

## **Annex 1 – Front-end prototype providers technical onboarding package**

This technical onboarding package provides an overview of workplan, the prototype scope, core functions and a description of the prototype settlement core to the front-end prototype providers. The aim of this prototyping exercise is to test how well the technology behind a digital euro integrates with prototypes developed by providers. There are no plans to re-use the prototypes in the subsequent phases of the digital euro project, if there will be any.

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

## Table of Contents

<b>1.</b>	<b>Work plan with the two front-end prototype scopes</b>	<b>4</b>
<b>2.</b>	<b>Prototype components and allocation of scope</b>	<b>7</b>
<b>3.</b>	<b>Functionality provided by the Wallet service</b>	<b>10</b>
3.1	Custodial wallet management	11
3.2	User Authentication and Transaction Authorization	11
3.3	Connectivity and lookup services among intermediaries	11
<b>4.</b>	<b>Offline solution</b>	<b>12</b>
4.1	High level architecture	12
4.2	Basic assumptions	12
4.3	Integration with other systems	13
<b>5.</b>	<b>Functional split</b>	<b>14</b>
5.1	Lifecycle management	15
5.2	Onboarding, Offboarding	15
5.3	Liquidity Management	15
5.4	Digital euro in circulation	16
5.5	Control of amount in circulation	17
5.6	Transaction initiation	18
5.7	Authentication	24
5.8	Fraud Monitoring	24
5.9	Pre-validation	24
5.10	Settlement instruction	25
5.11	Settlement validation	25
5.12	Settlement recording	25
5.13	Post settlement and settlement reporting	25
5.14	Business user master data	25
5.15	Intermediary master data	25
5.16	System master and parameter data	25
5.17	Transaction data in the digital euro environment	26
5.18	Reporting and analytical services	26

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

5.19	Billing and invoicing	26
5.20	User to application (U2A) interface	26
5.21	Application to application (A2A) interface	26
5.22	Non-functional requirements	26
<b>6.</b>	<b>Prototype settlement core</b>	<b>26</b>
6.1	High-level view of the solution	27
6.2	UTXO Data model	27
6.3	Transactions	28
6.4	Wallets	29
6.5	Technical assumptions	30

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

## 1. Work plan with the two front-end prototype scopes

The following tables give an overview of key deliverables, evaluation metrics and a timeline for the front-end prototype development.

Scope	ECB deliverables	Provider deliverables	Timeline
<b>Reduced front-end scope</b>	I) PSD2-like API description  II) High level technical description of Core settlement engine, including technical specification of transaction format and transaction flows, technical standards for messaging etc	I) High level architecture and functional overview	After contract signature as part of the onboarding package. Provider deliverable: 15 <sup>th</sup> of September
	I) Mock PSD2-like API endpoint providing mock responses to payment functions	TBD	October 2022
	I) Implementation of Wallet service using Client behind exposed PSD2-like API  II) Implementation of Core settlement engine	TBD	November 2022
		I) Delivery of end-user facing application/device that is able to interact with the PSD2-like API of the Wallet service and ability to make payments  II) Completion of user journey, fulfilment of industry standards  III) Coverage of all necessary functions of chosen use case  IV) Final report (it is expected that (integration) testing continues in Q1 2023, it might therefore be agreed to deliver a second version of the final report after closing the testing)	December 2022

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

Scope	ECB deliverables	Provider deliverables	Timeline
		V) Presentation of the solution VI) Access to the test-environment of the prototype	
<b>Full front-end scope</b>	I) PSD2-like API description II) High level technical description of Core settlement engine, including technical specification of transaction format and transaction flows, technical standards for messaging etc	I) High level architecture and functional overview	After contract signature as part of the onboarding package. Provider deliverable: 15 <sup>th</sup> of September
	I) Mock PSD2-like API endpoint providing mock responses to payment functions II) Mock Core-Settlement API layer	TBD	October 2022
	II) Implementation of Core settlement engine behind Core settlement API Layer	TBD	November 2022
		I) Delivery of end-user facing application/device that is able to interact with the PSD2-like or custom front-end provider API of the Wallet service and is able to conduct payments II) Completion of user journey, fulfilment of industry standards III) Coverage of all necessary functions of chosen use case IV) Final report (it is expected that (integration) testing continues in Q1 2023, it might therefore be agreed to deliver a second version of the final report after closing the testing) V) Presentation of the solution VI) Access to the test-environment of the prototype	December 2022

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

## 2. Prototype components and allocation of scope

The Digital Euro prototype consists of three layers of applications:

1. **Core settlement engine (ECB internal name: N€XT Core):** a settlement engine which processes transactions. (Section 6 provides a description of the prototype settlement core).
2. **Wallet service:** A collection of services which are responsible for (i) managing user wallets/accounts; (ii) receiving payment instructions from user-facing applications; (iii) converting payment instructions into transaction messages which can be sent to the core settlement engine for settlement; (iv) routing payment instructions (and notifications) between different wallet/account management services (the **Communication service**). A service which combines (i)-(iii) is called a **Client**. Functionality (iv) allows multiple Clients to communicate. It will only be implemented in case of need and to the extent necessary.
3. **User facing apps / devices:** devices or applications which interact with the user.

The **Core settlement engine** will be developed by the ECB. As outlined above, as a stand-alone component it is not a full end-to-end solution for the processing of retail payments. To ensure at least one end-to-end implementation of the digital euro prototype, the ECB will also build one example **Wallet service** including one Client implementation. For the prototyping, the Eurosystem will not provide a **Communication service** or encompassing payment scheme that would allow different payment solutions to interoperate. If front-end providers would be interested in demonstrating scheme-like functionalities, such as payments between wallets hosted by different providers, they are free to do so, but it would also require them to simulate multiple providers in their prototype and build a communication service.

Together with the Core settlement engine, a Wallet service comprises a full payment processing solution with which the apps and devices in the third layer can interact. Since the core settlement engine is an UTXO-based processor, the Wallet service will be responsible for crafting UTXO transactions and managing wallets (as opposed to accounts).

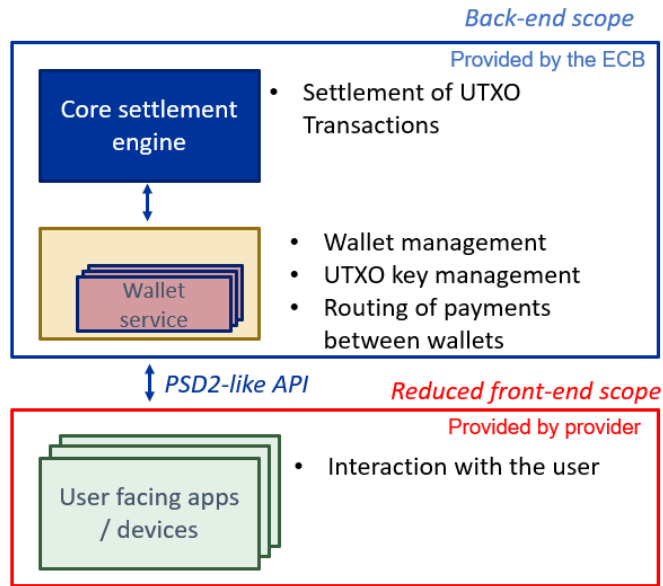
There are two possible scopes for the front-end prototype providers selected from the market.

First, user facing apps and devices define the **reduced scope for the front-end prototype**. Second, since it is foreseen that a Wallet Service would be developed by intermediaries in a production system, and since there is no reason to exclude having more than one, it will also be possible for front-end providers to develop a Wallet Service. This defines the **full (and preferred) scope for the front-end prototype**.

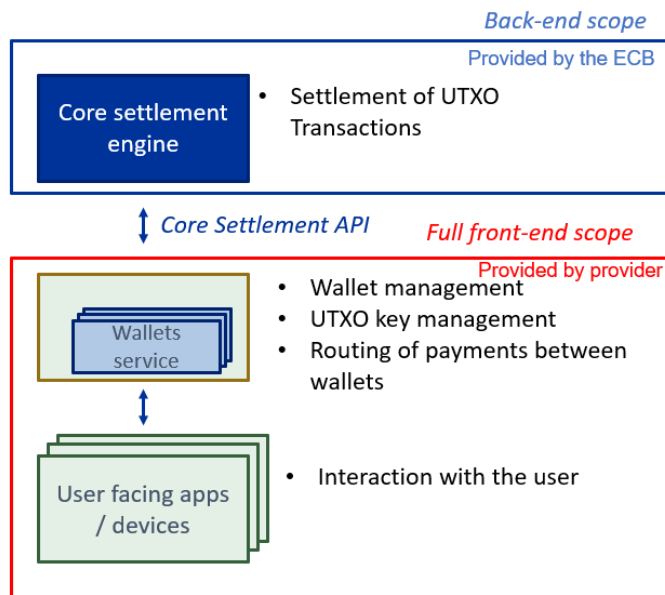
Hence there are two possible ways to draw the scope boundary between Eurosystem-developed back-end and the market-developed front-end prototypes:

- With a **reduced front end scope, the Eurosystem provides the Wallet Service component in addition to the Core settlement engine** (see Figure 1 below)
- With the **full front-end scope, the Eurosystem only provides the Core settlement engine** (see Figure 2 below).

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro



**Figure 1: The first allocation of back-end and front-end prototypes**



**Figure 2: The second allocation of back-end and front-end prototypes**

The Eurosystem can provide the Wallet service for those cases where the provider only intends to provide a reduced front-end scope; in that case the scope boundaries of Figure 1 will apply, otherwise the boundaries and allocations shown in Figure 2 will apply.

Figures 3 – 6 below will provide process flows for payer- and payee-initiated payments, with an indication of the scope for the full front-end scope and the reduced front-end scope.

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro



## Full front-end scope

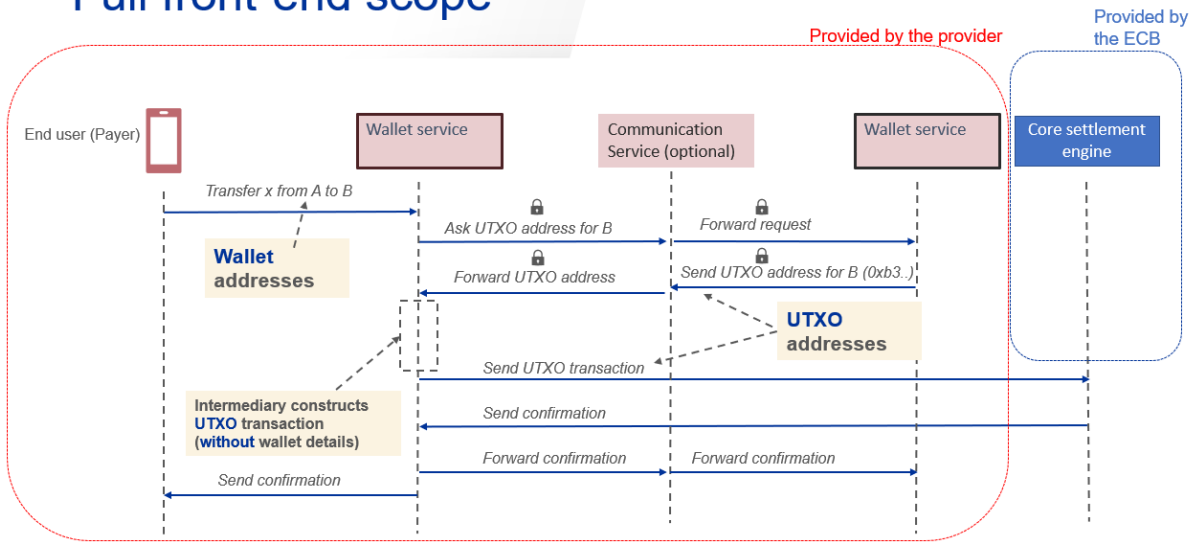


Figure 3: Full front-end scope payer-initiated process flow

## Reduced front-end scope

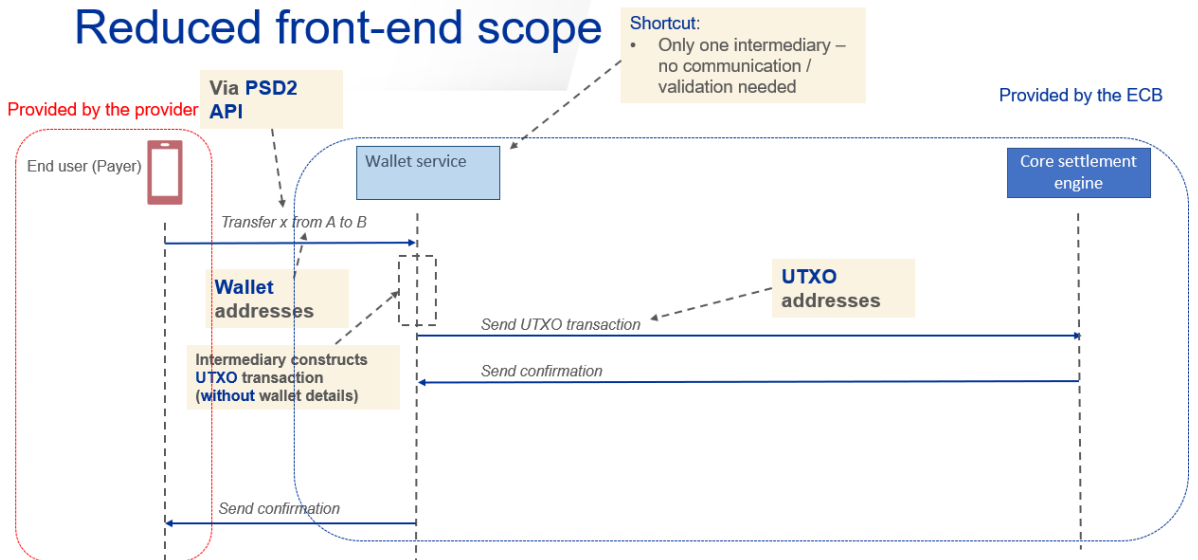


Figure 4: Reduced front-end scope payer-initiated process flow

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

## Full front-end scope: Payee-initiated

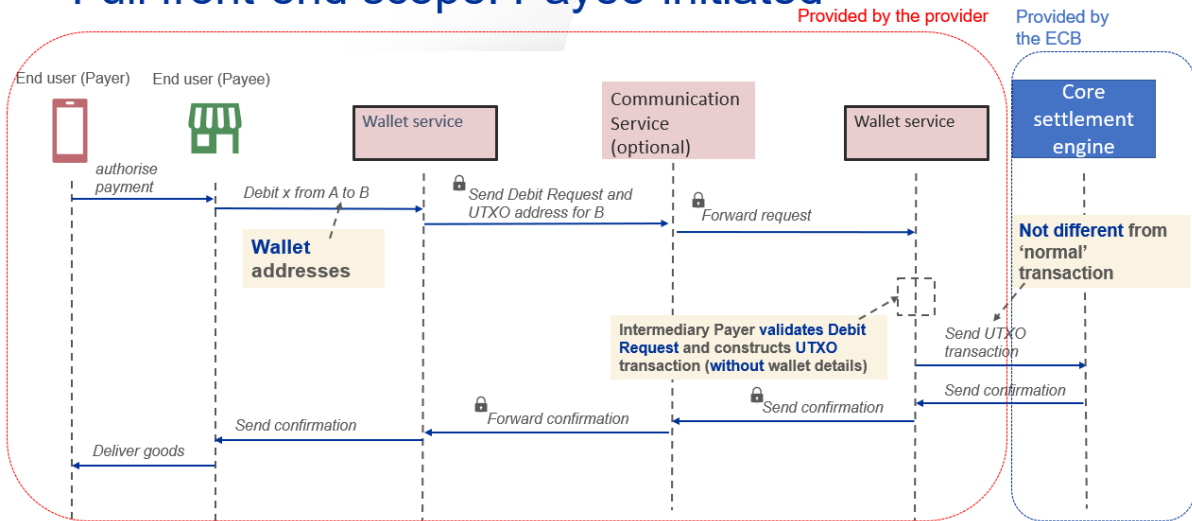


Figure 5: Full front-end scope payee-initiated process flow

## Reduced front-end scope: payee initiated

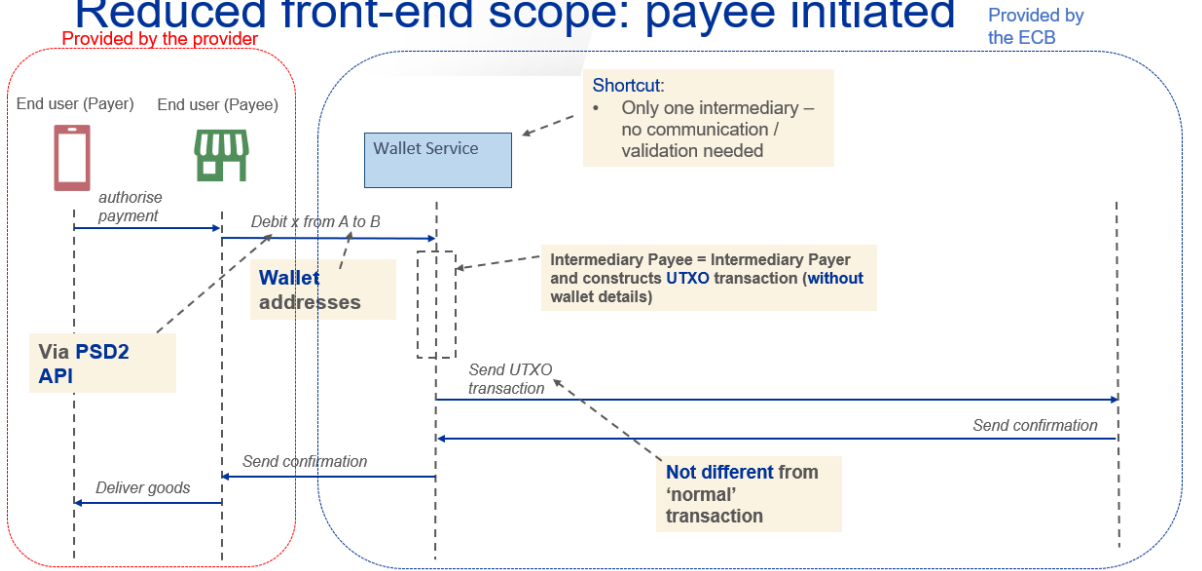


Figure 6: Reduced front-end scope payee-initiated process flow

### 3. Functionality provided by the Wallet service

We will now give more details on the Wallet service. As explained above, the ECB will build an example Wallet service for the reduced front-end scope. In the full front-end scope, providers may provide their own Wallet service. A Wallet service is an application with the responsibilities (i)-(iv) described in the previous section. In a Production architecture, it is intended to be operated by multiple intermediaries. For the prototype:

- There could be **several front-end prototype providers** that act as **wallet provider**, i.e. that host wallets and construct UTXO transactions and send them to the core settlement engine.

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

These are referred to as **Clients**, as they are directly involved in the initiation and execution of the transactions.

- We do not assume that the different Clients communicate with each other, but it can be optionally implemented by the provider.

Clients interact with the Core settlement engine through the Core Settlement API. They provide payment functionality by exposing a PSD2-like API to user-facing applications and devices (the API specifications will be provided in separate files).

The following paragraphs describe the basic functionalities in more detail. As stated in the beginning, the full scope of the prototyping exercise also involves developing an example Wallet Service. An initial design proposal is contained in the N€XT chapter of the Architecture Scenario Analysis.

### 3.1 Custodial wallet management

The Clients operating in the Wallet Service will manage user identities, not directly known to the Core settlement engine. The default model will be for each Client to be in charge of the **secure storage and management of the users' private key(s)**<sup>1</sup>, as this (or these) will be used to sign the transactions. This choice (as opposed to managing the user keys within the end-user devices) defines a so-called custodial wallet management model, where the intermediary operating the Clients hold the user's keys in custody on behalf of their owners. This model has several advantages, among which the most important is simplifying the end-user devices and applications and providing more assurance.

The Wallet service will also manage the computations required to create the transaction messages accepted by the core settlement engine in the proper format, which includes the cryptography operations that use the keys and that are necessary to manage the UTXOs of a transaction.

### 3.2 User Authentication and Transaction Authorization

The Client will provide a basic authentication service towards the user device and application layer that will be used by the front-end Clients to obtain a "temporary session token" to authenticate any user.

Towards the core settlement engine, the Client will need to sign transactions with the instructing user's private key and (possibly) authenticate the transaction as coming from the client.

### 3.3 Connectivity and lookup services among intermediaries

In case the prototype simulates multiple intermediaries, they need to support some basic interactions:

- A communication service to facilitate communication between the intermediary of the payer and the intermediary of the payee
- the lookup of an intermediary "address" based on the payee identification (as this is required, e.g. for the notification of an executed transaction, as the core settlement engine is not supposed to store or learn of payee details for privacy reasons)
- a PSD2-like proxy lookup service, for the discovery of a payee details (including the abovementioned intermediary address) through a user-friendly lookup key (e.g. email address or phone number)
- the inquiry of a payee service to obtain a dynamic address ("UTXO address") to be used in constructing a transaction

---

<sup>1</sup> This is a direct consequence of the custodial model chosen for the prototype activity

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

## 4. Offline solution

In addition to the use cases described in the previous chapter one part of the prototyping exercise is to showcase an end-to-end solution capable of *consecutive offline transactions* among devices, including the relevant feature to fund and defund the *wallets* from a *simulated* liquidity source.

The “Offline transaction” is defined as a “peer-to-peer validated transaction with finality operated in close proximity”. This excludes the *need* to involve a third party to either validate or settle the transaction.

### 4.1 High level architecture

Although the provider is expected to provide its own view of the end-to-end solution, this section describes a general reference architecture that is in line with some basic assumptions specified in the following sections.

The provider is expected to set up a system that includes:

- The devices (*wallets*) with which the transactions take place;
- An online platform running at the intermediary level, for funding and defunding operations (see next section);
- An online platform running at the Eurosystem level, dedicated to issuing and redeeming the “offline currency”.

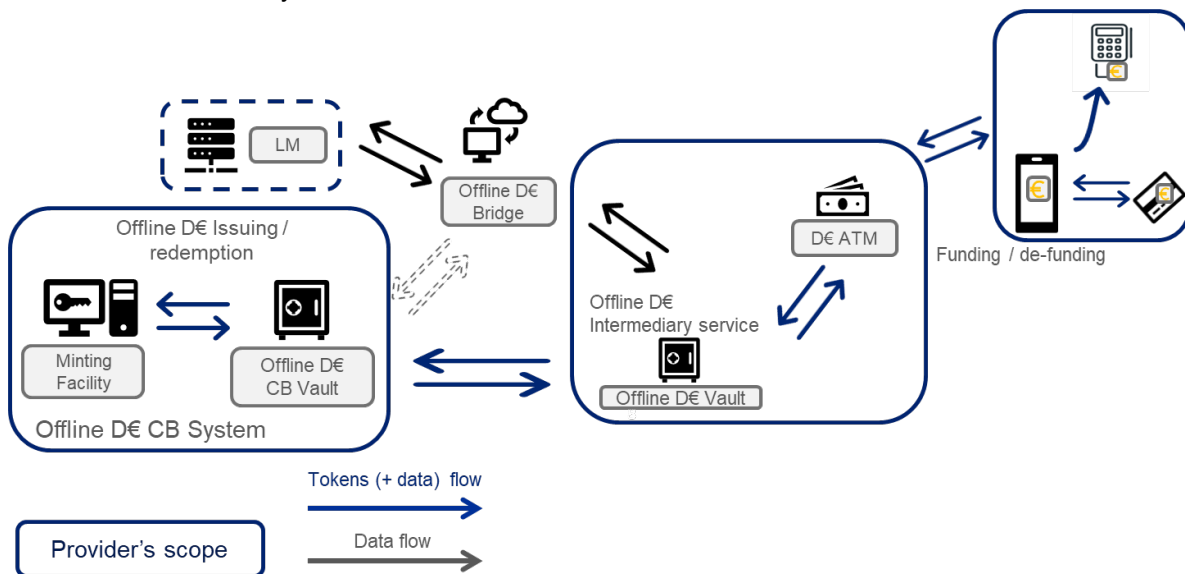


Figure 7 - Tentative high-level architecture for the offline solution

Depending on the design decision by the provider, a *vault* (i.e. a facility to securely store value tokens) may be required at the intermediary level and/or at the Central Bank (CB).

### 4.2 Basic assumptions

The offline solution can work independently from the online Digital Euro platform. For the *transaction* itself, this is based on the basic definition. The *wallets* will go online, and interact with an online backend for “reconciliation” (i.e. to offload transaction history, to refresh tokens, etc., but in any case not at the time of the transaction)

Offline transactions are *final*, and they cannot be revoked in *any* case, even in circumstances where a fraud is detected ex-post.

Offline funds can be spent, *even offline*, immediately. Transaction time is *comparable* with the one currently available with cards (for reference, *under 1 second*).

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

The security (i.e. non repudiation) of the operation should be ensured and be supported by an action of the payer (e.g. “unlocking the device” prior to executing the payment).

Funding (and defunding) operations demonstrations are among the main goals of the activity. We assume that these operations *only* go through intermediaries.

Eurosystem (ECB, in the context of the prototype) manages the total amount of offline digital euro in circulation, and offline funds can only be *created* at the ECB level.

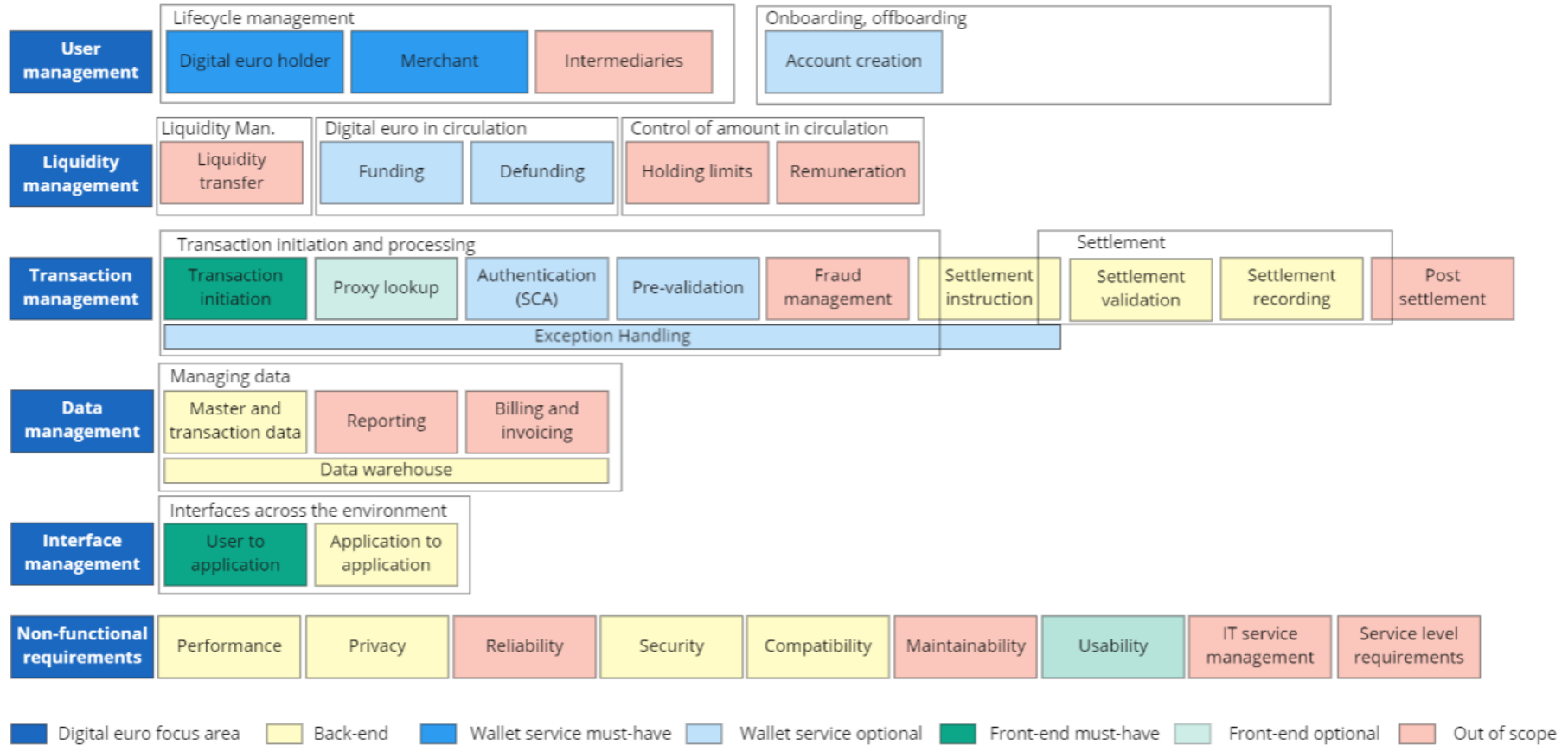
### **4.3 Integration with other systems**

In the scope of the prototype, only one possible integration for the offline solution is foreseen, specifically in orchestrating the synchronous “distribution” operation (i.e. the transfer of offline currency between the Central Bank and the requesting intermediary) or the dual request of moving back the funds (redemption).

It is assumed that this request can be triggered by the intermediary-level component of the offline solution, via a call to the Bridge component depicted in the tentative architecture.

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

## 5. Functional split



The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

## 5.1 Lifecycle management

A user manager should ensure that data used for identification of a digital euro holder is valid at all times. A user manager should also encourage a digital euro holder to promptly communicate other changes in personal data (such as change of digital euro holder's home address, loss or theft of ID document, etc.).

A user manager shall store in a secure location all the updated data provided by a digital euro holder.

### **Authenticate a digital euro holder**

*See chapter 5.7 Authentication for more information on the authentication process.*

A digital euro holder should be able to access at least digital euro balance and transactions history after passing online authentication. For some online interactions with the digital euro environment a digital euro holder might be required to reauthenticate.

### **Update digital euro holder's data**

A digital euro holder provides updated information, such as new home address, new proxy look-up data or new passport or ID card.

### **Deriving additional information from updated data**

*Deriving additional information from updated data is out of scope for the prototyping.<sup>2</sup>*

A user manager shall in addition to updating digital euro holder's data store also information when data was last modified and, if applicable, update the information regarding existence of a link between digital euro account/wallet and commercial bank account or change a digital euro account/wallet type.

### **Use of privacy enhancing tools**

*Use of privacy enhancing tools is out of scope for the prototyping.*

A user manager shall use privacy enhancing tools to protect an updated repository data.

## 5.2 Onboarding, Offboarding

*Onboarding and offboarding is out of scope for the prototyping.*

## 5.3 Liquidity Management

*Liquidity Management is out of scope for the prototyping.*

A liquidity transfer is the process, controlled and performed by the Eurosystem, to move central bank reserves between a supervised intermediary's main central bank reserves and its central bank reserves dedicated for the use in the digital euro environment, i.e. a transfer between the MCA (main cash

---

<sup>2</sup> For the functions that are marked out of scope, the ECB doesn't require those, but the providers can still decide to include those functions in their prototype.

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

account) and a D€ DCA (digital euro dedicated cash account). It is executed upon request by supervised intermediaries to satisfy the expected demand from end users.

#### **5.4 Digital euro in circulation**

*Controlling the digital euro in circulation is out of scope for the prototyping.*

##### **Funding**

Funding is the process, controlled and performed by the Eurosystem, of converting digital euro dedicated central bank reserves to newly issued digital euro at par and simultaneously moving it into the possession of an end user, creating a claim of an end user on the central bank. It is executed upon request by an end user via their wallet service. The reverse considerations apply to defunding.

##### *Funding in the prototype*

In the prototype, a mock-up facility will be implemented that enables wallets to be funded on request (without a payment in the other direction). This is to be specified in the APIs.

##### *Pure funding*

End users would be able to trigger a funding request via their wallet service. This could be done either manually, where the end users themselves initiate the funding, or automatically based on by a predefined event. The automated funding would be a value-added service offered by the wallet service. Examples for triggers for automated funding: position going below a predefined threshold, at specified intervals (e.g. every 1st of a calendar month), allocate certain percentage upon salary payment, etc.

##### *Manual funding*

It is a basic functionality that enables the end user to manually start the conversion of a given liquidity source into digital euro. The process could be fully digital (for commercial bank money) or involve a physical step (cash to digital). In the first case, the end user would instruct its wallet service to transfer funds from its commercial bank money holdings to the digital euro ones. In the latter case, the end user would deposit banknotes in exchange for digital euro funds, making it possible to also include the unbanked end users.

##### **Defunding**

##### *Manual Defunding*

Following the same logic as for manual funding, this basic functionality enables the end user to manually convert digital euro into commercial bank money or to withdraw cash in exchange for digital euro.

##### *Waterfall*

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro



Waterfall: functionality to ensure that end users can still receive a payment even if their post-transaction position exceeds the individual holding limit (mandatory defunding, triggered by an incoming payment that brings the payees holding above the allowed holding limits)

This functionality supports avoiding the rejection of an incoming payment that would result in exceeding the individual holding limit, should this be set by the Eurosystem. Given that such functionality works as an event-driven automatic defunding, it requires by design that the end user interested in using it agrees to establish a univocal link to a commercial bank money liquidity destination, e.g. a deposit account or credit card. Consequent to an incoming payment, the waterfall functionality refers to pushing received liquidity exceeding the established digital euro ceiling to the linked end user commercial bank money liquidity destination.

This functionality enables the end users to receive a digital euro payment regardless of its amount as holding limits would not result in payments being rejected. The waterfall functionality acknowledges financial stability considerations to avoid the holding of an excessive digital euro amount, while at the same time leveraging user convenience by guaranteeing end users that any incoming payment can be settled.

#### *Reverse Waterfall*

Reversed waterfall: functionality to ensure that end users can still make a payment even if the payment amount exceeds their current digital euro holdings (funding operations connected to an outgoing payment).

The reversed waterfall functionality foresees that the additional liquidity needed to reach the payment amount is automatically pulled from the linked end user source of commercial bank money liquidity. Given that it would work as an event-driven automatic funding, it requires by design that the end user interested in using it agrees to establish a univocal link to a commercial bank money source.

From a practical point of view, the reverse waterfall will result into two transactions, the first one to fund the digital euro position with sufficient liquidity, and a second one, simultaneously or immediately thereafter, to complete the digital euro payment in its full amount. Technically, these two events need to be linked to avoid race conditions, where one event happens before or without the second one. Also, more practical implementations possibly not needing these two events could be further explored, depending also on the final configuration of the system (e.g. with UXTOs there might be no need to move UTXOs from the supervised intermediary to the payer before making the transaction or with accounts there could be a single payment instruction resulting in the missing amount being directly deducted from the supervised intermediary's D€ dedicated cash account during settlement).

## **5.5 Control of amount in circulation**

*Holding limits and remuneration are out of scope for the prototyping.*

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

To reduce complexity, the holding limits would be applied per digital euro account, and not per end user.






















Generally applicable holding limits are specified by the Eurosystem, for each end-user type (e.g. natural persons would be able to hold up to xx€, business would be able to hold up to yy€, etc.). There would be no deviation within (sub)types (e.g. the same limits would apply for all natural persons regardless to their location).

## **5.6 Transaction initiation**

Refers to the requirements related to the instruction of the payment order to the back-end including collecting the required payment data. The transaction can be initiated either by the payer or by the payee via their wallet service. For the avoidance of doubt “payer initiated” is defined as the payer instructing its issuing wallet service to credit the payment instrument of the payee whereas “payee initiated” is defined as the payee instructing its acquiring wallet service to debit the payment instrument of the payer.

Below a high-level overview of the different payment scenarios envisaged including variables impacting the payment initiation. This includes certain functionalities that are in subsequent sections detailed-out, however, relevant at this stage to mention, as it facilitates the understanding of the payment initiation process.

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

	Form factor combination	Consumer device/hardware	Consumer interface	Data exchange tech	Transaction initiation
P2P	Proximity: QR-code-based payment via mobile device	 Mobile device	 App	 QR-code	Payer-initiated
	Proximity: NFC-based payment via mobile device	 Mobile device	 App	 NFC	Payer-initiated
	Remote: internet-based payment via mobile device or computer	 Mobile device or computer	 App or online interface	 Internet	Payer-initiated
POS	NFC-based payment via mobile device or card or wearables	 Mobile device, card or computer	 App or physical card	 NFC	Payee-initiated Payer-initiated
	Chip-based payment via card	 Card	 Physical card	 Chip	Payee-initiated
	QR-code based via mobile device or card (printed)	 Mobile device, card or computer	 App or physical card	 QR-code	Payer-initiated Payee-initiated
e-commerce	Internet-based payment via mobile device or computer	 Mobile device or computer	 Online interface (and potentially app)	 Internet	Payer-initiated Payee-initiated

**P2P use case**

The following description are soft requirements that can be amended by the front-end provider based on their experience.

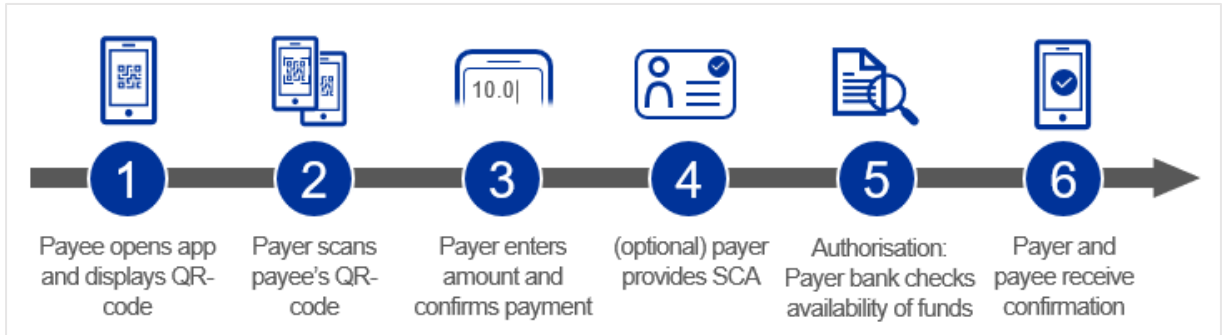
P2P payments are payer initiated, however, there are multiple options how the payment process can be started. Typically, P2P payments are carried out via mobile device or computer. Main differentiator in that regard is the environment (proximity or remote communication between payer and payee).

**Proximity: QR-code- and NFC-based payment via mobile device**

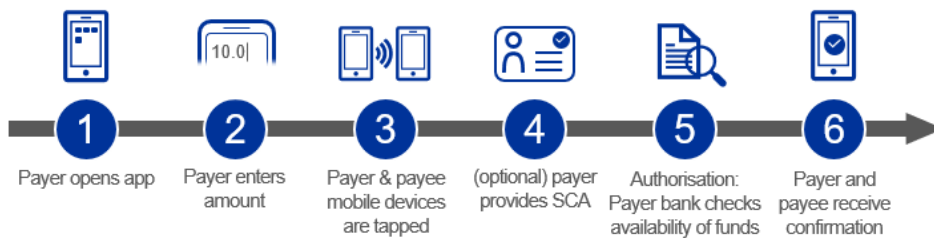
In general, two different processes are possible for QR-code based payments, depending on whether static or dynamic QR codes are used whereas for the NFC-based payment option one process can be identified.

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

- Static QR-code:** The payee is opening the app and is displaying the QR-code. The payer scans the QR-code with a mobile device, by which a prefilled form with relevant payment details is generated. The payer still must manually introduce the payment amount. If using a dynamic QR-code this step will not be necessary, as the amount and potentially other relevant information will be pre-filled by the payee. Afterwards, payer could be authenticated via SCA (this would usually depend on the amount transferred). Both parties receive a payment confirmation



- Dynamic QR-code:** The payee is opening the app and selects to generate a QR-code. The payee is prompted to specify the payment amount it likes to receive. Subsequently, the QR-code is created, containing (i) the payment information of the payee, and (ii) the amount the payee requests. The payer is now only scanning the dynamic QR-code and will see a prefilled form containing all relevant information of the transaction. Before accepting the form, SCA might also apply. Again, payer and payee receive a confirmation.
- NFC-based payment via mobile device:** The payer opens the app and specifies the amount it likes to transfer. Subsequently, the mobile devices are tapped and depending on the payment amount the payer might be requested to authenticate itself via SCA. As soon as the payer wallet service authorises the payment and confirms it to payee wallet service, both parties receive a confirmation.



**Remote: Internet-based payment via mobile device or computer**

In a remote scenario either the payer can instruct its issuing PSP to credit the payment instrument of the payee by entering a specific digital euro payee ID (can also be a proxy e.g. phone number or email address) or post receiving a request-to-pay message including payee details and the amount to be transacted.

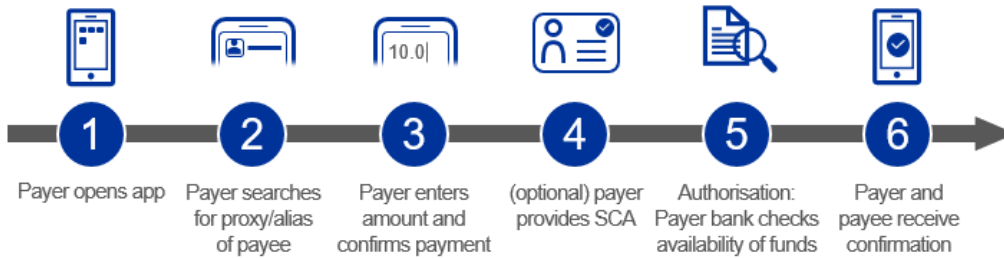
The end-user experience is different depending on whether a mobile device or a computer is used. The use of a mobile device provides a more convenient end-user experience for the majority of users, can

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

address proximity payments too and can support different technologies for payment initiation and authentication.

The examples shown are based on the use of a mobile device.

- Payer initiated:* the payer opens the app and search for the proxy/alias (e.g. phone number or email address) of the payee which is linked to its digital euro ID. Afterwards the payer specifies the payment amount and potentially authenticates itself via SCA. Both parties receive a payment confirmation after payment authorisation



- Payer initiated via payment request:* the payee is opening the app and selects the option to send a request-to-pay (RTP) to the payer. The payee would need to specify the amount it requests. Subsequently, the RTP is sent to the payer. This notification is usually done via e-mail or SMS. Then, the payer can open its app and check the payment request. The RTP contains a prefilled form with all relevant information of the transaction. The payer would just need to accept the RTP, after having potentially authenticated via SCA and would thereby trigger a credit transfer to the payee.



**C2B use case**

*The following description are soft requirements that can be amended by the front-end provider based on their experience.*

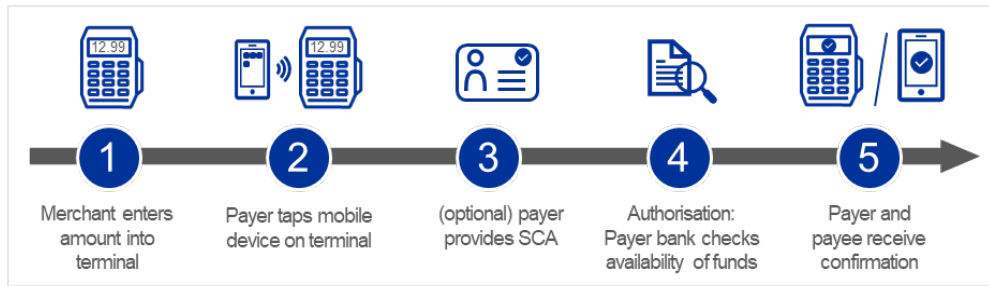
C2B payments are payer or payee initiated, hence, there are multiple options how the payment process can be started. Typically, C2B payments are carried out via mobile device, computer, wearable or physical card. Key differentiator in that regard is the environment either point-of-sale (POS) or e-commerce not only impacting how the payment is initiated but also by whom.

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

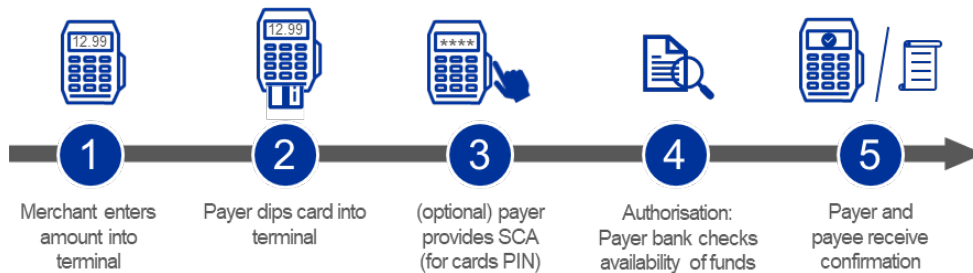
**POS: NFC-, chip, QR-code- based payment via mobile device, physical card, or wearable**

POS payments are by nature proximity transactions. Form factors supporting POS payments enable either payee or payer-initiated transactions, depending on the type of consumer device and the technology used. Physical cards, mobile devices and wearables are the consumer devices used for this use case. In terms of transaction initiation, mobile devices and wearables can support payee and payer-initiated payments, whereas physical cards support payee-initiated. In terms of authentication, mobile devices support different methods

- *Payee initiated: NFC-based payment via mobile device, card or wearable:* the merchant (payee) starts the payment transaction by entering the payment amount into the POS terminal. The payer taps its device (card/ mobile/ wearable) on the terminal or mobile device of the merchant and potentially authenticates via itself SCA. Authentication is done with the end-user device (e.g. via fingerprint on phone/ card or via PIN introduced on the app/ terminal). Merchant and payer receive a confirmation on the merchant’s terminal or mobile device that the payment was successful.



- *Payee initiated: Chip-based payment via card:* Alternatively, the payer can also insert its card into the terminal's card reader. The payment process is the same as above, except that the card would need to be dipped into the card reader. If SCA is required there are again multiple options on how to ensure two-factor authentication (e.g. PIN on terminal or fingerprint on card)

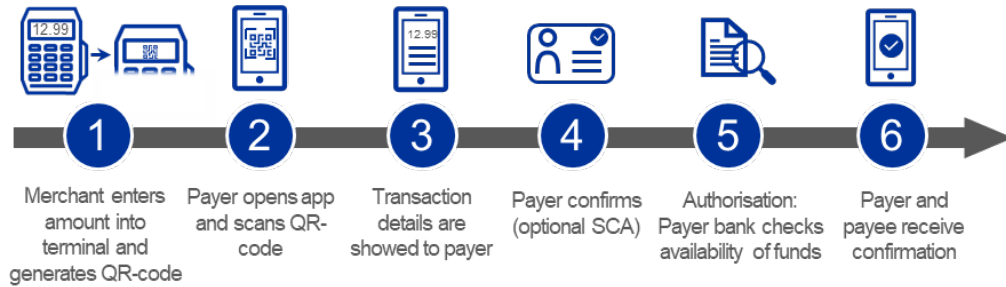


- *Payer initiated: QR-code-based payment via mobile device or wearable:* QR-codes are typically used via an app on the mobile device (wearable also possible depending on functionality) can be differentiated between two processes for QR-code-based POS payments, depending on who presents the QR-code: merchant (payee) or consumer (payer)

- Merchant presents QR-code: The merchant (payee) is either showing a static QR-code or it is generating a dynamic QR-code by entering the payment amount into its terminal/mobile

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

device. In case of the dynamic QR-code, the consumer opens its app and scans the QR-code. Transaction details are displayed, and the consumer confirms the transaction via SCA if required. This will trigger the payment and the transaction will be confirmed to both parties. In case a static QR-code is used the payer is required to enter the amount in-app.



- Consumer presents QR-code: The consumer (payer) opens its app and generates a QR-code. The information is retrieved by the merchant and a payment request message is generated. Transaction details will be displayed, and the consumer will confirm the transaction via SCA if required. The payment request will trigger the initiation of a credit transfer and transaction will be confirmed to both parties.



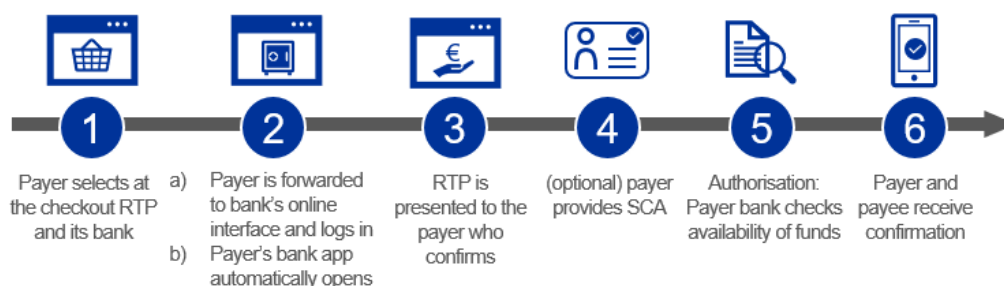
**E-commerce: internet-based payment via mobile device or computer supported by a web-interface or app**

*The following description are soft requirements that can be amended by the front-end provider based on their experience.*

E-commerce payments are by nature remote transactions. Form factors supporting e-commerce payments enable both payee and payer-initiated transactions, however, for the basic payment use case a payer-initiated transaction is the most practical option. Recurring transaction for which payee-initiated transactions similar to direct debits are the most practical option are explored at a later stage.

- *Payer initiated: internet-based payment via mobile device or computer supported by a web-interface or an app where the consumer initiated a credit transfer triggered via a RTP: the payer selects in the merchant’s check-out page “request-to-pay (RTP) via digital euro” as payment option and selects its issuer. The payer can access the RTP either via app or online interface. After having received the RTP message, a prefilled form with all relevant information of the transaction is presented to the payer. The payer only needs to accept the RTP message and would thereby automatically trigger the payment to the merchant, making it effectively a payer-initiated payment. The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro*

Depending on the payment amount SCA might be required. Finally, payer and payee are informed about the successful payment.



## 5.7 Authentication

*Authentication is out of scope for the prototyping.*

Authentication refers to the procedure, allowing to verify the identity of a wallet service user or the validity of the use of a specific payment instrument, including the use of the user's personalised security credentials. Current regulatory requirements (PSD2) mandate the application of strong customer authentication (SCA) to electronic payments. This is based on the use of two or more elements categorised as knowledge (something only the user knows), possession (something only the user possesses) and inherence (something the user is) that are independent. The combination of consumer device, data exchange technology and end-user interface delivery determine how authentication is performed and who can do it. To note that evolving electronic payments regulation will impact authentication requirements, and the respective standards are to be deployed in the context of digital euro transactions.

## 5.8 Fraud Monitoring

*Fraud Monitoring is out of scope for the prototyping.*

Wallet services might be obliged to perform risk management and fraud prevention and detection measures and to monitor the digital euro activities within their area of responsibility. Ideally, the digital euro specific risk and fraud management will be integrated into existing infrastructures and applications.

## 5.9 Pre-validation

*Authentication is out of scope for the prototyping in the sense that they will not be actively supported by the ECB. Wallet services or user facing apps can simulate the functions.*

Wallet services involved in the transaction process perform a set of pre-validation before a settlement instruction will be initiated. These pre-validations may include technical protocol validations as well as business validations such as presence of valid and active contract to accept digital euro payments by the payee, availability of funds on the commercial bank money account of the payee, limit checks or AML/CFT checks.

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro



### **5.10 Settlement instruction**

Process triggering the funds transfer after successful validation of a transaction.

### **5.11 Settlement validation**

Details requirements related to checking the payer's fund availability, eliciting the reservation of funds, and any other task that may be necessary for the validating entity or entities to assess whether such funds can be transferred from payer to payee

### **5.12 Settlement recording**

Details requirements related to the bookkeeping of funds, which mainly includes reserving funds (where necessary) and the actual transfer of funds from payer to payee.

### **5.13 Post settlement and settlement reporting**

*Post settlement and settlement reporting is out of scope for the prototyping.*

Processes related to reporting and investigations, after settlement has occurred. The digital euro settlement provider would in principle support a transaction status investigation process, which can be initiated by the payer (Originator Participant/Instructing Party).

### **5.14 Business user master data**

Business users' data like merchant country, merchant category code, supported initiation channel (eCommerce or POS) would need to be stored by the wallet service.

### **5.15 Intermediary master data**

The core settlement engine will not need to know the wallet services connected.

### **5.16 System master and parameter data**

Unique identification value. Any direct participant connected to the digital euro environment shall be identifiable. The unique identifier (such as EUIBAN) assigned to a participant must be known to the wallet service.

Merchant category code (MCC). To support fraud and risk management any business user shall be assigned to a merchant category code (MCC). Certain functions, additional data requirements or value-added services might be applicable to selected MCC only. Additional data requirements might apply to certain MCCs.

Country codes. The digital euro scheme should distribute valid payee and payer country codes to the scheme participants.

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

### **5.17 Transaction data in the digital euro environment**

To support lifecycle management of transactions a transactions history is required. The transaction history will be available in the wallet service.

### **5.18 Reporting and analytical services**

*Post settlement and settlement reporting is out of scope for the prototyping.*

The digital euro might have an own reporting engine to generate reports and statistics to monitor activities of wallet services.

### **5.19 Billing and invoicing**

*Billing and invoicing is out of scope for the prototyping.*

### **5.20 User to application (U2A) interface**

Users must be able to access data and functionality via an interface suitable for human interaction, i.e. graphical user interface like a web-based solution.

### **5.21 Application to application (A2A) interface**

The technical components of the digital euro environment exchange data between themselves and with external services (e.g. wallet services) via application to application interfaces (e.g. APIs).

### **5.22 Non-functional requirements**

*Non-functional requirements are optional for the prototyping in the sense that they will not be actively supported by the ECB. Wallet services or user facing apps can come up with relevant indicators which would be welcomed by the ECB.*

## **6. Prototype settlement core**

The settlement engine comprises a new, greenfield design by Eurosystem experts. It features a new type of transaction ledger, based on unspent-transaction outputs (UTXOs).

The settlement core performs the function described as the first application layer in Section 2. It is not an end-to-end prototype implementation of the digital euro. It is to be positioned as the settlement system of transactions used by (supervised) intermediaries. Intermediaries are responsible for constructing user wallets (the second application layer).

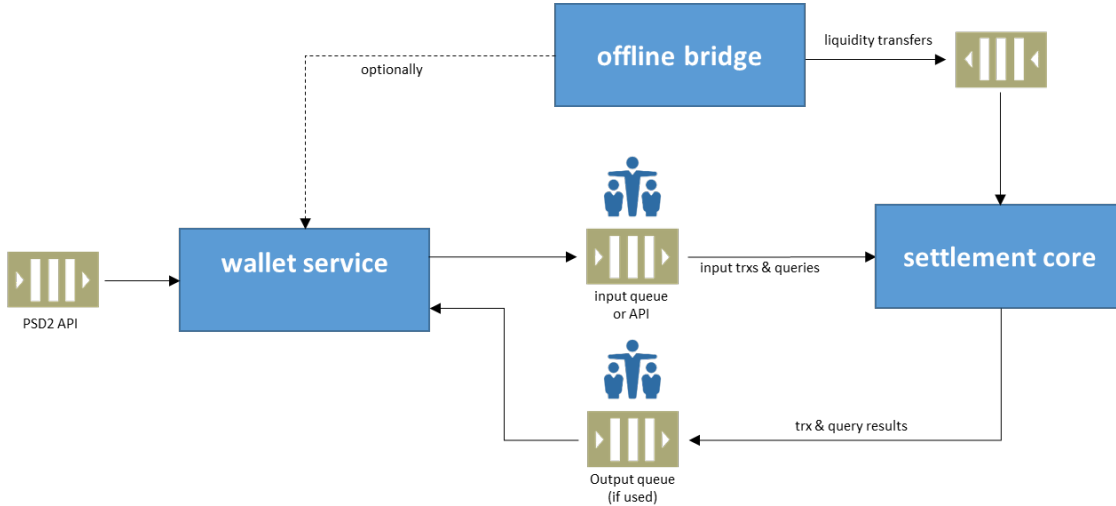
The following sections provide a high-level description of the settlement core. The details (especially on UTXO and transaction format) will only obtain their final form once the full specification of the settlement core API has been made available. The information in these sections may therefore be subject to change. The provided information is there to give an idea of the technical tasks that a wallet service is expected to fulfil as part of the prototype.

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

### 6.1 High-level view of the solution

The components of the prototype drawn in Figure 8 below.

Figure 8: The full back end of the prototype



The following components comprise the settlement core:

- The **input queue or API** for receiving transactions from intermediaries and passing them to the settlement core
- The **settlement core** which validates and processes transactions
- A **wallet service** which is responsible for constructing user wallets containing UTXO holdings
- A **bridge** component to offer compatibility with offline digital euro

### 6.2 UTXO Data model

The settlement core keeps track of the value and availability of digital euro holdings associated with a specific address. These holdings are referred to as *unspent transaction outputs*, or UTXOs. When a UTXO is spent by a transaction, it is no longer usable. In turn, new ready-for-use UTXOs are created by the transaction.

The address of a UTXO is used to verify that a transaction has been signed by the owner of the UTXO. In the proposed data model, it is given by a hash of a public key. Ownership is demonstrated by signing a transaction with the corresponding private key. Note that the meaning of the word ‘owner’ in this document and elsewhere will depend on the context in which it is used. It could designate an end user of digital euro, as the legitimate owner of their funds, but it could also designate the legitimate possessor of the private keys corresponding to their UTXOs (which would be the servicing intermediary).

The data contents of a UTXO are summarised in Table 1.

Table 1: The data contents of a UTXO

UTXO field	Content
<b>Token_id</b>	Unique identifier of the UTXO. It could be defined as <i>trx_id#i</i> where <i>trx_id</i> is the transaction where the UTXO was created as the <i>i</i> -th output token (à la Project Hamilton).
<b>Owner_address</b>	Hash of the public key of the owner
<b>Amount</b>	The value of the UTXO in eurocents

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

<b>Tx_ref</b>	Trx_id of the transaction where the UTXO was spent (only when Status = Unavailable)
---------------	---

### 6.3 Transactions

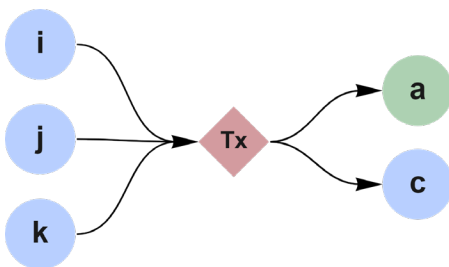
Transactions update the state of available UTXOs in the settlement core. A list of available UTXOs, *input UTXOs*, are marked 'Unavailable', and a list of new 'Available' UTXOs, *output UTXOs*, are created. From an accounting perspective, any such operation which preserves the sum of UTXO amounts could be considered valid. To process payments, it is also needed that the whole transaction is signed by the owner(s) of the input UTXO(s) (as described in the previous section), proving that the owner(s) agree(s) for the UTXO(s) to be spent in this transaction.

Transaction requests sent to the settlement core need to satisfy additional constraints, namely:

- All input UTXOs have the same address, which can be referred to as the address of the *payer*.
- There are only two output UTXOs, one of which has the same address as the owner and serves as 'change' back to the owner of the input UTXOs. The other address is the address of the *payee* of the (payment) transaction. The two output UTXOs can be ordered by letting the first (1) refer to the amount received by the payee, and the second (2) to the change.

An example of a transaction which satisfies these constraints is illustrated in Figure 9.

**Figure 9: A UTXO transaction with input amounts i,j,k and output (payee) amount a, with change  $c = i + j + k - a$ .**



The data used to build a transaction is summarised in Table 2. In this data, a reference to the instructing party (the intermediary of the payer) is included to be able to route back a response to this transaction. It can be checked that from this input data, a full UTXO transaction can be uniquely specified.

**Table 2: The components used to build a simple transaction request**

Transaction request field	Content
<b>Request version</b>	Version of the request – in this case it should indicate a simple transaction.
<b>Intermediary payer</b>	Routing address of the payer’s intermediary (instructing party)
<b>Public key payer</b>	The single public key of the payer (input) UTXOs
<b>Payment address payee</b>	Address of the payee UTXO
<b>Amount</b>	Amount of the payee UTXO
<b>Input token ids</b>	List of input UTXO ids (sorted by token_id)
<b>Amount input token ids</b>	List of amounts of the input token ids

Each transaction request will be given a transaction identifier which should be unique for the data that builds the request. For simple transactions, an identifier may be given as a hash of the concatenated  
The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

list of all inputs to a request (as above). Further, the transaction request should be accompanied by a timestamp and a valid signature of the data defining the request (compressed in the transaction id) with the private key corresponding to the payer's public key. The data which completes a transaction request is summarised in Table 3.

**Table 3: Additional fields completing a simple transaction request**

Transaction request field	Content
<b>Trx_id</b>	Hash of the concatenated input data for a transaction request (Table 2)
<b>Timestamp</b>	Timestamp of request
<b>Signature</b>	Signature of Trx_id with the private key corresponding to the payer's public key

Validation of transactions consists of the verification of one digital signature, with the possible additional verification of the sending party (the intermediary of the payer) for authentication.

#### 6.4 Wallets

User wallets combine UTXOs held on behalf of a single end user. The sum of the UTXO amounts in a wallet is the end user's *balance*. Wallets are not part of the settlement core but are created and maintained by (supervised) intermediaries (clients). In the division of tasks between intermediaries and the settlement core, the following is important:

- Addresses or aliases attached to wallets (like the end user's e-mail address or phone number) should be stored only by the intermediary (client) and not travel to the settlement core.
- For offering payments between wallets, from one wallet address to another, intermediaries (clients) are responsible for constructing a UTXO transaction to be sent to settlement core for a settlement attempt.
- Wallets show the user's balance.
- To create UTXO transactions from a payer's wallet to a payee, intermediaries are responsible to provide the necessary information to each other, like the payee's address in the settlement core (hash of their public key). This could be done in the form of a shared lookup service which relates payee's aliases or wallet addresses (e.g., phone number) to UTXO addresses and relevant intermediaries.

A potential data model that complies with the above points is given in Figure 10 and Figure 11.

**Figure 10: Data held by the end user of a wallet**

End user	Content
<b>UUID</b>	Unique identifier of the end user
<b>Personal data</b>	To be specified upon requirements e.g. on AML checks
<b>Type</b>	Person or Firm
<b>Commercial Id</b>	To be allowed to receive payments

**Figure 11: A potential data model of a Wallet**

Wallet	Content
<b>UTXOs</b>	Array of available UTXOs held in the wallet
<b>User ID</b>	UUID of the user of the wallet

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro

<b>Private keys</b>	Array of private keys
<b>Public keys</b>	Array of public keys

## 6.5 Technical assumptions

*The following assumptions are preliminary and may be subject to change during development*

**Amounts:** Digital euro amounts are encoded as 8-byte integers representing value in eurocents. This avoids the use of fractional arithmetic, which could be a source of error. It has been noted that this may affect the remuneration component (if implemented) since remuneration will be measured in fractions of eurocents. Yet, it will not create undue difficulties.

**Timestamps:** Timestamps are encoded as 8-byte integers representing milliseconds since 1/1/1970 (UTC time).

**Character encoding:** Characters are encoded in UTF-8 format.

**Hash algorithm:** The hash algorithm SHA-256 will be used to create condensed representations of data. This is a NIST-approved standard<sup>3</sup> and believed to be quantum resistant.

**Digital signatures:** Public and private key pairs and digital signatures will be using the same encryption standard used by Bitcoin, which is based on the **secp256k1** elliptic curve. This has been chosen because of its ubiquity in the open source community, which makes shared use easier for the multiple participants in the prototyping exercise.

---

<sup>3</sup> [FIPS 180-4 \(2015\)](#)

The design of the prototype does not pre-empt any technology decisions nor commit the Eurosystem to providing a digital euro