NextGenPSD2 XS2A Framework Implementation Manual

1 Table of Contents

2 About the Document

This document is a guide for the implementation of the NextGenPSD2 XS2A framework foropen banking compatible with Georgian legislation. The document is fully compatible with the relevant frameworks (1) and (2) and qualifies their requirements, taking into account Georgian legislation and the local context.

The purpose of the document is to facilitate the harmonization of open banking in Georgia to the highest possible degree, and, in keeping with this purpose, it establishes many such restrictions for which the NextGenPSD2 XS2A framework provides the opportunity of freer interpretation. A non-exhaustive list of issues additionally restricted by the document is as follows:

1) Acceptable methods of customer authentication (although (1) and (2) provide choices for strong customer authentication methods, the following document only requests OAuth2 authentication)
2) Document format – the following document declares it mandatory to support the JSON format

The document has also regulated some issues which were considered insufficiently regulated (or left beyond regulation) by (2). In certain cases, when issues clarified by this document establish requirements that are incompatible with the NextGenPSD2 XS2A framework, the issue is marked with the text "// deviation from the XS2A framework".

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" must be interpreted in accordance with (3).

The document follows a hierarchical structure. In cases when issues described by a certain chapter of the document (e.g. card account information services) are marked as "OPTIONAL", the necessary requirements provided in that chapter (and its subchapters) are considered necessary only if the relevant issue is implemented.

Issues marked as "SHOULD", "RECOMMENDED" and "SHOULD NOT" are also followed with specifying indications as to why it is recommended (or not recommended) to implement that issue in accordance with the document's directive.

3 Document History

Version	Date	Update
0.3	20.08.2020	Initial version.
0.4	01.09.2020	A detailed description of account information services was added. The description of strong customer authentication became more detailed.
0.5	09.09.2020	Comments from the National Bank of Georgia and different commercial banks were represented.
0.6	16.09.2020	Additional comments from the National Bank of Georgia were represented – "global agreement" was deemed impermissible and the rule for granting identifiers to system participants was defined. The rule for cross-checking the dates in messages was added.

4 Terms and Abbreviations

All terms used in this document which are defined in the Law of Georgia on "Payment Systems and Payment Services" have meanings defined by this same law, except for the cases where the term is defined differently in the corresponding chapter.

The following abbreviations may be used in the document:

ASPSP	Account Servicing Payment Service Provider
TPP	Third Party Provider – Payment Service Provider, except ASPSP
PISP	Payment Initiation Service Provider
AISP	Account Information Service Provider
PIISP	Payment Instrument Issuing Service Provider
CRL	Certificate Revocation List
OCSP	Online Certificate Status Protocol
PSU	Payment Service User

Table 1: Abbreviations used.

In none of the chapters of this document does the term "client" signify a person who takes advantage of any type of financial service. For the purposes of this document, in the majority of cases, the term "client" signifies a TPP (Third Party Provider).

5 TPP, ASPSP and Their Authentication Towards Each Other

The TPP and ASPSP SHALL connect with each other through a protected channel which simultaneously ensures the authentication of both sides towards each other. It is REQUIRED that this be implemented in accordance with (2) on the levels of both the communication channel and software.

5.1 Usage of Unique Identifiers

For identifying each other, the TPP and ASPSP SHALL use unique numbers which are composed based on the following principle:

PSDGE-NBG-suffix

Where "suffix" is defined as follows:

1. If, at the start of the payment service, the payment service provider (ASPSP, TPP) is participating in the real-time gross settlement (RTGS) system of the National Bank of Georgia, then the suffix in their identifier will automatically (without the need for an agreement with the National Bank of Georgia) be the same as their RTGS system participant identifier (thus, the suffixes will be PSDGE-NBG-BAGAGE22, PSDGE-NBG-CBASGE22, PSDGE-NBG-DISNGE22, etc.).
   1.1. If the payment service provider subsequently loses their RTGS system participant status, they will retain the right to use the aforementioned identifier for 2 years after the loss of status.
   1.2. If the payment service provider changes their RTGS system participant identifier, they will retain the right to use the old identifier for 2 years after the change.

2. In other cases, the suffix is the unique identifier granted to the payment service provider by the National Bank of Georgia. The following document does not regulate the rules for granting these identifiers by the National Bank of Georgia. If such a payment service provider, who has already been granted an identifier in accordance with the 2nd paragraph, later enters into the RTGS system, they will retain the right to use the initial identifier for 2 years after entering the RTGS system.
3. If the payment service provider changes their identifier in any of the cases defined in the previous paragraphs, they SHALL NOT use their old identifier, regardless of whether the 2-year term defined in the previous paragraphs has expired or not.

The client ID parameter used during the initiation of the Oauth2 protocol SHALL strictly match (also taking into account the cases of the symbols) the aforementioned identifier.

5.2 Protected Communication Channel Requirements

1. The communication channel SHALL be established in accordance with the following protocols: TLS, TLS 1.2 or higher.
2. At the moment of establishing the connection, both sides SHALL provide their X.509 certificates. For the purposes of this document, each side SHALL operate in accordance with one of the following scenarios:
   2.1. The X.509 certificate fully, technically satisfies the requirements of (4) for qualified certificates of website authentication, includes the identifier defined by chapter 5.1, and is issued by an issuing party which satisfies the requirements set forth by the Georgian legislation for the issuing parties of certificates needed for open banking.
   2.2. The X.509 does not technically satisfy the requirements of (4) and/or does not include the identifier defined by chapter 5.1, but its usage is possible in TLS sessions for client and server authentication and there is additional information, issued about the certificate, which satisfies the requirements of chapter 5.5 of this document.

During the realization of the OAuth2 protocol, the client (TPP) authentication SHALL be used, the tokens SHALL be issued in accordance with the requirements of (5), and certificates defined in the previous chapter SHALL be used in this process.

The ASPSP and TPP SHALL protect, with the certificates, all such hyperlinks which they exchange in the communication process defined in this document (e.g. the ASPSP returns the hyperlink to the TPP with the _links section, as is defined in the subchapters of chapter 9).

5.3 Additional Requirements for Dynamic Registration of OAuth2 Clients

The ASPSP SHALL support the dynamic registration of clients in accordance with (6) as well as (7) and the management of dynamic registration in accordance with (8).

During dynamic registration, the ASPSP SHALL grant the same identifier (client_id) to the TPP which strictly matches the TPP identifier defined by chapter 5.1.

Remark: This requirement is stipulated, firstly, for the purpose of meeting the requirements of chapter 5.5, but other possibilities for the TPP to provide additional information to the ASPSP might be added in future versions of the document.

5.4 Requirements for Protection on the Software Level

During the exchange of information, the TPP and ASPSP SHALL, in addition to messages, use digital signature technology, which will be in full convergence, technologically, with the qualified electronic seal technology, even in cases where the aforementioned is not a qualified electronic seal acknowledged by Georgian legislation. The exception SHALL be applicable only to the certificate issuer's status in Georgia and even in such a case, the certificate issuer must be agreed upon by the National Bank of Georgia and the ASPSP/TPP.

Remark: To keep the compatibility with (1) *and* (2)*, the aforementioned technical signature will henceforth be referred to as a "qualified electronic seal".*

An electronic seal SHALL be attached to responses sent from the TPP to the ASPSP, as well as from the ASPSP to the TPP. Issues pertaining to the attachment of this seal are discussed in chapter 7.4 of this document.

The ASPSP SHALL verify the qualified electronic seal implemented by the TPP cryptographically as well as in keeping with the rule outlined in chapter 5.6 of this document.

If the X.509 certificate does not technically satisfy the requirements of (4) and/or does not include the identifier defined by chapter 5.1, additional information SHALL be issued about it. The rules for the issuing and usage of additional information is regulated by chapter 5.5 of this document.

5.5 Usage of Non-standard Certificates

// deviation from the XS2A framework

The ASPSP and the TPP have the right, for the purpose of compatibility with this document, to use such certificates which do not satisfy the requirements of (4) and/or do not include the identifier defined by chapter 5.1 of this document. In this case, the appropriate requirements defined in this chapter and its subchapters SHALL be met.

The ASPSP SHALL support the dynamic registration of clients in accordance with (6) as well as (7) and the management of dynamic registration in accordance with (8).

The ASPSP has to offer the TPP the possibilities of dynamic registration and the management of such registration in an optional manner (e.g. only for such cases where the TPP wants to use a non-standard certificate or wants to send additional information – such as the address of a logo - to the ASPSP). In particular, if the TPP uses website authentication and qualified electronic seal certificates which satisfy the requirements of (4) and include identifiers defined by chapter 5.1, the ASPSP SHALL NOT make their dynamic registration obligatory, and the ASPSP SHALL collect all necessary information directly from the certificates.

During dynamic registration, the ASPSP SHALL assign the TPP the same identifier (client_id) which strictly matches the TPP's identifier as defined in chapter 5.1. In cases where the TPP uses non-standard certificates, the ASPSP SHALL collect its identifier and other information from the Software Statement (see 5.5.2) and not oblige the TPP to provide additional information.

5.5.1 Authorization Certificates for ASPSP and TPP Authentication

If the X.509 certificate used by the ASPSP for the establishment of a protected channel does not satisfy the requirements of (4) and/or does not include identifiers defined in chapter 5.1, an additional authorization (attribute) certificate SHALL be granted for it, in accordance with (9). The attribute certificate SHALL satisfy the following requirements:

1) Must be issued by the National Bank of Georgia (or possibly by other organizations as defined by the National Bank of Georgia) and signed with such certificates which the National Bank of Georgia will provide to every interested party at no charge.
2) Must be valid at the moment of its usage.
3) The term of validity must not exceed 3 hours.
4) Must be connected to the X.509 certificate with a cryptographical method.
5) Must include the identifier defined by chapter 5.1.
6) Must include the payment service provider role in accordance with chapter 5.2.2. of (4).

The cryptographical connection of the certificate SHALL include the existence of the Acinfo /Holder entry in the certificate, which will use baseCertificateID and objectDigestInfo. objectDigestInfo SHALL use the SHA-256 or SHA-512 hash functions.

The ASPSP SHALL publish the certificate on the following address:

https://server-hostname-and-port/.well-known/openbankinggeo/psd2.crt

Where "server-hostname-and-port" signifies the hostname and, appropriately, port of the server with which the communication is protected using a non-standard certificate.

If the TPP protects the hyperlinks provided to the ASPSP (e.g. user redirection hyperlink after authentication) using a non-standard certificate, the TPP SHALL protect every such hyperlink in accordance with the rule defined in this chapter.

5.5.2 Software Statement Usage on the Client Side

If the X.509 certificate used by the TPP for the establishment of a protected channel does not satisfy the requirements of (4) and/or does not include identifiers defined in chapter 5.1, a Software Statement SHALL be issued for it by the National Bank of Georgia, and the TPP will present it to the ASPSP during dynamic registration.

The document SHALL be electronically signed by the National Bank of Georgia (or by another organization as defined by the National Bank of Georgia) using such a certificate which the National Bank of Georgia

will provide to every interested party at no charge. Additional issues pertaining to digital signatures and the sharing of relevant certificates are not regulated by this document.

The Software Statement SHALL include all of the fields which are included in Image 1: Software Statement example.

```
{
        "software_id": "65d1f27c-4aea-4549-9c21-60e495a7a86f",
        "software_roles": [
                "PISP",
                "AISP"
        ]
        "iss": "https://nbg.gov.ge",
        "sub": "PSDGE-NBG-99999999",
        "nbf": 1300419270,
        "exp": 1300819380,
        "token_endpoint_auth_method": "tls_client_auth",
        "client_name": "Example Client",
        "client_name#ka": "საჩვენებელი კლიენტი",
        "client_uri": "https://client.example.net/",
    "jwks": {
        "keys": [
            {
                "x5t#S256": [
                        "ac58a191de026f4ab6fd3b04293238ee2b50fa5fcc775f4a8f73139f7941e9ae"
                ],
               "use": "tls"
            },
            {
                "x5t#S256": [
                        "2fe0ff296ac2c5620f016ea8285e13f8d0f0a62ddf15ec69d5eb931b966e6e9a"
                ],
               "use": "sig"
            }
        ]
    },
}
```


Image 1: Software Statement example.

Where:

1) The value of the software_id field is defined by the National Bank of Georgia at their discretion.
2) software_roles are acceptable authorization roles as defined by PSD2 (one or more). PISP, AISP, ASPSP, PIISP are acceptable values.
3) nbf is the time of the Statement entering into force (amount of seconds since January 1, 1970, UTC time-zone).
4) exp is the Statement expiry date (amount of seconds since January 1, 1970, UTC time-zone).
5) client_name and client_name#ka include client names in English and Georgian. The National Bank of Georgia MAY define a translation in another language as well, but the support of such a translation by the TPP and the ASPSP is OPTIONAL.
6) jwks includes the JSON Web Key Set structure – the hash codes of the certificates which can be used in order to establish a TLS session or apply a seal (accordingly, the tls and sig values in the "use" attribute). In this array, all cases SHALL include only one entry.

5.5.3 Transmission of Non-standard Certificates During the TPP Dynamic Registration

During dynamic registration, the TPP SHALL transfer to the ASPSP those certificates from its TLS and seal certificates which are non-standard. The ASPSP SHALL test the correctness of each certificate (through hash comparison) towards the Software Statement document (see 5.5.2).

5.5.4 Non-standard Certificate of the ASPSP's Electronic Seal

In cases where the ASPSP's electronic seal certificate is not standard, the ASPSP shall publish the Software Statement document on each of its own servers which apply electronic seals to responses using these certificates, at the following address:

https://server-hostname-and-port/.well-known/openbakinggeo/statement-identifier.json

1) "server-hostname-and-port" signifies the hostname of that server and, appropriately, the port.
2) The "identifier" is the ASPSP's unique identifier defined by 5.1.

The TPP SHALL download the aforementioned document and use it in order to test the permissibility, for the purpose of this document, of the electronic seal certificate used by the ASPSP.

5.6 General Requirements for the Certificate Verification Process

The requirements of this subchapter are relevant to every certificate used in this chapter.

1. During the establishment of the connection, the certificate SHALL be tested in accordance with the following criteria:
   1.1. The certificate is issued by an issuing party which is acknowledged in Georgia as a trustworthy issuer of open banking certificates.
   1.2. The validity of the certificate is not suspended or annulled. The testing SHALL be performed in accordance with the OCSP (Online Certificate Status Protocol).
   1.3. Each party of the connection (the ASPSP as well as the TPP) SHOULD have the ability to use OCSP Stapling. In cases where one of the parties of the connection, during the establishment of the

connection, notes that they have such an ability, the other party SHALL return all the needed OCSP responses. If not, the first party SHALL deny the establishment of a protected connection.

1.4. All appropriate mechanisms of testing the validity of certificates SHALL be observed, including the rule for testing the validity of OCSP certificates, which is defined by international standards and best practices.

5.7 General Requirements for Hyperlink Testing

In the process of requesting the ASPSP's services, the TPP SHALL test each hyperlink returned to them by the ASPSP and make sure that they are indeed communicating with the ASPSP.

The ASPSP SHALL test each hyperlink sent to them by the TPP and make sure that they are indeed communicating with the TPP.

6 Customer Authentication and Authorization

For the purposes of this document, customer authentication can be performed in two ways:

- General customer authentication in the channel between the TPP and the ASPSP.
- Strong customer authentication.

For authentication and authorization, the OAuth2 protocol - whose realization is fully compatible with the requirements of (10) - SHALL be used.

6.1 General Customer authentication in the Channel Between the TPP and the ASPSP

For general customer authentication in the channel between the TPP and the ASPSP, the OAuth2 SHALL be used as a "pre-step", as is outlined in chapter 4.3 of (2), for every message which the TPP transfers to the ASPSP. The exception is the consent transaction (see chapter 9.1), for which the requirements are regulated by this same chapter.

Accordingly, in every message sent by the TPP, the satisfaction of all the requirements outlined in (2) and noted with "if OAuth2 has been used as PSU authentication" SHALL be taken into account. The OAuth2 token SHALL be issued in the request header, Authorization parameter, with the type Bearer. The token SHALL be issued by the ASPSP directly.

As opposed to other services, in terms of consent transaction (see 9.1), the TPP SHALL use one of the following approaches:

1. Does not pass the OAuth2 token to the ASPSP while requesting consent.
2. Passes to the ASPSP a token issued by this same ASPSP during a different transaction (e.g. another consent transaction).
3. Passes to the ASPSP an OAuth2 token issued by an authorization server who has been previously agreed upon with the ASPSP.

The ASPSP SHALL support cases where the TPP does not provide a token during a consent transaction. In this case, the ASPSP has to assume that the request relates to an unidentified PSU, has to authenticate it using the OAuth2 protocol and receive the PSU's agreement.

The ASPSP SHALL NOT deny the consent transaction only because they were unable to read the token issued by the TPP, assumed that the said token is issued by an issuing party unknown to them or was issued by the TPP but was annulled for whatever reason. In this case, the ASPSP SHALL behave as if the TPP passed an empty token (see above).

Remark: If the token is issued by a party other than the ASPSP but there is an agreement upon its usage between the TPP and the ASPSP, the ASPSP SHALL NOT consider the existence of this token as strong customer authentication, regardless of how many authentication factors were used in the process of issuing this token and whether or not this was known to the ASPSP. But it is possible that the ASPSP transform the said token into strong customer authentication in accordance with the rule defined by **შეცდომა! კავშირის წყარო არ მოიძებნა.**.

6.2 Strong Customer authentication

For the purposes of this document, the ASPSP SHALL use the OAuth2 protocol with redirection. Other strong authentication methods defined in (2) (e.g. "embedded") are not supported. Accordingly, the ASPSP SHALL NOT request the initiation of the strong customer authentication process using any protocols other than OAuth2 from the TPP. The realization of the OAuth2 protocol SHALL be fully compatible with chapter 13 of (2) as well as the requirements of (10).

The ASPSP SHALL NOT request a confirmation of token receipt from the TPP, they must not return a 'confirmation' hyperlink type in the _links section (e.g. in accordance with chapters 5.3.1 and 6.3.1.1 of (2)). Despite this requirement, the TPPs SHOULD have support for such an ability for the purpose of full compatibility with (2).

6.3 Transforming General Customer authentication into Strong Customer authentication

The ASPSP MAY use their general authentication token (see 6.1) and consider its presentation as one factor authentication so as to simplify the strong authentication process for the TPP in accordance with the rule outlined in this chapter and only request the usage of one factor in the strong authentication process (for example, consider the presentation of a strong authentication token to be a confirmation of the fact that the user has gone through authentication with their name and password and strong authentication only requires confirmation with an SMS code).

In case of realizing such an optimization, in terms of expressing consent (see 9.1), the process SHALL be as follows:

1) The TPP begins the initiation of the consent transaction and passes the token in the Authorization header.

2) The ASPSP registers the consent transaction and connects it with the token outlined in the first step.
3) The ASPSP begins the OAuth2 strong authentication process in accordance with (2).
4) The authorization resource which the ASPSP returns to the TPP (see **შეცდომა! კავშირის წყარო არ მოიძებნა.**) requests authentication with a lessened (but necessary for strong authentication) amount of factors. For example, the ASPSP's authorization server does not offer the TPP to input their username and password and only requests that they send and confirm the SMS code.
5) After the authentication process is successfully carried out, the strong authentication OAuth2 token is generated.

7 API Structure

The exchange of data SHALL be fully compatible with (2), including the requirements defined by its chapter 4.4.

In cases where there is a link from (2) to expanded servers, for example, ones defined by (11) (e.g. using the TPP-Notification-URL parameter), during realization, it must be taken into account that this version of the document does not regulate such possibilities. Such connections will be regulated in future versions of this document and this might create danger to the compatibility of the already created interface with this document's future versions.

This document defines additional requirements which do not go against the aforementioned specifications and create limitations for issues where (2) assumes more freedom for the ASPSP.

7.1 API Resource Addresses

API resource addresses SHALL have the following form:

https://{provider}/{version}/v1/{service}{?query-parameters}

The fields have the following values:

- {provider} server host and path. The host and/or past MAY include the API version identifier, at the ASPSP's discretion.
- {version} the version of this document. The document version is indicated in paragraph 3.
- v1 – version, according to (2).
- {service} and {query-parameters} see (2).

This format is different from (2) only because of the {version} field which additionally links to this version of the document. The ASPSP MAY support services corresponding to different versions of this document.

The ASPSP SHALL clearly differentiate between the production and other (testing, demonstrational, etc.) versions of the service. The differentiation SHALL be indicated through the {provider} field.

7.2 API Request and Response Format

For the purposes of compatibility with this document, the JSON format of request and response SHALL be supported. In terms of this same interface, the ASPSP may support other formats defined by (2) (e.g. XML) and in such a case, this SHALL be clearly indicated in the ASPSP's interface documentation.

7.3 HTTP Response Codes and Additional Information About the Response

The ASPSP SHALL return not only the HTTP response codes (these codes have to be fully compatible with (2), in particular with the requirements defined in chapter 4.12), but also expanded information about status, in accordance with chapter 4.13.2 of (2).

7.4 Applying the Electronic Seal to the Request and Response

// deviation from the XS2A framework

For HTTP requests which are related to the API calls, the request initiator SHALL apply a qualified electronic seal similarly to chapter 4.2 of (2) (Signing Messages at Application Layer).

In calculating the hash code and creating the signature, the ASPSP as well as the TPP SHALL use only such algorithms which are accepted by the indicated standard.

The calculation of the hash code on the request body and the response body SHALL be carried out in accordance with the rule outlined in chapter 12.1 of (2) and the result SHALL be added to the header with the name "digest", as per the form defined by this same chapter.

For the purposes of compatibility with this document, new versions of the HTTP protocol signature format SHALL be used. In particular, (12) and, only appropriately - (13) SHALL be used.

For the purposes of compatibility with the following document, the 'date' field SHALL be entered into the list of attributes to be signed (the noted standards say SHOULD and not SHALL) in terms of request as well as response.

While applying a seal to the request, the seal SHALL cover the following headers:

1) Date
2) Content-Type (in case of data transfer)
3) Content-Length (in case of data transfer)
4) X-Request-Id
5) All headers with the "PSU" prefix, which, in accordance with (2), are sent to the ASPSP
6) Special (request-target) pseudo-headers, calculated in accordance with (12)
7) Digest

While applying a seal to the response, the seal SHALL cover the following headers:

1) Date
2) Content-Type (in case of data transfer)
3) Content-Length (in case of data transfer)

4) X-Request-Id
5) Digest

The TPP SHALL write their certificate, coded in the base64 format, in the TPP-Signature-Certificate header, and ASPSP SHALL do the same in the ASPSP-Signature-Certificate header.

The requesting party SHALL apply the electronic seal to the HTTP request and the responding party SHALL apply it to the response. To achieve this, the sealing party SHALL create a Signature header in accordance with (12) (in the request or the response). In this header the following values SHALL be used:

1) In the request, the keyId SHALL be generated in accordance with the requirements of chapter 12.2 of (2) (the same requirement applies to the ASPSP as well).
2) The keyId, Algorithm, Headers and Signature fields SHALL be included in the Signature header.
3) The Signature field in the Signature header SHALL be calculated in accordance with (12).

On the electronic seal, the parties SHALL NOT use a qualified timestamp because this, on one hand, increases the message signatory's dependence on outside services and, on the other hand, forces the message receiver to cross-check the correctness of the qualified timestamp.

Each participant of the system (ASPSP and TPP) SHALL adhere to time synchronization with the Coordinated Universal Time (UTC).

Each participant of the system (ASPSP and TPP) SHALL test the time field value in the received message before processing this message and cross-check it with the time in their own system. In cases where the time indicated in the received message is greater than the time in the participant's system and the time difference is greater than 2 (two) seconds, the following rule applies: the message SHALL NOT be processed.

8 Payment Initiation Service

The Payment Initiation Service is not regulated by this version of this document and it will be defined in future versions.

9 Account Information Service

In terms of account information services, (2) differentiates between card and other account information resources. For the purpose of compatibility with this document, account services (see (2), chapter 4.11.2) SHALL return information on those accounts also to which one or several plastic cards (debit or credit) are tied. The rule for returning card account information is regulated by chapter **შეცდომა! კავშირის წყარო არ მოიძებნა.** of this document.

This document differentiates between accounts available and accessible for open banking. In particular:

- An available account is an account which can be used through the XS2A interface, in accordance with Georgian legislation;

- An accessible account is such an available account for the usage of which there is any kind of confirmation given in terms of any kind of information distribution and this consent is active.

9.1 User Consent to Account Information Sharing

Only two scenarios of consent expression defined in chapter 6 of (2) are compatible with this document: detailed consent and consent offered by the bank. Scenario descriptions are provided in the subchapters. Global consent (which, for informational purposes, is described in chapter 9.1.2) is not compatible with this document and SHALL NOT be used.

Before reaching any concrete account/accounts, the TPP SHALL create a consent document, in which there will be provided a detailed description of what issue the PSU needs to consent to (some formats of the consent document are provided in the following subchapters of this chapter), and SHALL hand it over for registration to the ASPSP in accordance with the rule defined in chapter 9.1.7. The following actions are outlined in the aforementioned chapter.

The consent receipt process is as follows:

1. The TPP creates a consent object needed for consent registration (see below), as is described in the subchapters of the following chapter, as well as chapters 14.16 and 6 of (2).
2. The TPP requests the consent registration service from the ASPSP and transfers the consent object over to the ASPSP. The rule for this transmission is described in chapter 6.3.1.1 of (2) as well as the subchapters of the following chapter.
3. The ASPSP registers the consent object with itself and issues a registration identifier for it, as well as automatically makes one of the following decisions:
   3.1. Automatic rejection.
   3.2. Automatic approval.
   3.3. Strong customer authentication and consent request approval.
4. If the ASPSP's decision is automatic consent, the TPP can use the consent object identifier (received from the ASPSP) in account information sharing services (see subchapters corresponding to this chapter).
5. If the ASPSP's decision is strong customer authentication and consent request, they return a response to the TPP.

9.1.1 Detailed Consent Request Document

This scenario takes into account that the PSU has the ability to transfer account numbers to the TPP for any reason (e.g. the TPP received account numbers on the basis of previous consent, the TPP entered them manually, scanned a QR code containing the account number, etc.).

With the participation of the PSU, the TPP creates the consent document which could have, for example, the following format:

```
{
```

```
   "access":{
      "accounts":[
         {
            "iban":"GE00UT0000000101904917"
         },
         {
            "iban":"GE00UT0000000101904918"
         }
      ],
      "balances":[
         {
            "iban":"GE00UT0000000101904919"
         }
      ],
      "transactions":[
         {
            "iban":"GE00UT0000000101904920"
         }
      ]
   },
   "frequencyPerDay":5,
   "recurringIndicator":true,
   "validUntil":"2020-09-10"
}
```


The construction of this document falls completely under the TPP's functions. After this, the TPP requests the ASPSP's /v1/consents function (see chapter 6.3.1.1 of (2)) and transfers this object. The ASPSP registers the aforementioned request and returns different data, including consentId, which is necessary for the creation of authorization resources (see 10.1), for the following purpose of strong authentication.

When such a consent requires strong customer authentication, it SHALL happen on the ASPSP's side.

The ASPSP SHALL only show the consent object to the PSU visually and without the ability to edit it. If the PSU does not agree to receiving such a right, they have to clearly deny the transmission of this right.

The ASPSP SHALL keep the fact of consent expressed by the PSU in their system.

9.1.2 Global Consent Request Document

This is similar to the “detailed consent”, but the PSU gives consent to the TPP globally, concerning all information on all accounts, the TPP, on their side, creates the consent document which is transferred to the ASPSP for registration. The following process is managed by the ASPSP. In particular, they make the user go through strong authentication and afterwards, request that the user express general consent to transferring information to the TPP. For example:

```
{
    "access":{
        "allPsd2":"allAccounts"
    },
    "frequencyPerDay":3,
    "recurringIndicator":true,
    "validUntil":"2021-11-10"
}
```

The technical description of such consent expression is outlined in the “Consent Request for Access to all Accounts for all PSD2 defined AIS – Global Consent” paragraph in chapter 6.3.1.2 of (2). There is also an `"allPsd2":"allAccountsWithOwnerName"` variant, which means that the consent is given for returning the account owner’s name as well.

For the purposes of this document, the TPP SHALL NOT send the global consent request document and in the case when the ASPSP receives such a document from one of the TPPs, the ASPSP SHALL return an error.

9.1.3 Initiation Document for Consent Expressed in the ASPSP

In this scenario, the process of receiving consent from the PSU is fully carried out between the PSU and the ASPSP.

During the initiation of the scenario, as in all other cases, here too, the TPP begins, but as opposed to detailed consent (see chapter 9.1.1), in the JSON document which is transferred to the ASPSP, in the “access” attribute, “accounts”, “balances”, and/or “transactions” sub-attributes will be transferred, in a manner in which all transferred will include an empty array (“[]”), which will be a sign to the ASPSP that the TPP does not know exactly which accounts they require consent for from the PSU.

For example:

```
{
    "access":{
        "accounts":[
        ],
        "balances":[
        ],
```

```
        "transactions":[
        ]
    },
    "frequencyPerDay":12,
    "recurringIndicator":true,
    "validUntil":"2020-10-15"
}
```

The ASPSP SHALL request strong customer authentication.

The ASPSP SHALL request from the PSU detailed or general consent for such data for which the TPP transferred an empty array. The ASPSP SHALL NOT make general consent mandatory – they have to allow the PSU not to express consent for any of the accounts.

The ASPSP SHALL keep the fact of consent expressed by the PSU in their system.

In cases where one of the sub-attributes – “accounts”, “balances” and/or “transactions” – includes a non-empty array but a specific list (see 9.1.1), the ASPSP SHOULD not allow the PSU to change the corresponding data in the process of expressing consent.

The TPP may find out about the consent later, through a corresponding service request. The technical description of such consent expression is outlined in the “Consent Request Without Indication of Accounts – Bank Offered Consent” paragraph in chapter 6.3.1.2 of (2).

9.1.4 Consent Document for Requesting Information on All Available Accounts

The following constitutes a special scenario wherein the TPP needs to receive access to the list of all available accounts of the PSU (except balances and transactions). This scenario SHALL be supported.

```
{
    "access":{
        "availableAccounts":"allAccounts"
    },
    "recurringIndicator":false,
    "validUntil":"2017-08-06",
    "frequencyPerDay":"1"
}
```

The ASPSP MAY additionally support the following modifications of this scenario.

- `"availableAccounts":"allAccountsWithOwnerName"` – All available accounts and the owner name for each one. This issue is additionally regulated by chapter 9.1.5.2.
- `"availableAccountsWithBalance":"allAccounts"` – All available accounts and their balances.
- `"availableAccountsWithBalance":"allAccountsWithOwnerName"` – All available accounts, their balances and owner names. This issue is additionally regulated by chapter 9.1.5.2.

The availableAccounts request SHALL be supported, with the allAccounts value.

For the purposes of this version of this document, the value 'false' SHALL be indicated in the combinedServiceIndicator field, because the payment initiation service is not defined in this version.

9.1.5 The Rule for Requesting the Sharing of Additional Information

In the information request consent document, (2) supports the additionalInformation sub-attribute in the 'access' attribute. The issue of this attribute's usage, as well as the usage of some of its alternatives, is defined in accordance with the rule outlined in the subchapters of this chapter.

The account owner name request SHOULD be accompanied with clearly stated consent. For example, if the ASPSP receives consent for the available accounts list in the form of `"availableAccounts":"allAccounts"` (as defined by chapter 9.1.4), they may not return the account owner's information to the TPP.

9.1.5.1 Previously Clearly Defined Account Owner Name

The ability to request the account owner's name (the ownerName field) SHALL be supported, as is defined by chapter 14.17 of (2).

9.1.5.2 Account Owner Name During Requesting Available Accounts

Variants with the suffix WithOwnerName (e.g. allAccountsWithOwnerName) in the availableAccounts, availableAccountWithBalance and allPsd2 attributes SHOULD be supported for the purpose of heightened compatibility with (2) as well as increasing the user's comfort.

The request of the ownerName field SHOULD be accompanied by clearly stated consent and in case of absence of such consent, the ASPSP SHOULD NOT return this information to the TPP.

9.1.5.3 List of Trusted Beneficiaries

For the purposes of compatibility with this version of the document, the ASPSP SHALL NOT support the sharing of the list of trusted beneficiaries in accordance with the rule defined in chapter 14.17 of (2) (attribute name "trustedBeneficiaries") and should the TPP request consent for this, the ASPSP SHALL return the corresponding error code.

9.1.6 Consent Expressed for Multi-Currency Accounts

The consent expressed for a multi-currency account SHALL include it as one account. The ASPSP SHALL transfer information to the TPP, in accordance with this document, in a way that it will be possible to identify the account as a multi-currency one, and it will also be possible to conclude which currency in particular can be used in this multi-currency account.

The ASPSP MAY support the “fragmentation” of the multi-currency account in a way that will allow the TPP to receive the PSU’s consent corresponding to a specific currency in connection with the multi-currency account. This includes, for example, indicating the currency code during the request of detailed consent (see chapter 9.1.1).

If the ASPSP supports the “fragmentation” of the multi-currency account, consent expressed towards one currency SHALL NOT imply consent towards another currency.

If the ASPSP does not support the “fragmentation” of the multi-currency account, during the request of detailed consent for the account (see chapter 9.1.1), the ASPSP SHALL make the PSU express consent regardless of whether the ASPSP received the account’s currency together with the account number.

9.1.7 Consent Request Service

The service is described in chapter 6.3.1 of (2). This service SHALL be supported for the purpose of compatibility with the presented document.

After the TPP creates the consent object and transfers it to the ASPSP, the ASPSP SHALL return one of the following responses:

1) Automatic rejection of the consent request.
2) Automatic approval of the consent request.
3) Demanding strong customer authentication.

In cases where the granting of the consent object requires consent expressed by several users (e.g. in a corporate context), for the purposes of this version of the presented document, the request SHALL be automatically rejected. So, for this version of the presented document, the ability defined by chapter 6.3.4 of (2) SHALL NOT be used. This will be supported in future versions.

9.1.7.1 Automatic Granting of the Consent Request

The ASPSP MAY directly return the consentStatus=valid value and not return ASPSP-SCA-Approach in cases where all of the following conditions are met:

1) The OAuth2 token is granted in Authorization, it is valid and with it, the PSU can be identified in accordance with the credential level defined by Georgian legislation, if such exists.
2) Georgian legislation grants the ASPSP the right to consider the presentation of such a token as consent expressed by the user for issuing accounts.

In any other case, for the purposes of this version of the presented document, the ASPSP SHALL NOT return consentStatus=valid and request strong customer authentication.

9.1.7.2 Consent Authorization on the Basis of Strong Customer authentication

In case the ASPSP needs strong customer authentication for the purpose of consent confirmation, the ASPSP SHALL return only the OAuth variant, in particular:

1) In the ASPSP-SCA-Approach field, REDIRECT SHALL be returned.

2) In the hyperlinks section (_links) the scaOAuth hyperlink SHALL be returned, pointing to the ASPSP's OAuth2 server metainformation, which, in its own right, SHALL be structured in accordance with (14).

In order for strong customer authentication to be carried out, the TPP needs to receive the "authorization resource". The "authorization resource" creation process will be fully directed by the ASPSP, through different configurations returned into the hyperlinks section. This is described in chapter 10.1 of this document. After the "authorization resource" is received, strong customer authentication and consent receipt SHALL happen in accordance with the rule outlined in chapter 10 of this document.

9.1.8 Consent Compliance with Requested Information

In cases where the consent is not enough, the request SHALL return CONSENT_INVALID and not a "lessend" document value (for example, if consent is not given for returning balances, the usage of the withBalance parameter has to result in the CONSENT_INVALID type of error).

9.1.9 Technical Consolidation of Consents

The ASPSP SHOULD not request strong customer authentication from the TPP based o the request if the TPP has already received several consents, the combination of which includes the consent requested by the TPP and if of all of these consents, each one's term and occurrence is equal to or higher than the term and multiplicity of the requested consent. In case the ASPSP has realized this functionality, the ASPSP SHALL grant a unique identifier to this consent just as they would grant one to any other consent.

9.1.10 Consent Management Services

The ASPSP SHALL support services defined by chapters 6.3.2, 6.3.3, and 6.4 of (2) (consent status, consent information, consent request).

9.2 Bank Account Information Services

9.2.1 Consent and Authorization

Each service SHALL receive Authorization and Consent-ID headers.

The Bearer token provided in the Authorization header SHALL be the same as the one used during the consent confirmation indicated in the Consent-ID header, if the said token is valid. If the token's validity has expired, then the Authorization header SHALL include a new token created as a result of extending the same token through the Refresh Token.

9.2.2 Account Number Tokenization

The ResourceID values (see chapter 14.19 of (2)) SHALL be tokenized with unique tokens so that the account numbers are not leaked from the URIs.

In the tokenization process, a one-sided algorithm SHALL be used so as to prevent the restoration of any of the characteristics of the account from the token. Usage of the UUID as a form of token entry is RECOMMENDED (see (15)).

9.2.3 List of Accounts

This service is described in chapter 6.5.1 of (2). For the purpose of compatibility with the presented document's current version, this service SHALL be supported.

If the identifier presented in the Consent-ID header indicates consent for the available accounts (see chapter 9.1.4), the ASPSP SHALL return all available accounts. In other cases, information SHALL be returned on accessible accounts for which consent was granted.

While returning information, the ASPSP SHALL NOT return that detailed information for the sharing of which consent was not expressed in the consent indicated in the Consent-ID header value. The following list indicates several examples:

1. The ASPSP SHALL NOT return the 'balances' elements for available accounts, if the granted consent type is not availableAccountsWithBalance (and is availableAccounts instead).
2. The ASPSP SHALL NOT return the 'transactions' elements for available accounts.
3. The ASPSP SHALL NOT return the 'balances' elements for an accessible account if the sharing of the balances of such an account was not clearly consented to (see chapters 9.1.1 and 9.1.3). The same rule applies to transactions and the account owner's name as well.
4. If the account is a multi-currency one and consent is given on the level of a multi-currency account, the currency "sub-accounts" as well as information about the multi-currency account itself SHALL be returned. The currency of the multi-currency account SHALL be provided as "XXX".
5. If the account is a multi-currency one and consent is given not for the entire account but only for one or several currencies within it, the multi-currency account information SHALL NOT be returned.
6. If consent exists, a hyperlink SHALL be returned for balances as well as transactions.

When, for the purpose of satisfying this chapter's requirements, the ASPSP is returning several entries corresponding to a multi-currency account, each one of them SHALL have the same value in the ResourceId field.

Examples of information to be sent to the ASPSP and the responses to be returned are provided in chapter 6.5.1 of (2).

9.2.4 Account Details

This service is described in chapter 6.5.2 of (2). For the purpose of compatibility with the presented document's current version, this service SHALL be supported.

Passing the ResourceID value returned by the transaction list (see chapter 9.2.3) in Account-ID parameter SHALL be supported.

If the account is a multi-currency one, the ASPSP SHALL return XXX in the currency.

The ASPSP SHALL NOT return the ‘balances’ hyperlink if the identifier issued in the Consent-ID header does not indicate consent for the sharing of such balances. The same applies to the ‘transactions’ hyperlink and additional information. If consent exists, the hyperlink SHALL be returned.

For those accounts in which the balance can be altered through electronic channels (including through card operations), the ‘balances’ element SHALL be returned to the account details, if the consent includes the sharing of balance information as well. For such accounts, at least the following balance types SHALL be returned inside the ‘balances’ element: interimAvailable, interimBooked.

9.2.5 Account Balance

This service is described in chapter 6.5.3 of (2). For the purpose of compatibility with the presented document’s current version, this service SHALL be supported.

Passing the ResourceID value returned by the transaction list (see chapter 9.2.3) in Account-ID parameter SHALL be supported.

If the account is a multi-currency one, the balance SHALL be returned in each currency defined in that multi-currency account.

At least the following balance types SHALL be returned: closingBooked and interimAvailable.

For the purpose of supporting future versions of (2), the ASPSP SHOULD return the ‘account’ component.

9.2.6 List of Account Transactions

This service is described in chapter 6.5.4 of (2). For the purpose of compatibility with the presented document’s current version, this service SHALL be supported.

Passing the ResourceID value returned by the transaction list (see chapter 9.2.3) in Account-ID parameter SHALL be supported.

If the account is a multi-currency one, the ASPSP SHALL return transactions in a currency for all “sub-accounts”.

The request of the transaction list with the dateFrom and dateTo parameters SHALL be supported.

The case of requesting the transaction list where only the dateFrom parameter is used SHOULD be supported.

The case of requesting the transaction list where the dateFrom parameter is not used and only the entryReferenceFrom parameter is used SHOULD be supported, and as a response to such a request, the ASPSP will return transactions which happened after the transaction indicated by entryReferenceFrom (the transaction, where, in its detailes, the entryReference would be equal to this parameter, see chapter 9.2.9).

The ASPSP SHALL support the dateTo parameter (requesting transactions up to the indicated date), but the transfer of this parameter during the request should be OPTIONAL so that the TPP can requested before the moment of the service request.

The 'booked' value in the bookingStatus parameter SHALL be supported during the service request. The 'pending' value in the bookingStatus parameter SHALL be supported during the service request for such an account whose balance can be altered through electronic channels, online (including through card operations). The 'both' and 'information' values are OPTIONAL.

If consent is expressed for sharing balances and bookingStatus=information is not requested, the 'balances' section SHALL be formed in accordance with the following rule:

1) If, as bookingStatus, "booked" or "both" is requested (if supported), the openingBooked and closingBooked types of balances SHALL be returned.
2) If, as bookingStatus, "pending" or "both" is requested (if supported), the interimBooked and interimAvailable types of balances SHALL be returned.
3) Each balance SHALL reflect information on the presented portion (and not the entire statement).

9.2.7 Transaction Information Pagination

If the amount of transactions to be returned is greater than 50, the ASPSPs SHALL return responses in a divided format. The TPP and the ASPSP may agree, on the basis of a contract, to annul such a request or alter the limit.

The ASPSP SHALL support at least the following scenario for the return of the response in a divided format:

1) There should be 50 or less transactions in a portion.
2) For the first portion, 'first' and 'next' types of hyperlinks are indicated in the hyperlinks section (_links).
3) For the interim (neither first nor last) portion, 'first' and 'next' hyperlinks are indicated in the hyperlinks section.
4) For the last portion, only the 'first' hyperlink is indicated in the hyperlinks section.
5) The 'first' hyperlink always indicates exactly the same address which was used to request the transactions list.

In the 'next' hyperlink which they return to the TPP, the ASPSP SHOULD transfer the entryReferenceFrom parameter to their own service.

9.2.8 Informational "Transactions" Support

Supporting standing payment orders or other informational transactions defined in (2) for the TPP (the bookingStatus=information parameter is OPTIONAL for the purposes of this version of the presented document and the volume of the returned information is not regulated by this document). While supporting this parameter, the ASPSP SHALL fully abide by (2). But specific transactions created in terms of standing payment orders SHALL be given to the TPP in the same form which is observed for any other transaction in the account (in cases of bookingStatus=booked, pending, or both).

9.2.9 Minimal Information Requirements for Account Transactions

For any transaction, the ASPSP SHALL return through the service at least the following fields from the list defined in chapter 14.24 of (2):

Name	Type	Necessity	Comment
entryReference	Max35Text	REQUIRED	Transaction identifier which the TPP will appropriately give to the PSU and the usage of which will also be possible for the partial return of information if it is supported by the ASPSP
transactionAmount	Amount	REQUIRED	Transaction sum and currency, formatted in accordance with the rule defined in chapter 14.3 of (2) (a debit transaction is marked with a negative value, a credit transaction – with a positive value)
valueDate	ISODate	REQUIRED	The day when: 1) The sum became accessible to the account owner (in case of a credit transaction) or 2) Stopped being accessible (in case of a debit transaction)
bookingDate	ISODate	Conditional	The day when the transaction was reflected in the balance. In the account transaction list (“booked” list), this field SHALL be returned.
currencyExchange	ReportExchangeRate ტიპის მასივი	Conditional	If the transaction was conducted in a different currency, this value SHALL be returned
remittanceInformationUnstructured	Max140Text	Conditional	The ASPSP SHALL return one of these two values if such information exists in their system and it is accessible to the PSU in any other way. If the ASPSP and the TPP have not agreed otherwise, the ASPSP SHALL return remittanceInformationUnstructur
remittanceInformationUnstructuredArray	Max140Text ტიპის მასივი	Conditional	

			edArray, so as to fully reflect the information protected in their system. If there is an agreement with the TPP, the ASPSP has the right to return the remittanceInformationUnstructured value and make the entire information accessible in accordance with chapter 9.2.10.
additionalInformation	Max500Text	Conditional	The ASPSP SHALL return this information if it exists in their system and is accessible to the PSU in any other way.
transactionId	String	Conditional	SHALL be returned only in cases where the returning of additional information happens with a separate request (e.g. the ASPSP transmits remittanceInformationUnstructured)
_links	Links	OPTIONAL	The ASPSP MAY return the transactionDetails hyperlink type so that it is easier for the TPP to have access to transaction details in accordance with chapter 9.2.10.

Table 2: Fields in Transaction Information

Every other field which is defined in (2) is OPTIONAL if not returning them is not against Georgian legislation. The ASPSP SHOULD return all such fields which they make accessible to the PSU using other channels.

9.2.10 Transaction Details Service

This service is described in chapter 6.5.5 of (2). The support of this service is OPTIONAL if the ASPSP fully returns the information defined in chapter 9.2.9 (e.g. if remittanceInformationUnstructed and not remittanceInformationUnstructuredArray is being returned). If the information is not fully returned by the ASPSP (these rules are also defined in chapter 9.2.9), then this service SHALL be supported.

9.3 Card Account Information Services

Card account services are provided in chapter 6.6 of (2). For the purposes of compatibility with the presented document as well as the users' comfort, these services SHOULD be supported.

Accordingly, all necessary requirements outlined in the subchapters of this chapter are only necessary if the card account services are supported by the ASPSP.

In case of supporting these services, the ASPSP SHALL support access to these accounts in the form of access to regular bank accounts as well (see 9.2).

9.3.1 General Outline of Card Account Access

Since, for the presented document's current version, (2) does not support the existence of several cards for one card account, for the purpose of preserving compatibility, the following requirements SHALL be supported in card account (card-accounts) services (see chapter 6.6 of (2)):

1. In card account services, the ASPSP MAY not rely on the consent expressed for the bank account identified through an IBAN and request consent separately for cards associated with this bank account. If the ASPSP chooses this path, the following requirements SHALL be satisfied:
   1.1. The ASPSP SHALL return card information while returning the "available accounts list" (during which they will return the reference with type maskedPan);
   1.2. In case of card accounts, the ASPSP SHALL support each form of consent expression which they support for bank accounts. In the process of managing the consent, the ASPSP SHALL show the card number to the PSU in a masked manner so that it is not difficult for the PSU to identify the specific card and make an informed decision about approval or denial;
   1.3. After "consent expressed in the bank", the ASPSP MAY transfer card numbers to the TPP not in a masked manner (maskedPan), but in a tokenized form (pan). Although, in this case, the ASPSP SHALL ensure the effective sharing of this tokenized value and its unambiguous connection with a specific card to the PSU (e.g. by showing it in the internet-bank) during the term of the corresponding consent's validity;

2. The value of the resourceId field (and, accordingly, the account-id parameter) used in card account services SHALL uniquely point to a specific card and stay unchanged, at least during the term of validity of the active consent expressed for this card. This value SHALL NOT match the tokenized identifier of the corresponding bank account in the bank account access (accounts) services.

3. The value which will be used as resourceId/account-id, SHALL NOT reveal the card number (Primary Account Number, PAN). It SHALL be tokenized in accordance with modern standards and best practices for card data in a way that, simultaneously, the requirements of (2) are not violated.

4. // deviation from the XS2A framework
   Each card account's information SHALL return in the hyperlinks section (_links) with the "account" indicator type, which will point to the details of the corresponding bank account (see chapter 6.5.2 of (4)). The account identifier SHALL be tokenized and the token SHALL remain unchanged, at least during term of validity of consents expressed for both the card account and the bank account itself. In cases where the card is attached to a multi-currency bank account, the hyperlink SHALL return information on the multi-currency account. The hyperlink SHALL be returned regardless of whether or not the user has expressed consent for this specific account (e.g. in cases where the ASPSP

differentiates between consent expressed for the account through the IBAN and consent expressed for the card).

9.3.2 Rule for Returning Information on Card Accounts

For the purpose of compatibility with this document, each requirement defined by chapter 9.2 for bank accounts also applies to card accounts.

10 Strong Customer Authentication and Expressing Consent

10.1 The Concept and Creation Process of Authorization Resources

An authorization resource is a hyperlink which expressly describes the unit for which consent is given.

- For the purposes of this version of the presented document, this unit is the consent request document (for sharing account information), which is already registered with the ASPSP in accordance with the rule outlined in chapter 9.1.7.
- In future versions of the presented document, other units (e.g. single payment, payment cancellation, etc.) will be added gradually.

When the consent expression unit (e.g. consent request document for sharing account information) is registered in the ASPSP's system, the ASPSP SHALL direct the authorization resource creation process through the hyperlinks section (_links).

In particular, for the purposes of this version of the presented document, the ASPSP SHALL support only the following variants in the hyperlinks section:

1) SHALL return the scaStatus hyperlink type (*remark: do not confuse with the scaStatus attribute!*)

   This means that the ASPSP has all appropriate information to begin strong customer authentication and scaStatus includes the authorization resource hyperlink directly. After receiving this hyperlink, the TPP SHALL directly begin the OAuth authorization process.

2) SHALL return the startAuthorization hyperlink type.

   This means that the ASPSP has all appropriate information to begin strong customer authentication, but the creation of the authorization resource has to be clearly requested by the TPP. Authorization resource creation is described in chapter 10.2. after successfully creating the authorization resource, the TPP SHALL directly begin the OAuth authorization process.

3) SHALL return the startAuthorisationWithAuthenticationMethodSelection hyperlink type.

   This means that the ASPSP could not automatically decide which type of authentication to make the customer go through, the TPP has to request the explicit creation of the authorization resource, during which the ASPSP will return the selectAuthenticationMethod hyperlink (and the

corresponding additional information). Authorization resource creation is described in chapter 10.2, the rule for processing the selectAuthenticationMethods hyperlink is described below.

4) SHALL return the selectAuthenticationMethods hyperlink type.

   In this case, the ASPSP SHALL return, in the same response, a non-empty scaMethods array and the TPP SHALL offer the user to select one of the methods, and the selected method SHALL be transmitted to the selectAuthenticationMethod in the manner that is described in chapter 7.2.3 of (2).

   If the authentication method selection is successful, after this is finished, the ASPSP SHALL return the scaStatus hyperlink and the OAuth2 server parameters, as is outlined in the presented chapter, and the TPP SHALL directly begin the OAuth authorization process.

   This method SHOULD be used instead of returning the startAuthorisationWithAuthenticationMethodSelection hyperlink type.

The authentication method selection through the selectAuthenticationMethods hyperlink (with the startAuthorisationWithAuthenticationMethodSelection interim stage or directly) MAY be used by the ASPSP in order to transform general customer authentication (which is depicted with the OAuth2 token provided in the Authorization header) into strong customer authentication, as is described in chapter 6.3 of this document, particularly, return such parameters of the OAuth protocol which ensure authentication only with the second factor. Although, to achieve this, the ASPSP MAY use some other method which is compatible with the requirements of this chapter.

The ASPSP SHALL NOT return other hyperlinks of the startAuthorisation* type other than the one described in the presented chapter.

10.2 Authorization Resource Creation Service

If the authorization resource was not automatically created during the registration of the unit for which consent has to be expressed (e.g. registration of the consent request document for account information sharing), which is indicated in the scaStatus hyperlink type returned in the hyperlinks section, the TPP SHALL create the authorization resource by an explicit request.

The authorization resource creation service is described during the 7.4 service call in (2) (supporting this service and the usage of the explicit creation approach to authorization resources is RECOMMENDED). In particular, the ASPSP SHALL return the address of this service through the startAuthorisation hyperlink type.

1) startAuthorisation – if the request was carried out successfully, the authorization resource SHALL be created and returned to the TPP.

2) startAuthorisationWithAuthenticationMethodSelection – if the request was carried out successfully, the ASPSP SHALL return the selectAuthenticationMethods hyperlink type and a non-empty scaMethods array in the response to the TPP. The rule for processing such cases is described in chapter 10.1.

10.3 Expressing Consent after Strong Customer Authentication

Requirements for strong customer authentication are provided in chapter 6.2 of this document. After the customer goes through authentication, the ASPSP SHALL request that the customer express consent and the minimum consent requirements are provided in this chapter. The ASPSP MAY broaden these requirements at their own discretion.

10.3.1 Expressing Consent to Account Information Sharing

If, during the request of consent for account information sharing, the TPP provided specific account numbers or masked card numbers (thus, requested detailed consent, as defined in chapter 9.1.1), the ASPSP SHALL show the PSU all corresponding accounts and cards for which consent is expressed.

If the ASPSP supports card account services (see chapter 9.3) and the TPP has provided a tokenized card number in the consent expression process, the ASPSP SHALL transform it into a masked number and show it to the PSU in this way so that the PSU can clearly see which card's corresponding account they're giving consent for to the TPP.

If the TPP has provided a consent request for receiving all available accounts (see chapter 9.1.4), the ASPSP SHALL inform the PSU (in the form of information popping up on the screen) about all of the accounts and cards whose information will be shared with the TPP.

If, in the process of expressing consent, the TPP requests access to account balances or transactions (e.g. during requesting detailed consent), the ASPSP SHALL show the account balance (current balance or balance at the beginning of the current operational day) to the PSU during the process of consent expression.

11 Sources

1. **The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface.** NexGenPSD2 XS2A Framework, Operational Rules. Version 1.3 December 21, 2018.

2. —. NexGenPSD2 XS2A Framework, Implementation Guidelines. Version 1.3.6 February 3, 2020.

3. **Bradner, Scott.** *RFC 2119 - Key words for use in RFCs to Indicate Requirement Levels.* March 1997.

4. **ETSI.** TS 119 495 - Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Qualified Certificate Profiles and TSP Policy Requirements under the payment services Directive (EU) 2015/2366. ETSI TS 119 495.

5. ***RFC 8705 (OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens).***

6. ***RFC 7591 (OAuth 2.0 Dynamic Client Registration Protocol).***

7. ***OpenID Connect Dynamic Client Registration 1.0.***

8. ***RFC 7592 (OAuth 2.0 Dynamic Client Registration Management Protocol).***

9. ***RFC 5755 (An Internet Attribute Certificate Profile for Authorization) .***

10. ***Financial-grade API - Part 1: Read-Only API Security Profile.***

11. The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface. ***NextGenPSD2 XS2A Framework, Extended Services, Resource Status Notification Service.*** **მარტი 1, 2019.**

12. Signing HTTP Messages, draft-ietf-httpbis-message-signatures-00. [Online] https://tools.ietf.org/html/draft-ietf-httpbis-message-signatures-00.

13. Signing HTTP Messages, draft-cavage-http-signatures-12. [Online] https://tools.ietf.org/html/draft-cavage-http-signatures-12.

14. ***RFC8414 (OAuth 2.0 Authorization Server Metadata).***

15. ***RFC 4122 (A Universally Unique IDentifier (UUID) URN Namespace).***