

FIDO for PSD2

Providing for a satisfactory customer journey

September, 2018

1 Introduction

When PSD2 is deployed in Europe, users will be able to take advantage of services offered by Third Party Providers (TPPs) to trigger payments or to view account information. These users will typically start interacting on the TPP's user interface. However, at the point when a TPP will request from an Account Servicing Payment Service Provider (ASPSP) access to a user's account(s), the PSD2 Regulatory Technical Standards (RTS) for Strong Customer Authentication (SCA) require that the user be strongly authenticated by the ASPSP and demonstrate that he/she has provided consent for the operation that the TPP is requesting to execute.

The Strong Customer Authentication requirement introduces challenges in the customer experience as there are no longer just two parties involved, the user and its bank, but three: The end user journey starts and ends on the TPP's user interface.

TPPs will interface with the ASPSPs via open APIs. A number of standardization bodies have released drafts of such Open APIs, for example the Open Banking Implementation Entity (OBIE) in the UK, STET in France and the Berlin Group for various European countries.

These specifications describe how Strong Customer Authentication should be implemented and several models have been defined, if not (yet) fully specified: the redirection, decoupled and embedded models. At the time of this paper's release, a potential delegated model is also being discussed. These models vary in the way the user interacts with the TPP and the ASPSP and have a deep impact on both the user experience and the security of the user's financial accounts.

This paper examines the advantages and drawbacks of the different SCA compliant authentication models and outlines how FIDO compliant solutions deliver the best user experience in any of these models, in way that meets the needs of TPPs and ASPSPs.

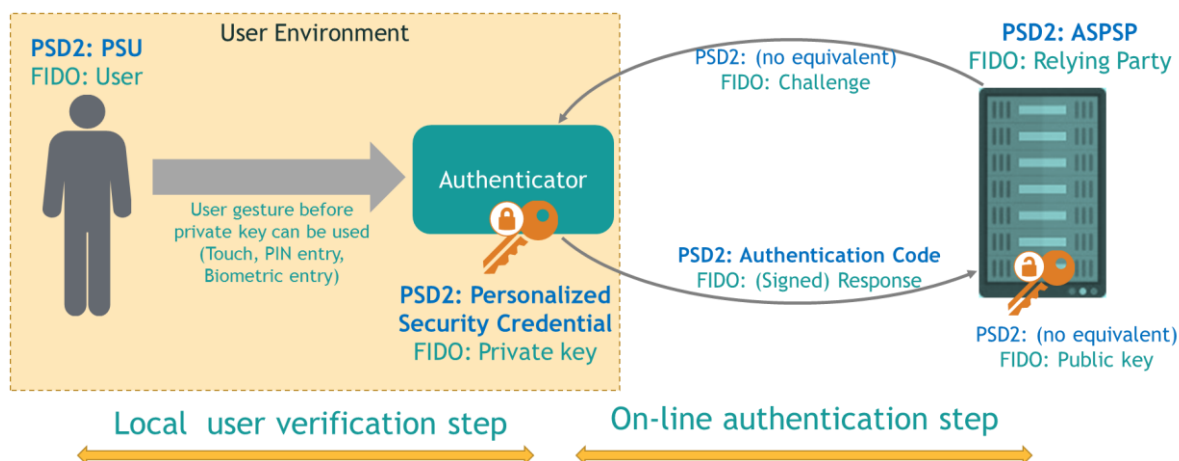
2 Glossary of terms

AISP	Account Information Service Provider. For example, a provider of account aggregation services.
ASPSP	Account Servicing Payment Service Provider. Typically, the bank holding the accounts
eIDAS	Electronic Identification And trust Services (REGULATION (EU) No 910/2014). Referred to in the RTS for the provision of qualified certificates to identify TPPs
IDP	Identity Provider in a federated identity ecosystem
OTP	One Time Password
PISP	Payment Initiation Service Provider
PSU	Payment Service User. The user providing consent to a TPP to access its accounts
RTS	Regulatory Technical Standard. In the context of this white paper, RTS refers to the RTS on Strong Customer Authentication and Common and Secure Communication
SCA	Strong Customer Authentication
TPP	Third Party Provider: an AISP or a PISP
XS2A	Access to Account (for the purpose of initiating a payment or retrieving account information)

3 The basics of FIDO Authentication

3.1 FIDO authentication

The figure below illustrates the basic two step user authentication mechanism provided by the FIDO standards. The figure also maps PSD2 terminology with terminology used in the FIDO standards:



How FIDO Works

To authenticate with FIDO, the Payment Service User (PSU) must have a FIDO authenticator that can either be integrated in a general purpose device (e.g. Smartphone, Laptop) or be a separate device (e.g. Security Key, smart card).

User verification

The first step of FIDO authentication is the user verification step that is performed off-line, locally, by the authenticator. This user verification step can be:

- A verification of user presence whereby the user makes a pro-active gesture with the authenticator (for example, touches a security key or taps an NFC card on a reader).
- The verification of a PIN code or of biometric data by the authenticator. In this case, the local user verification constitutes one of the authentication factors mandated by the RTS.

The local user verification step is a pre-requisite for the on-line authentication step.

On-line authentication








The on-line authentication step proves the possession of the FIDO authenticator and constitutes a second factor of authentication mandated by PSD2. In this step, the ASPSP server sends a message to the authenticator which is then cryptographically signed by a private key stored in the authenticator. The signed response is returned to the ASPSP and its positive verification serves as proof of possession.

The FIDO standards are based on public key cryptography. The private key is the Personalized Security Credential described in the RTS. It is part of a key pair randomly generated by the Authenticator itself and is not known to any other party. At the generation time, the associated public key is sent in a protected way to the ASPSP.

The Authenticator maintains dedicated Personalized Security Credentials (private keys) for each ASPSP. For example, if the PSU has accounts at ASPSP1 and ASPSP2, the Authenticator would store different Personalized Security Credentials for ASPSP1 and ASPSP2, each being restricted for use with the respective ASPSP.

3.2 Authenticators

FIDO authenticators exist in several implementations and are classified as shown in the table below:

	Bound authenticators	Roaming authenticators
Multi Factor authentication (possession + knowledge/inherence)	 PC with TPM & biometric capture  Smart phone with TEE & biometric capture	 Smart card with PIN or fingerprint sensor  Security key with PIN or fingerprint sensor
2 nd factor (Login & Password + possession factor)	 PC with TPM only	 Smart card  Security key

Example of FIDO authenticators

Deployment and reach

The reach of the SCA solution is a key aspect of PSD2 compliance: the mandate for a possession factor requires that ASPSPs deploy devices to all of their users. This may mean that multiple devices have to be deployed and supported by the ASPSP’s authentication server.



Multi-channel/multi-device based authentication

The power of FIDO standards - and the value they offer to ASPSPs and TPPs - is that they ensure that a FIDO certified application will securely interoperate with any FIDO certified device, whichever its form factor, at a cost point that reflects an open, competitive, interoperable marketplace of standards-compliant commercial products.

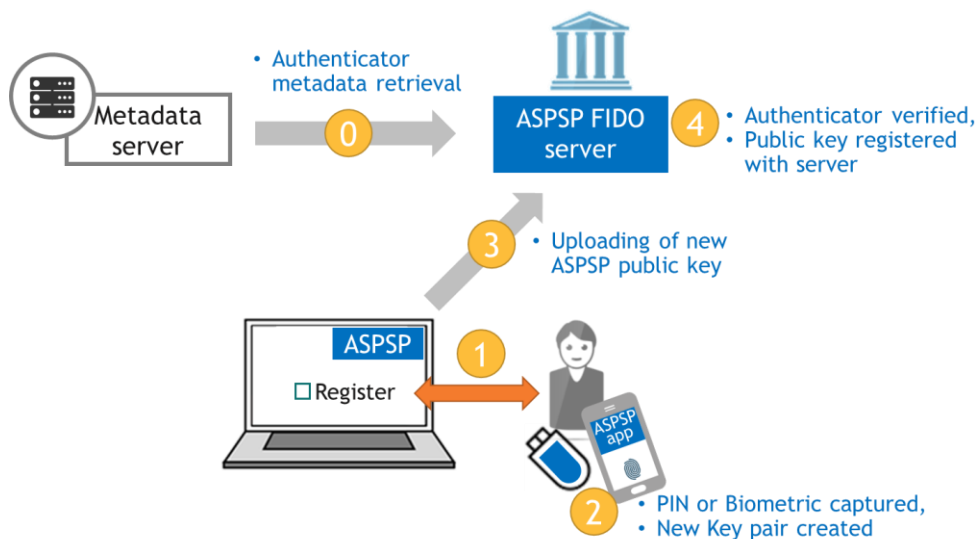
3.3 Registration

When a user registers with an ASPSP, he/she may use the FIDO authenticator provided by the ASPSP but also use an authenticator already in his/her possession. Note that many smartphones and laptops ship from the factory with FIDO authenticators already built in, making FIDO authentication a natural, low-friction approach to fulfill SCA requirements. For example, the FIDO Alliance has recognized Windows 10 and Android as reference implementation platforms compliant with FIDO standards, and Chrome, Firefox, and Edge as reference implementation web browsers compliant with FIDO standards.

As part of the registration process, the ASPSP will verify that the user’s authenticator is genuine and matches its policy (for example in terms of biometrics supported, of biometrics accuracy, of security environment, etc.). This is achieved by means of retrieving, from a trusted provider, the characteristics or “metadata” of the authenticator.

Authenticator characteristics may be available to ASPSPs through Metadata servers such as FIDO’s public MDS server (see <https://fidoalliance.org/mds>). This is a free, open, global registry of all FIDO certified authenticator metadata made available to all FIDO compliant applications so that they may build risk-based policy into their implementation, e.g. setting a higher level of trust when the device is protecting the FIDO private keys with a restricted operating environment.

Upon user registration, the authenticator will generate a personalized security credential (public/private key pair) specific to the ASPSP and the public key will be uploaded to the ASPSP’s FIDO server:



FIDO Registration

3.4 Authentication code

With FIDO, the Authentication Code described in the RTS is provided by the signed response that the FIDO authenticator calculates upon receiving a challenge message from the ASPSP FIDO server. It is computed using the Personalized Security Credential in a way that it can be verified with the public key. The Authentication Code can only be generated by the Authenticator - never by any other party.

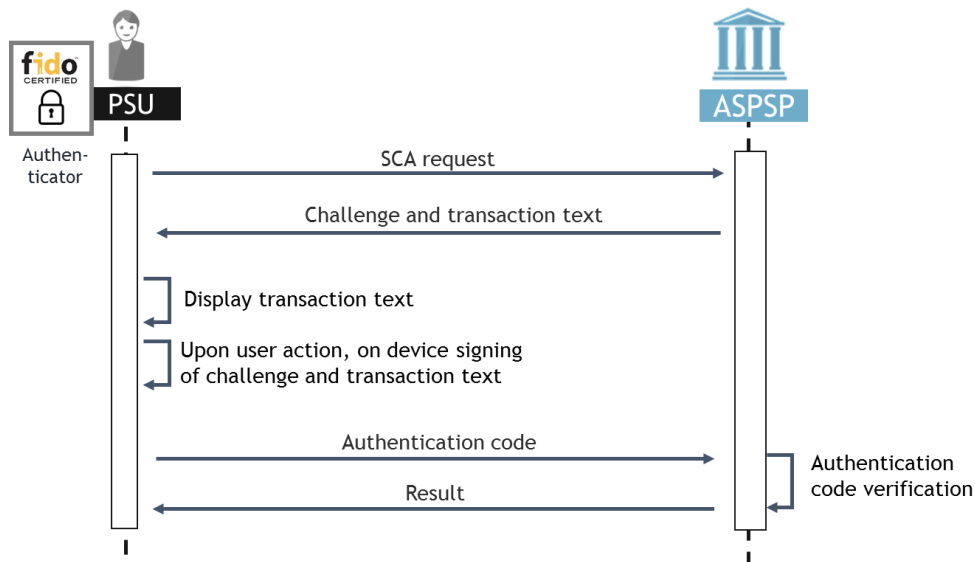
The verification of the authentication code by the ASPSP proves the possession of the device, i.e the FIDO authenticator. Moreover, if the device is a multifactor authenticator, the signed response will only be

generated upon a positive local user verification. The verification of the authentication code by the ASPSP will thus prove the possession factor as well as the knowledge or inherence factor, as mandated in Article 4 of the RTS.

Dynamic linking

For remote payments, the RTS require that the transaction amount and payee be dynamically linked to the authentication code. The FIDO standards support this requirement in two ways:

1. The message sent by the FIDO server can include the amount, payee ID and other data. The signed response will then cryptographically link this data to the authentication code.
2. FIDO Transaction Confirmation can be used, when supported by the authenticator: Such authenticators will be able to display the transaction text to the user and ask for user approval. Successful approval is securely indicated to the ASPSP. The ASPSP can cryptographically verify that the transaction text displayed to the user is identical to the original transaction text provided by the ASPSP. This concept implements the “What-you-see-is-what-you-sign” model.



Dynamic Linking with FIDO

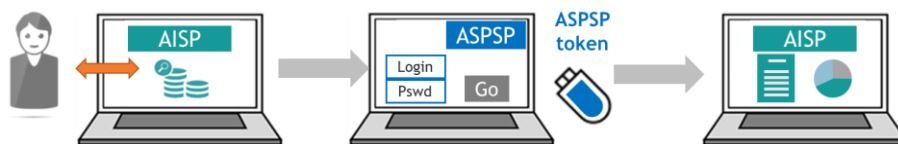
4 The different authentication models

The way the PSU is authenticated has a direct impact on the customer journey, when using the services of a PISP or AISP. It may also have an impact on the Open APIs and the interactions between TPP and ASPSP. This is the reason why API specification bodies such as the Berlin Group have analysed the customer journey and defined four different authentication models:

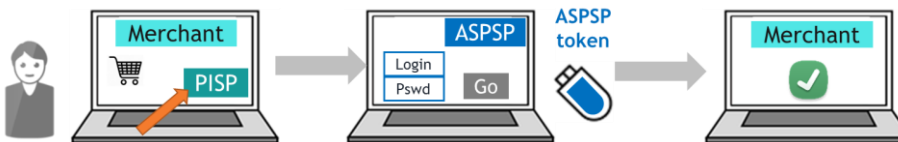
1. Redirection model
2. Decoupled model
3. Embedded model
4. Delegated model

4.1 The redirection model

The redirection model is an approach whereby the PSU starts interacting with a TPP and is redirected on the same device to a web interface of the ASPSP for authentication.



Example for an account aggregator



Example for a payment initiator

In this model, the ASPSP manages the authentication interactions with the PSU and handles the SCA autonomously. The Open APIs used by the TPP to interface with the ASPSP are not used for SCA operations.

App to app variant

When the customer journey is initiated on a mobile device, app-to-app redirection is a considered model, for example by the Open Banking Implementation Entity (OBIE) that are to specify this model for the UK Open APIs.



App to app redirection, on a smartphone

Advantages of the redirection model

Of all the models which ASPSPs and TPPs can choose from, the redirection model is the most secure and most proven for all parties involved. The key virtue of the redirection model is that the user is trained to only give their credentials to the service that registered those credentials in the first place. This avoids the pitfalls of training users to give their credentials to 3rd parties resulting in the unintended consequence of making them, and the financial ecosystem overall, more vulnerable to social engineering attacks.

- For ASPSPs, redirection offers the ability to be in full control of user authentication. The ASPSP can re-use the authentication method it provides to its users when they access their account directly with the ASPSP.

The ASPSP is also in control of its schedule and may implement its SCA solution as part of its own compliance plan, without dependence on other parties. Moreover, as the model is independent from other parties, it will work with any TPP connecting through the Open APIs.

- For TPPs, redirection offers the use of proven standards, as well as the ability to make clear to consumers which role is played by a TPP in a transaction and which role is played by the ASPSP.
- For users, they have the comfort and security of authenticating from the interface of the ASPSP, which they may be used to and find more trustworthy.
- While other models under consideration may prove both secure and commercially viable in the future, the redirection model is already recognized as an industry best practice easily deployed quickly at scale and built on a suite of well-established public industry standard protocols.

The currently published APIs, for example from the Berlin Group, OBIE or STET support the redirection model.

4.2 The decoupled model

The user experience of the decoupled model approach to SCA is similar to that of the redirection approach. The difference is that the ASPSP asks the PSU to authenticate e.g. via the ASPSP's dedicated mobile app or any other application or device which is independent from the online banking frontend.

In some cases, the decoupled model may improve the user experience as, when the PSU initiates the service from a browser, he/she will stay in the TPP interface of the browser.

However, a common vulnerability of all decoupled methods, even when FIDO is being used, is opening the user to social engineering attacks know as session hijacking and/or man-in-the-middle. This is because the website session on the laptop is not bound to the mobile app session on the phone. Therefore a user visiting a fake website could be tricked into providing that attacker with a valid SCA-compliant authorization. A best practice would be for the ASPSP to provide contextual information to the user on the decoupled device.

For example, for payment initiation, the user's mobile phone should display the transaction amount and payee so that user authentication and authorization is granted only for that transaction.



Example for payment initiation, in the decoupled model

For account aggregation, the user’s mobile phone should display information on the TPP that the user is authorizing to access its accounts.



Example for account aggregation, in the decoupled model

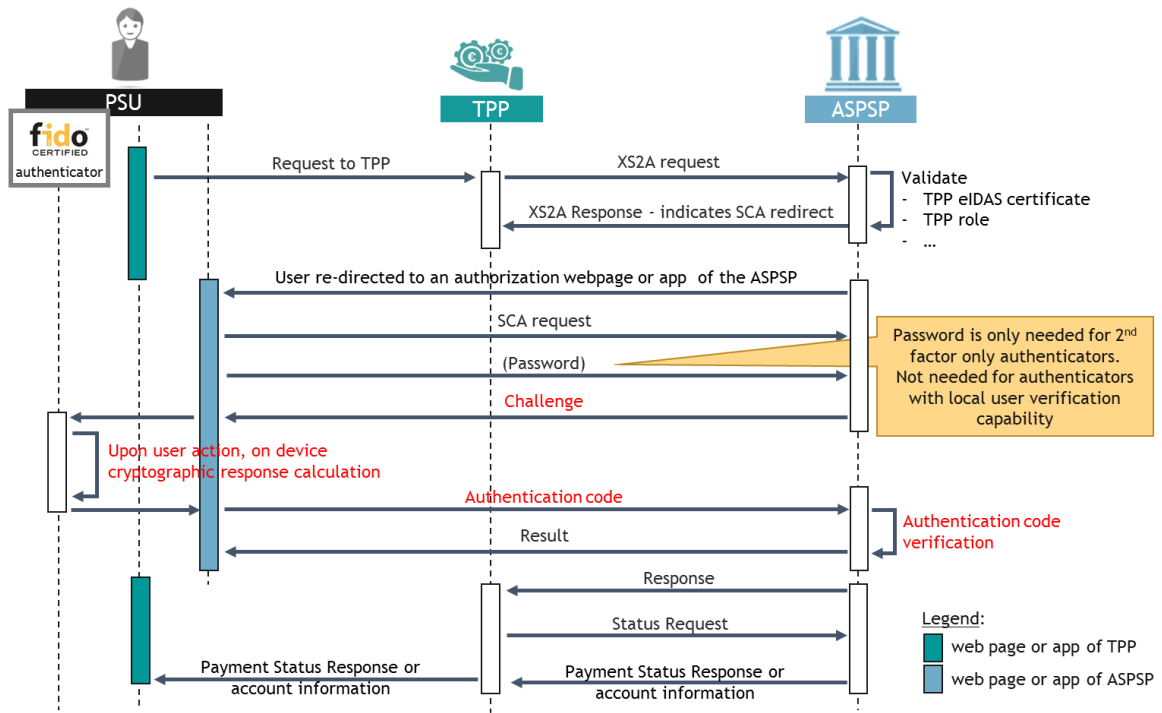
The pre-requisite of the decoupled model is of course that the PSU has a smartphone to provide consent by means of the SCA functionality of the ASPSP’s app. As not all users will have a smartphone, the decoupled approach cannot be considered alone.

4.3 Using FIDO in the redirection or decoupled model

The FIDO standards are designed to work in these models. Both the FIDO client application and server are operated by the ASPSP which is the “Relying Party” in the FIDO terminology.

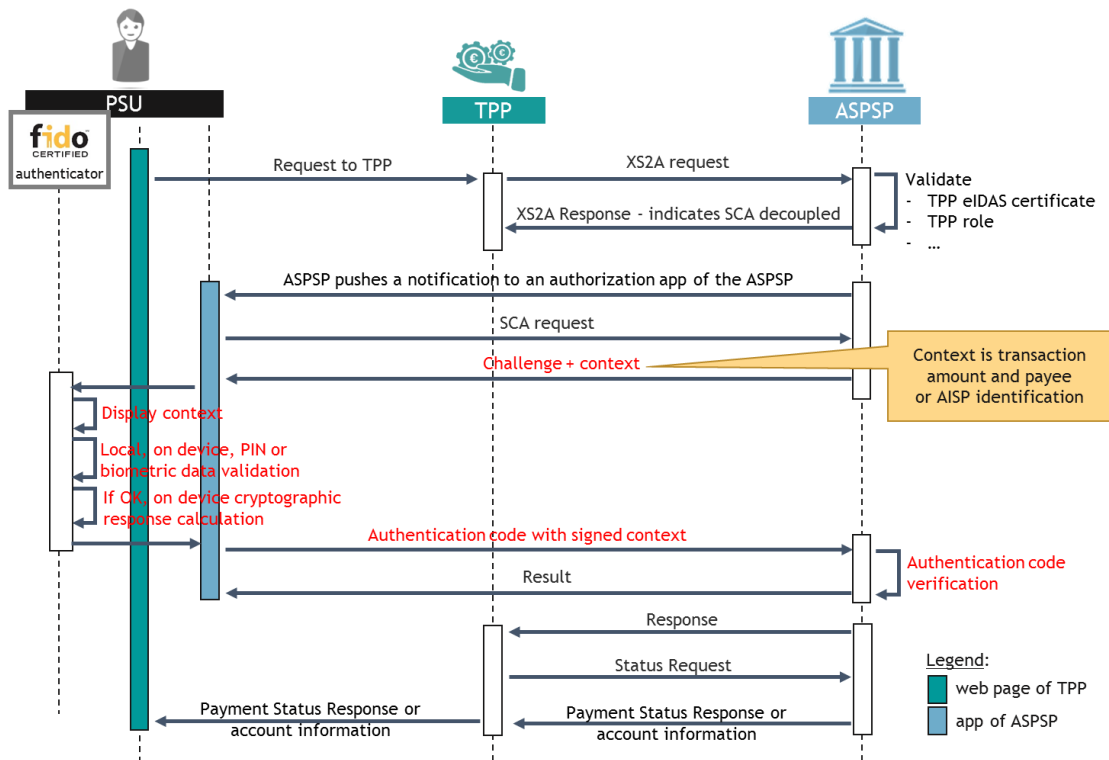
The FIDO standards allow ASPSPs to provide a very simple customer experience with a simple touch, fingerprint scan or facial recognition to authenticate the PSU - while eliminating obstacles to an excellent user experience.

The simplified sequence diagrams below illustrates the interactions between the three parties, the sequences in red corresponding to a high level description of the FIDO authentication:



FIDO authentication in the redirection model

For the Decoupled model, the FIDO standards can be used to digitally sign the context presented to the PSU to mitigate the risk of social engineering attacks described above:



FIDO authentication in the decoupled model

4.4 The debate on the user experience

A number of European Fintechs that are proposing to become TPPs have raised concerns on the user experience when using the redirection or decoupled models.

As the PSU typically starts accessing the service via the interface provided by the TPP, the redirection or decoupled approach means that the PSU leaves this interface, is switched to a different user interface to authenticate, before returning to the TPP interface.

For AISP use cases, the user could be redirected multiple times to as many ASPSPs as there are accounts to aggregate. Each bank being autonomous, may choose a device and a user experience quite different from one another which could lead to a cumbersome user journey.

More generally, TPPs are concerned if the redirection model is poorly implemented, for example with multiple screens on the ASPSP interface.

Is redirection an “obstacle?”

The **European Commission**, in their finalized version of the RTS, took the TPP concerns into account and noted in Article 32-3 of the RTS that use of the redirection model may be considered an “obstacle to the provision of payment initiation and account information services.”

However, the **European Banking Authority (EBA)**, clarified in their June 13, 2018 *Opinion of the EBA on the Implementation of the RTS on SCA and CSC*,¹ in item 49,

“... that the RTS do not state that redirection per se is an obstacle to AISPs and PISPs providing services to their PSUs. Instead, the RTS state that it ‘may’ be so, if the ASPSP implements it in a manner which is restrictive or obstructive for AISPs or PISPs.”

The EBA further stated in their consultation associated to the Opinion paper², in item 40:

“... any method of access may be an obstacle depending on how it has been implemented and National Competent Authorities should consider the user experience, whether the access method accommodates all methods of authentication and how this impacts on the user experience or if it creates delays and friction in the customer journey when assessing an exemption application for a dedicated interface that provides for access using only a single method of access.”

The focus is thus not on the model itself, but rather on the impact on the user experience of the way in which an ASPSP implements the model.

One of the most notable innovations driven by the FIDO Alliance and its members has been the advent of single-gesture, passwordless multi-factor login experiences that are both more secure and more convenient than older, legacy approaches to multi-factor authentication. The FIDO model enables the delivery of an excellent user experience using any of these models.

To this point, we offer a comparison of two different login experiences using the redirection model: one using FIDO authentication and a second via passwords and OTP:

- The first experience is based on a redirection model where a PSP redirects the PSU to an ASPSP’s authentication page, which then prompts the PSU to authenticate via FIDO. Because FIDO

¹ See

<https://www.eba.europa.eu/documents/10180/2137845/Opinion+on+the+implementation+of+the+RTS+on+SCA+and+CSC+%28EBA-2018-Op-04%29.pdf>

² Consultation Paper On Conditions To Be Met Under Art 33(6) Of RTS On SCA & CSC,

<https://www.eba.europa.eu/documents/10180/2250578/CP+on+draft+Guidelines+on+the+conditions+to+be+met+to+benefit+from+an+exemption+from+contingency+measures+under+Article+33%286%29%20of+Regulation+%28EU%29%202018389+%28RTS+on+SCA+%26+CSC%29%20%28EBA-CP-2018-09%29.pdf>

supports “single gesture” authentication, the only step the PSU needs to take to authenticate is to place her finger on a sensor or take a selfie; that biometric is then matched locally on the PSU’s device, which then unlocks the second factor: a private key that is used to sign a cryptographic challenge presented by the ASPSP. Once authenticated, the PSU is then routed back to the PSP’s interface.

The entire process takes less than two seconds, and the only thing the PSU is asked to do is present a biometric. The signing of the cryptographic challenge takes place entirely behind the scenes, without anything being demanded of the PSU.

- The second experience is based on a redirection model that requires a PSU to take multiple steps to authenticate:
 1. First, the PSU is redirected to the ASPSP’s authentication page
 2. Then, the PSU must enter a username and password into the ASPSP application
 3. Then the PSU is prompted to enter the second factor: an OTP. To do so, she must launch a separate OTP app or retrieve an OTP code sent by SMS, then view and remember the 6-digit code
 4. Then, she must return to the ASPSP login page and key in the 6-digit code without error, or without that code expiring

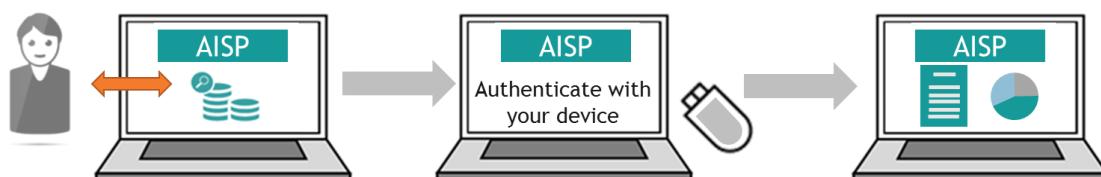
The entire process may take upwards of 15 seconds, and requires the PSU to take multiple steps to authenticate.

The use of FIDO authentication in the redirection model delivers a customer journey and user experience with less obstacles, delays and friction.

Moreover, use of the redirection model is far more secure here, given that OTP codes are often phished just like passwords, as has been documented by NIST and other security experts. FIDO authentication - given its use of public key cryptography - is resistant to phishing attacks and other tools used to compromise authentication.

4.5 The embedded model

When applying the embedded model approach, the SCA of the PSU is executed entirely through the user interface offered by the TPP. The authentication of the PSU is nevertheless still done by the ASPSP.



Example for account aggregation



Example for payment initiation

4.5.1 Challenges of this model

The embedded model presents several challenges to overcome, starting with the user verification step.

User Verification Step

Verification of the knowledge and/or inherence (biometric) factor(s) is defined in this paper as the user verification step.

In the embedded model, the knowledge and/or inherence factor(s) could be verified by the ASPSP but, if this was to be done via the TPP's interface, it would require transmission of this user data by the TPP to the ASPSP for verification, i.e. on-line to the ASPSP server.

This would introduce several difficulties:

1. The central storage of such user data with its potential for data breaches and with the liabilities linked to GDPR compliance.
2. The risks involved with the TPP handling and transmitting this user data.
3. Potential obstacles to the user journey.

Indeed, for a fluid customer journey, the PSU would ideally go through the user verification step only once even when accessing multiple accounts. This could be possible if the user verification method is managed locally through the TPP's application and is common to all ASPSPs.

The "OEM Pay" (Apple Pay, Samsung Pay, Google Pay) are examples of user verification performed by a third party, in this case the OEM (Fingerprint verification, FaceID, Iris scan), with the user authentication nevertheless performed by the bank.

Proof of possession step

Possession of the authentication device, in the embedded model, may be proven by the ASPSP in various ways:

1. The ASPSP could send an unpredictable number, such as a One Time Password (OTP) to the PSU's device where it would be displayed. The PSU would then enter this OTP in an appropriate field of the TPP's interface and the TPP would send it back to the ASPSP for verification.

With this method, the communication channel to provide the OTP to the user should be independent from the TPP in order to provide a universal solution and it should be secure so that the OTP cannot be intercepted or misused through a Man-in-the-Middle attack.

While this implementation seems straightforward it presents a number of challenges:

- The obvious channel to provide the OTP would be via SMS. However, this channel is not secure enough and is prone to fraud as has been documented in several publications, for example from NIST³
 - It does not provide for a friendly user experience as the PSU may have to receive and enter several OTPs when using account aggregation services
 - The method will likely be implemented together with an on-line user verification step, implying a shared secret prone to hacking as mentioned above
2. A better method to implement the embedded model consists in using a device, in conjunction with the TPP's user interface, that generates a cryptographic response to a challenge sent by the ASPSP.

³ Use of SMS was also restricted in the United States by NIST due to a variety of documented weaknesses in use of SMS OTP as a second factor. See <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf>

The verification of this cryptogram by the ASPSP provides the proof of possession described in the RTS.

Moreover, if the implementation is such that the cryptographic response can only be calculated upon positive user verification, the ASPSP will have the assurance, when verifying the response, that the user is properly authenticated per the requirements of PSD2.

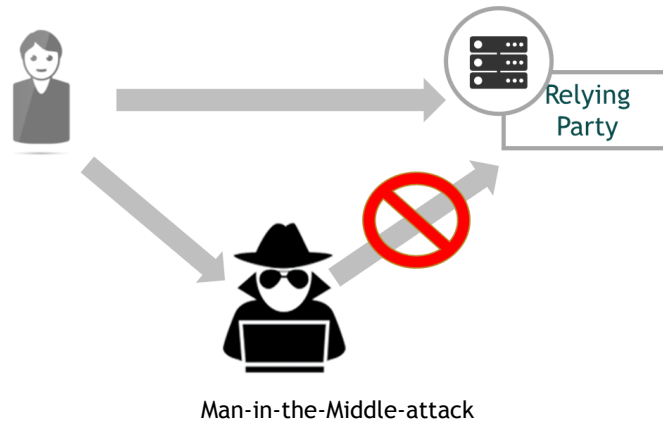
This method does imply that the device used contains securely the ASPSP keys required for the cryptographic calculation.

This latter method is the one supported by the FIDO standards which is described in the following sections of the paper.

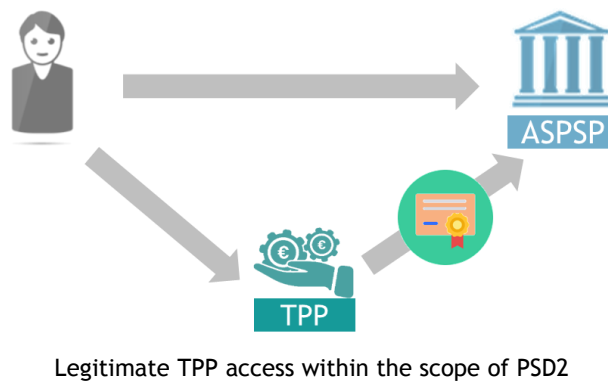
4.6 Using FIDO in the Embedded Model

4.6.1 Foreword

A model in which user access to a relying party’s resources is provided through some other party is problematic *in a general internet environment*, as it corresponds to a man-in-the-middle attack scenario. The security community is working on making it easy for relying parties to detect such a scenario and the FIDO standards include mechanisms to detect and prevent man-in-the-middle attacks.



On the other hand, PSD2 includes provisions to ensure ASPSPs identify legitimate TPPs, using eIDAS qualified certificates, as described in Article 34 of the RTS.



This specific regulatory aspect allows the FIDO standards to operate in the embedded model, within the scope of PSD2.

4.6.2 FIDO implementation

FIDO Authentication has always adhered to privacy-by-design principles as described in our published Privacy Principles https://fidoalliance.org/wp-content/uploads/2014/12/FIDO_Alliance_Whitepaper_Privacy_Principles.pdf.

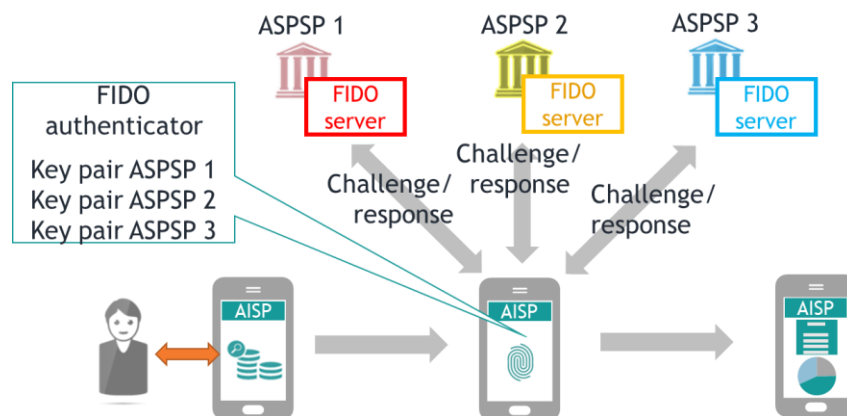
One important privacy consideration is the careful management of FIDO credential public keys so service providers do not abuse their privileged access to these public keys by, for instance, sharing public keys with other service providers without the informed consent of the user. Therefore a casual reader of FIDO specifications and certification requirements may incorrectly conclude that TPPs would be prohibited from sharing FIDO credential public keys with ASPSPs.

However, upon close review of the PSD2 requirements and the “embedded model” use cases, the FIDO Alliance believes that the legal binding between the ASPSP and the TPP, provided by the regulator and technically enforced by the eIDAS certificates, generally should be sufficient to establish cross organizational sharing of FIDO credential public keys as guided by privacy considerations enforced by law across Europe through GDPR. That being said, we want to clarify that even within the confines of this government-created legal framework, great care must be taken by all parties to not abuse their access to this component of the FIDO credential.

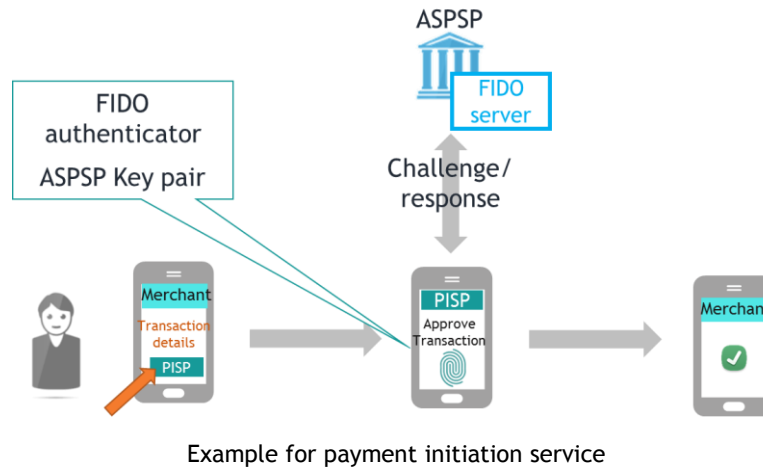
Therefore, while the proposed implementation is not what the FIDO standards were originally designed for, the regulatory context combined with the specifics of FIDO authentication allow for:

- ASPSP key pair generation within the authenticator, through the user interface of the TPP and uploading of the public key to the ASPSP
- User verification from within the user interface of the TPP
- Authentication code calculation with the ASPSP private key for SCA by the ASPSP, through the user interface of the TPP, the TPP being de facto a third party “in the middle”
- Authentication verification by the ASPSP with its public key in the usual fashion

The PSU authentication journey, with this method, consists of: PSU opens TPP application, scans finger (or takes selfie or enters PIN) and accesses the service. Behind the scenes, the FIDO authenticator based TPP application will connect to each required ASPSP for the purpose of SCA.



Example for account aggregation service. The FIDO Authenticator holds personalized security credentials (key pairs) for each ASPSP the PSU needs to access to



4.6.3 Registration

Preliminary aspects

For ASPSP keys to be generated within the authenticator, the PSU must be enrolled with the TPP.

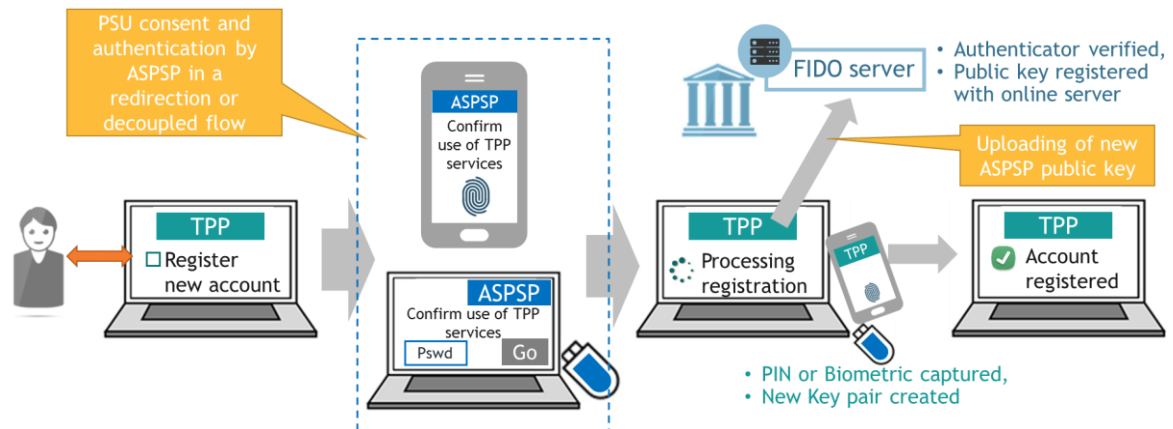
However, some TPPs, PISPs in particular, may not want to have users of their services enrolled with them. For example, a PISP that has integrated with an on-line merchant to provide payment initiation services will want *any* user shopping with this merchant to be able to use its services and will not want to oblige such users to go through a registration step.

If the PSU is not enrolled with the TPP, this TPP may not be able to provide value added services through its user interface, nor will it be able to participate in the FIDO flows that enable the embedded model. If the TPP will not have a direct relationship to the user then SCA compliant user authentication is best performed through a redirection or decoupled model.

Registration of PSU in the embedded model

The FIDO standards allow ASPSP/account registration, in the embedded model, to start from the TPP interface. However the ASPSP will need to verify the identity of the PSU once. This is best achieved through use of the redirection or decoupled models described earlier in this document.

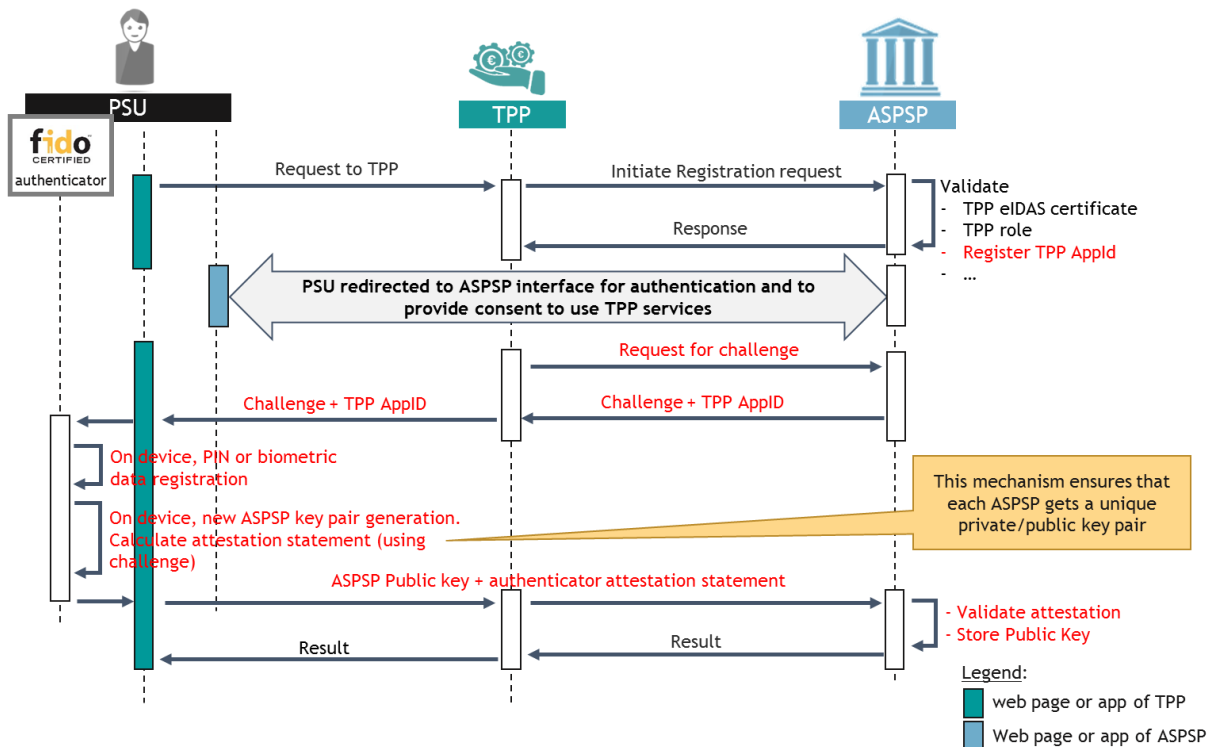
This step can also be used to obtain explicit consent from the PSU to use the services of the TPP in the embedded model, in order to mitigate privacy concerns.



Example of user journey for ASPSP registration

This registration procedure should be repeated with each ASPSP every time the PSU desires to add an account to its TPP application.

Simplified sequence diagram



An important step required by the FIDO standards is the registration of the TPP’s application ID in the FIDO server of the ASPSP.

Subsequently, the identity of the TPP application will be verified by the FIDO server during authentication through the recorded AppID. This allows ASPSPs to accept trustworthy TPPs and reject malicious applications from misusing the open banking API.

Privacy aspects

The registration step described above should be repeated for each new account/ASPSP that the PSU desires to register with the TPP's application.

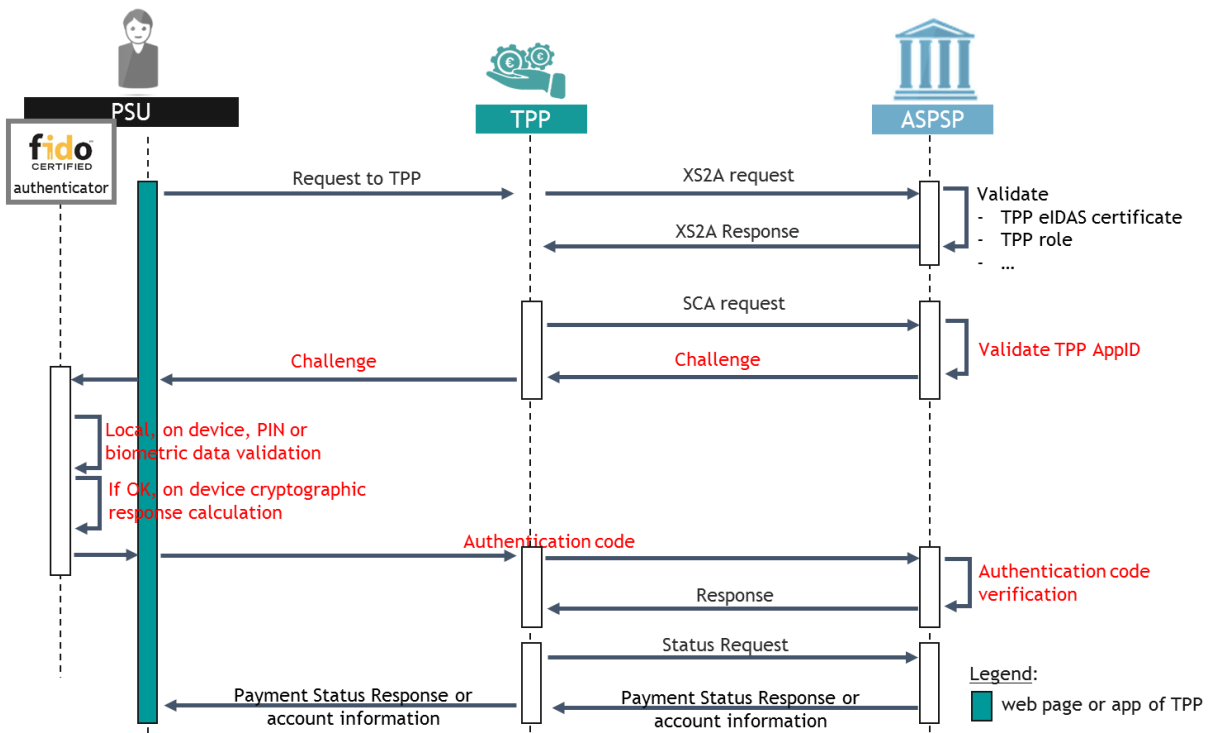
The FIDO standards provide safeguards to prevent a TPP from providing to an ASPSP the public key previously generated for another ASPSP. The TPP using a FIDO certified authenticator will have to generate a new key pair every time the PSU registers a new account with a new ASPSP.

More generally, the TPP should not relay the public key generated for an ASPSP to any other party than the intended ASPSP.

4.6.4 FIDO authentication in the embedded model

The challenge response mechanism of FIDO is used with the TPP acting as a pass-through authorized third party.

Simplified sequence diagram



User Verification Caching

The PSU may, in certain use cases, have to authenticate with multiple ASPSPs, for example for account aggregation purposes. While this authentication, from a user's perspective, would typically be as simple as a finger swipe per ASPSP, there is the possibility to reduce it further down to one single user verification for all ASPSPs.

The FIDO standards support this need through the User Verification Caching mechanism. This mechanism allows a FIDO authenticator to memorise - cache - for a period of time that the user verification was positively processed. During that time, the challenge/response part of the authentication procedure does not require a new user verification step.

Note that the time elapsed since the user verification took place is communicated to each ASPSP who have the means to control that it does not exceed a time limit they have set for their respective key at registration. This enables ASPSPs to keep control of the "freshness" of user verification, for example to ensure proper user consent.

4.6.5 Prerequisites for the parties

Clearly, in the embedded model described in this paper, the use of the FIDO standards cannot be a TPP decision or ASPSP decision alone. All parties must agree to adopt these standards.

More specifically, ASPSPs must agree to the user verification step being triggered by the TPP application and performed by the FIDO authenticator.

ASPSPs will need to record in their FIDO servers, the AppID of a TPP application connecting for the first time when the PSU registers with this ASPSP. In effect, ASPSPs will "white list" the TPPs that connect to them through the embedded authentication model.

Impact on the Open APIs

As can be seen from the sequence diagrams shown above, the implementation of the embedded model using the FIDO standards requires new APIs that are not needed for the redirection/decoupled model.

Typically, APIs are needed to support the challenge response mechanism. Also new data fields will be required to handle the registration phase when the TPP connects for the first time to an ASPSP or, more generally, to support the FIDO standards (transmission of policy, of authenticator attestation certificate, etc.).

The Berlin Group released new specifications that include an API to send and receive a challenge and response: <https://www.berlin-group.org/psd2-access-to-bank-accounts>.

4.7 The delegated model

In the delegated model, PSU authentication is performed by the TPP, not the ASPSP. This would ensure a smooth user experience as the TPP would handle the entire interactions with the PSU. This model presents a number of challenges in terms of providing trust to the ASPSP and in terms of liabilities in case of fraudulent access to the PSU's account.

A way forward could be for the ASPSP to perform risk management based on authentication information that the TPP would transmit, the ASPSP having the final decision whether to step to PSU authentication or not.

FIDO and EMVCo recently announced their collaboration to define in detail how EMV 3-D Secure (3DS) messages may be used to pass FIDO authenticator attestation data and signatures to the issuing bank, when the merchant performed PSU authentication using a FIDO authenticator.

Indeed, the FIDO standards can perfectly be used in the delegated model. In this case the FIDO "Relying Party" is the TPP and both TPP client and server interact during the PSU authentication process.

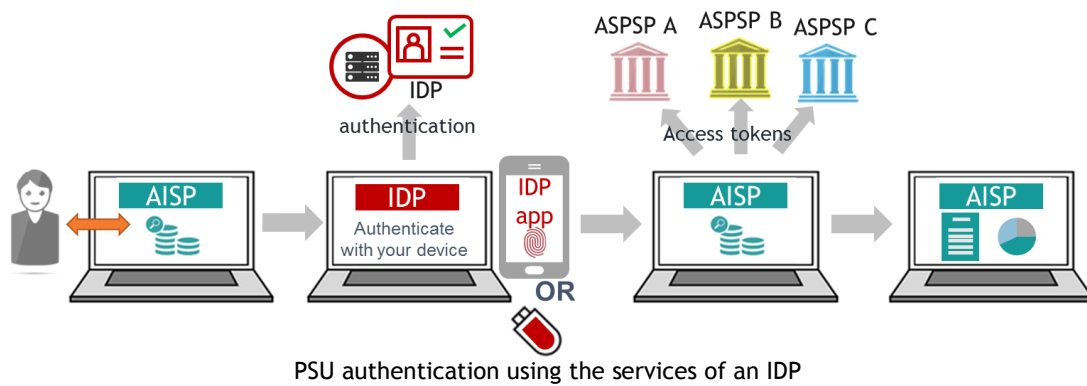
The FIDO Alliance is currently investigating the delegated model in the scope of PSD2.

5 ID Federation

The question of multiple authentications, in particular for account aggregation services, may be addressed by using federated identity and an authorization framework.

In such a system, the participating ASPSPs will recognize a trusted Identity Provider (IDP) as being tasked with authenticating the PSU once and subsequently delivering access tokens to the TPP to authorise it to access the different PSU's accounts.

The customer journey would then be illustrated as follows:



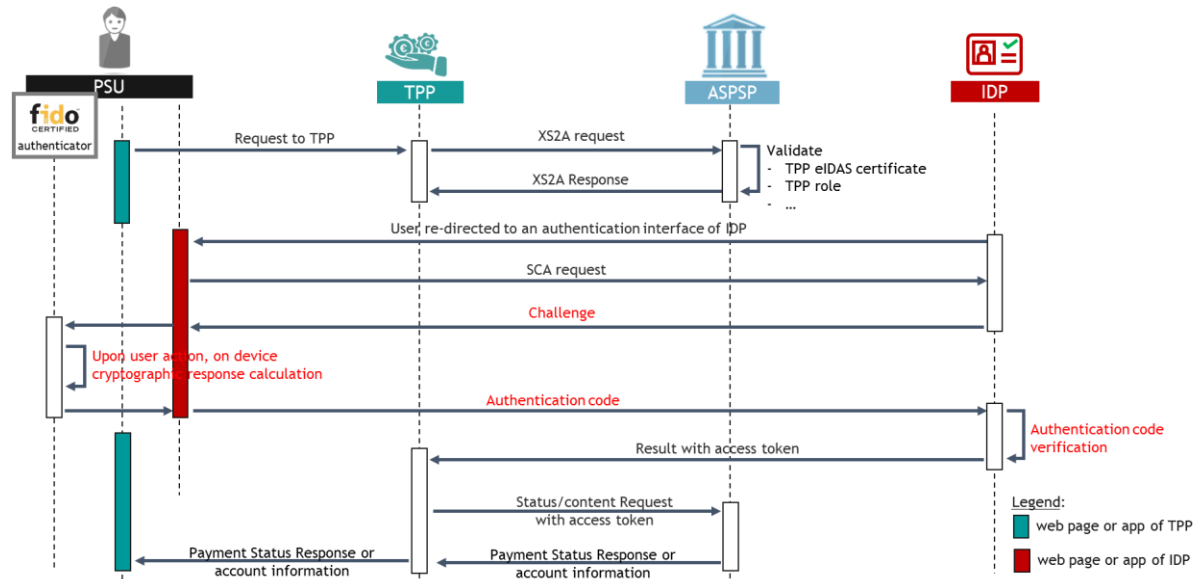
The example above, in the redirection model, may of course be supported in the decoupled model using an app proposed by the IDP to the PSU.

Federation protocols, such as OAuth 2, SAML and OpenID Connect are proposed in the Open API standards to support the authentication by the trusted IDP and the delivery of the required access tokens to the TPP.

ID federation technologies and FIDO standards are complementary. The use of the federation system extends the benefits of FIDO authentication to ASPSPs without requiring FIDO to be directly integrated by them. A full white paper on this subject available here:

https://fidoalliance.org/wp-content/uploads/Enterprise_Adoption_Best_Practices_Federation_FIDO_Alliance.pdf

Simplified sequence diagram



6 Conclusion

The redirection, decoupled, embedded and delegated authentication models have been reviewed in this white paper.

The FIDO standards work well in implementing any of these models - and are ideally suited to address concerns about “obstacles” to an excellent customer experience associated with some of the models. They offer privacy by design, compliance to the RTS and alignment with authorization frameworks such as OAuth 2, and their unique cryptographic properties remove vulnerabilities common to all “shared secret” methods, such as one-time-passcodes.

All of these capabilities make FIDO compliant solutions the clear choice for implementing excellent, low friction user experiences compliant to PSD2’s Strong Customer Authentication requirements.