FIDO & PSD2

Providing for a satisfactory customer journey

April, 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 user will have to 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 the user experience.

With the Regulatory Technical Standards officially published in the Official Journal on March 13th, 2018, the countdown for PSD2 readiness has started and banks will have to make choices of their Strong Customer Authentication implementation soon.

This paper examines the advantages and drawbacks of the different authentication models and proposes the FIDO standards as a solution to simplify the user experience, for any of these models, in a way that meets the needs of TPPs and ASPSPs.

2 Glossary of terms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AISP</td>
<td>Account Information Service Provider. For example, a provider of account aggregation services.</td>
</tr>
<tr>
<td>ASPSP</td>
<td>Account Servicing Payment Service Provider. Typically, the bank holding the accounts</td>
</tr>
<tr>
<td>IDP</td>
<td>Identity Provider in a federated identity ecosystem</td>
</tr>
<tr>
<td>OTP</td>
<td>One Time Password</td>
</tr>
<tr>
<td>PISP</td>
<td>Payment Initiation Service Provider</td>
</tr>
<tr>
<td>PSU</td>
<td>Payment Service User. The user providing consent to a TPP to access its accounts</td>
</tr>
<tr>
<td>RTS</td>
<td>Regulatory Technical Standard. In the context of this white paper, RTS refers to the RTS on Strong Customer Authentication and Common and Secure Communication</td>
</tr>
<tr>
<td>SCA</td>
<td>Strong Customer Authentication</td>
</tr>
<tr>
<td>TPP</td>
<td>Third Party Provider: an AISP or a PISP</td>
</tr>
<tr>
<td>XS2A</td>
<td>Access to Account (for the purpose of initiating a payment or retrieving account information)</td>
</tr>
</tbody>
</table>
3 The different authentication models

3.1 The redirection model

The redirection model is an approach, defined by the API standardisation bodies, whereby the Payment Service User (PSU) starts interacting with a TPP and is redirected to a web interface of the ASPSP for authentication. The PSU may also be redirected to the ASPSP’s mobile app for the purpose of authentication (See next section: decoupled model).

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.

Simplified flows
Advantages of the redirection model

The redirection model presents the advantage that the ASPSP is in full control of the way to handle user authentication. If it has a solution already in place, it may be used provided it complies with the RTS. 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. Besides, as the model is independent from other parties, it will work with any TPP connecting through the Open APIs.

Additionally, some users may be more comfortable with authenticating from the interface of the ASPSP which they may be used to and find more trustworthy.

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

The FIDO standards: A perfect match with the redirection model

The FIDO standards, as discussed in a previous white paper, https://fidoalliance.org/wp-content/uploads/FIDO-PSD2-white-paper-FINAL.pdf, are ideally suited to implement the redirection model. They meet all the requirements of the RTS and can help ASPSPs reduce their cost of deployment.

Indeed, 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.

Thanks to its interoperability model, the FIDO standards offer a means of achieving this reach with lower costs than for non-standardized implementations.

Also, FIDO solutions are frequently combined with authorization frameworks such as OAuth2.0 or OpenID Connect. These frameworks are proposed in the redirection model defined in the current open APIs and the FIDO standards can be jointly used with them, to facilitate limited access to bank accounts for TPPs, while ensuring strong customer authentication. See section 4 for more information on the use of FIDO standards within authorization frameworks.

3.2 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.
Example when accessing the service from a browser

Example when accessing the service from a smartphone

Simplified flows

Decoupled model versus redirection model

The decoupled model improves on the user experience as, when the PSU initiates the service from a browser, he/she will stay in the TPP interface of the browser. The decoupled model is also more suitable when the whole customer journey is on a smartphone, as illustrated above.

The pre-requisite 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.

Here again, the FIDO standards help ASPSPs with the deployment of this model as they support multiple channels, on PC, on smart phones, using multiple devices. A universal FIDO server operated by the ASPSP will
be able to support strong customer authentication originating from a PC with FIDO implemented in a USB security key or smart card as well as from a smartphone with an embedded FIDO authenticator.

### 3.3 The question of 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. This may lead to a cumbersome user journey, especially for account aggregation services, as illustrated hereafter:

The risk of the redirection/decoupled model

### A recognised issue

The European Commission, in their finalized version of the RTS, recognise this issue 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.” We note that EC officials have also stated this language does not mean that every instance of a redirection implementation will in fact be considered an obstacle; it will ultimately be up to each country’s National Competent Authority (NCA) to determine if an interface that uses the redirect is an obstacle. The fact that it has been flagged, however, means there will likely be a need in some countries to pursue other models.

The Euro Retail Payment Board (ERPB) also makes the recommendation, in their Final Report on Payment Initiation Services dated November 17th 2017, that “The PSU should not be required to access an ASPSP webpage as a part of the authentication process or any other relevant function as this would limit the PISP in the innovative design of its customer interfaces”. 

---

©FIDO Alliance 2018
Identity federation

Using the services of an Identity Provider (IDP) in a federated identity system is a solution to resolve the potential problem of multiple authentications/multiple devices highlighted above. Indeed the user would authenticate only once to the IDP and the TPP would be provided with access tokens to get access, transparently to the user, to the different ASPSPs.

The section 4 of this white paper describes how FIDO can help with the implementation of Strong Customer Authentication in a federated identity system.

3.4 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.

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

User Verification Step

In PSD2, SCA consists in verifying a knowledge factor (e.g. a PIN) and/or an inherence factor (e.g. biometrics) and/or a possession factor (a device) with a mandated minimum of two factors.

Verification of the knowledge and/or 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, i.e. on-line to the ASPSP server.

This would introduce several difficulties: One is related to the central storage of such user data with its potential for data breaches and with the liabilities linked to GDPR compliance. Another is linked to the TPP handling and transmitting this user data. And another relates, again, to the user journey.

Indeed, in an ideal implementation, the PSU would go through the user verification step only once even when accessing multiple accounts.

Example of user experience when using a browser
For better security and a fluid customer journey, it is therefore preferable that the user verification method be managed locally through the TPP’s application and be common to all ASPSPs.

The “OEM Pay” (Apple Pay, Samsung Pay, Android Pay) are examples of user verification performed by a third party, in this case, the OEM (Fingerprint verification, FaceID, Iris scan).

**Proof of possession verification step**

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

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 seen in 3D Secure implementations.
- 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.

Another method to implement the embedded model consists in using a device, in conjunction with the TPP’s user interface, to generate a cryptographic response to a challenge sent by the ASPSP. 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.

---

1 Use of SMS was also restricted in the United States by NIST due to a variety of documented weaknesses in use of SMS as a second factor. See [https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf](https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf)
A Challenge/Response API is supported in the Open API specification of the Berlin Group. [https://www.berlin-group.org/psd2-access-to-bank-accounts](https://www.berlin-group.org/psd2-access-to-bank-accounts).

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

### 3.5 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. However, 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.

This paper does not propose to cover the implementation of this model, as FIDO’s belief is that the embedded model, as described in section 5, provides a satisfactory user experience to the benefit of the PSU, the TPP and the ASPSP.

### 4 FIDO and 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:

![Customer Journey Diagram]

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.

Moreover the federated system may already be in place and the authentication services offered by the IDP used and trusted by the PSU.

See appendix for a summary of how FIDO integrates in federation protocols.
5 FIDO and the embedded model

The FIDO standards can be used to implement the embedded model as described in section 3.4, thanks to their management of user verification, of cryptographic keys, as well as to their support, in the FIDO server, of multiple application IDs, to identify connecting FIDO clients.

A FIDO authenticator can generate and securely store multiple keys for each registered ASPSP. These keys are used to calculate cryptograms verified by each ASPSP with their respective key to prove the possession factor.

This cryptogram generation can be conditioned to the positive verification of the user’s PIN or biometric data, locally by the FIDO authenticator. The user verification method depends on the Authenticator and the same method would be used for all registered ASPSPs.

FIDO Authenticators can support the concept of user verification caching. This concept allows multiple authentications to be performed in a defined time window following a single user verification (see user verification caching below).

Based on these features, the PSU journey, as illustrated in the examples hereafter, would look like this: 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 key pairs for each ASPSP the PSU needs to access to

Example for payment initiation service
5.1 Reminder of the basics of FIDO

Key generation at registration

When a user registers with a Relying Party (a service provider), the Relying Party 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.).

The authenticator will then generate a new public/private key pair specific to the Relying Party and the public key will be uploaded to the Relying party’s FIDO server as shown in the diagram below.

User authentication

When the user wishes to access the service, the Relying Party will authenticate the user through a usual challenge/response mechanism. Depending on the user’s authenticator, the user verification will first be performed, using a PIN, password or biometric match. This authentication process is illustrated below:

5.2 ASPSP/account registration 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 could be achieved through the redirection or decoupled models described earlier in this document:
Example of user journey for ASPSP registration

It should be noted, in the example above, that the smartphone used by the PSU to authenticate itself to the ASPSP could be the same device used in conjunction with the TPP application, both being based on the FIDO standards.

Simplified flows

This sequence diagram only describes the flows related to the FIDO standards.

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.
5.3 Customer authentication in the embedded model

The FIDO standards allow TPPs and ASPSPs to implement SCA in the embedded model with a very simple and convenient user experience as was illustrated above.

Moreover this implementation is perfectly compatible with the application of the exemptions to SCA defined in the RTS, as shown in the following sequence diagram:

Simplified flows

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 single 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 key at registration. This enables ASPSPs to keep control of the “freshness” of user verification, for example to ensure proper user consent.

5.4 Dynamic linking in the embedded model

When the PSU is shopping on-line, using the services of a PISP, the RTS require that the transaction amount and payee be dynamically linked to the authentication code. This can be achieved if the cryptogram, generated by the PSU’s device, signs the amount and payee ID.
Moreover the PSU must provide consent for this transaction by approving the transaction data as displayed. In the embedded model this transaction data will be displayed on the TPP’s user interface.

The FIDO standards support dynamic linking through the Transaction Confirmation feature.

Using FIDO Transaction Confirmation, the authenticator displays the transaction text to the user and asks the user for 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.

5.5 Other important aspects of the solution

ASPSP policy

When using the FIDO standards, an ASPSP does not necessarily need to deploy authenticators, as authenticators already in the field may be accepted to implement SCA, provided that they comply with the ASPSP’s policy.

To make use of this clear benefit, the FIDO standards allow ASPSPs to define a policy for the authenticators they will accept for use by TPPs.

The ASPSP policy will ensure that authenticator characteristics - or Metadata - match certain criteria such as secure environment to store keys, cryptographic algorithms supported, biometrics supported, false acceptance/rejection rates, etc.

Authenticator characteristics may be available to ASPSPs through Metadata servers such as FIDO’s public MDS server.
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.

5.6 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 the Challenge and Response.

6 Conclusion

The pros and cons of the redirection, decoupled and embedded authentication models have been reviewed in this white paper.

The FIDO standards offer an ideal solution to implement any of these models. Their privacy by design, compliance to the RTS and alignment with authentication frameworks such as OAuth 2 make them particularly suitable to implement Strong Customer Authentication.

The FIDO standards provide ASPSPs with versatility and flexibility: They may decide to use the services of an IDP or implement their own authentication solution. Whichever the choice, the FIDO standards can be used to authenticate PSUs in the redirection and decoupled models with a maximized reach, supporting a multiplicity of devices. They can also allow ASPSPs to propose the embedded model to TPPs that integrate FIDO authenticators in their solutions.
Appendix: Integrating FIDO & Federation Protocols

The information in this appendix is extracted from a full white paper on this subject available here: https://fidoalliance.org/wp-content/uploads/Enterprise_Adoption_Best_Practices_Federation_FIDO_Alliance.pdf

The high-level integration between a federation protocol flow and a FIDO-based authentication is shown below. This call flow assumes that the User registered FIDO credential with the Authentication Authority:

1. User accesses Application Provider (via User Agent) and needs to authenticate.
2. Application Provider redirects the User Agent to the FIDO-enabled Authentication Authority with a request that the user be authenticated. This redirect allows the Application Provider to specify that the user authentication be FIDO-based (either explicitly or implicitly) according to its preferences.
3. User Agent accesses the Authentication Authority federation endpoint. The Authentication Authority determines that the FIDO authentication specified in the previous request is both relevant and possible by confirming that the policies of the Authentication Authority match with the preferences of the Application Provider.
4. The Authentication Authority sends a FIDO Server challenge to FIDO Client (via User Agent). This challenge may indicate a particular FIDO Authenticator - as determined by the Application Provider’s preference and the Authentication Authority’s own policies.
5. FIDO Client locates FIDO Authenticator, and the user is authenticated
6. FIDO Client returns FIDO authentication response (via User Agent)
7. FIDO Server validates the FIDO response and reports same to Authentication Authority which looks up the user and then redirects the User Agent back to the Application Provider with an authentication assertion.
8. User Agent calls Application Provider with the authentication assertion (or an artifact that the Application Provider can exchange against the Authentication Authority for the actual authentication assertion, step not shown). The authentication assertion can include the specifics of the FIDO-based authentication or more generally a policy identifier for that authentication - as previously discussed.

The bolded steps are those by which respectively,

a) the Application Provider is able to indicate to the Authentication Authority its expectations for how the user should be authenticated and
b) the Authentication Authority is able to indicate to the Application Provider details about how the user was actually authenticated.