THE Berlin GROUP

A EUROPEAN STANDARDS INITIATIVE

Joint Initiative on a PSD2 Compliant XS2A Interface

NextGenPSD2 XS2A Framework

Security Bulletin

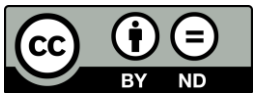
Version 1.0

06 November 2019

License Notice

This Specification has been prepared by the Participants of the Joint Initiative pan-European PSD2-Interface Interoperability* (hereafter: Joint Initiative). This Specification is published by the Berlin Group under the following license conditions:

- "Creative Commons Attribution-NoDerivatives 4.0 International Public License"

  This means that the Specification can be copied and redistributed in any medium or format for any purpose, even commercially, and when shared, that appropriate credit must be given, a link to the license must be provided, and indicated if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. In addition, if you remix, transform, or build upon the Specification, you may not distribute the modified Specification.

- Implementation of certain elements of this Specification may require licenses under third party intellectual property rights, including without limitation, patent rights. The Berlin Group or any contributor to the Specification is not, and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.
- The Specification, including technical data, may be subject to export or import regulations in different countries. Any user of the Specification agrees to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import (parts of) the Specification.

---

* The 'Joint Initiative pan-European PSD2-Interface Interoperability' brings together participants of the Berlin Group with additional European banks (ASPSPs), banking associations, payment associations, payment schemes and interbank processors.

Contents

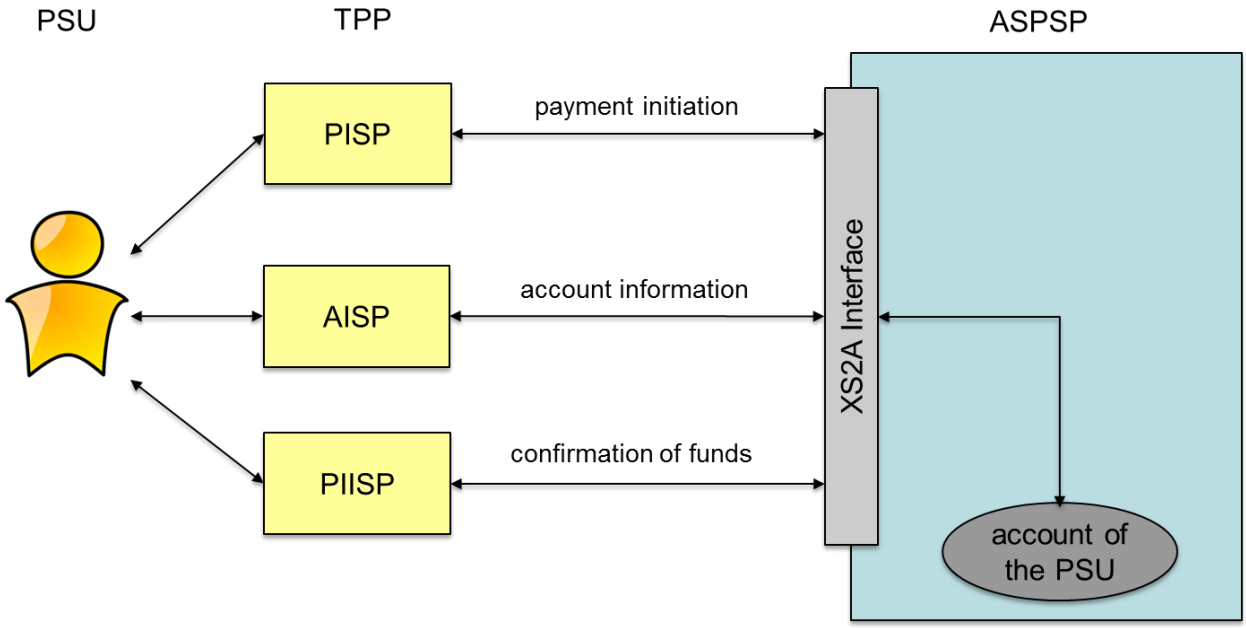
1 Introduction

1.1 Background

The Berlin Group started to publish its XS2A Framework in Version 1.3 on 15 October 2018 with later updates on the Implementation Guidelines leading to a Sub Version 1.3.4 on 5 July 2019. This framework consists of the two Documents

- [XS2A OR]: Operational Rules and
- [XS2A IG]: Implementation Guidelines.

The following account access services are covered by this framework:



Every XS2A interface might be affected by potential fraud and attacks, like the ASPSP online channels itself. ASPSPs have evolved several mitigation measures to counter attacks in their online channels.

The NextGenPSD2 XS2A Implementation Guidelines foresee the support of several mitigation measures to counter fraud in line with the corresponding requirements of [EBA-RTS]. Many of these mitigation measures are related to processing of SCA methods or SCA exemption processing. These measures might further depend on the chosen SCA Approach of the ASPSP, i.e. the choice between an

- Embedded SCA Approach,
- Decoupled SCA Approach,
- Redirect SCA Approach, or the

- Integrated OAuth SCA Approach,

as defined in [XS2A IG].

The NextGenPSD2 Framework supports in addition to the different SCA Approaches as such the transport of major data as detectable in the PSU – TPP interface. This sort of PSU related device and interface data is used by ASPSPs already today in online channels for fraud detection.

Chapter 2 is explaining why this data was introduced into the Implementation Guidelines and how the absence of related data might lead to a higher risk score in the risk management systems of the ASPSPs. Thus, some of these mitigation measures are addressing the specific situation of introducing a Third Party Provider between the PSU and the ASPSP by forwarding PSU device or PSU –TPP interface related data to the ASPSP.

In [FAPI-CBPIA] specific attacks on redirect based protocols implemented within the PSD2 context have been published, where session fixation is addressed. The NextGenPSD2 TF has decided to define a specific mitigation measure in the NextGenPSD2 XS2A interface for session fixation in addition to mitigation measures defined in Section 2 applicable only in the SCA Redirect Approach and Integrated SCA OAuth Approach for the future version 2 of the XS2A Framework. This mitigation measure supports an authorization code as defined in the OAuth2 protocol generated during the SCA processing on the redirection authorisation page, which is then to be used in a subsequent authorisation confirmation step. This approach is defined in Chapter 3 of this document. Depending on the ASPSP's requirements, this mitigation measure might already be implemented in the current version 1.3.4 of the Implementation Guidelines. In this case, this needs to be documented explicitly in the ASPSP's documentation.

1.2 Change Log

Version	Change/Note	Approved
1.0	Initial Version	6 November 2019

2 Support of Existing Mitigation Measures

Already version 1.3.x of [XS2A IG] supports some (technical) mitigation measures which (by proper implementation) may support to minimise the chance of success of potential attacks at the XS2A interface. These are:

- Shortening the time frame for executing a necessary authentication of the PSU.
- Check and evaluate information known about the PSU/TPP interface.
- Execute an OAuth 2 pre-step.
- Accept only complete transactions at the XS2A interface.

Many attacks at online channels are based already today to some extent on social engineering. Technical parts of the attacks can only be successful if the attacked PSU cooperates due to his deception by a "convincing story" of the attacker. Due to the changing eco system for accessing accounts managed by an ASPSP (for initiating payments or for retrieving account information) these attacks might become more likely and PSU might become more easily vulnerable by such attacks. For this reason it is recommended that an ASPSP in addition to technical counter measures increases his efforts for informing and raising of the awareness of the PSU about potential attacks.

2.1 Shortening the time frame for authentication

For many attacks the attacker needs the cooperation of the attacked PSU, because the authentication of the PSU is a requirement for the transaction to be executed successfully. For this cooperation the attacker has to convince the PSU to execute the strong customer authentication within a time frame beginning with the start of the (either implicit or explicit) authorisation process for the transaction at the XS2A interface (see for example section 5.1 of [XS2A IG]) and ending after some interface specific time out. The duration of this time frame is determined by the ASPSP. By shortening this time frame the chances of success for the attacks is decreased.

2.2 Check and evaluate information about the PSU/TPP interface

Version 1.3.x of [XS2A IG] enables to include information about the PSU/TPP interface into request messages to be sent by the TPP to the XS2A interface. Parameters of the header may contain the following information:

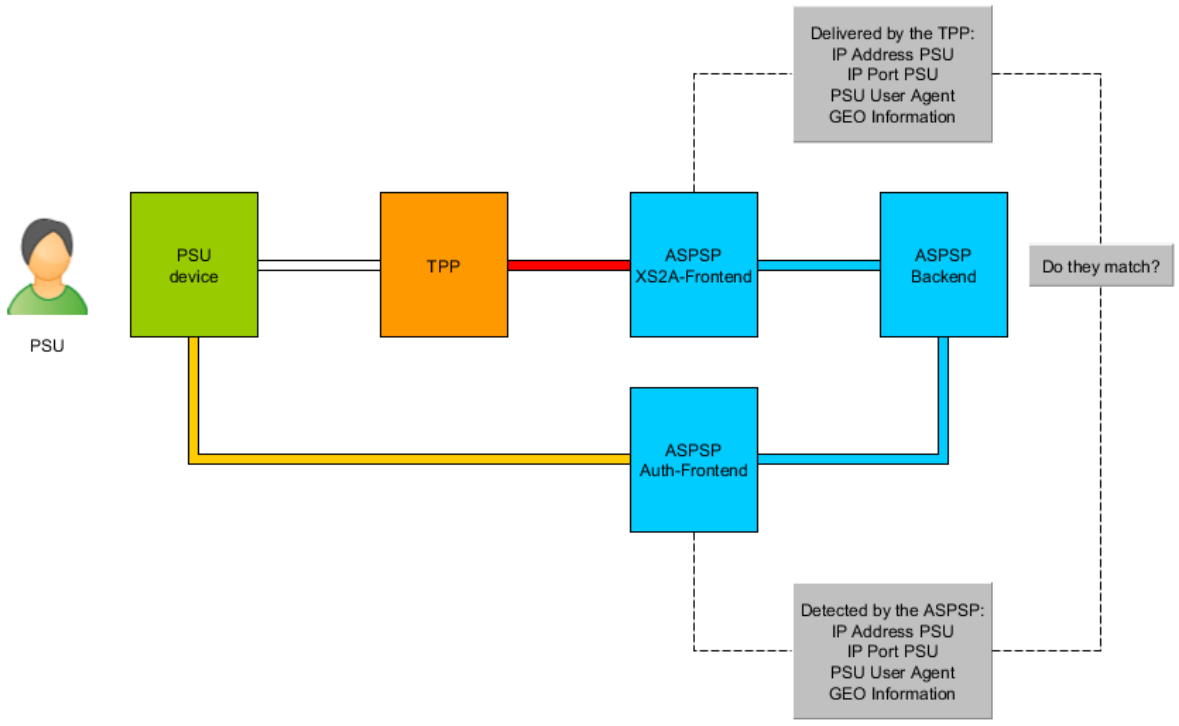
- IP address of the PSU for accessing the TPP.
- IP Port number of the PSU for accessing the TPP.
- Key information about the browser type and operating system used by the PSU to access the TPP.
- Device Identification.
- Information about the GEO location of the PSU while accessing the TPP.

The IP address of the PSU has to be delivered mandatorily by the TPP as part of the "Payment Initiation Request" and the "Establish Consent Request". For all other information it is optional to deliver these parameters. Also for other request messages it is optional to

include this information. See section 5.2 (for payment initiation) and 6.2 (for account information) of [XS2A IG] for an overview of these header parameters.

This information about the PSU/TPP interface will be used by the ASPSP as input for his fraud detection and risk management systems. Some ASPSPs use this information also to exclude some authentication methods (for example some ASPSPs do not allow to receive an OTP by SMS on the same smartphone used also for the transaction itself). For this reason it is highly recommended that a TPP includes all of this information into his request messages. Missing information may result in an assessment of the user device as not useable for the authentication method or in a classification of the current transaction as a "higher risk transaction". By this the probability of a rejection of that transaction due to the result of fraud detection and/or risk management might be increased.

Transactions at the XS2A interface may include a redirection of the PSU to a frontend of the ASPSP. This is the case if the redirect approach or the integrated OAuth approach for executing a strong customer authentication is used. For these transactions the information about the PSU/TPP interface delivered by the TPP should be used by the ASPSP to detect possible attacks to the sessions used at the PSU/TPP interface and the PSU/ASPSP interface. The following picture shows the context:



By comparing the information about the PSU/TPP interface delivered by the TPP and the information about the PSU/ASPSP interface detected by the ASPSP itself, the possibility of the success of attacks as described in [FAPI-CBPIA] can be reduced.

2.3 OAuth 2 Pre-Step

The NextGenPSD2 Interface allows the ASPSP to define an OAuth2 Pre-Step. Depending on the detailed definition of this Pre-Step, another mitigation measure is defined to counter potential attacks on the NextGenPSD2 protocol. Since this pre-step is not further detailed within [XS2A IG], this mitigation measure is not further detailed in this document neither.

2.4 Complete transactions for submission at the XS2A interface

Further measures should be taken by an ASPSP to complicate the automated generation of mass attacks according to the attack scenario described in [FAPI-CBPIA]. For example if the PSU ID and/or the account number of the attacked PSU are not already demanded as part of the submission of a transaction at the XS2A interface of the ASPSP, the attacker can generate much more easily a fraudulent transaction and submit this at the XS2A interface, hoping that the PSU will provide the missing information (as part of the authorisation process) and after that will authorise the transaction. In this case the attacker can generate and submit a huge number of fraudulent transactions automatically, hoping that at least some of the attacked PSUs might complete and authorise some of these transactions. To avoid this kind of automated attacks transactions should be determined as completely as possible as part of their submission at the XS2A interface.

3 Additional Mitigation Measure (Authorization Code)

Version 2 of the NextGenPSD2 Framework will contain further mitigation measures specifically against the attack for redirect based SCA architectures described in [FAPI-CBPIA]. This solution is following the solution proposal as defined in OAuth2 using an access token resp. a confirmation code for a confirmation command of the TPP after the transaction has been authorized by the PSU via a redirection to the ASPSP authentication server. This solution will be available for the Integrated OAuth NextGenPSD2 Interface solution as well as for a plain redirect SCA approach. The ASPSP will inform the TPP about the extended process step by providing an additional hyperlink with tag "confirmation" together with either the hyplink with tag "scaOAuth" or "redirect".

3.1 Impact on Flows

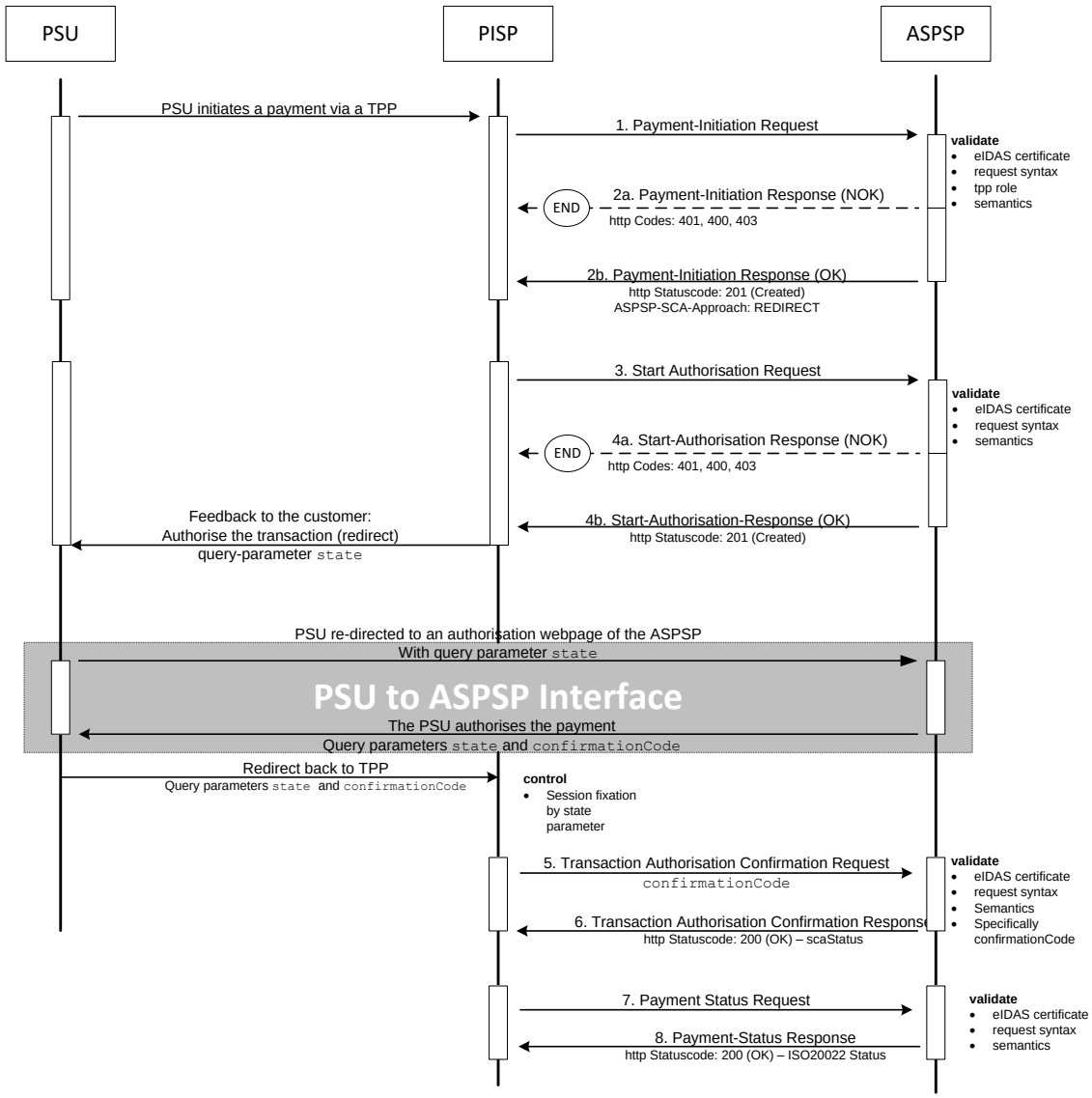
In the following sections, the impact on the flow is defined. It has been done exemplary for the process of payment authorisation, but could also be applied to consents for account information or authorising signing baskets.

The change to the existing flows is that instead of the TPP sending a Payment Status Request Message (GET command on status resource), the TPP will need to send a Transaction Confirmation Request Message (PUT command with an access token to the authorisation resource), as shown in the following flows.

The functional difference of the two solutions is that the payment will not be executed by the ASPSP in the new solution as long as the Transaction Confirmation Request Message has not been performed.

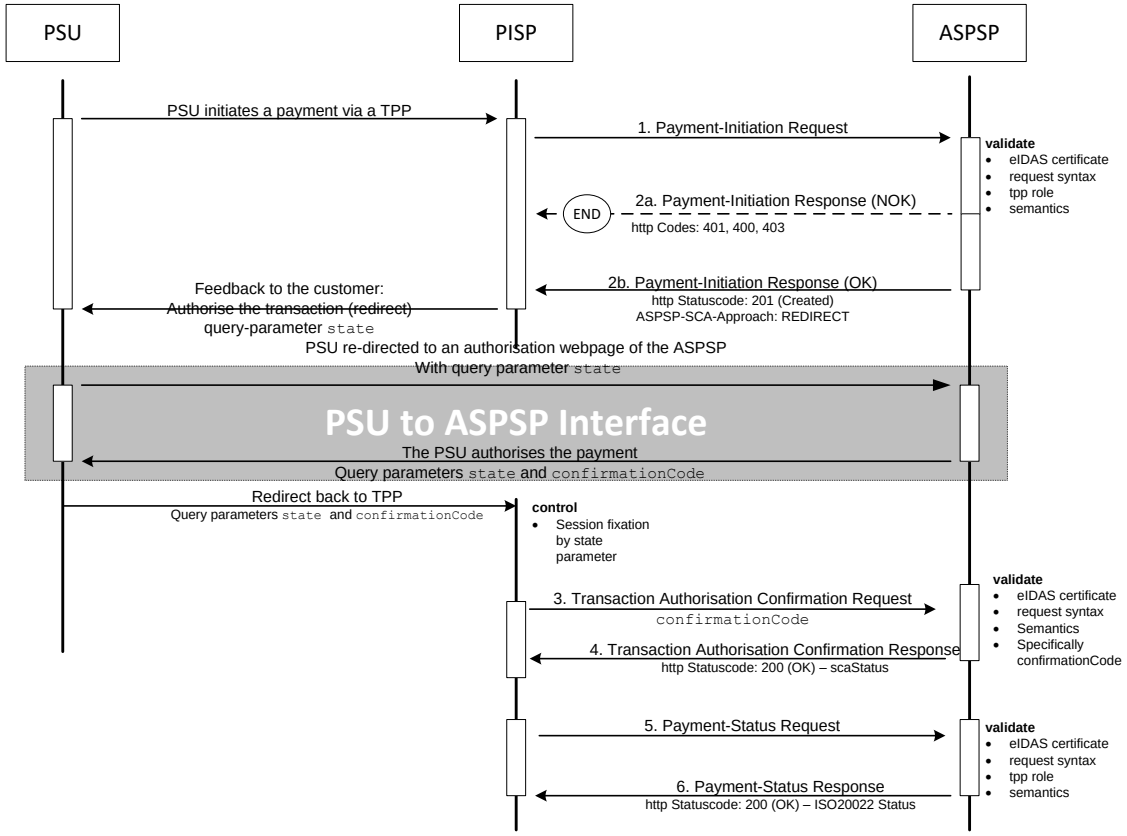
3.1.1 Redirect SCA Approach: Explicit Start of the Authorisation Process

If the ASPSP supports the Redirect SCA Approach, the message flow within the payment initiation service is simple. The Payment Initiation Request is followed by an explicit request of the TPP to start the authorisation. This is followed by a redirection to the ASPSP SCA authorisation site. An authorisation confirmation request might be requested by the TPP after the session is re-redirected to the TPP's system and after the TPP's control on session fixation. In the end, a payment status request might be needed by the TPP to control the exact status of the payment initiation.



3.1.2 Redirect SCA Approach: Implicit Start of the Authorisation Process

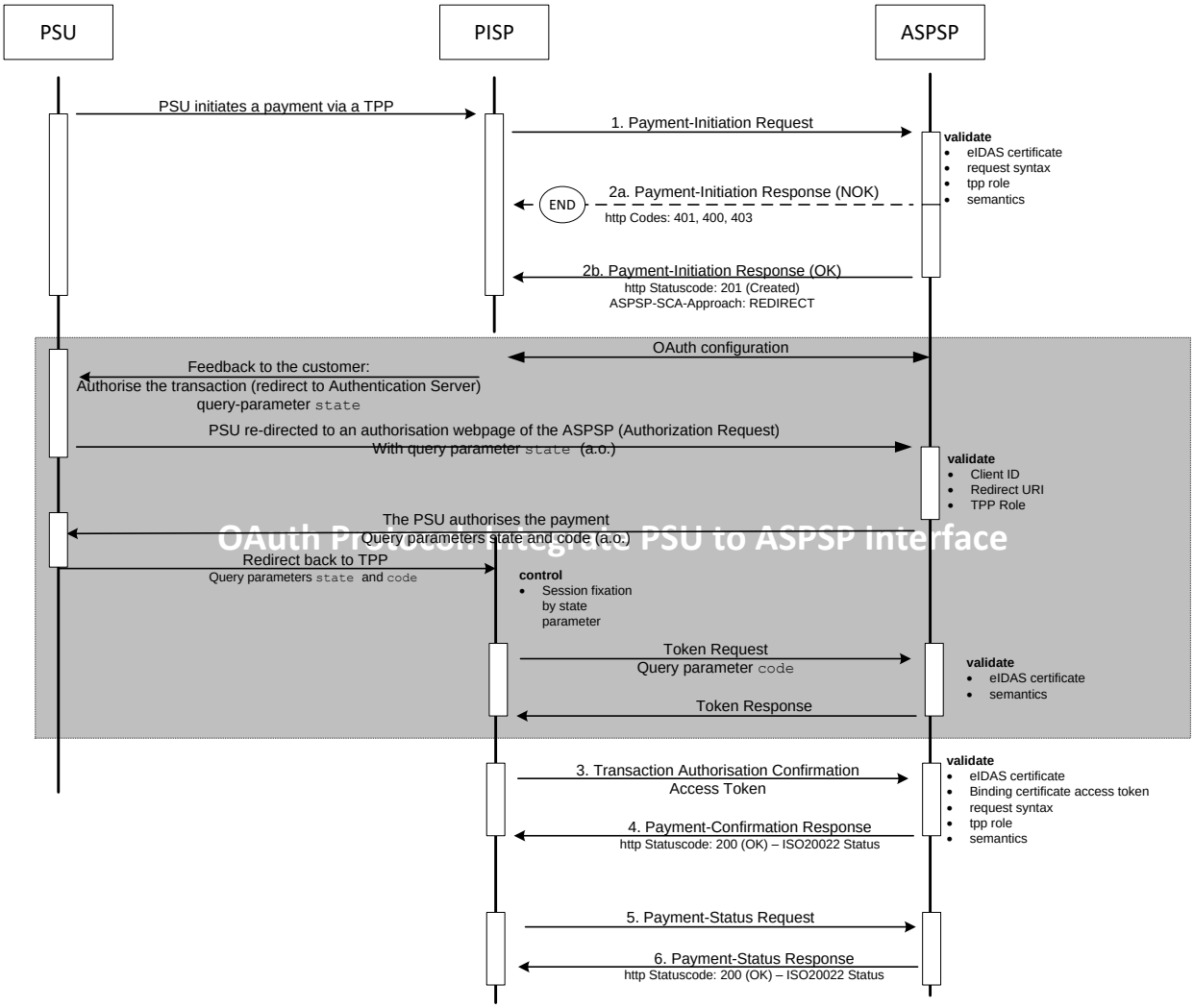
ASPSPs might start the authorisation process implicitly in case of no additional data is needed from the TPP. This optimisation process results in the following flow (which is exactly the Redirect SCA Approach flow from the version 1.0 and 1.1 of the Implementation Guideline before authorisation sub-resources have been established). In this case, the redirection of the PSU browser session happens directly after the Payment Initiation Response. An authorisation confirmation request might be requested by the ASPSP after the session is re-redirected to the TPP's system and after the TPP's control on session fixation. In the end, a payment status request might be needed by the TPP to control the exact status of the payment initiation.



3.1.3 OAuth2 SCA Approach: Implicit Start of the Authorisation Process

If the ASPSP supports the OAuth2 SCA Approach, the flow is very similar to the Redirect SCA Approach with implicit start of the Authorisation Process. Instead of redirecting the PSU directly to an authentication server, the OAuth2 protocol is used for the transaction authorisation process. An authorisation confirmation request might be requested by the TPP after the session is re-redirected to the TPP's system and after the TPP's control on session fixation. In the end, a payment status request might be needed by the TPP to control the exact status of the payment initiation.

Remark: The OAuth2 SCA Approach with explicit start of the Authorisation Process is treated analogously.



It is further recommended for ASPSPs and TPPs in this case to follow the Security Best Practice definitions as defined in [OA-SecTop]. This reference will also be added in the next version of the Implementation Guidelines.

3.2 Impact for integration into the current implementation

This functionality could be offered by the ASPSP already based on [XS2A IG]. This section describes an integration into the current version.

3.2.1 Impact on hyperlinks

The hyperlink with tag "confirmation" might be added by the ASPSP to the response body in the following sections of [XS2A IG]:

Section 5.3.1, 6.4.1, 6.4.4, 7.1, 7.2.3

The entry for the hyperlink is defined as follows:

"confirmation": Might be added by the ASPSP if either the "redirect" or "scaOAuth" hyperlink is returned in the same response message. This hyperlink defines the URL to the resource which needs to be updated with

- a confirmation code as retrieved after the plain redirect authentication process with the ASPSP authentication server or
- an access token as retrieved by submitting an authorization codeafter the integrated OAuth based authentication process with the ASPSP authentication server.

The corresponding confirmation command is defined in the new section 7.6.

The ASPSP might decide to use this function depending on the underlying transaction to be authorised. Since the communication to the TPP is dynamically given during the transaction, the differentiation of the solution by the ASPSP technically does not need to be communicated to the TPP beforehand.

3.2.2 Impact on authorisation status element scaStatus (Section 14.15)

If the confirmation of the transaction authorisation is mandated by the ASPSP by providing a hyperlink with tag "confirmation", then a new code "unconfirmed" will be introduced to the SCA Status data type:

Code	Description
unconfirmed	"Authorisation is technically successfully finalised by the PSU, but the authorisation resource needs a confirmation command by the TPP yet.

Remark: This definition implies that the current available scaStatus "finalised" is still telling the TPP that the full authorisation process (including potentially a confirmation) is successfully completed.

3.2.3 Impact on Error Handling (Section 14.11.1)

New code for Message Code in case where a confirmation is processed but the preceding SCA method failed.

Message Code	HTTP Response Code	Description
SCA_INVALID	400	Method Application on authorisation resource blocked since SCA status of the resource equals "failed".

3.3 New Section 7.6: Confirmation of Authorisation

This call is used, when in the preceding call the hyperlink of type "confirmation" was contained and if a redirection authentication method has been applied. Before the call can be submitted by the TPP, an authorization code, respectively a confirmation code needs to be retrieved by the TPP after the SCA processing in a redirect to the ASPSP authentication server.

In case of the integrated OAuth SCA Approach, the overall procedure is described in Section 12 of [XS2A IG].

In case of the Redirect SCA Approach, the procedure is described in the following sub section.

3.3.1 Retrieving the Confirmation Code in Redirect SCA approach

The TPP needs to fix the session of the PSU on the TPP browser with a nonce, where part of it is a unique state parameter.

In preparation of sending the authorization request, the TPP shall

- create a one-time use XSRF token to be conveyed to the ASPSP in the “state” parameter and,
- bind this value to the current session in the user agent.

Note: In case of the integrated OAuth SCA Approach, the TPP has to generate in addition a nonce for the challenge parameter. This has also to be bound to the session of the user agent.

3.3.1.1 Requirements on HTTP request of PSU browser

The TPP needs to forward the state parameter as query parameter to the PSU, which will lead to a GET HTTP request of the PSU browser as required as follows:

Query Parameter PSU Authorisation Request (GET command)

Attribute	Type	Condition	Description
state	string	mandated	state parameter as defined by the TPP as a unique parameter and bound to the PSU/TPP session.

Example

```
GET ASPSP-Redirect-URI?state=1234567er
```

After the customer authentication has taken place on the ASPSP server, the ASPSP responds with the same state parameter and a unique confirmationCode bound to the authorisation resource as query parameters. The confirmationCode will only be contained if SCA has been successfully performed.

Query Parameter PSU Authorisation Response (GET command response)

Attribute	Type	Condition	Description
state	string	mandated	state parameter as used in the corresponding request.
code	string	conditional	unique authorisation code of the ASPSP, bound to the related transaction, in case of Integrated OAuth SCA Approach.
confirmationCode	string	conditional	unique authorisation code of the ASPSP, bound to the related transaction, in case of Redirect SCA Approach.

Example in case of Redirect SCA Approach

```
http 302?state=1234567er&confirmationCode=2256ffgh
```

3.3.2 Confirmation Call Pre-Condition

When retrieving the GET command from the PSU browser, the TPP must check whether the state parameter is linked to the current session.The “state” value is linked to the current session in the user agent. If the check is positive then the TPP further processes

- within context of the Integrated OAuth SCA Approach with retrieving the access Bearer token as described in Section 13 of [XS2A IG] and then proceed as described in Section 3.3.3.
- within context of the Redirect SCA Approach directly as described in Section 3.3.3.

If the check fails, the transaction must be stopped by the TPP.

3.3.3 Authorisation Confirmation Call

Call in the context of a Payment Initiation Request

```
PUT /v1/{payment-service}/{payment-
product}/{paymentId}/authorisations/{authorisationId}
```

Updates the payment initiation authorisation sub-resource data on the server by an authorization code, if requested by the ASPSP.

Call in the context of a Payment Cancellation Request

```
PUT /v1/{payment-service}/{payment-product}/{paymentId}/cancellation-
authorisations/{cancellationId}
```

Updates the payment initiation cancellation authorisation sub-resource data on the server by an authorization code, if requested by the ASPSP.

Call in case of an Account Information Consent Request

```
PUT /v1/consents/{consentId}/authorisations/{authorsationId}
```

Updates the account information consent authorisation data on the server by an authorization code, if requested by the ASPSP.

Call in the context of a Signing Basket Authorisation Request

```
PUT /v1/signing-baskets/{basketId}/authorisations/{authorisationId}
```

Updates the signing basket authorisation data on the server by an authorisation code, if requested by the ASPSP.

Path Parameters

Attribute	Type	Description
payment-service	String	The possible values are “payments”, “bulk-payments” and “periodic-payments”
payment-product	String	The payment product, under which the payment under paymentId has been initiated. It shall be checked by the ASPSP, if the payment-product is matching the payment initiation addressed by paymentId.
paymentId, basketId or consentId	String	Resource identification of the related payment initiation, signing basket or consent resource.
authorisationId	String	Resource identifciation of the related Payment Initiation, Signing Basket or Consent authorisation sub-resource.
cancellationId	String	Resource identification of the related Payment Cancellation authorisation sub-resource

Query Parameters

No specific query parameters.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Conditional	Authorization Bearer Token as retrieved by the TPP in case the integrated OAuthSCA Approach as described in Section 12 of [XS2A IG].

Request Body

Attribute	Type	Condition	Description

Attribute	Type	Condition	Description
confirmationCode	String	Conditional	Confirmation Code as retrieved by the TPP from the redirect based SCA process as described in Section 3.3.1

Response Code

HTTP response code is 200.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
scaStatus	SCA Status	Mandatory	Value "finalised" if the transaction authorisation and confirmation was successful. Value "rejected" if the transaction authorisation was not successful.
_links	Links	Mandatory	A list of hyperlinks to be recognised by the TPP. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. **Remark**: All links can be relative or full links, to be decided by the ASPSP. Type of links admitted in this response, (further links might be added for ASPSP defined extensions): "status": The link to retrieve the Status of the corresponding transaction resource.

Attribute	Type	Condition	Description
psuMessage	Max512Text	Optional	

Example for integrated OAuth solution

Request

```
PUT https://api.testbank.com/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/authorisations/123auth456
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Authorization:          Bearer 1234567
```

Response

```
HTTP/1.x 200 OK
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json
{
 "scaStatus": "finalised",
 "_links":{
   "status":  {"href":"/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/status"}
   }
```

Example for redirect solution

Request

```
PUT https://api.testbank.com/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/authorisations/123auth456
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
{ "confirmationCode": "2256ffgh"}
```

Response

```
HTTP/1.x 200 OK
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json
{
 "scaStatus": "finalised",
```

```
 "_links":{
   "status":  {"href":"/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/status"}
   }
}
```


4 References

[XS2A-OR] NextGenPSD2 XS2A Framework, Operational Rules, The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface, version 1.3, published 21 December 2018

[XS2A-IG] NextGenPSD2 XS2A Framework, Implementation Guidelines, The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface, Version 1.3.4 published 5 July 2019

[EBA-RTS] Commission Delegated Regulation (EU) 2018/389 of 27 November 2017 supplementing Directive 2015/2366 of the European Parliament and of the Council with regard to Regulatory Technical Standards for Strong Customer Authentication and Common and Secure Open Standards of Communication, C(2017) 7782 final, published 13 March 2018

[FAPI-CBPIA] OpenID Foundation, Financial-grade API (FAPI) Working Group, Cross-Browser Payment Initiation Attack, https://bitbucket.org/openid/fapi/src/master/TR-Cross_browser_payment_initiation_attack.md, 3.01.2019

[OA-SecTop] OAuth 2.0 Security Best Current Practice draft-ietf-oauth-security-topics-13, Lodderstedt et al., 8 July 2019, https://tools.ietf.org/html/draft-ietf-oauth-security-topics-13