

## Digikukru I etapi analüüs

Version 1.0.1

21.12.2023

D-5-2910

## Kokkuvõte

---

EL on välja töötamas üle-euroopalist identiteeditasku lahendust, mille realiseerimine ja kasutuselevõtt muutub kohustuslikuks uue eIDAS määruse jõustumisega (eeldatavalt 2026+) ning liikmesmaad on juba arendamas oma lahendusi, tehes koostööd nii eIDAS ekspertgrupi tasemel ja suuremahuliste piiriüleste projektide koosseisus.

Paralleelselt teiste liikmesmaadega on Cybernetica RIA tellimusel alustanud tööd Eesti identiteeditasku analüüsiga. Projekti esimeses faasis käsitusallas oli koostada esialgne analüüs järgnevate teemade kohta

- Identiteeditasku tehnilise lahenduse arhitektuur,
- Identiteeditasku aktiveerimise analüüs ja
- Identiteeditasku kui elektroonilise autentimisvahendi kasutamise analüüs.

Käesolev dokument on kaaskirjaks dokumentide kogule, mis on valminud esimese faasi käigus. Dokumentide näol on tegu väljavõttega RIA valduses olevast Confluence'ist. See tähendab et dokumentide redigeeritavad originaalid asuvad tolles keskkonnas.

Dokumendid on koostatud inglise keeles.

Dokumentide kogus koosneb järgnevatest dokumentidest:

1. Business Object Model (ärikontseptsioonide sõnastik ja huvipoolte analüüs)
2. Lifecycle Diagrams (oluliste äriobjektide elukaarte kirjeldused)
  1. EE Wallet Instance Lifecycle (Eesti identiteeditasku instantsi elukaar)
  2. EE PID Attestation Lifecycle (Eesti PID atesteeringu elukaar)
3. Process Models (protsessimudelid)
  1. EE PID Attestation Issuance (Eesti PIDi väljastamine)
  2. EE PID Attestation Revocation (Eesti PIDi kehtetuks kuulutamine)
  3. EE Wallet Instance Revocation (Eesti identiteeditasku kehtetuks kuulutamine)
  4. Authenticate using EE Wallet & EE PID Presentation (autentimine Eesti identiteeditaskuga)
  5. EE Relying Party Registration (Identiteeditaskust sõltuvate osapoolte registreerimine)
4. Architecture of the EE Wallet (Eesti identiteeditasku arhitektuuri konseptsioon)
5. Data Models (Eesti identiteeditasku kontseptuaalsed andmemudelid)
6. Credential Presentation Interface (Identiteeditasku PID esitamise liidese kirjeldus)

LISA 1: EE PID Attestation Issuance (PID atesteeringu väljastamine mitme seadme vahel – laiendatud protsess)

LISA 2: EE PID Attestation Issuance same-device (PID atesteeringu väljastamine ühes seadmes – laiendatud protsess)

LISA 3: BPMN Guide (BPMN skeemide lugemisujuhend)

# 1. Business Object Model

This page aims to cover the highest level of analysis for the EE Wallet. For this purpose the page is split into two sections: *Business Concepts* and *Stakeholders*. The first section describes various key terminology which is used throughout the analysis - the domain. While it mainly acts as a glossary it also aims to capture some high-level requirements (eg: *wallet is an eID means at LoA high*) and assumptions made to limit the scope of this analysis (eg: *the wallet provider is a public sector entity*). The domain description is based on the European Digital Identity Wallet. The latter section describes the stakeholders for the EE wallet and their interest which act as the root source of requirements for the system.

## Business Concepts

### Credential

For the purposes of this analysis the term credential is used to mean any document presented using an EUDI wallet the most notable example of which is the PID presentation.

### EE

Country code **EE** (Estonia) is used in front of the other terms listed in this section signifies the Estonian instance of that term. For example **EE Wallet Provider** is the **Wallet Provider** in Estonia.

### EUDI Wallet Instance

Instance of an European Digital Identity Wallet Solution that belongs to and controlled by the user [ARF1.3] which acts as an eID means at LoA high [EIDAS2]. Wallet instances will also have other capabilities, but those are out of the scope of this analysis.

### EUDI Wallet Provider

The entity which provides European Digital Identity Wallet Solution to the users.

Providing a Wallet Solution would entail both releasing a mobile application and administering the back-end services supporting the application. As EUDI Wallet Instance is required to act as a LoA high eID means, the Wallet Provider would also have to coordinate the provision of a PID Attestation into the Wallet Instance. Depending on the policy decisions of a specific member state the PID Attestation could be provisioned by the Wallet Provider themselves.

A Wallet Provider could be either a public sector body or a private sector entity mandated or recognized by an EU Member State [EIDAS2]. For the purposes of this analysis it is assumed that in Estonia the Wallet Provider will be a public sector body - any process described herein might not be suitable for a private sector entity as a different trust model might apply.

### EUDI Wallet Solution, EUDI Wallet

European Digital Identity Wallet Solution is the entire product/service owned by an EUDI Wallet Provider, offered to all users of that solution [ARF1.3].

### Electronic Identification Means (eID means)

Something which contains person identification data and is used for authentication to an online or, where appropriate, to an offline service [EIDAS2].

### Key Attestation

During PID Attestation issuance the PID Provider associates the PID Attestation with some public key provided to it by the Wallet Instance. PID Provider needs to somehow be able to determine that the matching private key is protected to a sufficient degree, eg: PID Provider could require that keys are protected by specialized hardware. An EE Wallet Instance provides proof of this in the form of a *key attestation* - a statement signed by the Wallet Provider which specifies the security guarantees for the key pair. In practice this statement could be a simple x509 certificate.

For the same purpose the ARF proposes a similar concept called *wallet attestation*. This would essentially be a credential like any other proving that the wallet itself is secure, therefore any keys it provides can be assumed to be protected sufficiently. Such an indirect approach offers if not weaker than definitely harder to analyze security guarantees. For EE Wallet therefore, the more direct approach of using key attestations has been chosen.

*Note that in a scenario where PID Attestation is provided directly by the Wallet Provider, this concept could be dropped as the Wallet Provider is themselves capable of determining the security guarantees of the keypair.*

### Level of Assurance (LoA)

EU has defined three assurance levels for eID schemes: low, substantial and high [LoA]. Each level corresponds to a set of requirements for an eID scheme, while the highest LoA, high, includes all the lower level requirements. LoA high essentially requires absolute certainty about the identity of the user being authenticated.

In Estonia the examples of LoA high eID means are: the Estonian ID-card, Mobile-ID and Smart-ID.

### Person Identification Data (PID)

A set of data enabling the identity of a natural or legal person\*, or a natural person representing a legal person to be established. [EIDAS2]

In the context wallets, to ensure cross-border interoperability, for natural persons the following minimal PID Attributes dataset is mandatory [ARF1.3]:

- Family name
- Given name
- Date of birth
- Age over 18 (*Assertion: true/false*)
- Unique identifier
- Metadata
  - Issuance date
  - Expiry date
  - Issuing authority
  - Issuing country

*\*For the current version of the analysis only PID for natural persons is in scope.*

## Person Identification Data Attestation (PID Attestation)

An electronic attestation of PID issued into an EUDI Wallet Instance. The classical equivalent in the Estonian eID ecosystem are ID-card/Mobile-ID/Smart-ID certificates. Some notable differences are as follows.

- PID Attestations will use new data formats instead of x509 certificates.
- PID Attestations might contain a different dataset than what is currently included in certificates.
- PID Attestations would be usable offline in lieu of physical documents.

## Person Identification Data Presentation (PID Presentation)

A dataset derived from a PID Attestation which is presented to a Relying Party using an EUDI Wallet to authenticate identity or other attributes. A presentation has the following properties.

- PID Presentation might contain only a subset of the PID dataset as opposed to the PID Attestation. Eg: only the 'age over 18' value is presented. This is called selective disclosure.
- PID Presentations must prove that the person presenting them is the same as the underlying PID Attestation was issued to - in other words they must add a proof of user binding. In offline settings this could be done using biometrics, eg: using a document photo. In online settings the standard is to use a cryptographic signature created by the wallet.

## Person Identification Data Provider (PID Provider)

The entity which issues (signs) PID Attestations to EUDI wallets.

PID Providers are responsible for [ARF1.3]:

- verifying the identity of the EUDI Wallet User in compliance with LoA high requirements,
- issuing PID to the EUDI Wallet in a common format and
- proving the validity of PID Attestations to Relying Parties.

## Relying Party (RP)

Natural or legal person, that relies on European Digital Identity Wallets for authenticating the identities or other attributes of end-users.

*For the current version of the analysis only remote online presentation to Relying Parties is considered.*

## Relying Party Registry

Registry where the Relying Party registers if they intend to rely upon the European Digital Identity Wallets for the provision of public or private services. Registration is done in the Member State where the relying party is established [EIDAS2].

## Relying Party Registry Provider

Provider of national Relying Party Registry.

## User

Natural or legal\* person using the European Digital Identity Wallet. [ARF1.3]

*\*Only natural persons are in scope of this analysis. Processes described herein might not be suitable for legal persons.*

## Trusted List of Attestation Providers

Repository of information about authoritative entities in a particular legal or contractual context which provides information about their current and historical status. Trusted Lists can be implemented in different ways. [ARF1.3]



## eIDAS expert Group

The eIDAS expert group assists the European Commission in preparation of various policies, acts, etc. The members of the group include an authority from each of the Member States. [XPRT-GROUP]

This group has been mandated by the European Commission to develop a common Union Toolbox for the European Digital Identity Wallets [TOOLBOX-MANDATE]. For this purpose they are leading the development of the The European Digital Identity Wallet Architecture and Reference Framework or ARF [ARF1.3]. The ARF has been developed somewhat independently from the legislative process by not prejudging it - therefore any requirements within may or may not be aligned with the outcome of the legislative negotiations. However, the document does set out many interoperability requirements which the EE wallet should follow:

- PID Attestations must be presentable both in ISO/IEC 18013-5:2021 and the W3C VC Data Model 1.1
- PID Attestations must use signatures and encryption formats as detailed in JOSE RFCs and COSE RFCs and in accordance with SOG-IS ACM

## Estonian Information System Authority (Riigi Infosüsteemi Amet - RIA)

RIA which currently acts as the Supervisory Body for Estonia under eIDAS 1 is also leading the procurement of an EDIW in Estonia. Under the new eIDAS regulation they might act in multiple roles. This analysis assumes that they'd be the entity acting as the Wallet Provider for Estonia.

Regarding wallet registration, RIAs main concern is to minimize the barrier for entry for users. The users should be able to start using the wallet as an eID means as easily as possible. One way to streamline this is combining the wallet activation and PID attestation issuance into an atomic step (from the viewpoint of the user). RIA also feels that a person with only an ID-card and a mobile phone (ie: no access to a desktop computer and an id-card reader) should be able to activate a wallet and have a PID issued to it. For this purpose having the wallet also act as an NFC ID-card reader app would be sufficient.

While it has not been decided who will issue PID Attestations in Estonia RIA have suggested that using the minimal mandatory dataset would be sufficient.

## Estonian Ministry of Interior / Estonian Police and Border Guard Board (Siseministerium / Politsei- ja Piirivalveamet - PPA)

In Estonia the issuance of personal identification documents is the responsibility of PPA. This leads them having some stake in the issuance of PID Attestations as these could be considered such documents. Historically, the issuance of electronic certificates which the PID Attestation resembles has been outsourced to private sector entities. Then again, as PID Attestations are also planned to be an alternative to physical documents in face-to-face settings the policy of outsourcing might no longer be acceptable.

In this analysis it is currently assumed that PPA will act as the sole PID Provider. However there are many alternate setups choosing of which would require a political decision:

- RIA could act as both the PID and Wallet Provider. This would simplify many processes and would thus likely be the cheapest overall option. It would also perhaps be the easiest to comprehend for end users. Any customer support would then be handled by a single entity. However, having RIA issue personal identification documents could be construed as a change in policy.
- PPA could also act as both the Wallet and the PID provider. This would have the same benefits as the previous option.
- Some third party such as an eIDAS Trust Service Provider could act as the PID provider.
- PPA could act as the registration authority while a Trust Service Provider could handle the issuance of attestations. This would be most similar to how ID-cards are currently issued.

PPA has currently been able to provide the following input:

- It might be necessary to support PID Attestation provision in a physical setting (user is identified at a service office).
- Smart-ID would likely not be suitable as a basis for issuing a PID Attestation as it is operated by a private sector entity.

## End User

End users in this context refer to any person that uses the wallet and also any person that has the right to apply for a wallet. The three main interests of end users are convenience, security and privacy.

The main interest that will likely drive the adoption of wallets by the end users is how convenient they are to use. In this regard the strongest competitor in Estonia is Smart-ID. Some notable strengths that Smart-ID has are that it is somewhat easier to sign up for than Mobile-ID and compared to the ID-Card it is much less burdensome to use as it does not require having access to multiple physical devices (ID-card, ID-card reader, computer) to use. Convenience for users is also determined by the utility of the wallets - which RPs accept the wallet as an eID means.

For users the main concern over security relates to identity theft. Users must be guaranteed that their identity could not be assumed by an attacker. Notably, even users that have opted not to use wallets need protection in this regard. For example, in a scenario where an attacker has temporarily gotten hold of a victims LoA high eID means, they'd be able to create a wallet in the victims name. As a mitigation of this, the users should always be notified when a wallet has been issued under their name. Phishing attacks during both registration and authentication are also a major concern for users. Security best practices and relevant security measures should be used to mitigate such threats.

Different users might have very different privacy concerns. In an ideal scenario however, no other party - either public or private sector - would obtain any data about the transactions the user makes using their wallets. Users should also be able to minimize the data they provide to RPs. For this selective disclosure of attestations would be used. Users with privacy concerns should also be able to opt out of using a wallet. In practice however, this might be difficult to guarantee. It is realistic to assume that not using a wallet would cause users considerable disadvantage making it not feasible for them to opt out of using them. For example, it might be difficult to ensure that all RPs provide alternatives to wallets when using their services.

## Relying Parties

Relying Parties are the stakeholders who will gain the most value from the implementation of wallets as for the end users the underlying goal when using a wallet is to access a service that an RP provides. Indeed, the main reason for an RP to accept wallets would be that it reduces the cost for providing their service. This could be directly due to reducing the overhead on running an authentication service. The benefit could also be indirect. The increased security that wallets provide is likely to reduce damages that RPs incur due to security incidents - this is probably most evident in the financial sector where securing payments clearly reduces fraud. Furthermore, having a LoA high eID mean available is a precondition for running many digital services, for example most state portals.

Another reason that would push RPs to use wallets is needing to comply with the legislation mandating that RPs have to interface with wallets. However, currently it is unclear what would be the benefits for the Union for such a mandate. One proposed reason is that some very influential potential RPs such as Google and Facebook are currently also acting as Identity Providers which could potentially compete with EUDI Wallet. Such RPs might not be willing interface with wallets as it might damage their interests as identity providers.

## References

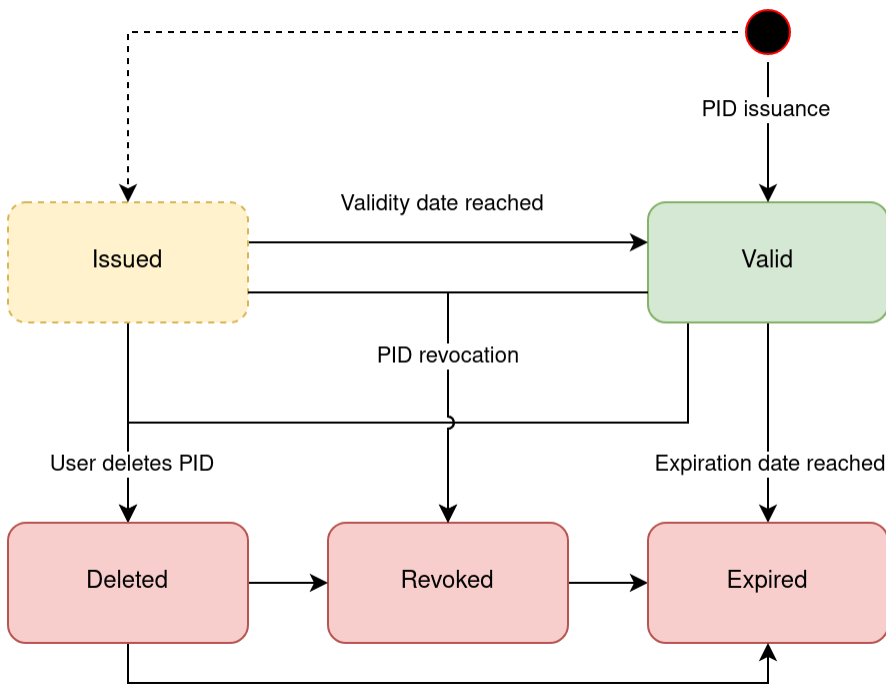
- [EIDAS1] - [https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L\\_.2014.257.01.0073.01.ENG](https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG)
- [EIDAS2] - <https://www.europarl.europa.eu/cmsdata/278103/eIDAS-4th-column-extract.pdf>
- [ARF1.3] - confidential
- [LoA] - <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02015R1502-20220711>
- [XPRT-GROUP] <https://ec.europa.eu/transparency/expert-groups-register/screen/expert-groups/consult?lang=en&groupId=3032>
- [TOOLBOX-MANDATE] <https://eur-lex.europa.eu/legal-content/GA/TXT/?uri=CELEX:32021H0946>

## 2. Lifecycle Diagrams

Pages under this space describe the different stages in the lifecycles of the core business objects *EE PID Attestation* and *EE Wallet Instance*. These diagrams put into context the various processes described within this analysis.

# EE PID Attestation Lifecycle

## EE PID attestation lifecycle



State	Transition to	Description
Valid	Described in the PID issuance process: 3.1 EE PID Attestation Issuance <i>TBD - explain transition from state Issued</i>	In this state a PID is valid and can be used for authentication.  <i>The EE PID Provider issues PID Attestations with validity periods starting immediately after the issuance. Therefore the attestations skip the Issued state described in the ARF.</i>
Issued	Described in process TBD.	In this state a PID Attestation is not yet valid for authentication, but it will become valid at some later time.  <i>This state might be dropped after the re-issuance process has been analyzed. In that process a new PID could be provisioned alongside a PID that is about to expire. That new PID could start its validity at the moment the old PID expires.</i>
Revoked	Described in PID revocation process: 3.2 EE PID Attestation Revocation	In this state the PID Attestation has been revoked before reaching its expiration date. This PID is not valid for authentication - an RP must always validate that a PID Attestation presented to them is not revoked.  <i>Technically it is possible to revert the revocation of a PID Attestation, however currently it has not been identified as a requirement.</i>
Deleted	PID transitions into this state when the user deletes the PID from their device, either by uninstalling the Wallet application or deleting the PID from their wallet.	In this state the PID Attestation nor the cryptographic keys it refers to no longer exists and therefore it can not be used for authentication. No further steps are needed to revoke the PID.
Expired	Once the PID reaches its expiration date, the PID will automatically enter this state.	This is the final state for any PID in which a PID is no longer valid for authentication. An RP must always validate that a PID Attestation presented to them is not expired by checking the expiration date.  If a revoked PID expires, the PID Provider does not need to keep publishing its revocation information as the PID will be considered invalid regardless.

## EE PID Attestation validity period

There can be different type of PID Attestation validity periods which can affect the PID Attestation issuance, revocation, presentation processes and mechanisms. In this project PID Attestations with short term or long term validity periods were under initial discussion and their main differences are brought out below.

### Short term PID Attestation

This type of PID Attestation is valid only within a short time frame - it is requested from the PID Provider shortly before its presentation to the Relying Party. Having a separate revocation mechanisms for this type of PID Attestation could be potentially avoided by having the PID Attestation expire shortly after its issuance. Example validity period of a short lived PID Attestation could be 5-15 minutes, but it is possible to have different validity periods (ARF currently limits max validity period to 24 hours for short term attestations, but does not currently determine max validity period specifically for the short term PID Attestation. Max validity period for short term PID Attestation would need to be further analysed and discussed). This also means that when the short term PID Attestation is presented, then the Relying Party can skip the revocation check as the PID Attestation is issued shortly before the RP receives it.

It is also possible that this type of PID Attestation is only presented to one or limited number of RPs as opposed to PID Attestation with long validity period. This could be beneficial for privacy in some anonymous authentication use cases, as it could mitigate risk of the RP profiling the user internally or colluding with other RPs to track/profile the user. This is only relevant if a subset of PID Attributes (most notably the age over 18 attribute) are presented during the anonymous authentication that do not enable to learn the true (legal) identity of the user. If in the mentioned anonymous authentication use case some PID Attributes, that do not disclose the identity of the user, would be presented based on a long term PID Attestation, then tracking/profiling could be done, for example, by matching some PID Attestation unique identifiers over time (for example, public key associated with the PID Attestation during its issuance). Downside for the short term PID Attestations is that the PID Provider would learn when and how often the user presents the PID Attestation, but at the same time PID Provider would not learn to whom it is presented to.

## Long term PID Attestation

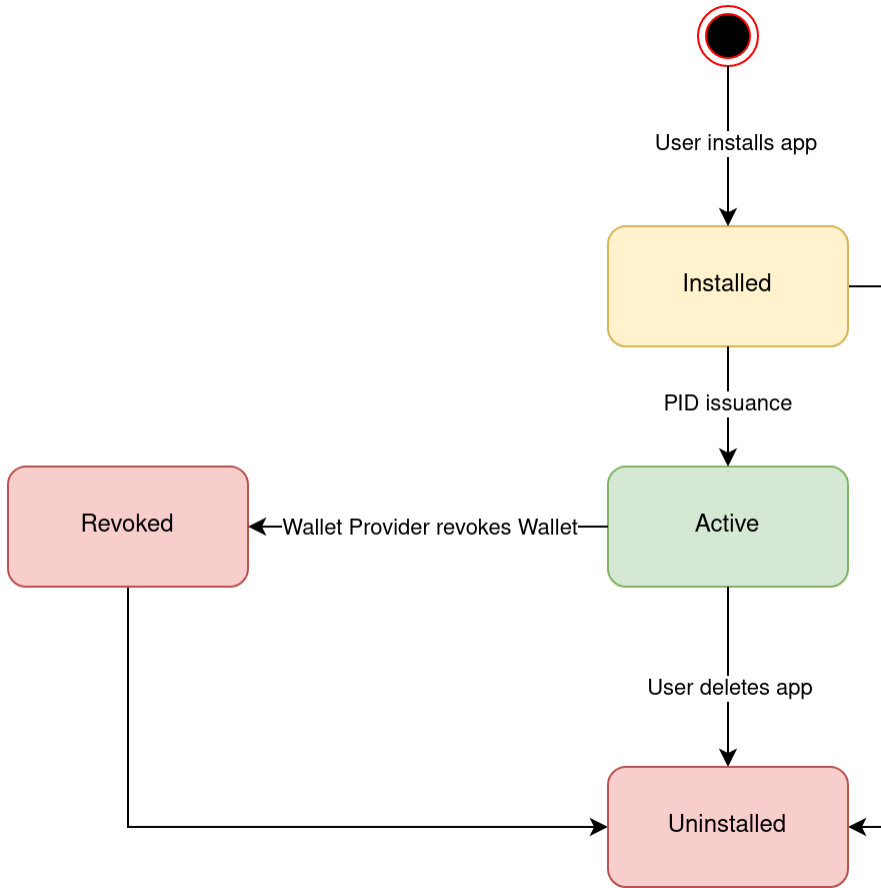
This type of PID Attestation is valid for a longer time period. After it is requested from the PID Provider it is stored in the Wallet Instance. For example, the validity period of this kind of PID Attestation could be days, weeks or years. During this validity period PID providers may decide that the PID needs to be revoked, therefore there needs to be a revocation mechanism specified for such attestations. This means the PID Provider must make the revocation information public which is some extra work compared to the PID Attestations with short validity period, but on the other hand, the PID Provider does not need to issue PID Attestations so often.

If the user uses the long term PID Attestation, then the PID Presentations released to the Relying Parties over time will be based on the same PID Attestation. That can have some potential privacy implications that were discussed in the previous paragraph when anonymous authentication is required and the RP does not learn the true (legal) identity of the user based on the PID Attributes it receives. However, it is possible that long term PID Attestation could still be used in the privacy-focused cases, but then a batch of long term PID Attestations would be issued to the user's Wallet Instance and each PID Attestation from the batch can only be used once. If the user runs out of PID Attestations, then a new batch needs to be requested. But in cases where the Relying Party is required to learn the true (legal) identity of the user to offer its services, then the long term PID Attestation usage is not a problem, as then the short term PID Attestation does not add any value in regards to privacy because the RP knows the true (legal) identity of the user interacting with them anyway.

This analysis focuses on the use cases where the RPs have the right or even a requirement to learn the true (legal) identity of the user during the user authentication based on the PID Attestation and therefore the short term PID Attestations or long term PID Attestations issued in batches (PID Attestation used only once) are not currently under consideration as it is not seen they add extra value. Therefore this analysis focuses on EE PID Attestation with long validity periods that can be used repeatedly. This would match the current practices for the Estonian ID-card, Mobile-ID and Smart-ID certificates which all use long validity periods (5 years). If the previously described anonymous authentication use cases are required, then the short term PID Attestations or batches of long term PID Attestations could be further analyzed.

---

# EE Wallet Instance Lifecycle



State	Transition to	Description
Installed	User installs the wallet application	In this state the wallet is a fresh application which contains no personalized data.
Active	The wallet is activated during the PID issuance process: 3.1 EE PID Attestation Issuance	In this state the wallet has been personalized - it has been associated with a unique identity. Associating the wallet with a person allows that person to manage their wallet (eg: request revocation) even when they lose access to the phone their wallet instance is installed on.  The wallet in this state may or may not contain a valid PID, but the wallet should be functional nevertheless. This might become relevant once the analysis includes other credentials in scope.
Revoked	Wallet is revoked by the Wallet Provider: 3.3 EE Wallet Instance Revocation	In this state the Wallet has become permanently unusable as the Wallet Provider has revoked it.
Uninstalled	The user uninstalls the wallet application which also deletes any locally held data.	The wallet no longer exists and any attestations within have been deleted making them unusable.  Note that the attestations issued into the wallet nor the wallet instance itself do not need to be revoked after the app is uninstalled as they become unusable when the associated key material is deleted.

## 3. Process Models

The subchapters in chapter 3 describe the proposed business process models for the main processes related to PID Attestation issuance, use, and revocation and also processes related to the Wallet Instance activation and revocation. Below is a list of the described processes.

- EE PID Attestations issuance (includes EE Wallet Instance activation by the EE Wallet Provider).
  - Two more detailed versions of this process are included as annexes *A1 EE PID Attestation Issuance (cross-device)* and *A2 EE PID Attestation Issuance (same-device)*.
- EE PID Attestation revocation.
- EE Wallet Instance revocation.
- Authentication using the EE Wallet & EE PID Attestation.
- EE Relying Party Registration.

The described process models demonstrate only the successful flow, with the exception that some more important alternative paths and exceptions are brought out separately either on the process model diagrams or in the textual descriptions.

# 3.1 EE PID Attestation Issuance

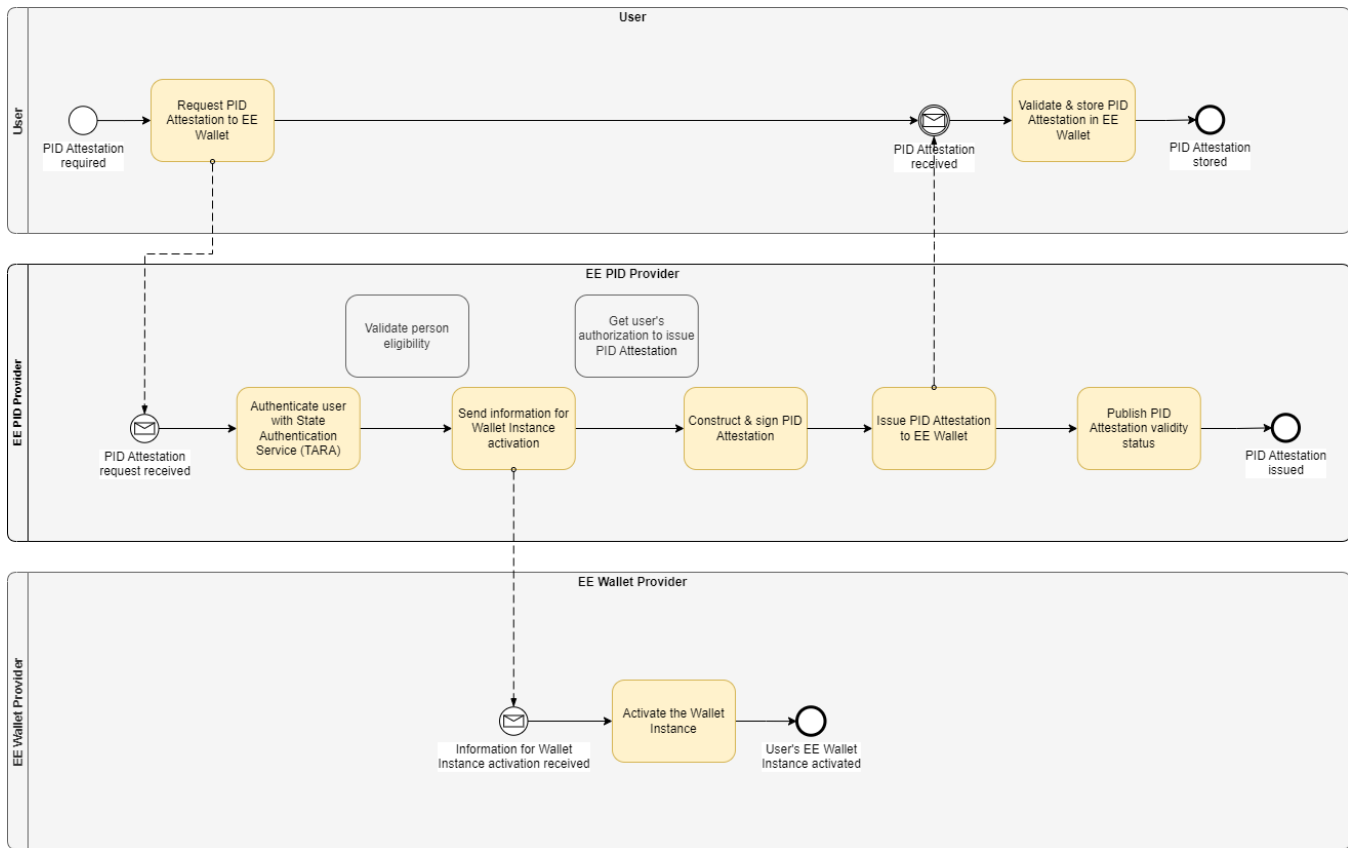
## Scope

This chapter describes the proposed business process for issuing an EE PID Attestation to the EE Wallet Instance. Activating the Wallet Instance by the Wallet Provider is also part of this process flow. Please note that separate steps for the establishment of trust between the actors is not currently described on this diagram. This might be changed at some point when the ARF specifies in more detail how these trust relations should work and discussions with stakeholders participating in this process are conducted. For example, Trusted List(s) could be leveraged as one option to establish some trust relations.

There are also more detailed PID Attestation Issuance process models that describe different flows depending on whether the user is interacting with the PID Provider and the Wallet Instance on different devices or on the same device. These flows also propose more details about different aspects of the process, but that requires making more assumptions and thus they are also more likely to be subject to change in the future.

- A1 EE PID Attestation Issuance (cross-device) - Describes PID Attestation issuance flow where the user interacts with the PID Provider and the Wallet Instance on the **same device**.
- A2 EE PID Attestation Issuance (same-device) - Describes PID Attestation issuance flow where the user interacts with the PID Provider on **one device** and with their Wallet Instance on **another device**.

## Process Model



## Process Description

The table below describes the activities (steps) of the process model.

Name of the activity	Description
----------------------	-------------

<p>Request PID Provider to issue PID Attestation to EE Wallet</p>	<p>User requests a new PID Attestation to be issued by the PID Provider into their EE Wallet Instance.</p> <p><i>There are different flows that the user can take to achieve this. The flows differentiate based on whether the user is conducting the PID Attestation Issuance only on one device or uses two devices. More detailed processes describing the same and cross device flows can be seen here (note that since they propose more details then likelihood for some changes in the future are higher than for the more general PID Attestation issuance process depicted above) :</i></p> <ul style="list-style-type: none"> <li>• A1 EE PID Attestation Issuance (cross-device)</li> <li>• A2 EE PID Attestation Issuance (same-device)</li> </ul>
<p>Authenticate user with State Authentication Service (TARA)</p>	<p>PID Provider uses the State Authentication Service (technical name is TARA) for user authentication. PID Provider retrieves user identity data from the authentication result. Identity data is used for creating the PID Attestation.</p> <p>Whether the PID Provider accepts all the main Estonian eID means (ID-card, Mobile-ID, Smart-ID) to be used by TARA for user authentication, is yet to be decided. However, the Estonian ID-card shall be one of the options and it is proposed that when the user wants to identify themselves in TARA with an ID-Card, then in addition to using the smart-card reader such as currently done in Estonia, it should also be possible to authenticate with the ID-card using the NFC capabilities of the mobile device and the ID-Card. This means the user does not have to insert an ID-card to a reader hardware, but instead could use the ID-card with the same device that they have installed their EE Wallet Instance on by leveraging the NFC technology.</p>
<p>Validate person eligibility</p>	<p><i>Potential step, necessity and details of this step are to be decided, not explicitly specified in ARF.</i></p> <p>PID Provider validates that the person applying for the PID Attestation is eligible to receive one.</p> <p>Examples of potential checks:</p> <ul style="list-style-type: none"> <li>• Confirm person is an Estonian citizen</li> <li>• Confirm person has a resident permit</li> <li>• Confirm person has active legal capacity</li> </ul> <p>Another example is that that the PID Provider validates person data (or queries extra data) by contacting the Population Registry.</p>
<p>Send information for Wallet Instance activation</p>	<p>PID Provider sends the user identity data and information about the user's Wallet Instance to the Wallet Provider. This is necessary so that the Wallet Provider can activate the Wallet Instance.</p> <p>The exact details, about the how the Wallet Instance data is received by the PID Provider and what it includes, depend on the technical implementation mechanisms. One possible option is that the Wallet Instance prepares the necessary information together with the Wallet Provider, that is then passed on to the PID Provider. This option is also depicted in the more detailed process models:</p> <ul style="list-style-type: none"> <li>• A1 EE PID Attestation Issuance (cross-device)</li> <li>• A2 EE PID Attestation Issuance (same-device)</li> </ul>
<p>Get user's authorization to issue PID Attestation</p>	<p><i>Potential step, necessity and details of this step are to be decided, not explicitly specified in ARF.</i></p> <p>The PID Provider asks the user to continue the process by confirming PID Attestation issuance to the Wallet and receives user's confirmation.</p> <p>For example, the user could sign an acceptance contract that documents the user agreement to PID Attestation issuance. Requiring a signature could potentially also stop some phishing attacks. A victim might erroneously grant access to the attacker during the authentication step, but requiring a signature shortly after, might help the victim realize that they are being manipulated.</p>
<p>Activate the Wallet Instance</p>	<p>Wallet Provider has received information about user's identity and user's Wallet Instance from the PID Provider.</p> <ul style="list-style-type: none"> <li>• If the user is issuing the PID Attestation to their freshly installed Wallet Instance the first time, then the Wallet Provider needs to locally associate the user's identity with the Wallet Instance. This is needed for example when the user loses access to their wallet and wants to revoke it, then they can contact the Wallet Provider and request their Wallet Instance to be revoked. After this association is completed the Wallet Instance is activated.</li> <li>• If the Wallet Provider determines that the user is issuing the PID Attestation to a Wallet Instance that has previously already been activated, then the Wallet Provider checks that the identity provided by the PID Provider is the same that has been associated with the Wallet Instance previously. This check is required as it is proposed that only one identity can be associated with one Wallet Instance throughout the Wallet Instance lifecycle.</li> </ul> <p><b>Note</b> that the current version of ARF describes: <i>"In general, the EUDI Wallet Provider does not need to know the true identity of the User. An alias, for example an e-mail address, should be sufficient. However, the EUDI Wallet Provider may request the true identity of the User to be able to offer additional services. It is up to the EUDI Wallet Provider to determine the conditions for creating an online account, and to the User to accept or refuse these conditions."</i> This means that it is also possible to set up the user account without the Wallet Provider knowing the identity of the user, for example, by using username/password type of solutions or leveraging the possibility of creating an user account based on a user pseudonym. Based on the preliminary discussions, the assumption in this project is made that the user account is created based on the real identity of the user. This enables the Wallet Provider to use strong authentication means to authenticate the user when the user wants to perform Wallet management activities like revocation of a Wallet Instance. If any future updates on ARF determine new aspects about this or some privacy related aspects arise, then it could be subject to change. Future changes could also affect when the Wallet Instance activation is performed, for example, it could be separated and take place prior to the PID Attestation Issuance process.</p>

Construct and sign PID Attestation	PID Provider constructs and signs the PID Attestation. PID Attestation must be bound to a user, meaning that only the user whom the attestation is issued to, can present it. The option currently proposed to achieve this is that the PID Attestation is associated with the key pair generated by the Wallet Instance (for the PID Attestation). Key Pair consists of a public and private key. Public key will be shared with the PID Provider and it is currently proposed this is done by means of Key Attestation. It is important to note that the Key Attestation can only be generated if the Wallet Instance is activated and has not been revoked.
Issue PID Attestation to Wallet Instance	PID Provider sends the PID Attestation to the Wallet Instance.
Publish PID Attestation validity status	PID Provider makes the information about the PID Attestation status available. This information is required by the Relying Parties when they want to check that the PID Attestation presented to them is still valid and has not been revoked by the PID Provider. It could potentially also be used by the Wallet Instance to check PID Attestation revocation status.
Validate & store PID Attestation in EE Wallet	<p>User validates that the details in the PID Attestation are correct and consents to storing it in the EE Wallet.</p> <p>It is also possible that after the user's EE Wallet Instance received the EE PID Attestation it validates that the PID Provider, who issued the PID, is an legitimate EE PID Provider and has the right to issue PID Attestations. One option for this validation to be conducted is contacting the EE Trusted List of Providers, that would include information necessary to perform the validation. But this depends on the trust mechanisms used between the parties and is not currently specified.</p> <p>Secondly, the Wallet Instance could also validate that the received EE PID Attestation is not revoked. This validation could be done using the same mechanisms as RPs use during PID presentation. Wallet Instance could query the received EE PID Attestation status from the source where the EE PID Provider published this information.</p> <p>If the EE PID Attestation's <i>valid from</i> date arrives some time after its issuance, then the Attestation should not be presentable before that date. Whether such PID Attestations are a going to be used is to be decided.</p>

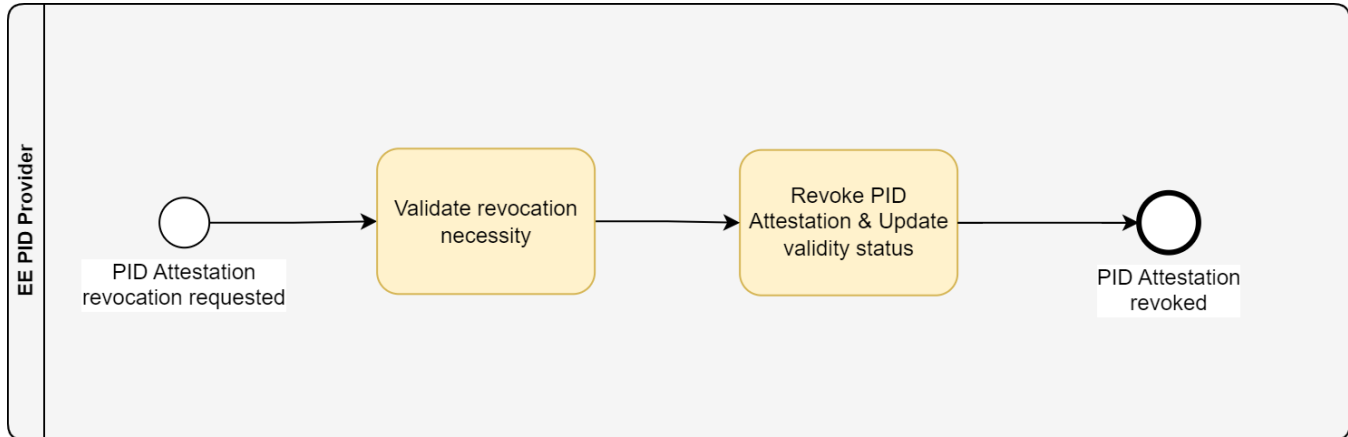
## 3.2 EE PID Attestation Revocation

### Scope

This chapter describes the business process of PID Attestation Revocation. In scope currently is only the revocation of the PID Attestation, how its revocation could affect other type of attestations issued to the Wallet Instance is currently not under consideration. In addition, it is currently assumed that PID Attestation can only be revoked and not suspended as suspension shall not be performed according to ARF.

### Process Model

PID Attestation revocation process can have multiple different triggers that start the process but the revocation process itself is similar regardless of what triggered it. Examples of possible triggers are described in the next section. The different parties that could request the PID Attestation revocation are not depicted on the model itself but are also brought out separately in the next section which describes the example triggers.



### Triggers

Below is an example list of triggers that could start the PID Attestation revocation process:

- **PID Attribute value change** - PID Provider learns that a value of some attribute in the PID Attestation has changed and that causes a need to revoke the PID Attestation. For example, a person's name could change. This information could potentially come from another party, for example, the person notifies when they changed their name or it comes from some registry.
- **User lost access to the Wallet Instance** - PID Provider learns that the user has lost access to their Wallet Instance and requests to revoke their PID Attestation.
  - User could contact the PID Provider directly and request PID Attestation revocation.
  - Wallet Provider could act as an intermediary. User contacts the Wallet Provider and asks to revoke the Wallet Instance or attestations. PID Provider will receive the information that the Wallet Instance has been revoked and therefore the PID Attestation should be revoked. Different mechanisms are possible to achieve this, possibly some intermediaries are used.
- **PID Provider initiated revocation** - PID Provider determines a need to revoke the PID Attestation. For example, because the PID Provider learns that fraudulent activity was performed by the user during the that PID Attestation issuance or some technical issues require PID Attestation revocation.

List of triggers need to be defined and agreed with the EE PID Provider and other relevant stakeholders.

### Process Description

In the table below, each activity in the process model is described.

Name of the activity	Description
Validate revocation necessity	PID Provider has received a revocation request that specifies which PID Attestation needs to be revoked and the reason for revoking the PID Attestation. PID Provider validates that the reason for the PID Attestation revocation is justified. Making that decision depends on the business rules that will be defined for the EE PID Attestation revocation.

<p>Revoke PID Attestation &amp; Update validity status</p>	<p>PID Provider revokes the PID Attestation and updates the validity status by marking the PID Attestation status as revoked. When the RP-s query the revocation status of the revoked PID Attestation, they will know that the PID Attestation is no longer valid.</p> <p>One option is that the PID Attestation revocation solution is operated by a third party, but even in this case the PID Provider shall still be the one that makes the decision about the revocation and this decision is communicated to the party operating the revocation solution.</p> <p><i>It is a possibility that the Wallet Instance, where the PID Attestation had been issued, learns about the revocation and will not allow user to present the revoked PID Attestation to the RPs. However, this does not imply that the RPs would not have to check the revocation status themselves.</i></p>
--	--

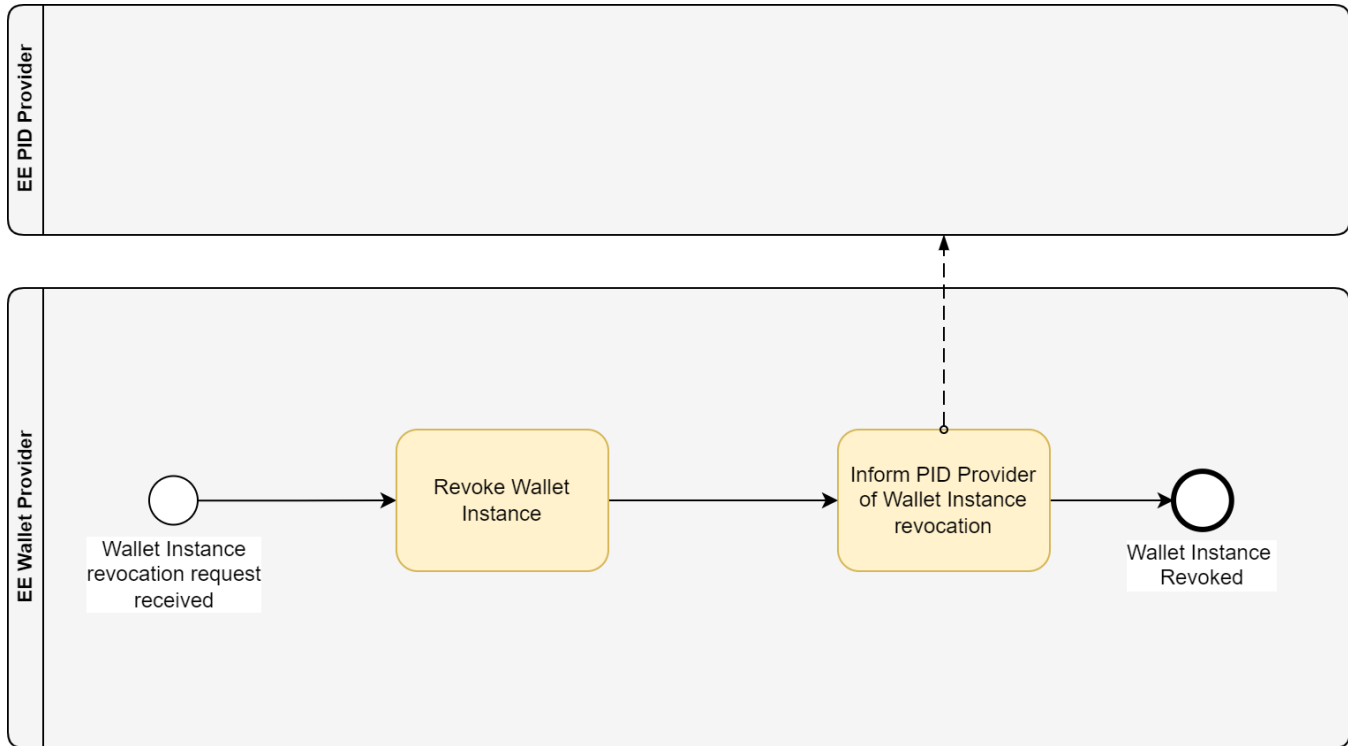
# 3.3 EE Wallet Instance Revocation

## Scope

This process model covers the revocation of an EE Wallet Instance. It is only considered how the Wallet Instance revocation affects the PID Attestation validity status. How a Wallet Instance revocation could affect other types of attestations is not currently in scope.

## Process Model

Wallet Instance revocation process can have multiple different triggers that start the process but the revocation process itself is similar regardless of what triggered it. Examples of possible triggers are described in the next section. The different parties that could request the Wallet Instance revocation are not depicted on the model itself but are also brought out separately in the next section which describes the example triggers.



## Triggers

Below is an example list of triggers that can start the Wallet Instance revocation process.

- **User request** - User requests the Wallet Provider to revoke their Wallet Instance.
  - User authenticates to the Wallet Provider online and requests their Wallet Instance to be revoked, for example, due to losing the device where the Wallet Instance was installed on. User account at the Wallet Provider, that enables the user to interact with the Wallet Provider, was created during the first time issuing a PID Attestation to the Wallet Instance.
  - User requests Wallet Instance revocation/deletion in their Wallet application.
  - User contacts the Wallet Provider by calling a helpline and requests Wallet Instance to be revoked.
- **User deleting their user account at the Wallet Provider** - If the user deletes their account at the Wallet Provider then the Wallet Provider also revokes all active Wallet Instances associated with the user.
- **Wallet Provider detects a need for revocation** - Wallet Provider detects internally that a Wallet Instance needs to be revoked, for example, when some security risks involving the Wallet Solution or a specific Wallet Instance are detected.

## Process Description

In the table below, each activity in the process model is described.

Name of the activity	Description
----------------------	-------------

Revoke Wallet Instance	Wallet Provider has received a Wallet Instance revocation request which specifies the Wallet Instance that needs to be revoked. Wallet Provider revokes the Wallet Instance. Key Attestations can no longer be issued for the revoked Wallet Instance. Exact mechanisms used for the revocation of the Wallet Instance are not clearly defined in the current version of ARF and are therefore also not further elaborated in this analysis.
Inform PID Provider of Wallet Instance revocation	<p>Wallet Provider informs the PID Provider about the PID Attestations that are associated with the revoked Wallet Instance. PID Provider should revoke those PID Attestations. Exact mechanisms used for informing the attestation providers are not clearly defined in the current version of ARF and are therefore also not elaborated on in this analysis.</p> <p>ARF notes that it shall be implemented in a way that the Wallet Provider would not learn about the attestations present on the user's Wallet Instance so that the Wallet Provider cannot create a user profile based on the attestations present on the Wallet Instance.</p>

# 3.4 Authenticate using EE Wallet & EE PID Presentation (Remote flow)

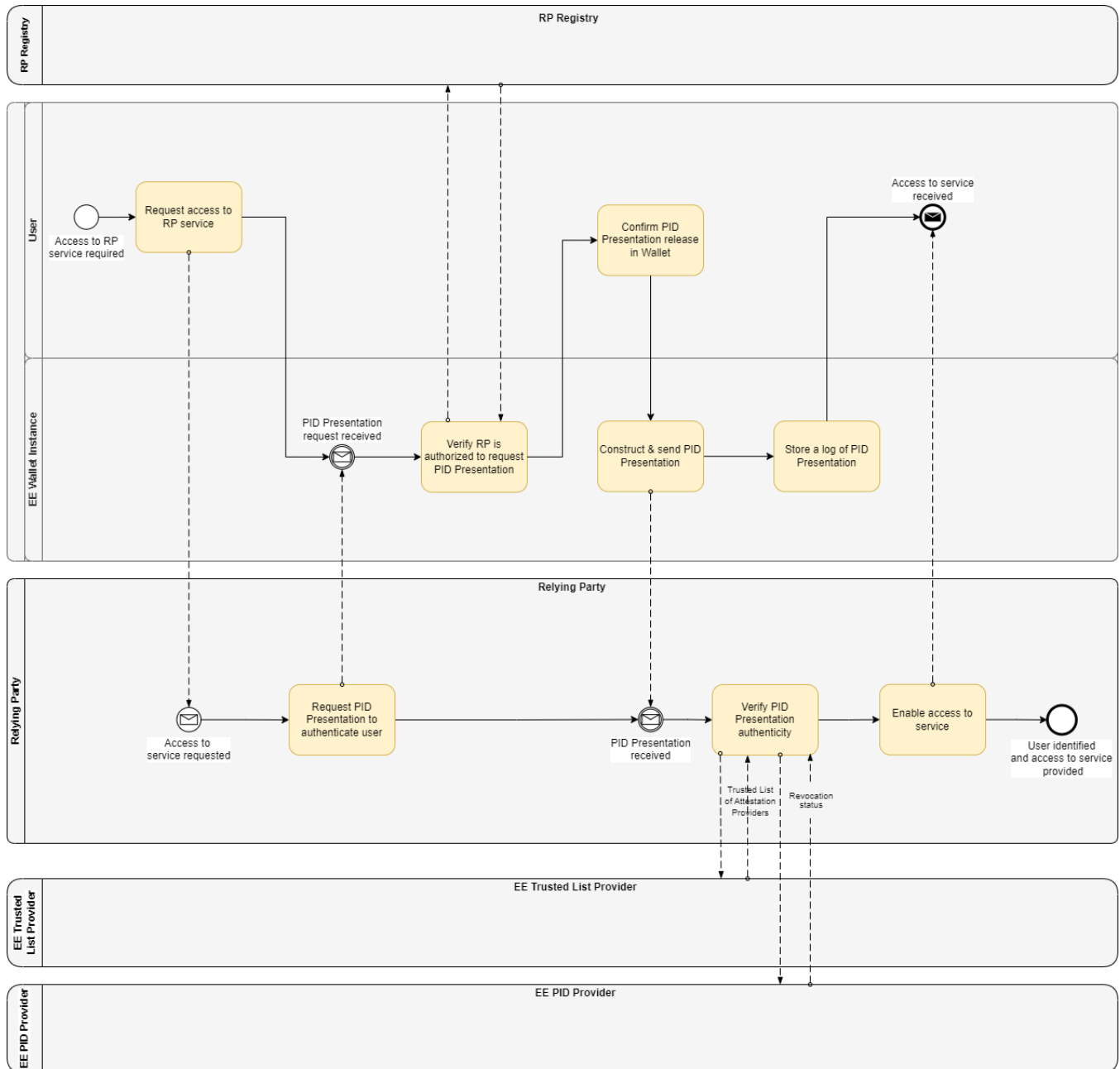
## Scope

This chapter describes the process of a natural person presenting an EE PID Presentation from their EE Wallet to a Relying Party (RP) in order to authenticate themselves so that the RP learns the legal identity of the user. This process describes a remote flow, meaning that the data exchange between the user's Wallet and the RP takes place over the Internet. Cases where a RP is another natural or legal persons European Digital Identity Wallet are currently out of scope.

Remote online presentation can be performed either by:

- same-device flow, meaning that the user interacts with the RP and their EE Wallet Instance on the same device;
- cross-device flow, meaning that the user interacts with the RP and their EE Wallet Instance on different devices.

## Process Model



## Process Description

In the table below, each activity in the process model is described.

Name of the activity	Description
Request access to RP service	User has accessed the RP web portal or application and chooses to authenticate using their EE Wallet in order to use the RP services.
Request PID Presentation to authenticate user	<p>RP has received a request from the user to access its services. RP requests a PID Presentation from the EE Wallet Instance in order to authenticate and identify the user and enable access its services. RP also signs the PID Presentation request.</p> <p>It is possible that in some cases the RP might request only a subset of PID attributes and differentiate between which requested PID attributes are mandatory (have to be received by RP) and which are optional (user can choose to decline sharing them).</p> <p>The ARF foresees that the RP might want to authenticate that it is communicating with an authentic EUDI Wallet Instance. However, for the EE Wallet we currently believe that such a check is unnecessary. It would be the responsibility of the PID Provider to only issue PID Attestations into authentic wallets. The RP should only rely on the trust relationship it has with the PID Provider and not verify the authenticity of the Wallet Instance. One reason for this is that it might not be technically feasible for the RP to verify a Wallet Instance's authenticity. Such verification might also introduce new privacy concerns.</p> <p>The process diagram above depicts the activities performed the user and the EE Wallet Instance in the same-device authentication flow, meaning the user interacts with the RP and their EE Wallet Instance on the same device. If the user performs cross-device flow, meaning the user interacts with the RP and their EE Wallet Instance on separate devices, then before the RP requests the PID Attestation it is necessary to perform extra steps for establishing communication with the EE Wallet Instance. One option is that the RP displays a QR code and the user scans the QR code with their Wallet. Wallet can read the necessary information from the QR code to establish communication with the RP and the RP can send the PID Attestation request.</p>
Verify RP is authorized to request PID Presentation	<p>EE Wallet Instance has received a PID Presentation request from the RP.</p> <p>EE Wallet Instance:</p> <ul style="list-style-type: none"> <li>Authenticates the RP. According to ARF, if the authentication fails then the Wallet Instance shall not release PID Attestation to the RP.</li> <li>Verifies the RP is authorized to request the PID Presentation and also if the RP is allowed to request all of the requested PID attributes or only a subset of the PID attributes.</li> </ul> <p>One option for performing RP authentication and authorization is contacting the RP Registry that holds the information to authenticate the RP and information about what attributes the RP is allowed to request. This option is also depicted in the process diagram above. It is possible that the information form the RP Registry is obtained real time or at some point in time before the PID Presentation process, the former option is currently described in the process diagram.</p>
Confirm PID Presentation release in Wallet	<p>User confirms the release of the PID Presentation to the RP based on the information that the EE Wallet Instance displays, including the RP name and list of requested PID attributes. User shall also be authenticated by the Wallet Instance, it is not specified in this analysis how it is done, but different mechanisms are possible to achieve this (for example by using user authentication options present or connected to the device where the Wallet Instance is installed on, the Wallet implementing its own PIN code or biometrics capture, combination of previous two options etc).</p> <p>Regarding the mandatory attributes requested by the RP, the user can either accept or decline sharing them. If user declines, then the PID Presentation should not be sent to the RP.</p> <p>Regarding the optional attributes requested by the RP, the user can choose which ones to share with the RP.</p>
Construct & send PID Presentation	EE Wallet Instance constructs the PID Presentation and sends it to the RP. PID Presentation includes at least the PID Attributes which were marked as mandatory by the RP in the PID Presentation request. Proof that the PID Presentation is sent by the same person and thus also the same device that the PID Attestation was originally issued to is also generated by the EE Wallet Instance and included in the PID Presentation.
Store a log of PID Presentation	EE Wallet Instance logs the details of the PID Presentation transaction, like the information about the RP and information about what PID Attributes were presented. User can later view these logs in their Wallet.

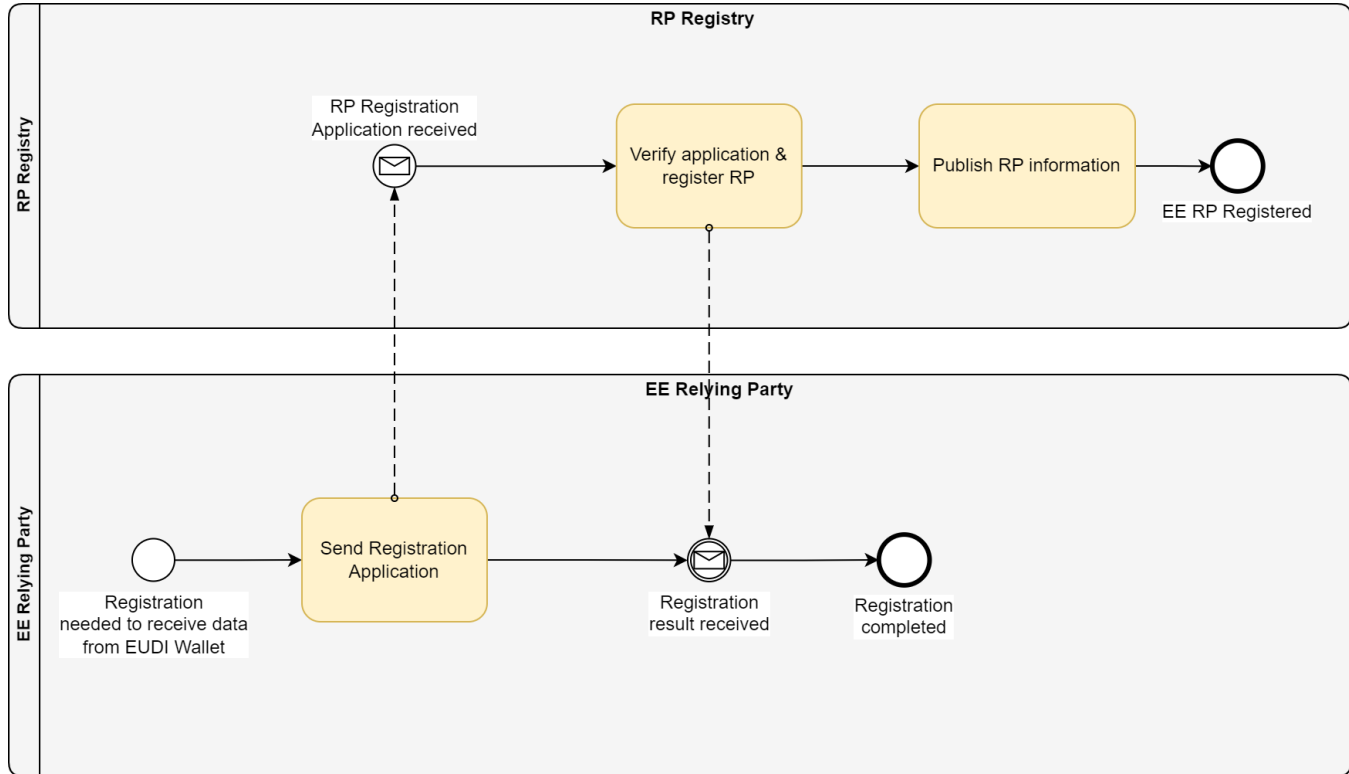
<p>Verify PID Presentation authenticity</p>	<p>RP verifies the authenticity of the received PID Presentation by performing the following activities.</p> <ul style="list-style-type: none"> <li>• RP verifies that the PID Provider is trusted. According to ARF, there are different options to achieve this: <ul style="list-style-type: none"> <li>◦ Retrieve information about PID Provider from a Trusted List of Attestation Providers - Provider of this Trusted List is a trusted party who has assessed that the PID Provider qualifies to issue PID Attestations. This information is documented in a Trusted List of Attestation Providers and signed by the the Trusted List Provider. Trusted List of PID Providers is available for the RPs. In the process diagram this option is depicted to achieve trust. ARF determines that there shall be one Trusted List of Attestation Providers only containing PID Providers, but it is not yet determined who will provide this list.</li> <li>◦ Establish trust with PID Provider separately - For this alternative option the RP would have to assess and decide themselves to trust the PID Provider. A self-signed PID Provider certificate is obtained directly from the provider in order to verify the PID Presentations. This approach is not prohibited for PID Attestations according to the current version of ARF.</li> </ul> </li> <li>• RP verifies that the PID Attestation is not revoked by the PID Provider - Depending on the revocation list mechanism, the information that enables revocation checking might not be obtained during the process real time, but is obtained previously. For example the revocation list could be downloaded periodically. The process model currently depicts the real time option for requesting the revocation status.</li> <li>• RP verifies the PID Provider signature over the PID Attestation.</li> <li>• RP verifies that the PID Presentation is presented by the same person that the PID Attestation was originally issued to.</li> </ul> <p>ARF mentions that the RPs should have a possibility to check if the Wallet Instance is not revoked by checking the revocation status of the Wallet Instance Attestation. As the analysis focuses only on the PID Attestation type of attestations then it does not propose to use Wallet Instance Attestation as it is described in ARF, but instead Key Attestation concept is used during the PID Attestation issuance. It is also currently proposed in this analysis that the PID Provider would learn, without delay, about the revocation of the EE Wallet Instance and can therefore immediately revoke the related PID Attestation. This means that it would not be necessary for the RPs to perform separate Wallet Instance revocation check when a PID Attestation is presented to them.</p>
<p>Enable access to service</p>	<p>Authenticity of the PID Presentation is verified and the RP has obtained the required identity (user's legal identity) information. RP enables the user to access their service.</p>

# 3.5 EE Relying Party Registration

## Scope

When a Relying Party relies on the European Digital Identity Wallets to offer a service, then it needs to notify the Member State where they are established. This kind of mechanism is also needed for EE Relying Parties and the proposed business process for that is described below.

## EE Relying Party Registration Process



## Process Description

This section describes the activities of the process model.

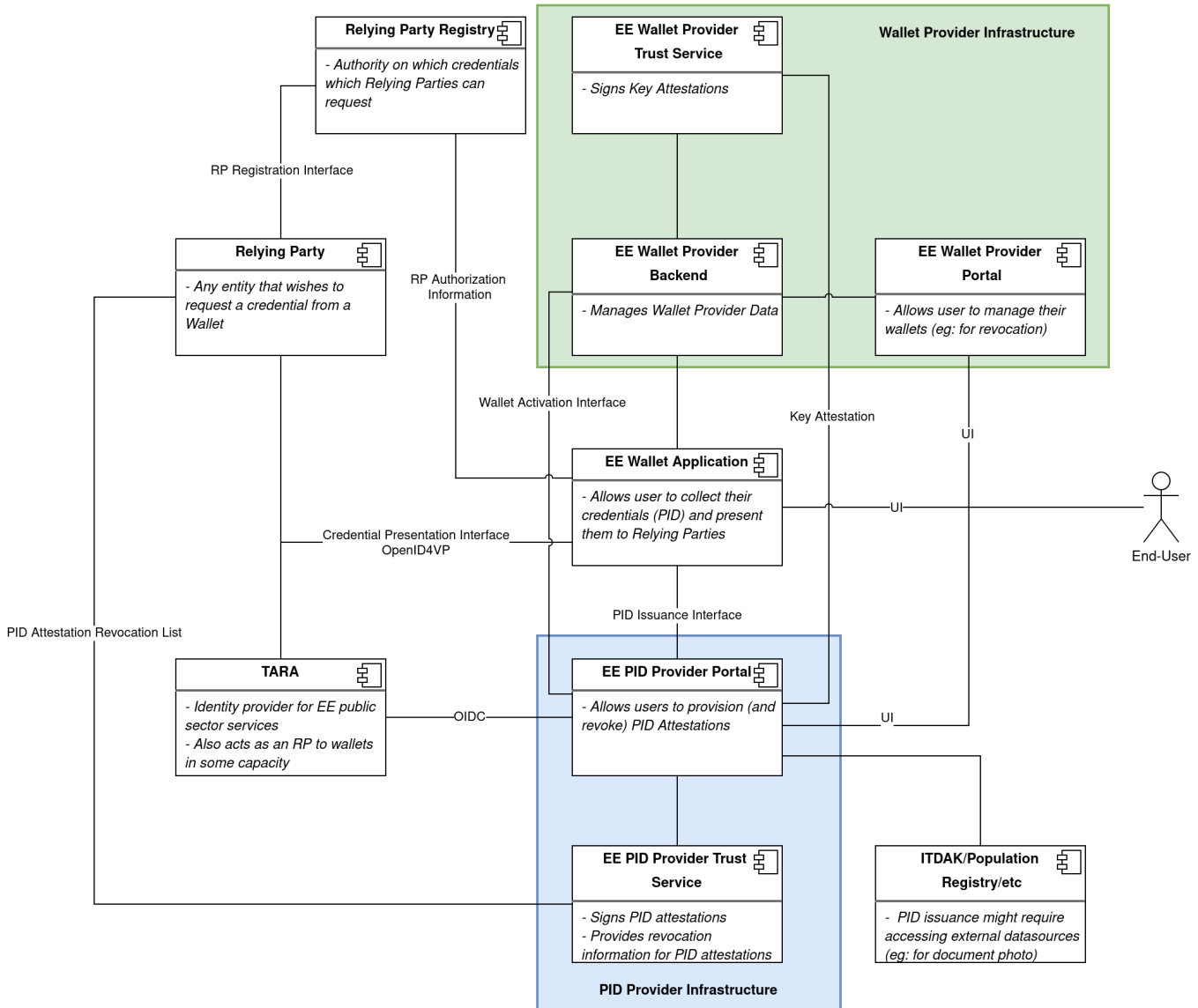
Name of the activity	Description
Send Registration Application	<p>EE Relying Party has detected a need to use data from the EUDIW Wallet in order to offer its service. EE Relying Party creates and sends a Registration Application to the RP Registry. According to current draft version of eIDAS 2, the Registration Application includes at minimum:</p> <ul style="list-style-type: none"> <li>the information necessary to authenticate to European Digital Identity Wallets, which as a minimum includes:               <ul style="list-style-type: none"> <li>the Member State in which they are established and</li> <li>the name of the relying party and, where applicable, its registration number as stated in an official record together with identification data of that official record;</li> </ul> </li> <li>contact details;</li> <li>the intended use of the European Digital Identity Wallet, including the data to be requested.</li> </ul> <p>In the current scope only PID Attestation can be requested by the RP.</p>
Verify application & register RP	<p>RP Registry checks the application and verifies that the RP qualifies to be registered. Specifics about how exactly the verification procedure is conducted is currently undefined. This can be further specified when ARF has defined more concrete steps and the stakeholders involved also agree on the specifics. After the RP is registered, the RP is notified of successful registration.</p>

Publish  
RP  
information

RP Registry makes the information available to the European Digital Identity Wallets. Information that can be used to check if the RP is authorized to request data from the EUDI Wallet users.

# 4. Architecture of The EE Wallet

This page describes the architecture - the components and the interfaces between them - of the systems surrounding the EE Wallet. The following diagram summarizes the architecture into a single image while the descriptions below provide more context. As currently the system analysis for the EE wallet has not been concluded this architecture represents more of a potential vision than a set in stone proposal. Thus, any decisions described here are open for change.



## Components

### EE PID Provider Infrastructure

The internal architecture of EE PID Provider's systems cannot currently be described to full detail as much is still open regarding how the responsibilities between the PID and Wallet Provider will be shared in Estonia. Nonetheless, some initial assumptions about which systems the PID provider needs to deploy can be made.

### EE PID Provider Trust Service

EE PID Provider will need to handle PID attestation signing keys. Having access to these keys will allow creating valid PID attestations. Therefore, these keys have to be protected in a manner such that they can not be duplicated and such that the usage of these keys can be audited/monitored. It is therefore reasonable to encapsulate handling of these keys into a separate component. To protect the keys, this component would use a Hardware Security Module (HSM). For monitoring the component would need to keep a strict audit log. Storing information about which PID attestations have been issued would also likely be the responsibility of this component.

The PID Provider also needs to publish revocation information for PID attestations. This revocation information would also include signed data which also involves handling some signing keys. For this similar security measures have to be taken. Whether this component publishes revocation information directly or hands it over to a different PID Provider component is currently an open question.

## **EE PID Provider Portal**

The PID Provider will also need to have an end-user facing component. The portal would act as a registration authority - it would authenticate end-users and let them request provision of PID attestations. In addition the user might also use the portal to request PID revocations.

For authentication it is currently foreseen that the portal relies of the State Authentication Service TARA as a LoA high eID means. For the issuance of PID the system might also need to request data from external data-sources such as the Population Registry.

During the PID provision process, the PID provider needs to communicate with different Wallet sub-components. The details of the interfaces other than the *PID Issuance Interface* between the Wallet Application and the Portal have currently not been decided. It is reasonable to assume that while the diagram above depicts the portal implementing all these interfaces this will be changed in the future. For example, the PID Provider Trust Service could communicate with the Wallet Provider Directly. Regarding the PID Issuance Interface a bit more can be said. It will definitely be an interface directly between the wallet application and the EE PID Provider Portal. The protocol it will use has not been decided. One promising candidate is the OpenID4VCI protocol, but as this is designed to be a general purpose protocol (for situations where the wallet provider and credential issuer are not expected to have prior agreements) it might be more prudent to design a simplified issuance protocol for the EE Wallet. In such a case, reusing the design principles from the OpenID4VCI protocol would still likely be beneficial.

## **EE Wallet Provider Infrastructure**

By and large, the Wallet Provider infrastructure mirrors that of the PID provider's infrastructure. However, as the requirements for the Wallet Provider are somewhat better understood than those of the PID provider, the components of the Wallet Provider have been described in slightly more detail.

### **EE Wallet Provider Trust Service**

The Wallet Provider will need to issue key attestations and possibly wallet attestations. As these are signed proofs, this again involves the handling of signing keys. These keys require the same level of protection as the PID provider's signing keys thus the same principles (HSM and audit log) are reused in this component.

### **EE Wallet Provider Backend**

The separation of the Wallet Provider backend from the Wallet Provider Trust service is somewhat arbitrary and might not turn out to be the correct approach. That being said, it is imagined that if the trust service component is responsible for managing keys (both key attestation signing keys and the keys the attestations are signed for) then the backend is responsible for managing the wallet instance data.

This component might also include some sort of administration interface which could mainly be used for revocation of wallet instances.

### **EE Wallet Provider Portal**

The Wallet provider would also need to provide an end-user facing component other than the wallet application itself. This allows the user to get an overview of all the wallets in their possession and possibly revoke them.

## **EE Wallet Application**

EE Wallet Application is a mobile application released by the wallet provider. As its instances are run on end-user devices it is somewhat separate from rest of the wallet provider's infrastructure. Each such instance would be in the control of a single end-user. Personal data of that end user such as PID attestations, attestation usage logs and other credentials would be kept strictly within the instance installed in the user's device.

Using this application the user can present their credentials to relying parties. For this, the wallet implements the Credential Presentation Interface which is the only interface that is described in this documentation in more detail.

Another important consideration is that the wallet provider would have to provide multiple different wallet applications for different platforms (Android, iOS) while the functionalities of these applications should be identical.

As an additional functionality the Wallet Application should also act as an eID app capable of communicating with Estonian ID-cards using NFC technology. For this purpose the work done for the m-voting project, especially the NFC-ID library could be reused.

## **Other Wallet Related Components**

### **Relying Parties**

Any entity that user provides credential to using their wallet would be considered a Relying Party. The State Authentication Service TARA would be a specific example of one such Relying Party another common type of RPs are financial institutions.

### **Relying Party Registry**

As eIDAS requires the Relying Parties to authenticate to themselves when requesting credentials from the wallet, there needs to be a party in the ecosystem which acts as a registration authority for RPs. This party would also have to publish an authoritative list of registered Relying Parties to wallet instances. As the mechanism for this needs to be interoperable cross-borders within Europe, this component has currently not been analyzed further.

### **Trusted List Provider(s)**

While this is not currently depicted on the diagram above there needs to be a mechanism for the Relying Parties to establish trust with - obtain the public keys of - PID Issuers. Similarly to the RP Registry this needs to be interoperable within Europe. Therefore, this is a potential candidate to be added onto the ecosystem diagram in future iterations.

# 5. Data Models

This page serves as a complement to the Architecture of The EE Wallet Ecosystem page. It describes what data parties in this ecosystem would store - their data models. As the architecture of the wallet gets more specific this page is expected to mature into complete logical data models.

## EE PID Provider Trust Service

### PID Attestation Data

Field	Type	Constraints	Description
Id	String	unique not null	Persistent identifier of this entity.
RevocationStatus	String	not null	Exact mechanism to store revocation status is still to be determined, but some data about it needs to be stored.
PidAttestation	Blob	unique not null	For future proofing the PID attestation should be stored in its entirety in its raw form.

## EE Wallet Provider Backend

### Wallet Instance Data

Field	Type	Constraints	Description
Id	String	unique not null	Persistent internal identifier which should not be shared with other parties.
UserIdentificationNumber	String	not null	The national identification number of the wallet's user. Mainly used during revocation.
DeviceAttributes	?	not null	It is likely that some description of the device that the wallet installed is on needs to be stored. This data would be used when the user requests the revocation of the wallet (to identify which wallet instance needs to be revoked in the case where the person has multiple wallets).
IsRevoked	Boolean	not null	Indicates the revocation status of the wallet.
LastActivation	DateTime	not null	Wallet instances can be abandoned by the end user. Having a last activation date stored will allow such wallet instances to be deleted from the wallet providers database.
PidIdentifier	String	not null	During wallet revocation the wallet provider should notify the PID provider that the PID issued into the wallet should also be revoked. This could be done using the PID identifier. Alternatively the PID provider could store a wallet identifier.  Note: This decision depends on how data is compartmentalized between the Wallet and PID Providers. For privacy reasons it might be important for the wallet provider to know which PID is stored in the wallet or vice versa.
ExternalId	String		A temporary identifier that can be shared with the PID provider to associate the wallet instance with an identity.

## EE Wallet Instance Data Model

The main data wallets store are different types of credentials. The structure of the only such analyzed credential - PID - is described below. For online presentation this is stored in the SD-JWT VC format. In future an mdoc type credential might also be stored for offline presentation. The PID attestation fields are based on the ARF PID rulebook. The EE PID is currently envisioned to only use the minimal PID dataset.

### PID Attestation

SD-JWT VC claim	mdoc data element identifier	Type	Description
sub	unique_id	String	An unique identifier of the subject of the PID as determined by the PID provider. In Estonia the national identification code would be used.
family_name	family_name	String	Subject's family name. It is mandatory to include but can be an empty string if the subject does not have a family name.
given_name	given_name	String	Subject's given name
birthdate	birth_date	String	Date of birth of the subject formatted YYYY-MM-DD

picture	portrait	?	A document photo of the subject. This could be used for proof of user binding in an offline scenario.  <i>The ARF PID Rule Book 1.1 recommends using this claim for document photo. However this recommendation is likely a mistake, as its registered type in IANA is an URL.</i>
age_over_18	age_over_18	Boolean	A value which indicates whether the subject is over 18. This field could be useful when the PID is selectively disclosed.
iat	issuance_date	jwt: <a href="#">NumericDate</a> mdoc: tdate or full-date	The moment at which the PID starts its validity period.
exp	expiry_date	jwt: <a href="#">NumericDate</a> mdoc: tdate or full-date	The moment at which the PID ends its validity period.
iss	N/A	jwt: HTTPS URL	An URL of the PID issuer.
cnf	N/A?	String	The public key of the key pair which is used for user binding.
status	<i>undecided in ARF</i>	Object	Contains the information on how PID attestation revocation should be checked.
type	N/A	String with the value " <a href="#">eu.europa.ec.eudiw.pid.1</a> ".	Has the value " <a href="#">eu.europa.ec.eudiw.pid.1</a> ". This is used to indicate the namespace/type of the credential.  For mdoc format the same value is used for document type and namespace (as specified in ISO /IEC 18013-5)

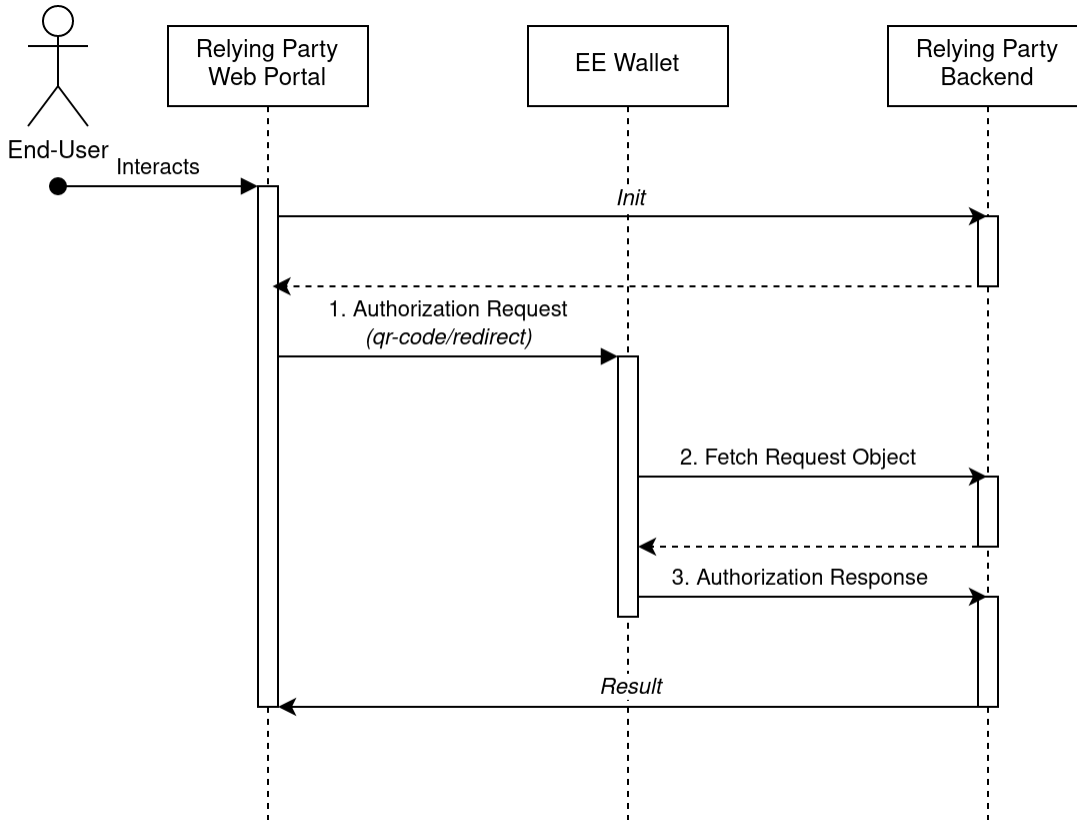
#### Example (SD-JWT VC):

```
{
  "alg": "RS256",
  "typ": "JWT"
}
{
  "sub": "PNOEE-38001085718",
  "family_name": "JÖEORG",
  "given_name": "JAAK-KRISTJAN",
  "birthdate": "1980-01-08",
  "age_over_18": true,
  "iat": 1262304000,
  "exp": 1420070400,
  "iss": "https://pid-issuer-url.ee",
  "status": {
    // this would contain information on how to check whether the attestation has been revoked
  },
  "cnf": {
    // this would contain the public key of the subject - the key that is used for presentation
  },
  "type": "eu.europa.ec.eudiw.pid.1"
}
```

# 6. Credential Presentation Interface

Based on the ARF and documentation released by other member states (such as [Italy](#)) it is currently assumed that the interoperability with the EUDI wallet ecosystem will require conforming to the [OpenID4VC High Assurance Interoperability Profile with SD-JWT VC](#) (HAIP). For credential presentation HAIP specifies the usage of the [OpenID4VP](#) credential presentation protocol. OpenID4VP itself is built on top of the common [OAuth 2.0](#) authorization protocol. Among other requirements HAIP also specifies the usage of the JWT-SD credential format.

This page describes the HAIP conformant interface/protocol EE Wallet uses for credential presentation. The general flow of the protocol is depicted on the following diagram.



*Note: Both HAIP and OpenID4vp are still under development which leads to some gaps in this interface description. These caps shall be filled at some later point, when the underlying protocols are further developed. This work will likely result in some messages of the protocol being somewhat restructured.*

## 1. Authorization Request

Credential presentation starts by the end user requesting access to some resource controlled by the Relying Party for which the Relying Party requires the user to authenticate. The details of that exchange are RP specific and are therefore out of the scope of this document.

Following the initial exchange the RP directs the user to provide a credential using a wallet. For this the RP constructs an Authorization Request as a URI which, similarly to other OAuth 2.0 based protocols, is used to redirect the End-User. In a same-device flow this redirection is done automatically by directing the device to open the URI. In a cross-device flow the URI is displayed to the user as a QR-code which the user has to manually scan using their wallet application.

The Authorization Request URI is constructed by adding the following parameters encoded using the `application/x-www-form-urlencoded` format to the query component of the wallet's authorization endpoint.

*Note: The 'wallet's authorization endpoint' has to be common among EU member states and shall be specified in this document at some later time.*

Parameter	Mandatory	Description
<code>request_uri</code>	+	<p>An absolute URI as defined by <a href="#">RFC 3986</a> that the wallet will query for the Request Object. The Request Object cannot be included in this URI directly as it is likely to exceed URI length limits.</p> <p>For security considerations (eg: to avoid session hijacking) the RP should construct the <code>request_uri</code> such that:</p> <ul style="list-style-type: none"> <li>• it is only used once,</li> <li>• has a short validity period (less than 1 minute) and</li> <li>• has sufficient entropy by including a cryptographically random value of 128 bits or more.</li> </ul>

client_id	+	The value must match the client_id provided in the Request Object.
-----------	---	--

**example:**

```
https://wallet.example.com/authorize?
client_id=s6BhdRkgt3
&request_uri=https%3A%2F%2Fwww.example.org%2Frequest.jwt
%2FGkurKxf5T0Y-mnPFCHqWOMiZi4VS138cQO_V7PZHAd
```

## 2. Fetching Request Object

Upon receiving the Authorization Request, the Wallet sends an HTTP GET to the request\_uri from the Authorization Request to receive the referenced Request Object. An RP must respond with media type application/oauth-authorization-request+jwt and with the response body containing only the Request Object.

The Request Object is a **JWS** signed **JWT** whose JWT Claims Set holds the following parameters:

Parameter	Type	Mandatory	Description
response_type	String	+	Must have the value vp_token.
response_mode	String	+	Must have the value direct_post.
response_uri	String	+	An absolute URI as defined by RFC 3986 that the wallet will send an HTTP POST request with the Authorization Response in the request body.
client_id_scheme	String	+	<p>The value must be either x509_san_dns or verifier_attestation.</p> <p><i>TBD: This mechanism is currently under specification in HAIP. How this will be utilized will also depend on the design of the RP registry.</i></p> <p><i>From OpenID4VP draft 20 some clues are provided for the semantics of these values:</i></p> <p><i>verifier_attestation: This Client Identifier Scheme allows the Verifier to authenticate using a JWT that is bound to a certain public key as defined in (#verifier_attestation_jwt). When the Client Identifier Scheme is verifier_attestation, the Client Identifier MUST equal the sub claim value in the Verifier attestation JWT. The request MUST be signed with the private key corresponding to the public key in the cnf claim in the Verifier attestation JWT. This serves as proof of possession of this key. The Verifier attestation JWT MUST be added to the jwt JOSE Header of the request object (see (#verifier_attestation_jwt)). The Wallet MUST validate the signature on the Verifier attestation JWT. The iss claim value of the Verifier Attestation JWT MUST identify a party the Wallet trusts for issuing Verifier Attestation JWTs. If the Wallet cannot establish trust, it MUST refuse the request. If the issuer of the Verifier Attestation JWT adds a redirect_uris claim to the attestation, the Wallet MUST ensure the redirect_uri request parameter value exactly matches one of the redirect_uris claim entries. All Verifier metadata other than the public key MUST be obtained from the client_metadata or the client_metadata_uri parameter.</i></p> <p><i>x509_san_dns: When the Client Identifier Scheme is x509_san_dns, the Client Identifier MUST be a DNS name and match a dNSName Subject Alternative Name (SAN) [RFC5280] entry in the leaf certificate passed with the request. The request MUST be signed with the private key corresponding to the public key in the leaf X.509 certificate of the certificate chain added to the request in the x5c JOSE header [RFC7515] of the signed request object. The Wallet MUST validate the signature and the trust chain of the X.509 certificate. All Verifier metadata other than the public key MUST be obtained from the client_metadata parameter. If the Wallet can establish trust in the Client Identifier authenticated through the certificate, e.g. because the Client Identifier is contained in a list of trusted Client Identifiers, it may allow the client to freely choose the redirect_uri value. If not, the FQDN of the redirect_uri value MUST match the Client Identifier.</i></p>
client_id	String	+	See client_id_scheme.
presentation_definition	Object	+	A Presentation Definition object (see below) describing the presentation that the RP expects to receive. In general this object follows the data model of DIF Presentation Exchange 2.0.0 (DIF) however HAIP defines a more specific JSON schema for this object in its appendix B.
nonce	String	+	<p>It is used to securely bind the Verifiable Presentation(s) provided by the Wallet to the particular transaction.</p> <p>Not suitable for maintaining state between request and response callback as this value is not mirrored back in error responses.</p> <p>An RP must create a fresh, cryptographically random value with sufficient entropy for every Authorization Request.</p>
iss	String		<i>This value is recommended to be used by rfc9101, but in the context of an OpenID4VP transaction it is unclear what the significance of this field would be. This will likely later be changed/specified in HAIP.</i>
aud	String		<i>This value is recommended to be used by rfc9101, but in the context of an OpenID4VP transaction it is unclear what the significance of this field would be. This will likely later be changed/specified in HAIP.</i>
state	String		A value passed back to the RP unmodified in the Authorization Response. Can be used by the RP to maintain state between the request and response callback.

### Presentation Definition

Parameter	Type	Mandatory	Description
-----------	------	-----------	-------------

id	String	+	Any value selected by the RP. The RP should select a unique ID for the desired context. This value is not used by the wallet, rather it is mirrored back in the <code>presentation_submission</code> object.
input_descriptors	Object[]	+	An array of Input Descriptors Objects (see below) which describe the presentation the RP is requesting. Unless a <code>submission_requirements</code> object is present a wallet will return a presentation matching all the Input Descriptors or if this is not possible an error response.
submission_requirements	Object[]		An array of Submission Requirements Objects (see below). This allows an input descriptor to be made optional.  <i>It is currently unclear for requirement using this field fulfills. HAIP seems to indicate that it is used by the RP to specify which fields of a credential it deems optional. For this purpose however, there is the field 'optional' under the input descriptor (see below) constraints. This will likely later be changed/specified in HAIP at some later time.</i>

### Input Descriptor Object

Parameter	Type	Mandatory	Description
id	String	+	A value selected by the RP which must be unique among other Input Descriptor Object IDs within the same request. This value is not used by the wallet, rather it is mirrored back in the <code>presentation_submission</code> object.
name	String		A human-friendly name which describes what the requested presentation represents. This is used to display information to the End-User.  <i>It seems erroneous that an RP would be allowed to control this information during the authentication flow. A proper approach would get this information from the RP registry. In any case, if this field is used to display text to the end user, there does not seem to be any translation support specified.</i>
purpose	String		A human friendly description of the purpose for which the RP is requesting the presentation for. This is used to display information to the End-User.  <i>The same note applies here as for the 'name' field.</i>
group	TBD		If present must match a "from" value of one of the Submission Requirements Objects.
format	Object		A JSON object with a single property " <code>vc+sd-jwt</code> " with the value "{}" (an empty JSON Object). This indicates the data format of the PID requested.
constraints	Object	+	A Constraint object (see directly below).
constraint.limit_disclosure	String		The value must be either <code>required</code> or <code>preferred</code> with the default value being <code>preferred</code> .  <i>TBD - HAIP says that this field is mandatory to support, however considering the semantics in DIF requiring it to be mandatory does not make much sense. DIF says that this is used by the RP to indicate to the Wallet that the Wallet should ONLY return fields explicitly requested by the RP. This is different from the "optional" property, which is used to specify whether presenting a field is optional.</i>
constraint.s.fields	Object[]	+	An array where each object describes a claim the RP is expecting to see in the presentation.
constraint.s.fields[*].path	String[]	+	Must contain exactly one string with a <a href="#">JSONPath</a> expression which describes one claim the RP is expecting to see in the presentation.
constraint.s.fields[*].filter	Object		A <a href="#">JSON Schema</a> descriptor which describes the data type the RP expecting to see for the claim specified in the "path" parameter above.  <i>HAIP contains an odd restriction for this: "filter MUST only contain type elements of value string and const elements.", this would make it impossible to request elements with type 'boolean' or 'number'. The schema provided in HAIP's the annex does also not conform to this restriction. This will likely later be changed/specified in HAIP at some later time.</i>

### Submission Requirement Object

Parameter	Type	Mandatory	Description
rule	String	+	Must have the value <code>pick</code> .  <i>TBD - there is a typo in HAIP which makes it unclear whether pick is the only value supported. If HAIP is updated needs to be verified.</i>
count	Number		If present must be greater than zero.
from	String	+	Must match one of the group strings specified for one or more Input Descriptor Objects. Essentially this acts as an identifier for the submission requirement object.
name	String		<i>This field seems to be used to display information to end-users. Has the same problems as the 'name' and 'purpose' fields in the input descriptor object.</i>

## 3. Authorization Response

Upon having parsed the Request Object, the Wallet sends an HTTP POST request to the `response_uri` from the Request Object containing an Authorization Response with parameters encoded in the body of the request using the `application/x-www-form-urlencoded` content type. The response can be either a success or an error response.

### Success Response

Parameter	Type	Mandatory	Description
vp_token	Object	+	A presentation conforming to the presentation_definition submitted in the request. Examples of these are described on the Data Models page
presentation_submission	Object	+	An Presentation Submission object (see below) describing how the presentation returned should be interpreted.
state	String		Matches the value provided by the RP in the Request Object. Is not included in the response if missing from the Request Object.

### Presentation Submission Object

Parameter	Type	Mandatory	Description
id	String	+	An identifier for the presentation submission as chosen by the wallet.
definition_id	String	+	An id matching the id of the presentation submitted in the Request Object.
format	String	+	Always has the value vc+sd-jwt indicating the data format of the submitted credential.
descriptor_map	Object[]	+	An array of Descriptor Map objects (see below).
descriptor_map[*].id	String	+	An id matching an input descriptor object id from the Request Object.
descriptor_map[*].format	String	+	Denotes the data format of the claim.
descriptor_map[*].path	String	+	A JSONPath string expression denoting the claim in relation to the submitted credential.

### Error Response

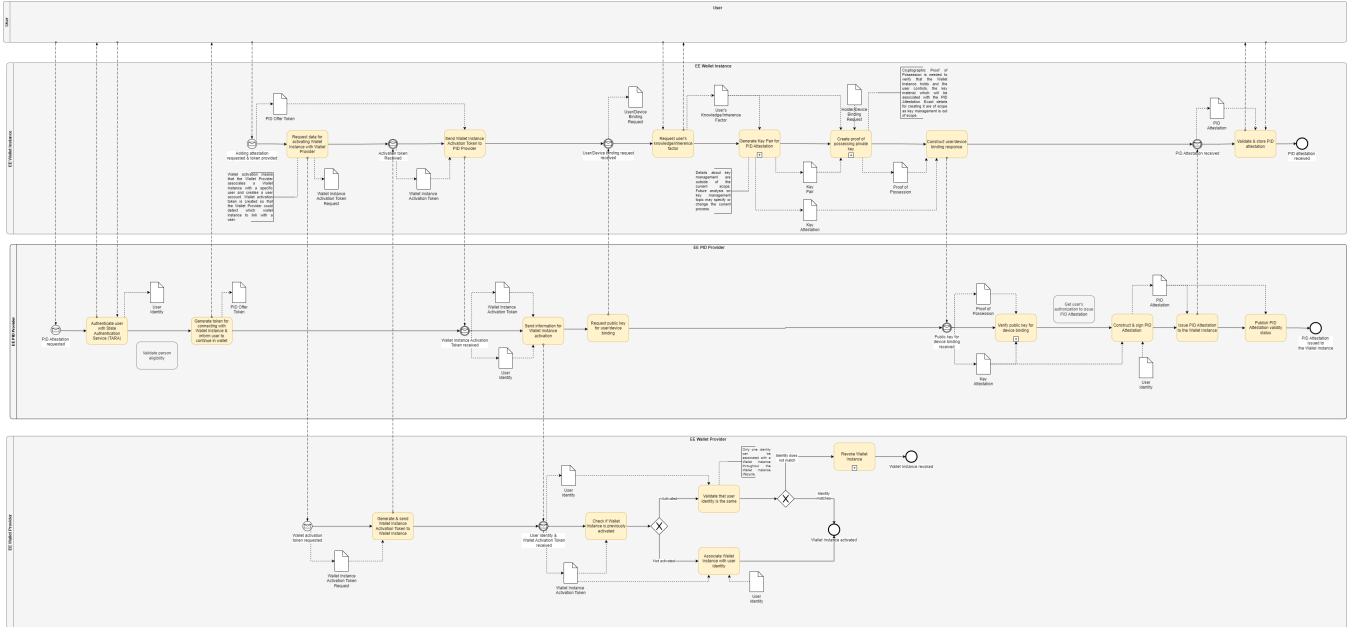
Parameter	Type	Mandatory	Description
error	String	+	A single error code with the following possible values: <ul style="list-style-type: none"> <li>invalid_request - The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed.</li> <li>invalid_client/unauthorized_client - TBD</li> <li>vp_formats_not_supported - The wallet does not support any of the VP formats requested by the RP.</li> <li>invalid_request_uri - The request_uri in the authorization request returns an error or contains invalid data.</li> </ul>
error_description	String		Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.
state	String		Matches the value provided by the RP in the Request Object. Is not included in the response if missing from the Request Object.

# A1 EE PID Attestation Issuance (cross-device)

## Scope

This process model proposes the flow for the EE PID Attestation issuance to the EE Wallet Instance. In this process flow the user interacts with the PID Provider and the Wallet Instance on the separate devices. Process flow where the user interacts with the PID Provider and the Wallet Instance on the same device is described separately. The process model also describes how during the PID Attestation issuance the EE Wallet Provider locally associates the user with the EE Wallet Instance and therefore creates the user account (*This procedure is proposed as one of the steps in the ARF for EUDI Wallet Instance activation*). Please note that separate steps for the establishment of trust between the actors is not currently described on this diagram. This might be changed at some point when the ARF specifies in more detail how these trust relations should work and discussions on the topic with stakeholders participating in this process are conducted. For example, Trusted List(s) could be leveraged as one option to establish some trust relations.

## Process Model



## Process Description

This section describes the activities and data objects of the process model.

### Activities

In the table below, each activity in the process model is described.

Name of the activity	Description
Authenticate user with State Authentication Service (TARA)	<p>PID Provider uses the State Authentication Service (technical name is TARA) for user authentication. User Identity data that is later used to create the PID Attestation is the output of this step.</p> <p>If the PID Provider accepts all the main Estonian eID means (ID-card, Mobile-ID, Smart-ID) to be used by TARA for user authentication, is yet to be decided. However, the Estonian ID-card shall be one of the options and it is proposed that when the user wants to identify themselves in TARA with an ID-Card, then in addition to using the ID-card chip reader as it is done currently in Estonia, it should also be possible to authenticate with the ID-card by scanning it using the NFC capabilities of the mobile device and the ID-Card. This means the user does not have to insert an ID-card to a reader hardware, but instead could scan the ID-card with the same device that they have installed their EE Wallet Instance on by leveraging the NFC technology.</p>

<p>Validate person eligibility</p>	<p><i>Potential step, necessity and details of this step are to be decided, not explicitly specified in ARF.</i></p> <p>PID Provider validates that the person applying for the PID Attestation is eligible to receive one.</p> <p>Examples of potential checks:</p> <ul style="list-style-type: none"> <li>• Confirm person is an Estonian citizen</li> <li>• Confirm person has a resident permit</li> <li>• Confirm person has active legal capacity</li> </ul> <p>Another example is that that the PID Provider validates person data (or queries extra data) by contacting the Population Registry.</p>
<p>Generate token for connecting with Wallet Instance &amp; inform user to continue in wallet</p>	<p>Since this is a cross-device flow then the user interacts with the PID Provider in one device and with the Wallet Instance in another device. The Wallet Instance and the PID Provider must establish connection and for that the PID Provider generates a PID Offer Token. The user acts as a middleman and as result of user action the token is passed on to the Wallet Instance. Exact details of the token and mechanism depend on the technical choices that may also later change the currently proposed process.</p> <p>For example, the PID Provider could offer the user a PID Offer Token in form of a QR code that the user would scan with the Wallet Instance. The QR code would contain the necessary information for the Wallet Instance and the PID Provider to establish connection.</p>
<p>Request data for activating Wallet Instance with Wallet Provider</p>	<p>Wallet Instance requests a Wallet Instance Activation Token from the Wallet Provider. This token is used later in the process by the Wallet Provider in order to associate the user with a specific Wallet Instance and therefore create a user account, a procedure that is called <i>Wallet Instance activation</i>.</p>
<p>Generate &amp; send Wallet Instance Activation Token to Wallet Instance</p>	<p>Wallet Provider generates the Wallet Instance Activation Token and sends it back to the Wallet Instance.</p> <p>Wallet Instance Activation Token contains the necessary information for the Wallet Provider to later in the process determine which Wallet Instance must be associated with a given user.</p>
<p>Send Wallet Instance Activation Token to PID Provider</p>	<p>Wallet Instance has received the Wallet Instance Activation Token from the Wallet Provider and sends it to the PID Provider. Token is later used by the Wallet Provider for associating the Wallet Instance with user's identity.</p> <p>PID Offer Token is used to establish connection with the PID Provider.</p>
<p>Send information for Wallet Instance activation.</p>	<p>PID Provider has established connection with the Wallet Instance and received the Wallet Instance Activation Token. PID Provider now sends User Identity data, received from the TARA authentication they conducted earlier, to the Wallet Provider. PID Provider also forwards the Wallet Instance Activation Token.</p>
<p>Check if Wallet Instance is previously activated</p>	<p>Wallet Provider has received the user's identity data and Wallet Instance Activation Token from the PID Provider. Wallet Provider will use the Wallet Instance Activation Token to check if the Wallet Instance has been previously activated.</p>
<p>Validate that user identity is the same</p>	<p>If from the previous step it was discovered that the Wallet Instance has been already activated, then the Wallet Provider checks that the User Identity, provided by the PID Provider, is the same as has been previously associated with the Wallet Instance.</p> <ul style="list-style-type: none"> <li>• If the identities do not match, then the Wallet Provider revokes the Wallet Instance, since it is currently proposed that only one identity can be associated with a Wallet Instance throughout Wallet Instance lifecycle. PID Attestation issuance process will also fail, because the Wallet Instance needs to be activated and not be revoked in order to provide a key pair for the PID Attestation.</li> <li>• If the identities match then the Wallet Instance remains activated.</li> </ul>

Associate Wallet Instance with user identity	<p>If in the step <i>Check if Wallet Instance is previously activated</i> it was detected that the Wallet Instance has not been previously activated, then the Wallet Provider locally associates the Wallet Instance (involved in the ongoing PID Attestation issuance process) with user's identity. Data for detecting which Wallet Instance to associate with the user's identity was received from the PID Provider (User Identity and Wallet Instance Activation Token data objects are provided by the PID Provider).</p> <p><b>Note</b> that the current version of ARF describes: <i>"In general, the EUDI Wallet Provider does not need to know the true identity of the User. An alias, for example an e-mail address, should be sufficient. However, the EUDI Wallet Provider may request the true identity of the User to be able to offer additional services. It is up to the EUDI Wallet Provider to determine the conditions for creating an online account, and to the User to accept or refuse these conditions."</i> This means that it is also possible to set up the user account without the Wallet Provider knowing the identity of the user, for example, by using username/password type of solutions or leveraging the possibility of creating an user account based on a user pseudonym. Based on the preliminary discussions, the assumption in this project is made that the user account is created based on the real identity of the user. This enables the Wallet Provider to use strong authentication means to authenticate the user when the user wants to perform Wallet management activities like revocation of a Wallet Instance. If any future updates on ARF determine new aspects about this or some privacy related aspects arise, then it could be subject to change. Future changes could also affect when the Wallet Instance activation is performed, for example, it could be separated and take place prior to the PID Attestation Issuance process.</p>
Request public key for user /device binding	<p>PID Provider requests a public key from the Wallet Instance. PID Provider will later bind the public key to the PID Attestation being issued. Binding the public key to the PID Attestation is required to achieve user/device binding, a mechanisms that the Relying Parties can use to cryptographically verify that the PID Attestation is presented from the same user/device it was originally issued to.</p> <p>In addition, the PID Provider also needs to verify that the Wallet Instance controls the private key associated with the public key it provides. For that the PID Provider also requires proof of possessing the private key ('Proof of Possession' data object). For example, the PID Provider could include a random unique value in the request and by signing this value the Wallet Instance can prove they control the private key.</p>
Request user's knowledge /inherence factor for key material	<p>Wallet Instance requests the user's knowledge/inherence factor. This will be used to protect the private key that is generated for the PID Attestation. Exact details about the knowledge/inherence factor depend on the key management mechanisms.</p>
Generate Key Pair for PID Attestation	<p>Wallet Instance creates a Key Pair for the PID Attestation. Key Pair consists of a public and private key. Public key will be shared with the PID Provider by means of Key Attestation. It is important to note that the Key Attestation can only be generated if the wallet is activated and has not been revoked.</p> <p>Specifics about key management are outside of the current project scope. Future analysis on key management topic may specify or affect the current process.</p>
Create proof of possessing private key	<p>Wallet Instance generates proof of possessing the private key. Cryptographic Proof of Possession is needed to verify that the Wallet Instance holds and the user controls the key material that will be associated with the PID Attestation. For example, to create the Proof of Possession, the Wallet Instance could sign the random unique value provided in public key request from the PID Provider.</p>
Construct user /device binding response	<p>Wallet Instance sends the Key Attestation and Proof of Possession to the PID Provider.</p>
Verify public key for device binding	<p>PID Provider validates the Key Attestation and verifies that the private key used for creating the Proof of Possession matches the public key in the Key Attestation.</p>
Get user's authorization to issue PID Attestation to wallet	<p><i>Potential step, necessity and details of this step are to be decided, not explicitly specified in ARF.</i></p> <p>The PID Provider asks the user to continue the process by confirming PID Attestation issuance to the Wallet Instance and receives user's confirmation.</p> <p>For example, the user could sign an acceptance contract that documents the user agreement to PID Attestation issuance.</p>
Construct & sign PID Attestation	<p>PID Provider constructs the PID Attestation which will be issued to the Wallet Instance. Public key from the Key Attestation is also added to the PID Attestation. PID Provider also signs the PID Attestation as the issuer.</p>

Issue PID Attestation to the wallet	PID Provider sends the PID Attestation to the Wallet Instance.
Publish PID Attestation validity status	PID Provider makes the information about the PID Attestation status available. This information is required by the Relying Parties, when they want to check that the PID Attestation presented to them is still valid and has not been revoked by the PID Provider. It could potentially also be used by the Wallet Instance to check PID Attestation revocation status.
Verify & store PID Attestation	<p>User validates the details in the PID Attestation are correct and consents to storing it in the EE Wallet.</p> <p>It is also possible that after the user's EE Wallet Instance received the EE PID Attestation it validates that the PID Provider, who issued the PID, is an legitimate EE PID Provider and has the right to issue PID Attestations. One option for this validation to be conducted is contacting the EE Trusted List of Providers, that would include information necessary to perform the validation. But this depends on the trust mechanisms used between the parties and is not currently specified.</p> <p>Secondly, the Wallet Instance could also validate that the received EE PID Attestation is not revoked. This validation could be done using the same mechanisms as RPs use during PID presentation. Wallet Instance could query the received EE PID Attestation status from the source where the EE PID Provider published this information.</p> <p>If the EE PID Attestation's <i>valid from</i> date arrives some time after its issuance, then the Attestation should not be presentable before that date .Whether such PID Attestations are an option is to be decided.</p>

## Data Objects

In the table below, each data object from the process model is described.

Name of the Data Object	Description
User Identity	<p>User Identity is received as the result of authenticating the user with the TARA service. It contains information about the person's identity that will be used as input for creating the PID Attestation. It is also forwarded to the Wallet Provider so that the Wallet Provider can locally associate the user with a specific Wallet Instance and therefore create a user account.</p> <ul style="list-style-type: none"> <li>• Family name</li> <li>• Given name</li> <li>• Date of birth</li> <li>• Person identification code</li> </ul>
PID Offer Token	<p>PID Offer Token is generate by the PID Provider. Token includes the necessary information for the Wallet Instance and the PID Provider to establish connection.</p> <p>Example of a possible form factor for the PID Offer Token is a QR code, which the Wallet Instance scans.</p>
Wallet Instance Activation Token Request	<p>Wallet Instance Activation Token Request is created by the Wallet Instance and contains information that the Wallet Provider needs to know about the Instance. Exact details are to be determined, but the request could contain data about the device that the Wallet Provider can later store when activating the Wallet Instance.</p>
Wallet Instance Activation Token	<p>Wallet Instance Activation Token is generated by the Wallet Provider and contains the necessary information for the Wallet Provider to determine which wallet instance must be associated with a given user.</p>
User /Device Binding Request	<p>User/Device Binding Request is created by the PID Provider to request a public key that will be associated with the PID Attestation. The request can also include a random unique value. This value could be used as input for proof of possessing the private key, the value would be signed using the private key.</p>
User's Knowledge /Inherence Factor	<p>This data object contains the user's knowledge or inherence factor provided as input by the user. Wallet Instance needs it to generate a key pair for the PID Attestation being issued. The user's knowledge or inherence factor is used to protect access to the private key that is generated.</p> <p>Example for a knowledge factor is a PIN code.</p> <p>Example for a inherence factor is a biometric scan.</p>

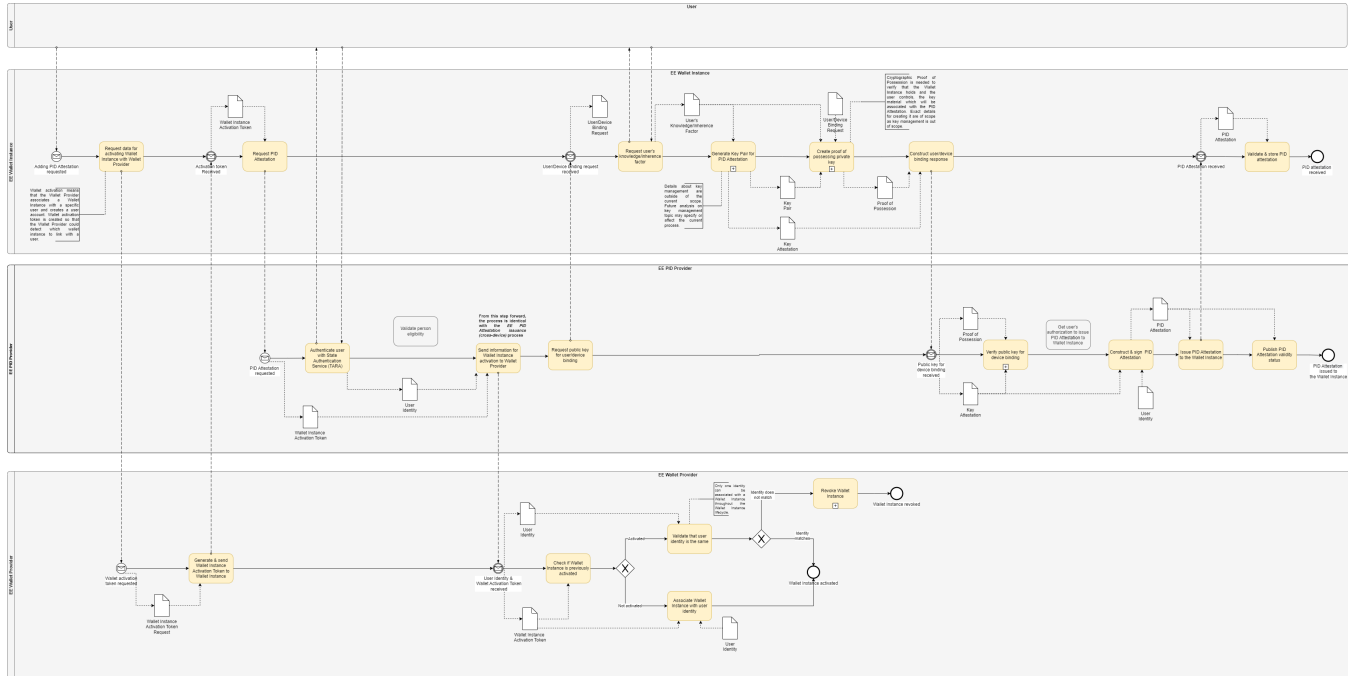
Key Pair	<p>Key Pair that is associated with the PID Attestation being issued.</p> <ul style="list-style-type: none"> <li>• Public key</li> <li>• Private key</li> </ul>
Key Attestation	<p>Key attestation contains the public key that is bound to the PID Attestation. It also contains other necessary data that is required to assess that the Key Pair was generated securely and is stored securely. Since key management is currently out of scope then exact details are not currently described. Future analysis on the topic could change the current proposal.</p> <ul style="list-style-type: none"> <li>• Public key</li> <li>• Attestation</li> </ul>
Proof of Possession	<p>Wallet Instance creates the Proof of Possession. One option to achieve this is signing the random unique value provided by the the PID Provider. Signature is given with the private key of the Key Pair created for the PID Attestation.</p>
PID Attestation	<p>PID Provider constructs the PID Attestation which will be issued to the Wallet Instance. PID Attestation is signed by the PID Provider.</p> <ul style="list-style-type: none"> <li>• Family name</li> <li>• Given name</li> <li>• Date of birth</li> <li>• Age over 18 (Assertion: true/false)</li> <li>• Unique identifier</li> <li>• Metadata <ul style="list-style-type: none"> <li>◦ Issuance date</li> <li>◦ Expiry date</li> <li>◦ Issuing authority</li> <li>◦ Issuing country</li> </ul> </li> </ul>

# A2 EE PID Attestation Issuance (same-device)

## Scope

This process model proposes the flow for the EE PID Attestation issuance to the EE Wallet Instance. In this process flow the user interacts with the PID Provider and the Wallet Instance on the same device. Process flow where the user interacts with the PID Provider and the Wallet Instance on different devices is described separately. The process model also describes how during the PID Attestation issuance the EE Wallet Provider locally associates the user with the EE Wallet Instance and therefore creates the user account (*This procedure is proposed as one of the steps in the ARF for EUDI Wallet Instance activation*). Please note that separate steps for the establishment of trust between the actors is not currently described on this diagram. This might be changed at some point when the ARF specifies in more detail how these trust relations should work and discussions on the topic with stakeholders participating in this process are conducted. For example, Trusted List(s) could be leveraged as one option to establish some trust relations.

## Process Model



## Process Description

This section describes the activities and data objects of the process model.

### Activities

In the table below, each activity in the process model is described.

Name of the activity	Description
Request data for activating Wallet Instance with Wallet Provider	Wallet Instance requests a Wallet Instance Activation Token from the Wallet Provider. This token is used later in the process by the Wallet Provider in order to associate the user with a specific Wallet Instance and therefore create a user account, a procedure that is called <i>Wallet Instance activation</i> .

Generate & send Wallet Instance Activation Token to Wallet Instance	<p>Wallet Provider generates the Wallet Instance Activation Token and sends it back to the Wallet Instance.</p> <p>Wallet Instance Activation Token contains the necessary information for the Wallet Provider to later in the process determine which Wallet Instance must be associated with a given user.</p>
Request PID Attestation	<p>Wallet Instance has received the Wallet Instance Activation token and sends it to the PID Provider and requests PID Attestation to be issued into the Wallet Instance.</p>
Authenticate user with State Authentication Service (TARA)	<p>PID Provider uses the State Authentication Service (technical name is TARA) for user authentication. User Identity data that is later used to create the PID Attestation is the output of this step.</p> <p>If the PID Provider accepts all the main Estonian eID means (ID-card, Mobile-ID, Smart-ID) to be used by TARA for user authentication, is yet to be decided. However, the Estonian ID-card shall be one of the options and it is proposed that when the user wants to identify themselves in TARA with an ID-Card, then in addition to using the ID-card chip reader as it is done currently in Estonia, it should also be possible to authenticate with the ID-card by scanning it using the NFC capabilities of the mobile device and the ID-Card. This means the user does not have to insert an ID-card to a reader hardware, but instead could scan the ID-card with the same device that they have installed their EE Wallet Instance on by leveraging the NFC technology.</p>
Validate person eligibility	<p><i>Potential step, necessity and details of this step are to be decided, not explicitly specified in ARF.</i></p> <p>PID Provider validates that the person applying for the PID Attestation is eligible to receive one.</p> <p>Examples of potential checks:</p> <ul style="list-style-type: none"> <li>• Confirm person is an Estonian citizen</li> <li>• Confirm person has a resident permit</li> <li>• Confirm person has active legal capacity</li> </ul> <p>Another example is that that the PID Provider validates person data (or queries extra data) by contacting the Population Registry.</p>
<p>From the next step, the process is identical with the A1 EE PID Attestation Issuance (cross-device) process.</p>	
Send Wallet Instance Activation Token to PID Provider	<p>Wallet Instance has received the Wallet Instance Activation Token from the Wallet Provider and sends it to the PID Provider. Token is later used by the Wallet Provider for associating the Wallet Instance with user's identity.</p> <p>PID Offer Token is used to establish connection with the PID Provider.</p>
Send information for Wallet Instance activation.	<p>PID Provider has established connection with the Wallet Instance and received the Wallet Instance Activation Token. PID Provider now sends User Identity data, received from the TARA authentication they conducted earlier, to the Wallet Provider. PID Provider also forwards the Wallet Instance Activation Token.</p>
Check if Wallet Instance is previously activated	<p>Wallet Provider has received the user's identity data and Wallet Instance Activation Token from the PID Provider. Wallet Provider will use the Wallet Instance Activation Token to check if the Wallet Instance has been previously activated.</p>
Validate that user identity is the same	<p>If from the previous step it was discovered that the Wallet Instance has been already activated, then the Wallet Provider checks that the User Identity, provided by the PID Provider, is the same as has been previously associated with the Wallet Instance.</p> <ul style="list-style-type: none"> <li>• If the identities do not match, then the Wallet Provider revokes the Wallet Instance, since it is currently proposed that only one identity can be associated with a Wallet Instance throughout Wallet Instance lifecycle. PID Attestation issuance process will also fail, because the Wallet Instance needs to be activated and not be revoked in order to provide a key pair for the PID Attestation.</li> <li>• If the identities match then the Wallet Instance remains activated.</li> </ul>

Associate Wallet Instance with user identity	<p>If in the step <i>Check if Wallet Instance is previously activated</i> it was detected that the Wallet Instance has not been previously activated, then the Wallet Provider locally associates the Wallet Instance (involved in the ongoing PID Attestation issuance process) with user's identity. Data for detecting which Wallet Instance to associate with the user's identity was received from the PID Provider (User Identity and Wallet Instance Activation Token data objects are provided by the PID Provider).</p> <p><b>Note</b> that the current version of ARF describes: <i>"In general, the EUDI Wallet Provider does not need to know the true identity of the User. An alias, for example an e-mail address, should be sufficient. However, the EUDI Wallet Provider may request the true identity of the User to be able to offer additional services. It is up to the EUDI Wallet Provider to determine the conditions for creating an online account, and to the User to accept or refuse these conditions."</i> This means that it is also possible to set up the user account without the Wallet Provider knowing the identity of the user, for example, by using username/password type of solutions or leveraging the possibility of creating an user account based on a user pseudonym. Based on the preliminary discussions, the assumption in this project is made that the user account is created based on the real identity of the user. This enables the Wallet Provider to use strong authentication means to authenticate the user when the user wants to perform Wallet management activities like revocation of a Wallet Instance. If any future updates on ARF determine new aspects about this or some privacy related aspects arise, then it could be subject to change. Future changes could also affect when the Wallet Instance activation is performed, for example, it could be separated and take place prior to the PID Attestation Issuance process.</p>
Request public key for user /device binding	<p>PID Provider requests a public key from the Wallet Instance. PID Provider will later bind the public key to the PID Attestation being issued. Binding the public key to the PID Attestation is required to achieve user/device binding, a mechanisms that the Relying Parties can use to cryptographically verify that the PID Attestation is presented from the same user/device it was originally issued to.</p> <p>In addition, the PID Provider also needs to verify that the Wallet Instance controls the private key associated with the public key it provides. For that the PID Provider also requires proof of possessing the private key ('Proof of Possession' data object). For example, the PID Provider could include a random unique value in the request and by signing this value the Wallet Instance can prove they control the private key.</p>
Request user's knowledge /inherence factor for key material	<p>Wallet Instance requests the user's knowledge/inherence factor. This will be used to protect the private key that is generated for the PID Attestation. Exact details about the knowledge/inherence factor depend on the key management mechanisms.</p>
Generate Key Pair for PID Attestation	<p>Wallet Instance creates a Key Pair for the PID Attestation. Key Pair consists of a public and private key. Public key will be shared with the PID Provider by means of Key Attestation. It is important to note that the Key Attestation can only be generated if the wallet is activated and has not been revoked.</p> <p>Specifics about key management are outside of the current project scope. Future analysis on key management topic may specify or affect the current process.</p>
Create proof of possessing private key	<p>Wallet Instance generates proof of possessing the private key. Cryptographic Proof of Possession is needed to verify that the Wallet Instance holds and the user controls the key material that will be associated with the PID Attestation. For example, to create the Proof of Possession, the Wallet Instance could sign the random unique value provided in the public key request from the PID Provider.</p>
Construct user /device binding response	<p>Wallet Instance sends the Key Attestation and Proof of Possession to the PID Provider.</p>
Verify public key for device binding	<p>PID Provider validates the Key Attestation and verifies that the private key used for creating the Proof of Possession matches the public key in the Key Attestation.</p>
Get user's authorization to issue PID Attestation to wallet	<p><i>Potential step, necessity and details of this step are to be decided, not explicitly specified in ARF.</i></p> <p>The PID Provider asks the user to continue the process by confirming PID Attestation issuance to the Wallet Instance and receives user's confirmation.</p> <p>For example, the user could sign an acceptance contract that documents the user agreement to PID Attestation issuance.</p>
Construct & sign PID Attestation	<p>PID Provider constructs the PID Attestation which will be issued to the Wallet Instance. Public key from the Key Attestation is also added to the PID Attestation. PID Provider also signs the PID Attestation as the issuer.</p>

Issue PID Attestation to the wallet	PID Provider sends the PID Attestation to the Wallet Instance.
Publish PID Attestation validity status	PID Provider makes the information about the PID Attestation status available. This information is required by the Relying Parties, when they want to check that the PID Attestation presented to them is still valid and has not been revoked by the PID Provider. It could potentially also be used by the Wallet Instance to check PID Attestation revocation status.
Validate & store PID Attestation	<p>User validates the details in the PID Attestation are correct and consents to storing it in the EE Wallet.</p> <p>It is also possible that after the user's EE Wallet Instance received the EE PID Attestation it validates that the PID Provider, who issued the PID, is an legitimate EE PID Provider and has the right to issue PID Attestations. One option for this validation to be conducted is contacting the EE Trusted List of Providers, that would include information necessary to perform the validation. But this depends on the trust mechanisms used between the parties and is not currently specified.</p> <p>Secondly, the Wallet Instance could also validate that the received EE PID Attestation is not revoked. This validation could be done using the same mechanisms as RPs use during PID presentation. Wallet Instance could query the received EE PID Attestation status from the source where the EE PID Provider published this information.</p> <p>If the EE PID Attestation's <i>valid from</i> date arrives some time after its issuance, then the Attestation should not be presentable before that date .Whether such PID Attestations are an option is to be decided.</p>

## Data Objects

In the table below, each data object from the process model is described.

Name of the Data Object	Description
User Identity	<p>User Identity is received as the result of authenticating the user with the TARA service. It contains information about the person's identity that will be used as input for creating the PID Attestation. It is also forwarded to the Wallet Provider so that the Wallet Provider can locally associate the user with a specific Wallet Instance and therefore create a user account.</p> <ul style="list-style-type: none"> <li>• Family name</li> <li>• Given name</li> <li>• Date of birth</li> <li>• Person identification code</li> </ul>
Wallet Instance Activation Token Request	Wallet Instance Activation Token Request is created by the Wallet Instance and contains information that the Wallet Provider needs to know about the Instance. Exact details are to be determined, but the request could contain data about the device that the Wallet Provider can later store when activating the Wallet Instance.
Wallet Instance Activation Token	Wallet Instance Activation Token is generated by the Wallet Provider and contains the necessary information for the Wallet Provider to determine which wallet instance must be associated with a given user.
User /Device Binding Request	User/Device Binding Request is created by the PID Provider to request a public key that will be associated with the PID Attestation. The request can also include a random unique value. This value could be used as input for proof of possessing the private key, the value would be signed using the private key.
User's Knowledge /Inherence Factor	<p>This data object contains the user's knowledge or inherence factor provided as input by the user. Wallet Instance needs it to generate a key pair for the PID Attestation being issued. The user's knowledge or inherence factor is used to protect access to the private key that is generated.</p> <p>Example for a knowledge factor is a PIN code.</p> <p>Example for a inherence factor is a biometric scan.</p>
Key Pair	<p>Key Pair that is associated with the PID Attestation being issued.</p> <ul style="list-style-type: none"> <li>• Public key</li> <li>• Private key</li> </ul>

Key Attestation	<p>Key attestation contains the public key that is bound to the PID Attestation. It also contains other necessary data that is required to assess that the Key Pair was generated securely and is stored securely. Since key management is currently out of scope then exact details are not currently described. Future analysis on the topic could change the current proposal.</p> <ul style="list-style-type: none"><li>• Public key</li><li>• Attestation</li></ul>
Proof of Possession	<p>Wallet Instance creates the Proof of Possession. One option to achieve this is signing the random unique value provided by the the PID Provider. Signature is given with the private key of the Key Pair created for the PID Attestation.</p>
PID Attestation	<p>PID Provider constructs the PID Attestation which will be issued to the Wallet Instance. PID Attestation is signed by the PID Provider.</p> <ul style="list-style-type: none"><li>• Family name</li><li>• Given name</li><li>• Date of birth</li><li>• Age over 18 (Assertion: true/false)</li><li>• Unique identifier</li><li>• Metadata<ul style="list-style-type: none"><li>◦ Issuance date</li><li>◦ Expiry date</li><li>◦ Issuing authority</li><li>◦ Issuing country</li></ul></li></ul>

# BPMN Guide

Business Process Model and Notation (BPMN) is a notation that is used to model processes in this analysis. This section describes the main elements of the BPMN that were used in the process models of this analysis.

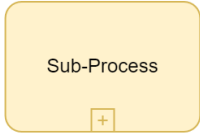


Describes the work being done in the process

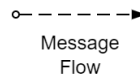


Data Object

Information produced as result of the process (activity) or required by the process (activity).



Informs that the activity holds more detail that is described in a separate sub-process. In this project mostly used to indicate that the activity includes details that are not analyzed in scope of the project



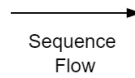
Message Flow

Represents messages from one process participant to the other



Start Event

Signals the start of the process



Sequence Flow

Connects flow objects in a sequential order



Message Start Event

Signals the start of the process and indicates that message received from other participant triggers the process to start



Data Association Flow

Shows relationships between data objects and flow objects



Message Intermediate Event

Indicates a message event that occurs between the start and end of the process. A message is received by other participant of the process



Exclusive Gateway

Used to indicate alternative paths within the process. Only one path can be taken during a given instance of the process.



End Event

Signals the end of the process



Message End Event

Signals the end of the process. Receiving a message marks the end of the process



BPMN pool describes an organization. The pool can (but does not have to) include lane(s), that depict specific participant(s) within the organization executing the activities in that lane. The pool can also not include any activities. In that case it is called a black box pool and does not show any internal details, but displays the incoming and outgoing messages from the pool.