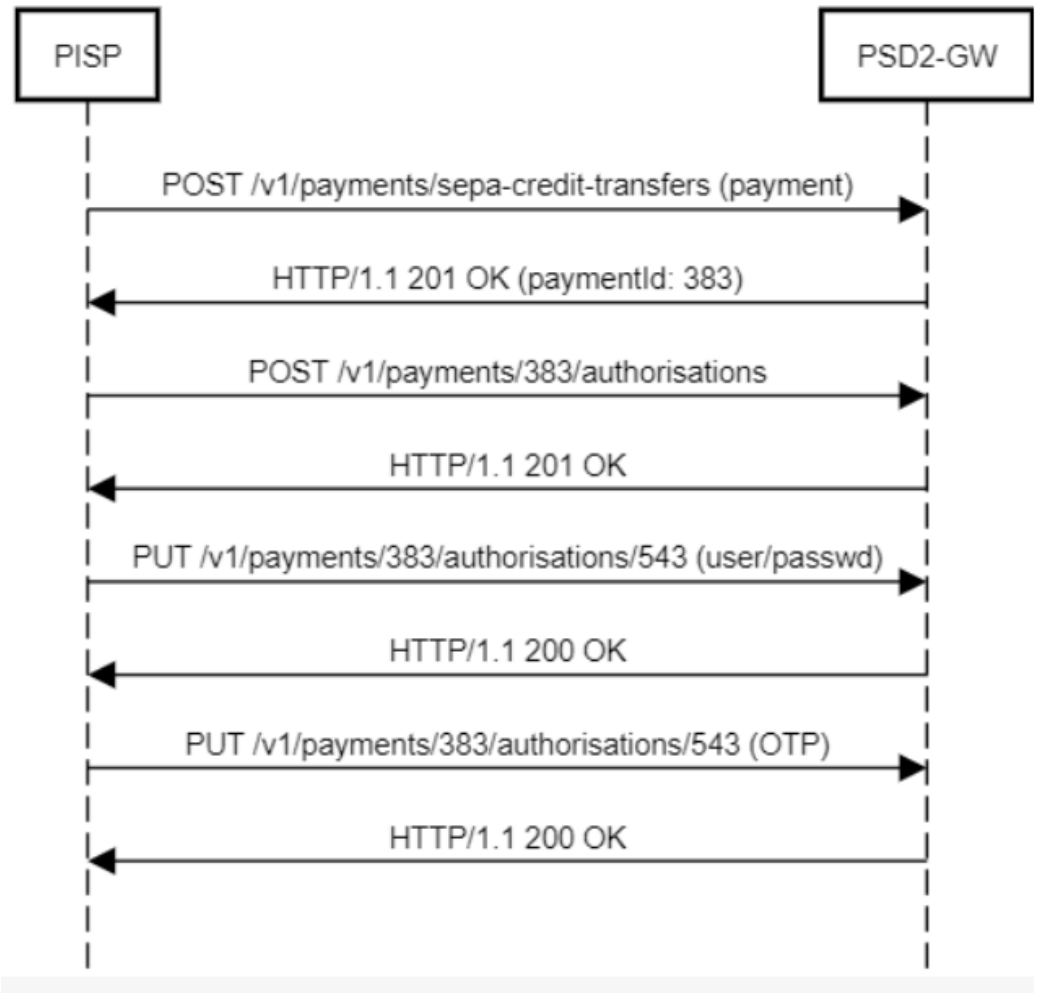
In order to create a single payment, according to the Berlin Group standard, the PISP must:

- Create a "Payment" resource;
- Associate it with an authorization resource, which will be used to identify and, if necessary, authenticate the PSU via SCA.

Refer to the Berlin Group standard documentation for defining entities in the JSON format.

In the following figure we present an example of the flow of a single payment authorized with embedded SCA with a "plain vanilla" challenge.

Figure 7



Detailed flow:

```
The PISP create a payment resource on gateway
PISP->PSD2-GW:POST /v1/payments/sepa-credit-transfers (payment)
{
    "instructedAmount":{
            "currency":"EUR",
            "amount":"123.50"

    },
    "debtorAccount":{
            "iban":"IT29G0326822300052763108812"

    },
    "creditorName":"Merchant123",
    "creditorAccount":{
            "iban":"IT29G0326822300052763108812"

    },
    "remittanceInformationUnstructured":"Ref Number Merchant"
Gateway response to PISP the correct creation of payment
PISP<-PSD2-GW:HTTP/1.1 201 OK (paymentId: 383)
```

```
{
    "transactionStatus": "RCVD",
    "paymentId": "383",
    "psuMessage": "",
    "_links": {
        "self": {
            "href": "/v1/payments/383"
        },
        "startAuthorisation": {
            "href": "/v1/payments/383/authorisations"
        }
    }
}
PISP create the authorisations resource on the payment
PISP->PSD2-GW:POST /v1/payments/383/authorisations
Gateway confirms to PISP the creation of authorisations resource
PISP<-PSD2-GW:HTTP/1.1 201 OK
{
    "scaStatus": "received",
    "_links": {
        "scaStatus": {
            "href": "/v1/payments/383/authorisations/543"
        },
        "updatePsuAuthentication": {
            "href": "/v1/payments/383/authorisations/543"
        }
    }
}
PISP pass user and password
PISP->PSD2-GW:PUT /v1/payments/383/authorisations/543 (user/passwd)
{
  "psuData": {
    "password": "passwd01"
  }
}
Gateway Responses to the PISP the challenge to perform for SCA purpose
PISP<-PSD2-GW:HTTP/1.1 200 OK
{
    "scaStatus": "scaMethodSelected",
    "chosenScaMethod": {
        "authenticationType": "SMS_OTP",
        "authenticationVersion": "v1",
        "authenticationMethodId": "SMS_OTP",
        "name": "SMS_OTP"
    },
    "_links": {
        "authoriseTransaction": {
            "href": "/v1/payments/383/authorisations/543"
        }
    }
}
PISP Requires the otp to the PSU and transmits it to gateway
PISP->PSD2-GW:PUT /v1/payments/383/authorisations/543 (OTP)
{
  "scaAuthenticationData": "12345678"
}
Gateway update result and status to PISP
PISP<-PSD2-GW:HTTP/1.1 200 OK
{
    "scaStatus": "finalised",
    "_links": {
        "scaStatus": {
            "href": "/v1/payments/383/authorisations/543"
        }
    }
}
```

The following table gives an overview of the generalizations of the Berlin Group's definition of JSON structures and of standards on SEPA payment products for single payments.

Data Element	Ty[5]pe	SCT EU Core	SCT INST EU Core	Target2 Paym. Core	Cross Border CT Core
endToEnd Identification	Max35Text	optional	optional	optional	n.a.
debtorAccount (incl. type)	Account Reference	mandatory	mandatory	mandatory	mandatory
debtorId	Max35Text	n.a.	n.a.	n.a.	n.a.
ultimateDebtor	Max70Text	n.a.	n.a.	n.a.	n.a.
instructedAmount (inc. Curr.)	Amount	mandatory	mandatory	mandatory	mandatory
transactionCurrency[6]	Currency Code	n.a.	n.a.	n.a.	n.a.
creditorAccount	Account Reference	mandatory	mandatory	mandatory	mandatory
creditorAgent	BICFI	optional	optional	optional	optional
creditorAgentName	Max70Text	n.a.	n.a.	n.a.	n.a.
creditorName	Max70Text	mandatory	mandatory	mandatory	mandatory
creditorId	Max35Text	n.a.	n.a.	n.a.	n.a.
creditorAddress	Address	optional	optional	optional	optional
ultimateCreditor	Max70Text	n.a.	n.a.	n.a.	n.a.
purposeCode	Purpose Code	n.a.	n.a.	n.a.	n.a.
chargeBearer	Charge Bearer	n.a.	n.a.	optional	optional
remittance Information Unstructured	Max140Text	optional	optional	optional	optional
remittance Information Unstructured Array	Array of Max140Text	n.a.	n.a.	n.a.	n.a.
remittance Information Structured	Remittance	n.a.	n.a.	n.a.	n.a.
requestedExecution Date	ISODate	n.a.	n.a.	n.a.	n.a.
requestedExecution Time	ISODateTime	n.a.	n.a.	n.a.	n.a.

2.5.3 Recurring Payments

The initiation of a recurring payment is realized in the XS2A interface by the initiation of a corresponding standing order, as it is supported today by the ASPSP in the client interface. The TPP can initiate a single payment along with administrative information about the frequency and duration of the recurring payments. The duration can be limitless.

This specific payment initiation needs to be authorized by the PSU with a SCA.

[5] Refer to Berlin Group Implementation Guidelines
[6] This is a data element to indicate a diverging interbank currency.

The flow diagram is identical to the single payment scheme, the only difference being some additional fields in the JSON structure of the payment entity to be processed (ref. 2.5.1):

Tag	Type	Usage	Description
startDate	ISODate	Mandatory	The first applicable day of execution starting from this date is the first payment.
executionRule	String	Optional	"Following" or "preceding" values are supported. This data attribute defines the behavior when recurring payment dates fall on a weekend or bank holiday. The payment is then executed either the "preceding" or the "following" working day. ASPSP might reject the request due to the communicated value, if rules in Online-Banking do not support this execution rule.
endDate	ISODate	Optional	The last applicable day of execution. If not given, it is an infinite standing order.
frequency	Frequency Code	Mandatory	The frequency of the recurring payment resulting from this standing order.
dayOfExecution	Max2Text	Conditional	"31" is “last”. The format follows the regular expression \d{1,2}. Example: The first day is addressed by "1". The date refers to the time zone of the ASPSP.

Future date payments and periodic payments are both payment types that are not directly executed after initiation. For both types of payments, ASPSPs might have reduced or no checks on customer profile or on funds availability since the actual payments are performed at a later date. The end status during the payment initiation process is then either "ACTC" or "ACCP" depending on the ASPSPs procedures on its online channels.

Whenever supported by each single ASPSP, the Fabrick gateway allows for the fundsAvailable data element to be included, in case a funds availability check has been performed during payment initiation.

2.5.4 Future Date Payments

The initiation of a “future date” payment is realized in the XS2A interface by the initiation of a corresponding single payment, as it is supported today by ASPSP in the client interface.

The only difference to the initiation of a regular payment is that the date is a future one, and is set through the appropriate “requestedExecution Date” field in the request.

2.5.5 Bulk Payments

Multiple payments, where a PSU first collects several payments and then performs a SCA to authorize the collection (bulk) of these payments is always realized as a bulk payment initiation in the XS2A interface. The collection of several payments may be performed in the interface between PSU and TPP.

2.6 AISP API Flows

2.6.1 Consent Management

According to the Berlin Group NextGEN PSD2 standard, a TPP may execute transactions as per the use case shown in the following figure and thus gain the right to execute further transactions on the account information service. Subject to PSU consent, the TPP can obtain the following rights for transactions (of the account information service):

- Get the list of reachable accounts of the PSU once;
- Get the balance for a list of accounts once, or multiple times;
- Get payment transaction information for a list of accounts once, or multiple times.

If the TPP is granted the right to access balance or payment transaction information for certain accounts, this will include automatically the right to retrieve detailed information about the related payment accounts. If the TPP is granted the right to execute a transaction multiple times, the validity period of the right is defined in days or the maximal period offered by the ASPSP. It is furthermore possible to define the permitted frequency of corresponding transactions on a per diem basis. The PSD2 and EBA-RTS requirements shall be observed during the entire validity period granted and for the allowed frequency of transactions.

Furthermore, the BG standard framework establishes that when the consent is established to get account information for a list of accounts, the actual technical transaction to retrieve the account data might be applicable only for a specific account, c.p. 2.6.2 “Accounts”.

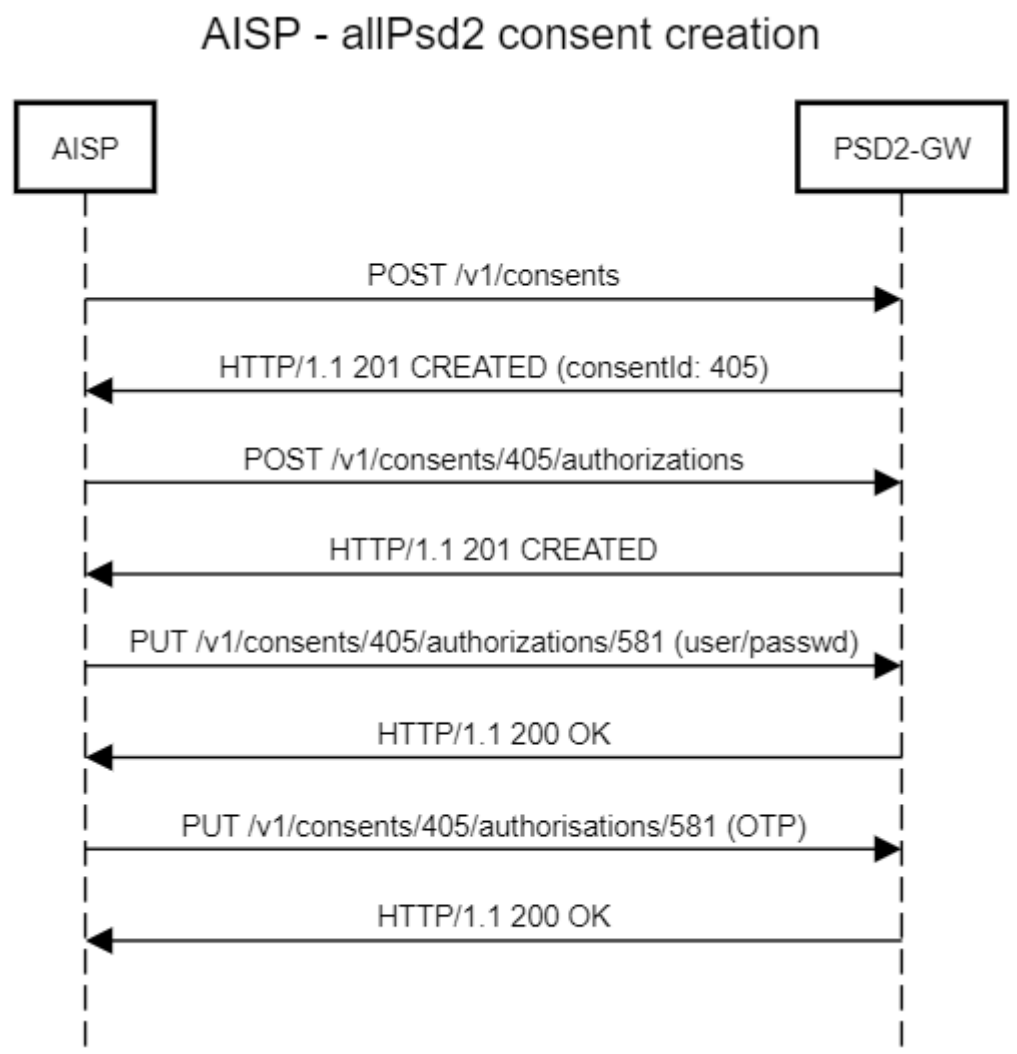
While the transaction at the XS2A interface is initiated by the TPP, it must first be initiated by the PSU at the PSU–TPP interface. However, the TPP has to inform the PSU clearly about the rights for which the PSU has to confirm its consent.

The ASPSP will reject the transaction if the TPP cannot be identified correctly at the XS2A interface and/or if it does not have an AISP role.

The Fabrick PSD2 Gateway allows banks in the Southbound interface and connection with themselves to not have to manage the consent data, as these will be engaged in the consent management flow only with regard to the identification and strong authentication of the customer.

The following figure shows only the information flow for the consent transmission using SCA with an embedded flow type approach:

Figure 8



The AISP transmits to the gateway the consent received by the PSU to access to an account

```
AISP->PSD2-GW:POST /v1/consents
{
    "access": {
        "balances": [{
            "iban": "IT29G0326822300052763108812"
        }],
        "transactions": [{
            "iban": "IT29G0326822300052763108812"
        }]
    },
    "recurringIndicator": true,
    "validUntil": "2019-11-01",
    "frequencyPerDay": "4"
}
```

The gateway responses to AISP about the creation of the consent resource and the auth resources

```
AISP<-PSD2-GW:HTTP/1.1 201 CREATED (consentId: 405)
{
    "consentStatus": "received",
    "consentId": "405",
    " links": {
        "self": {
            "href": "/v1/consents/405"
        },
        "startAuthorisation": {
            "href": "/v1/consents/405/authorisations"
```

```
        }
    }
}
AISP Create Authorizations resources
AISP->PSD2-GW:POST /v1/consents/405/authorizations
Gateway confirms to PISP the creation of authorisations resource
AISP<-PSD2-GW:HTTP/1.1 201 CREATED
{
    "scaStatus": "received",
    " links": {
        "scaStatus": {
            "href": "/v1/consents/405/authorisations/581"
        },
        "updatePsuAuthentication": {
            "href": "/v1/consents/405/authorisations/581"
        }
    }
}
AISP pass user and password
AISP->PSD2-GW:PUT /v1/consents/405/authorizations/581 (user/passwd)
{
  "psuData":{
    "password":"passwd01"
  }
}
Gateway describes the SCA challenge to the AIS in order to involve PSU in the SCA process
AISP<-PSD2-GW:HTTP/1.1 200 OK
{
    "scaStatus": "scaMethodSelected",
    "chosenScaMethod": {
        "authenticationType": "SMS OTP",
        "authenticationVersion": "v1",
        "authenticationMethodId": "SMS OTP",
        "name": "SMS OTP"
    },
    "_links": {
        "authoriseTransaction": {
            "href": "/v1/consents/405/authorisations/581"
        }
    }
}
AIS transmits the otp requested to the PSU
AISP->PSD2-GW:PUT /v1/payments/405/authorisations/581 (OTP)
{
  "scaAuthenticationData": "12345678"
}
Gateway responses that the consent is finalized and can be used
AISP<-PSD2-GW:HTTP/1.1 200 OK
{
    "scaStatus": "finalised",
    "psuMessage": "",
    "_links": {
        "scaStatus": {
            "href": "/v1/consents/405/authorisations/581"
        }
```

In order to perform operations related to consent management, AISs are enabled to access the following endpoints on the Northbound interface:

Endpoints/Resources	Method	Condition	Description	Engaging Southbound Immediately

consents	POST	Mandatory according to the BG. The product will be dependent on the offer of each single bank.	Create a consent resource, defining access rights to dedicated accounts of a given PSU-ID. These accounts are addressed explicitly in the method as parameters of a core function.	No (only for SCA and accounts validation)
consents	POST	Optional according to the BG.	As an option, an ASPSP may accept a specific access right from the access of all PSD2 related services for all available accounts. Another option an ASPSP may choose is to accept a command, whereby only access rights are inserted without mentioning the address account. The relation to accounts is then handled later between PSU and ASPSP. As a last option, an ASPSP may choose to accept a command with access rights: • to see the list of available payment accounts; or • to see the list of available payment accounts with balances	No
consents/{consentId}	GET , DELETE	Mandatory according to the BG	Reads the exact definition of the given consent resource {consentId} including the validity status / Terminates the addressed consent	No
consents/{consentId}/authorisations	POST	Mandatory according to the BG	Create an authorization sub-resource and start the authorization process. It may also transmit authentication and authorization related data. The ASPSP might render this access method unnecessary, since the related authorization resource will be created automatically by the ASPSP after the submission of the consent data with the first POST consents call.	Yes
consents/{consentId}/authorisations/{authorisationId}	PUT	Mandatory for Embedded SCA Approach, Conditional for	Update data on the authorization resource if needed. It may authorize a consent within the Embedded SCA Approach where needed. Independently from the SCA Approach, it can support the	Yes

		others	selection of the authentication method and a non-SCA PSU authentication.	
consents/{consentId}/authorisations/{authorisationId}	GET	Mandatory according to the BG	Read the SCA status of the authorization.	No

Detailed information about consent object, consent status and these endpoint parameters are available on NEXTGenPSD2_ImplementationGuidelines_V1.3_20181019.pdf. See chapter 2.1 for further reference.

2.6.2 Get list of reachable accounts

In this case, transactions can be used by a TPP to receive a list of reachable accounts of a PSU managed by the ASPSP. The term "reachable accounts" shall refer to online accessible payment accounts (according to articles 65, 66, 67 of the [PSD2]). ASPSPs support a large variety of account models. The ASPSP shall decide (in compliance with the PSD2) what accounts have to be treated as online accessible payment accounts and must therefore be reachable at the XS2A interface.

Because of this transaction type, the TPP will receive a list of account numbers. No further information about the accounts is returned. If the TPP has been granted the right to receive further information from the context of a previous transaction based on the "Establish account information consent" case, the TPP can use the obtained account numbers to receive further information about the accounts in additional transactions of the account information service.

The transaction at the XS2A interface is initiated by the TPP. It does not have to be initiated beforehand by the PSU at the PSU–TPP interface. However, the PSU must still have granted its consent during a preceding transaction.

The ASPSP will reject the transaction if the TPP cannot be identified correctly at the XS2A interface and/or if it does not have an AISP role. The ASPSP will also reject the transaction if the TPP does not carry the necessary rights for this transaction type from preceding transactions, as in the "Establish account information consent" case.

This approach, as outlined in the NextGenPSD2 standard, requires the creation of a consensus (and therefore the identification of the PSU) in order for such call to be made.

WARNING: this approach (as defined by the RTSs in which the function of access to the list of accounts is not included among those exempt from SCA in case of consent on designated

accounts) implies, necessarily, that in the event that the AIS is not already aware of the list of customer accounts and would like to get this list before the expression of customer consent it will:

Request a “non-recurring” consent to access the list of accounts and transmit it to the gateway

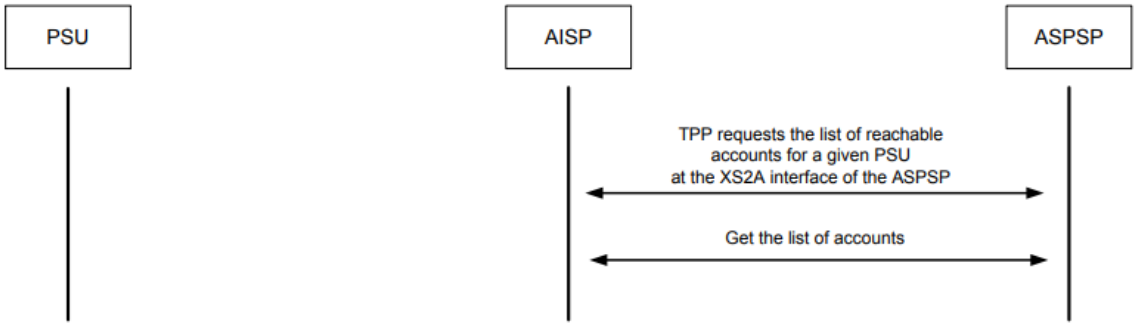
Obtain the list of accounts on which to model the request for access to data.

Request consent to access account information (balance and transactions) and transmit it to the gateway.

Another approach that can be used for AISs in order to obtain a good user experience is to request consent only once and in advance as "allPSD2" or referring to all the accounts available from the ASPSP. This will be feasible to the extent that a banking institution will allow for this method set out by the Berlin Group standard.

The following figure shows the correspondent information flow:

Figure 9



2.6.3 Get account details of a list of accessible accounts

In this case, transactions can be used to retrieve detailed information about all accounts of a PSU accessible by a TPP. Accessible accounts are defined as those accounts of a PSU for which a consent has been granted to the TPP to access these accounts for balances or transactions.

Details of these accounts can be:

- Hyperlinks to account information resources associated with these accounts;
- Alias identifiers under which these accounts are addressable;
- Types and names of the accounts; and
- The balance of the accounts if required as an additional data element and if the right has been granted to the TPP.

The transaction at the XS2A interface is initiated by the TPP. It does not have to be initiated by the PSU at the PSU–TPP interface beforehand. However, the PSU must still have granted its consent during a previous transaction. The ASPSP will reject the transaction if the TPP cannot be identified correctly at the XS2A interface and/or if it does not have an AISP role.

The ASPSP will also reject the transaction if the TPP does not have the necessary rights for this transaction type from a preceding transaction in the "Establish account information consent" case.

2.6.4 Account Transactions

In this case, the TPP can use transactions for receiving information about payment transactions of a specific account. Incidentally, the TPP will receive information about all payment transactions executed during the time-period indicated in the request

Note: Other balances will be provided in the "Get balance information for a given account" case.

Furthermore, the ASPSP may offer a delta report service. In this case, the ASPSP is delivering only the information about payment transactions since the last access of the TPP to a specified account information service, or it is delivering the information about payment transactions starting from next transaction of a payment transaction with a given transaction identification.

The transaction at the XS2A interface is initiated by the TPP. It does not have to be initiated by the PSU at the PSU–TPP interface beforehand. However, the PSU must still have granted its consent during a preceding transaction.

The ASPSP will reject the transaction if the TPP cannot be identified correctly at the XS2A interface and/or if it does not have an AISP role.

The ASPSP will also reject the transaction if the TPP has not been granted the necessary rights for this transaction type during a previous transaction according to the "Establish account information consent" case.

Call:

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

Reads account data from a given account addressed by "account-id".

Remarks: This account-id can be a tokenized identification due to data protection reasons, since the path information might be logged on intermediary servers within the ASPSP range of activities. This account-id can then be retrieved by the "GET Account List" call, cp. 2.6.2

Detailed information about query-parameters, body of request, body of response and these endpoint parameters are available at NEXTGenPSD2_ImplementationGuidelines_V1.3_20181019.pdf.

See chapter 2.1 for references.

2.6.5 Balance

In this case, the TPP can use transactions for receiving the balances for a given account. As a result, the TPP will receive detailed balances for the account identified in the request of this transaction. Also, booked order balances can be authorized, as well as other intermediary balances, depending on the implementation of the ASPSP. No further information about transactions of the accounts will be returned.

If the PSU has granted access to balances of several accounts, then a corresponding transaction has to be submitted for each account separately.

The transaction at the XS2A interface is initiated by the TPP. The transaction does not have to be initiated by the PSU at the PSU–TPP interface beforehand. However, the PSU must still have granted its consent during a previous transaction.

The ASPSP will reject the transaction if the TPP cannot be identified correctly at the XS2A interface and/or if it does not have an AISP role.

The ASPSP will also reject the transaction if the TPP has not been granted the necessary rights for this transaction type during a previous transaction in the “Establish account information consent” case.

2.7 PIISP Api Flow

2.7.1 Funds Availability

The TPP can use transactions according to this use case to receive confirmation about the availability of the requested funds on a specific account. As a result the TPP will only receive the answer YES or NO. No further information about the account will be returned. While the transaction at the XS2A interface is initiated by the TPP, it must first be initiated by the PSU by means of an e.g. card based payment transaction at a PSU – TPP interface, for example at a checkout point. The PSU – TPP interface is not within the scope of this document.

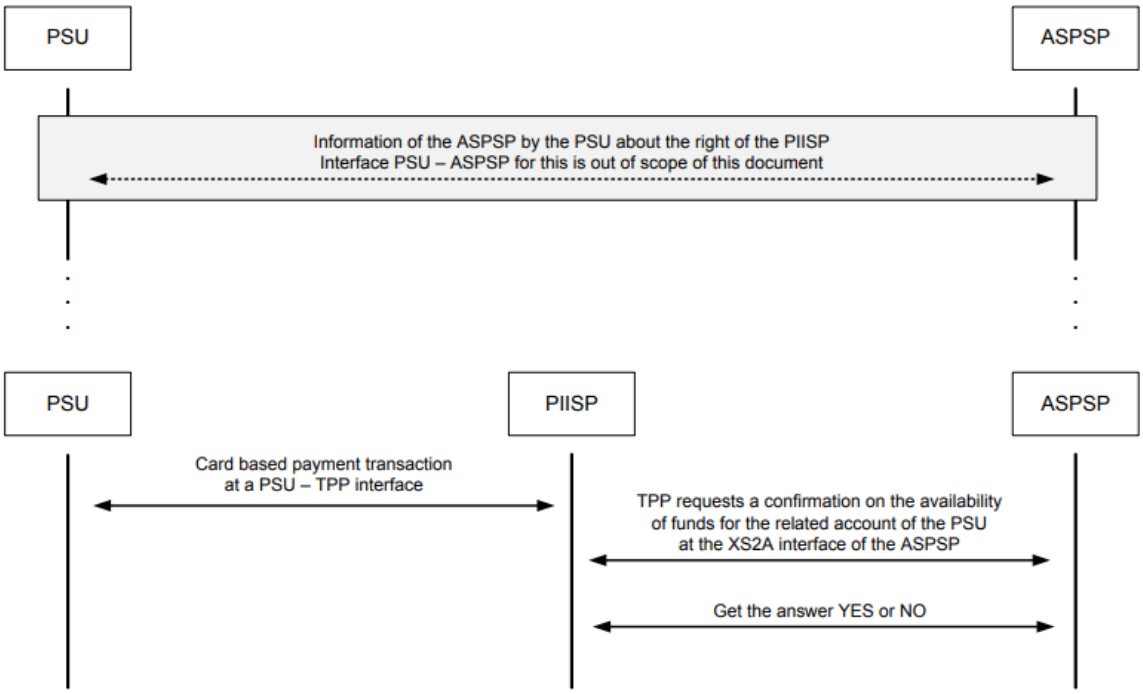
According to article 65 of [PSD2], the PSU has to inform the ASPSP about its consent to a specific request of the TPP prior to the transaction. The document at hand does not cover the interface between the PSU and the ASPSP required for the exchange of information.

The ASPSP will reject the transaction if the TPP cannot be identified correctly at the XS2A interface and/or if it does not have the role PIISP.

The ASPSP will also reject the transaction if the PSU has not previously informed the ASPSP about its consent to the corresponding transaction of the TPP.

The following figure shows only the very top-level information flow:

Figure 10



2.8 Fabrick delta fields comparing to BG standard

As explained in the preceding paragraphs, the Fabrick PSD2 Gateway implements the NextGenPSD2 protocol of the Berlin Group v1.3 in its interface with TPPs.

This protocol is constantly evolving, and the Fabrick PSD2 Gateway is committed to implementing the evolutions envisaged by the initiative.

It should also be noted that the implementation of some parts of the protocol may differ according to the choices made by individual ASPSPs and consequently the interface may not be homogeneous from the point of view of the enabled functions.

Furthermore, Fabrick has decided to enrich the functionality provided by the protocol to provide some additional features or complete the overall offer with some headers and ad hoc fields.

These enrichments and differences compared to the Berlin Group standard will be documented in detail for the TPPs on the mini-site dedicated to the PSD2 Gateway within the Fabrick website.

Appropriate documentation will also be made available, as well as the swagger of the entire dedicated interface.

As of today's date, following is a selected list of custom fields on the Fabrick PSD2 Gateway compared to th BG standard:

Fabrick introduced some special headers will be used in the response in any case where is requested to the TPP to get user/pwd or otp challenge in embedded approach. This header will be useful in order to enable the tpp to render a login form "familiar to the psu".

Header	Description
X-Psu-Username-Type	Username type (visualization supported values: Text | Password | Date)
X-Psu-Username-Name	Displayed username field name
X-Psu-Username-Hint	Displayed username field hint
X-Psu-Username-Description	Displayed username field description
X-Psu-Challenge-Type	Challenge type (visualization supported values: Text | Password | Date)
X-Psu-Challenge-Name	Displayed challenge field name
X-Psu-Challenge-Hint	Displayed challenge field hint
X-Psu-Challenge-Description	Displayed challenge field description

1. Some BG data structures has "optional" fields:

Structure	Data Element	Mandatory	Optional	Fabrick Implemented
Payment Data	endToEndIdentification		yes	yes
Payment Data	debtorAccount	yes		yes
Payment Data	debtorId			No[7]
Payment Data	ultimateDebtor			No
Payment Data	instructedAmount	yes		yes
Payment Data	transactionCurrency		yes	yes
Payment Data	creditorAccount	yes		yes
Payment Data	creditorAgent		yes	yes
Payment Data	creditorAgentName			No
Payment Data	creditorName	yes		yes
Payment Data	creditorId			No
Payment Data	creditorAddress		yes	yes
Payment Data	ultimateCreditor			No
Payment Data	purposeCode		yes	yes
Payment Data	chargeBearer		yes	yes
Payment Data	remittance Information Unstructured		yes	yes
Payment Data	remittance Information Unstructured Array			No
Payment Data	remittance Information Structured			No
Payment Data	requestedExecution Date	yes		yes
Payment Data	requestedExecution Time		yes	No

[7] Useful only for the SDD schema. Not in the PSD2 gateway functional perimeter, but offered by Fabrick through an API market.

2. Fabrick PSD2 gateway has added two "unexpected" error codes to those defined by the NextGENPSD2 standard to distinguish the origin of any unexpected errors between those generated by problems to be investigated in the gateway itself or when the call goes wrong within the software of the single ASPSP.
   These error codes are:
   - 500 INTERNAL_SERVER_ERROR (Generic unexpected error)
   - 502 ASPSP_SERVICE_FAILED (Unexpected error in the context of ASPSP)

2.9 Public Portal

2.9.1 Online Documentation

Fabrick Platform is the public portal through which consumers can access to the API documentation. It is a technological platform engaging both producers and consumers of banking and fintech services.

The plug-and-play technological infrastructure is "open-API"-based and it enables consumers to get access to the continuous evolving platform documentation. On the other side, producers are able to expose their APIs within the public portal. The Fabrick Platform website is reachable at .

For the purpose of the PSD2 Gateway a dedicated mini-site has been developed at .

The api documentation specific for the northbound interface are available at

fabrick

3 Environments

3.1 Development environment

Fabrick Platform offers to TPPs two distinct test environments called TEST and SANDBOX.

In the test environment, the client's credentials and the data returned are completely simulated.

In the sandbox environment, on the other hand, access to information is only possible using the customer's "real" credentials and performing an SCA with the same level of security required for the live environment. The sandbox environment contains actual data updated to the previous week while payments made remain in the test environment.

This approach allows the TPP to develop and test the connection in the TEST environment and to perform an "advanced" test including the complete authentication flow in the sandbox environment before going to live without having access to any confidential information.

In addition, in order to operate in the TEST environment, the TPPs are not required to provide an eIDAS certificate, but a valid X509 certificate is enough. The gateway provides client mutual authentication and it lets developers integrate all PSD2 cases. In this specific environment, the ASPSP is a bank simulation software that returns static information. It is also not necessary to provide any specific bank account.

In order for the environment to work effectively, a mock ASPSP has been created with three different test users. These users allow for the execution of dynamic tests for all PSD2 cases, such as:

- Consent creation;
- Access to account transactions (attended/unattended);
- Access to account balances (attended/unattended);
- Access to card transactions (attended/unattended);
- Access to card balances (attended/unattended);
- Payment initialization;
- Payment status.

Developers can also check errors, for instance:

- Error due to wrong username or password;
- Error due to wrong OTP entry;
- Error due to wrong Debtor IBAN;

- Error due to wrong Masked PAN;
- Be able to use different PSUs to receive different payments statuses (by using the get status)
- Insert cases of payment with exemption, or without exemption.

To perform the tests, the following data can be used.

Mock data:

UserName	Passwd	OTP	SCA	Debtor IBAN	Masked PAN	Payment ref	Payment status
user01	passwd01	12345678	No exemption	IT11V3615958524033917640757	455662******7413	11111111	ACCP
user02	passwd02	23456789	Full exemtion	IT78H8663926660736814299754	480580******2662	22222222	RCVD
user03	passwd03	34567890	No exemption	IT29E4223395989434205972721	455607******6337	null	RJCT

SCA

Operation	Revoke	SCA	authID
Payment service	FALSE	SMS	authidToken01
Payment service	TRUE	CHIP	authidToken02
Information service	N/A	[SMS, CHIP]	authidToken03

Test account URL: https://test-psd2gateway.fabrick.com/api

The test environment is available since Jan 3, 2019.

The sandbox one is available since March 14, 2019.

3.2 Live environment

The LIVE environment is the real life account accessible to licensed TPPs (AISPs, PISPs). This environment allows TPPs to perform API calls within a dedicated account with full enforcement of security protocols.

To get access to the LIVE environment, the TPP needs to provide an eIDAS certificate throughout a specific API call.

The LIVE environment will be available from June 1st, 2019.

3.3 Table of domains

The following table recaps domains and features for each environment:

Environment		Access	TPP Identification	ASPSP	Domain
DEVELOPMENT	TEST	TPP Authorized and TPP waiting for authorization	Any valid client certificate	In TEST environment the ASPSP is a bank simulation software that returns mocked-up informations. To access this environment, a X509 certificate will suffice.	test-psd2gateway.fabrick.com
	SANDBOX	TPP Authorized and TPP waiting for authorization	Any valid client certificate	In SANDBOX environment there's a set of endpoints for each single ASPSP on the https connection. A domain for each single ASPSP will be provided. This environment will be connected to each ASPSP of development environment.	sandbox-psdgw-{tennantname}.fabrick.com Example: https://sandbox-psdgw-bancapatrimoni.fabrick.com
LIVE		TPP Authorized	Any valid client certificate (on specific request can be exposed eidas certificate on a custom domain as the first level of bank-domain)	A certificate for each single ASPSP on the https connection. A domain for each single ASPSP will be provided. To operate in this environment a qualified certificate needs to be presented.	psdgw-{tennantname}.fabrick.com Example: https://psdgw-bancapatrimoni.fabrick.com

As of today's date the url table is as follows[8]:

ASPSP	Url live	Url sandbox
Banca Sella	psdgw-sella.fabrick.com	sandbox-psdgw-sella.fabrick.com
Banca Patrimoni	psdgw-bancapatrimoni.fabrick.com	sandbox-psdgw-bancapatrimoni.fabrick.com
Illimity	psdgw-illimity.fabrick.com	sandbox-psdgw-illimity.fabrick.com
Cartalis	psdgw-cartalis.fabrick.com	sandbox-psdgw-cartalis.fabrick.com
Sella Personal Credit	psdgw-sellapersonalcredit.fabrick.com	sandbox-psdgw-sellapersonalcredit.fabrick.com
Borsa del Credito	psdgw-borsadelcredito.fabrick.com	sandbox-psdgw-borsadelcredito.fabrick.com
Hype	psdgw-hype.fabrick.com	sandbox-psdgw-hype.fabrick.com
TIM (Hype)	psdgw-tim.fabrick.com	sandbox-psdgw-tim.fabrick.com
Smartika	psdgw-smartika.fabrick.com	sandbox-psdgw-smartika.fabrick.com
Soldo	psdgw-soldo.fabrick.com	sandbox-psdgw-soldo.fabrick.com
Paytipper	psdgw-paytipper.fabrick.com	sandbox-psdgw-paytipper.fabrick.com

4 Support and Help Desk

4.1 Tools used

A ticketing support system based on Jira Service Desk - Atlassian is employed for customer assistance, easing communications from customers on single tickets and within an organized workflow.
The service is 24/7.

4.2 Credentials

A unique issue from a specific customer is tracked with a case number.
The consumer can view open issues through the dashboard.
All cases are logged and the assigned Support Engineer is the point of contact until a resolution is found.

[8] Adherent members of the platform may vary from time to time. The table will be regularly updated on the Fabrick Platform web site

Our Support engineers utilize their personal product knowledge and experience to solve the problem and to avoid a repeat of the problem in the future.
The ticketing system is also connected to the internal back office in order to verify the status of the consumer.

4.3 Help desk and troubleshooting processes

There are different issue categories and all support requests are assigned a priority level between 1 and 4, based on their impact.

Categories/Project	Type of Consumer	Type of Issue
PSD2 Passive Gateway	Third Party Providers (PISPs/AISPs/PIISPs)	- Technical Support Ticket - Onboarding Support Ticket
PSD2 Active Gateway (+ product name)	AISPs	- Technical Support Ticket - Onboarding Support Ticket
Fabrick Platform APIs		- Technical Support Ticket - Onboarding Support Ticket
Fabrick Contact Us		- Product Information

Priority	Issue Cases	Resolution/Workaround
Priority 1	- Critical features not available	4 hours response time
Priority 2	- Performance greatly compromised - Unavailable functionality on working system (albeit with limitations)	2 days response time
Priority 3	- Error message with indication of an alternative solution - Minimum impact on performance - Wrong operation with a negligible impact - Doubts about the operation or configuration of the product being implemented	8 days response time
Priority 4	- Clarifications on product documentation or version notes - Request for product improvement	15 days response time

Workflow:

Create

Phase 1: Consumer Assistance Request + Ticket Creation

1. Consumer Assistance Request
    a. The consumer can request assistance:
        i. from the Fabrick website (dedicated link for the requests)
        ii. via e-mail (dedicated e-mail address)
        iii. via telephone (dedicated number)
2. Ticket Creation
    a. If the consumer uses the website channel, the ticket number will be generated automatically with a relative priority assigned;
    b. If the consumer uses the e-mail channel, a generic issue will be generated automatically;
    c. If the consumer uses the phone channel, the issue will be managed by a support engineer who will be responsible for opening the ticket in the customer support system.

Phase 2: Resolution/Workaround

A Support Engineer, who will verify the type of issue and will assign or modify a level of priority, will manage the Customer Assistance Request.
An automatic email will be sent to the consumer to notify the opening of the Assistance Request procedure as “IN PROGRESS”.

In case of a Priority 1, the procedure will automatically generate an SMS to the available mobile phone number and an email to the address provided, so that Support can take action in a timely fashion and in compliance with the guaranteed response times.

As the resolution/workaround is found, the operator will fill a response on the open request and will update the issue status to “RESOLVED”, generating an automatic email notifying the consumer.

In case further information is required from the consumer, the request will be updated to “UPDATE REQUIRED”.

It will be possible to delete a request, if necessary. In such case, the issue will be updated to “CANCELLED”, and the consumer will be notified via an automatic email.

As the request reaches completion, i.e. either “RESOLVED” or “CANCELLED” status, the request will automatically be updated to “CLOSED”.

Report/Statistics

Through the support procedure, it will be possible to generate reports that help monitor the response times.