EUROPEAN CENTRAL BANK

EUROSYSTEM

Digital euro pilot

Frontend implementation specifications

Distributing PSP

Disclaimer: This document is indicative and may be subject to modifications. The design, features, and scope of a digital euro may also differ if issued in the future.

Table of Contents

1. Introduction

This document provides detailed implementation specifications for the different domains and interactions between actors as presented in the diagram below.

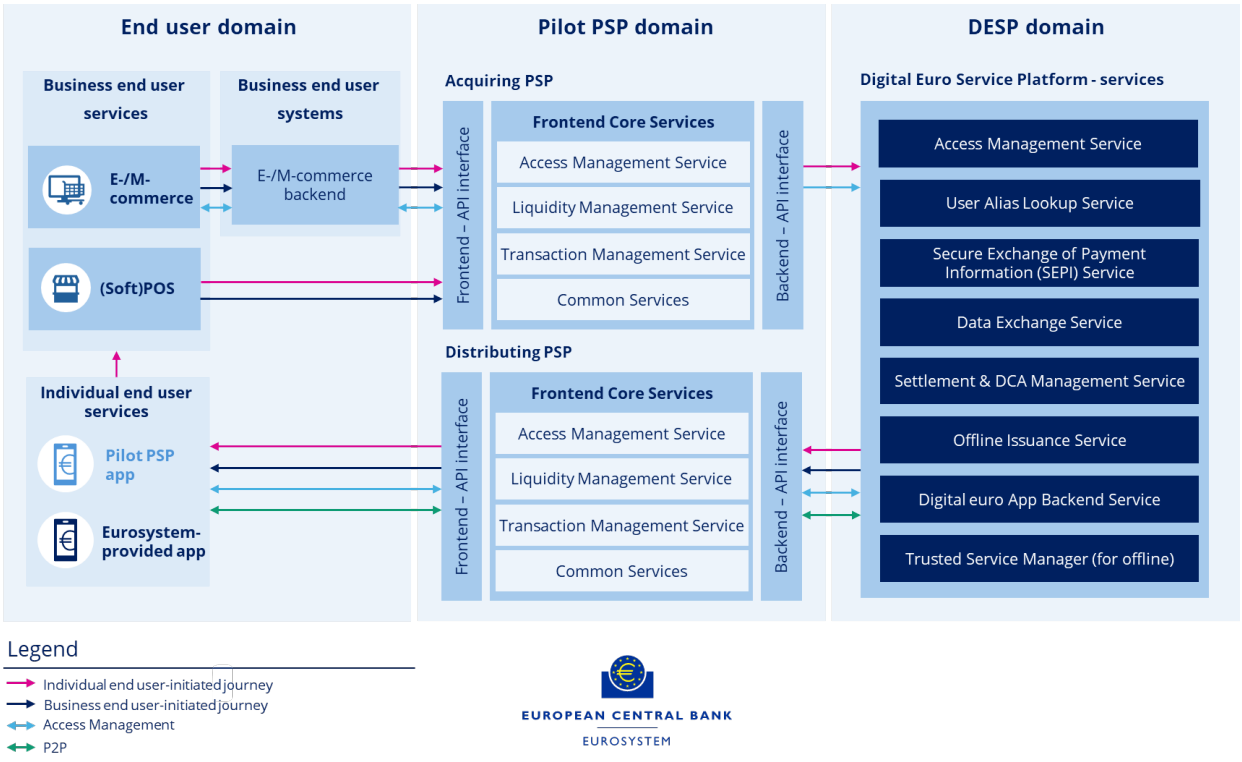

Figure 1 Digital euro pilot - functional architecture

This document is dedicated to distributing PSP requirements and focuses only on the following scope:

- Onboarding of an individual end user
- Individual end user life cycle management
- Offboarding of an individual end user
- Online manual funding from commercial bank money account – same pilot PSP
- Online manual defunding to commercial bank money account – same pilot PSP
- P2P payment with alias (payer initiated)
- P2P transaction with DEAN (payer initiated)
- P2B NFC Payment (payee initiated)

Offline transactions are out of scope of this document.

2. Structure of the document

The current document is organised according to Core Services as outlined below.

- Access Management Services
- Liquidity Management Services
- Transaction Management Services

Core Services are split into more granular services, and each service contains a set of unitary functions that must be implemented by pilot PSPs to support the pilot payment services. For each function, applicable business rules, supported message types and data elements are detailed.

Data are described according to the following information:

Element	Definition, annotation and example
Data element	Name of data element.
Description	Detailed presentation of the data that can include examples. If applicable, a set of possible values for data (attribute) is provided.
Type (Format)	Format of the data allowing the system to interact with the data and its value.
Length	Maximum number of characters.
Presence indicator	Indicator defining if a data is mandatory or not in a specific context: - M: Mandatory - O: Optional - C: Conditional
Standardised name	Name formatted according to a predefined conventions to ensure consistency and uniformity (ISO 20022). **Note**: No standard name is currently assigned to the data created for the needs of the beta digital euro (New for beta digital euro).

The list of possible types is presented below:

Data type	Description
String (STR)	Sequence of characters that can be a letter, a digit, a blank space, a punctuation mark (Alphanumeric)
Number (NUM)	Numeric characters only
Boolean (BOOL)	Represents the value True and False
String Set (SSET)	List of applicable attributes
Number Set (NSET)	List of applicable numbers
Number range	A numeric range for a data
Date	Date format YYYY-MM-DD
UTC Time	UTC time including milliseconds: HHMMSSsss
Datetime UTC	Date and time format in UTC YYYY-MM-DDThh:hh:sssZ
Binary (BIN)	Method of encoding data using sequences of bits
UUID	Universally Unique Identifier xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx

3. Overview of processes

For better context understanding, graphical representations illustrate the use cases and introduce the services and unitary functions.

The diagrams presented below:

- are conceptual views centred on flow initiators (individual end users & business end users).
- show responsibility split across end users, pilot PSPs, and DESP.
- highlight key capabilities (e.g., initiation, validation, settlement, risk & dispute handling.)

Their purpose is not to be sequence diagrams. The sequences are reflected in the end-to-end process flows.

3.1. Access Management processes

3.1.1. Individual end user onboarding

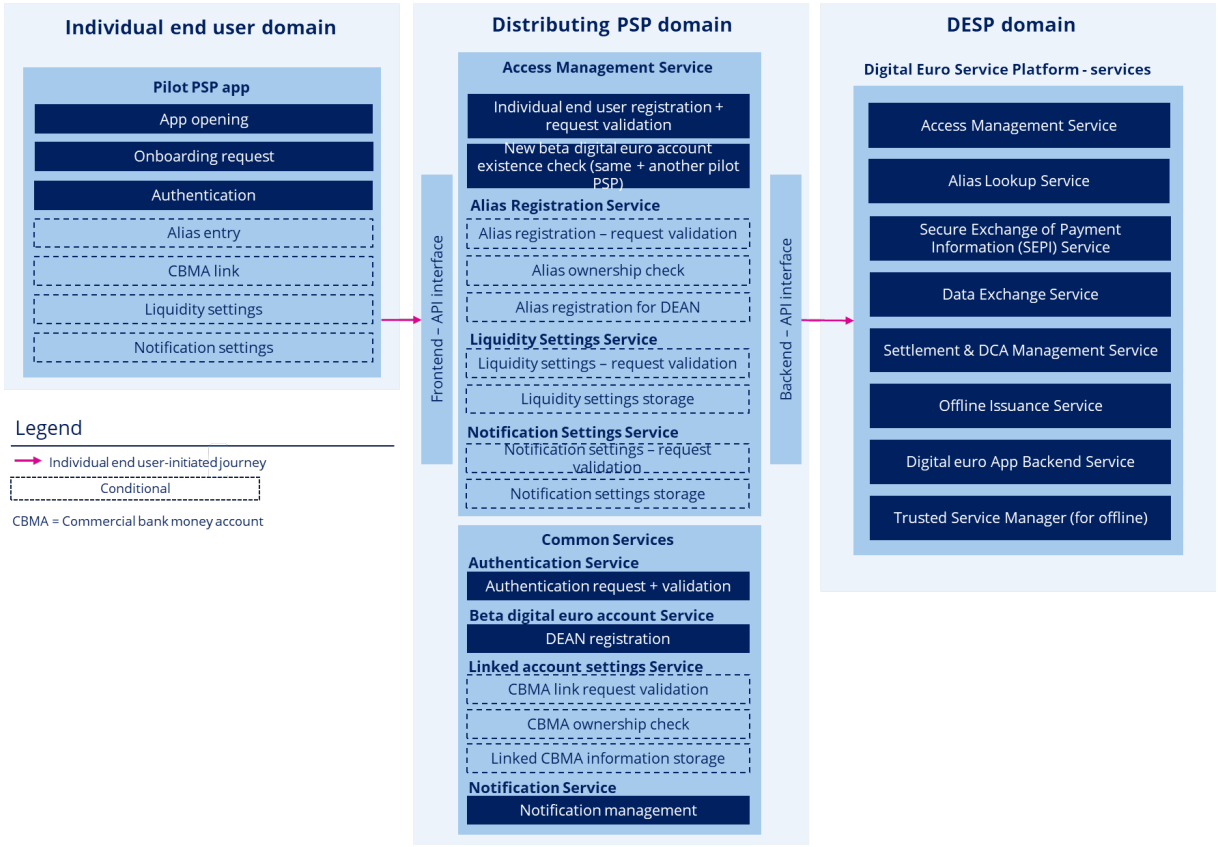

Figure 2 Individual end user onboarding

3.1.2. Individual end user lifecycle management

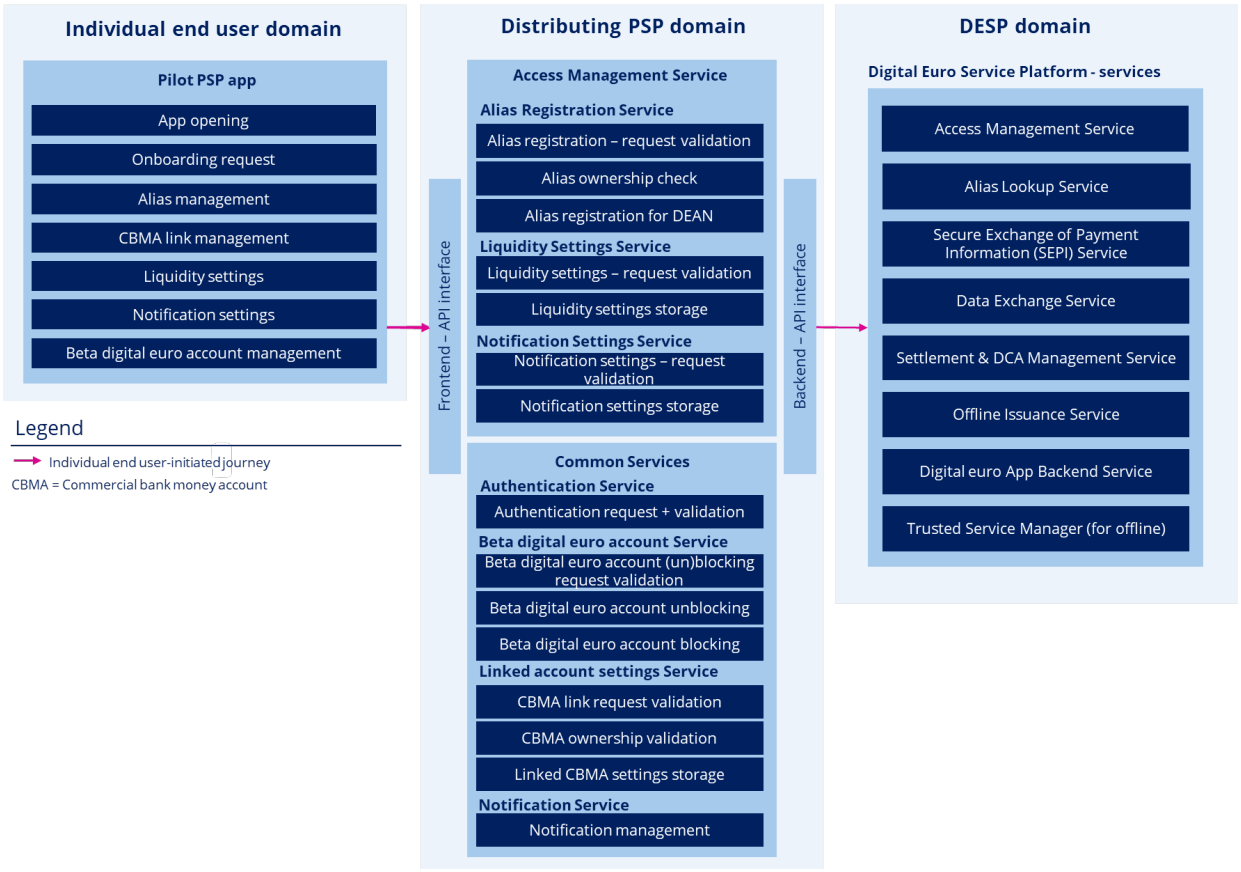

Figure 3 Individual end user lifecycle management

3.1.3. Individual end user life cycle management – NFC specificities

3.1.3.1. NFC enrolment

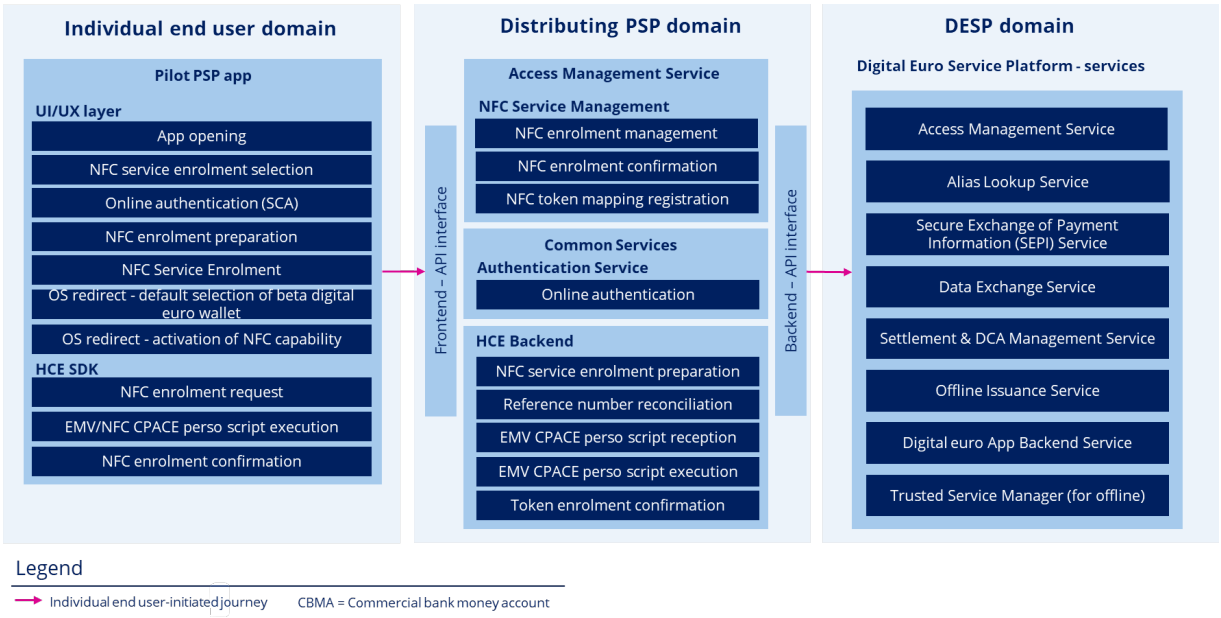

Figure 4 NFC enrolment

3.1.3.2. Key replenishment enabled by active internet connection

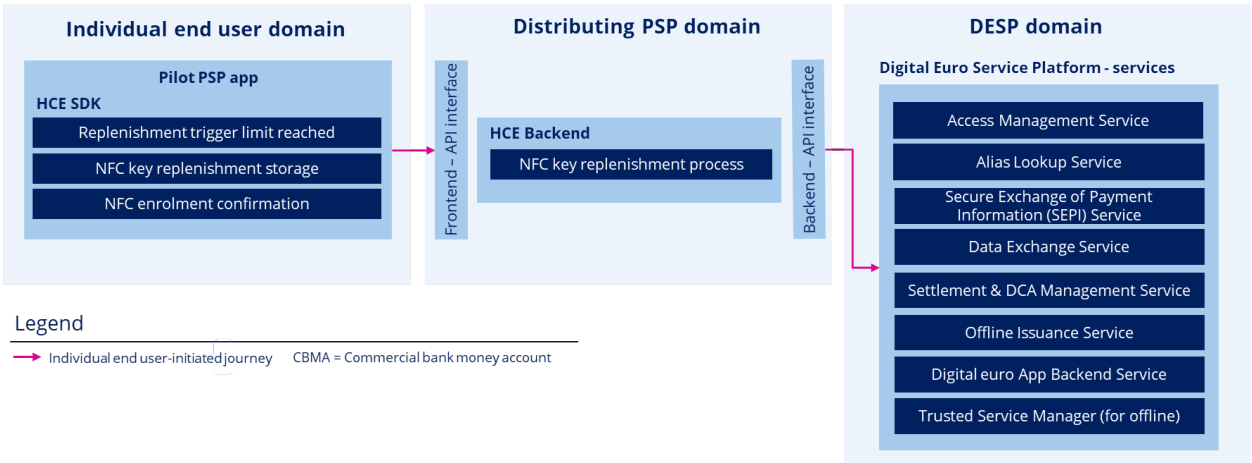

Figure 5 Key replenishment enabled by active internet connection

3.1.3.3. Key replenishment alert when internet is unavailable

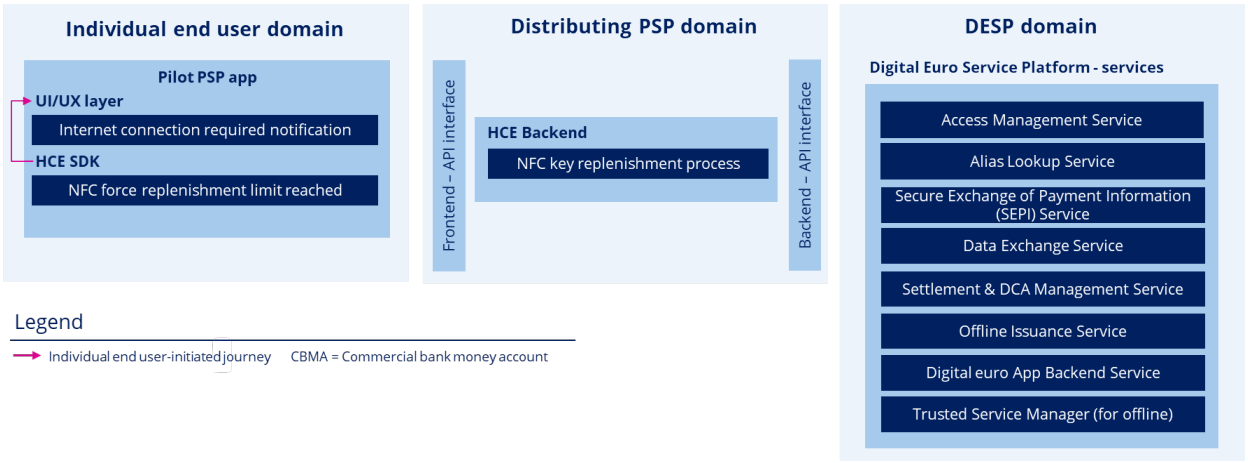

Figure 6 Key replenishment alert when internet is unavailable

3.1.3.4. NFC termination by app

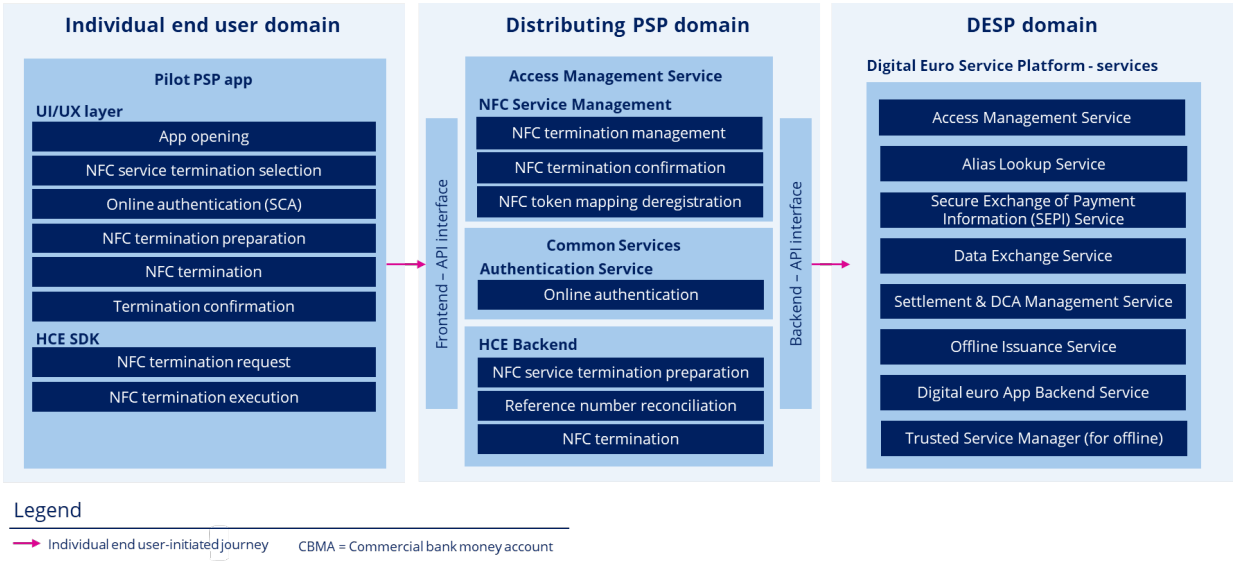

Figure 7 NFC termination by app

3.1.3.5. NFC termination by the distributing PSP

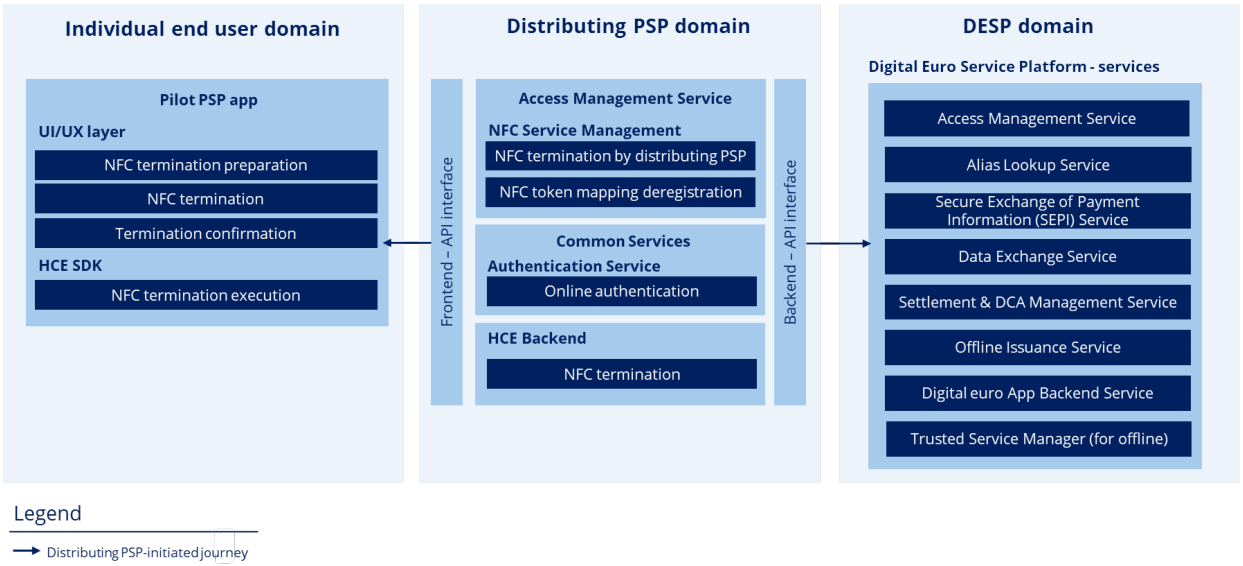

Figure 8 NFC termination by the distributing PSP

3.1.4. Individual end user offboarding

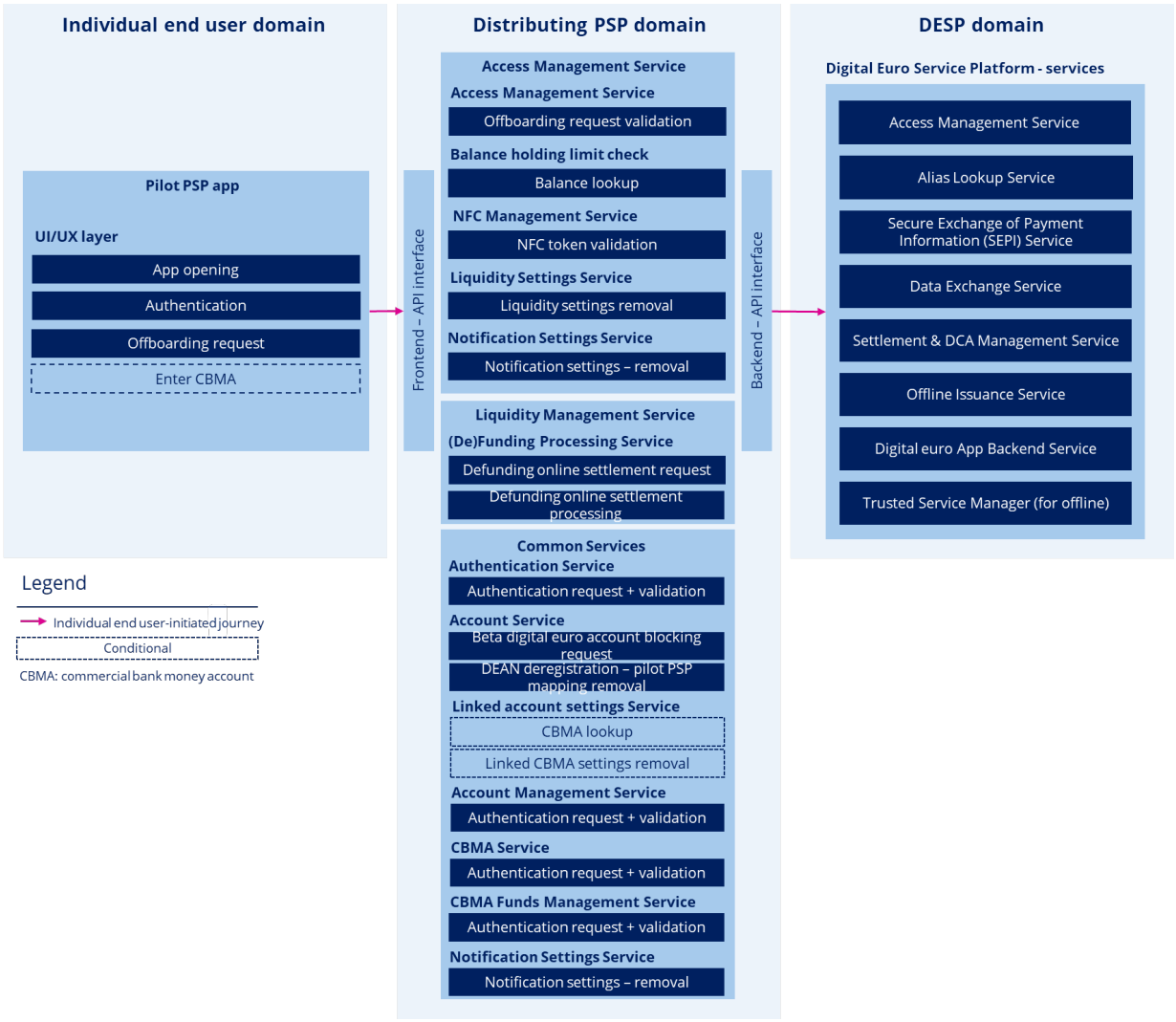

Figure 9 Individual end user offboarding

3.2. Liquidity Management processes

3.2.1. Online manual funding from commercial bank money account – same pilot PSP

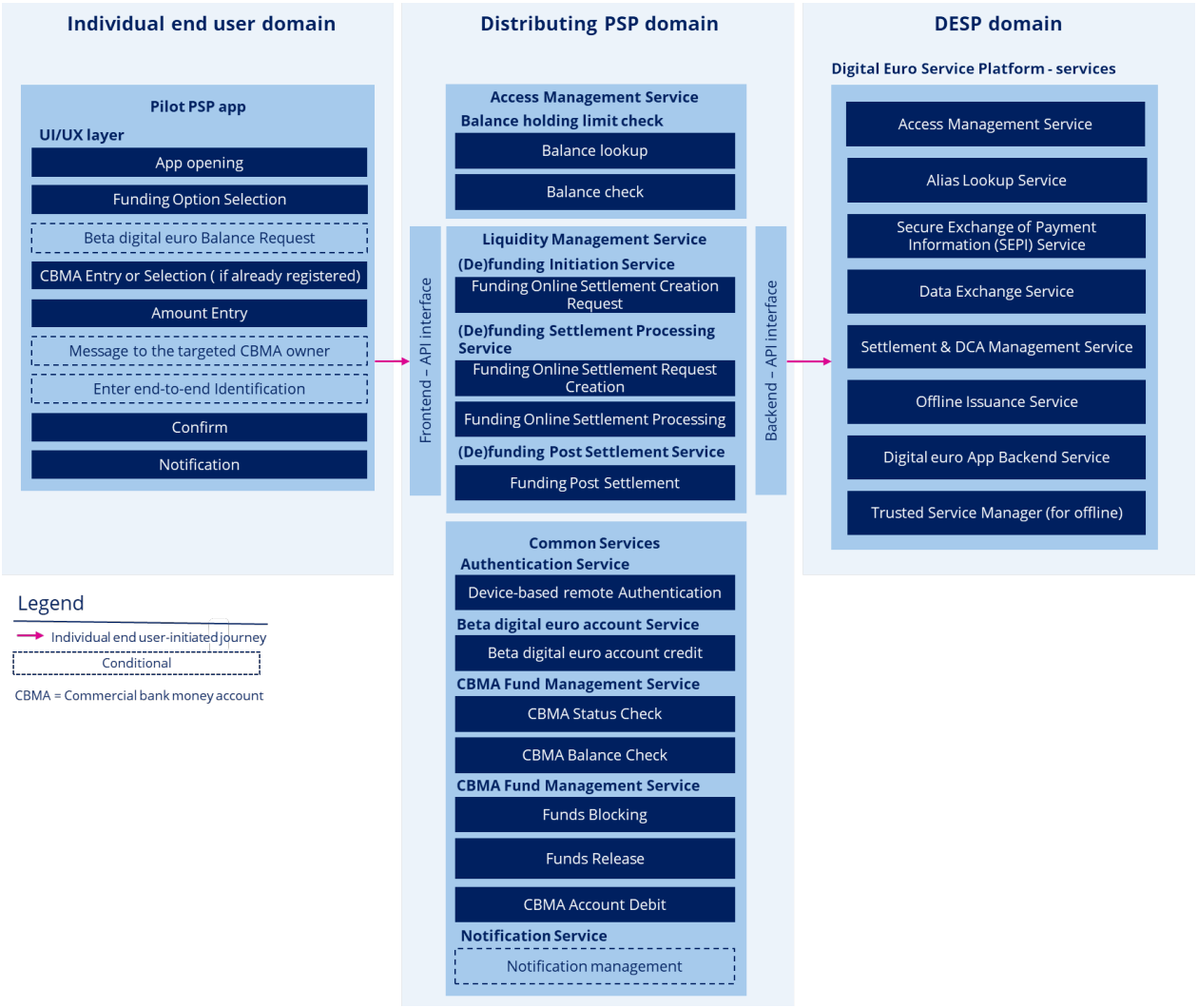

Figure 10 Online manual funding from commercial bank money account - same pilot PSP

3.2.2. Online manual defunding to commercial bank money account – same pilot PSP

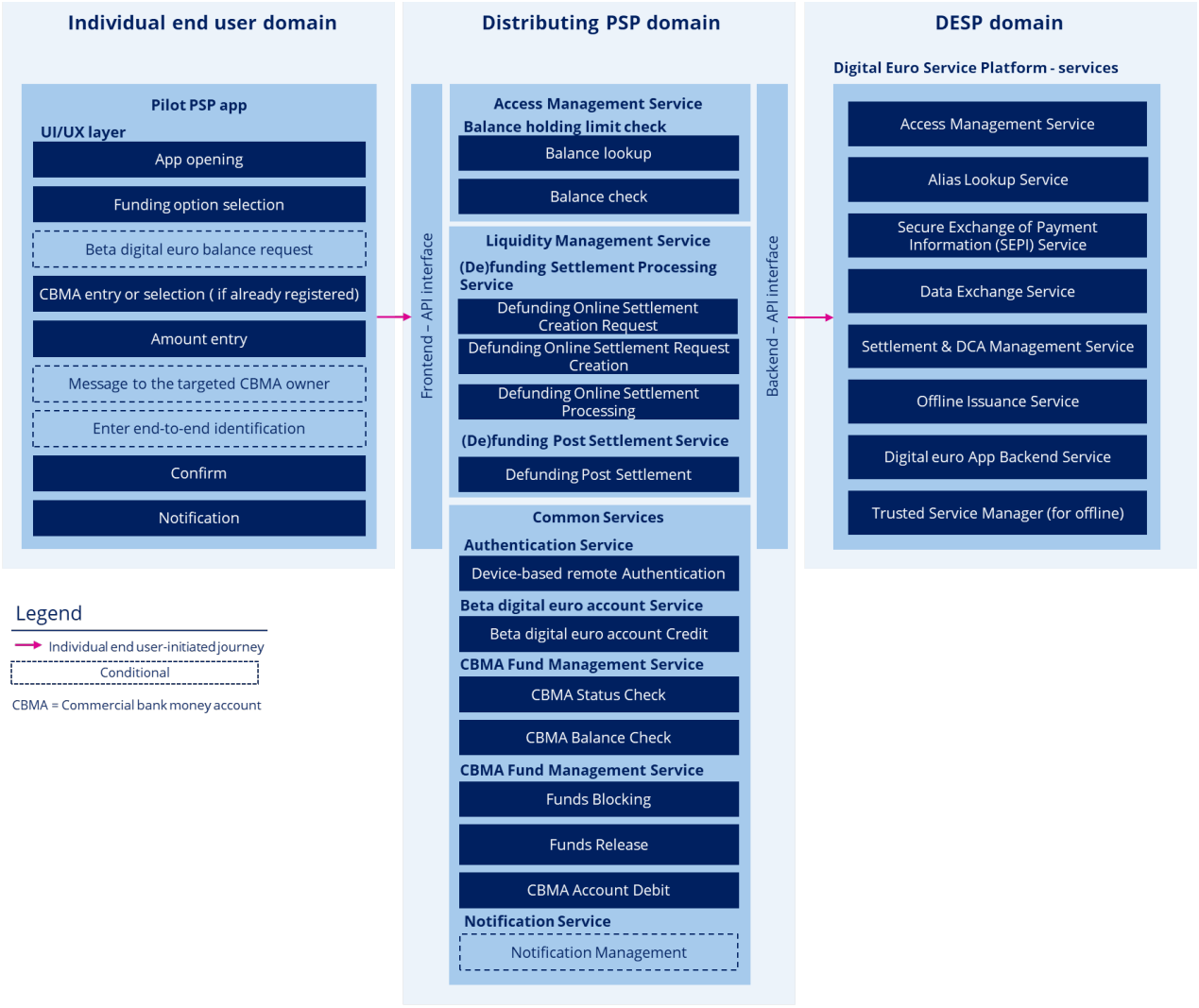

Figure 11 Online manual defunding to commercial bank money account - same pilot PSP

3.3. Transaction Management processes

3.3.1. P2P transaction with DEAN

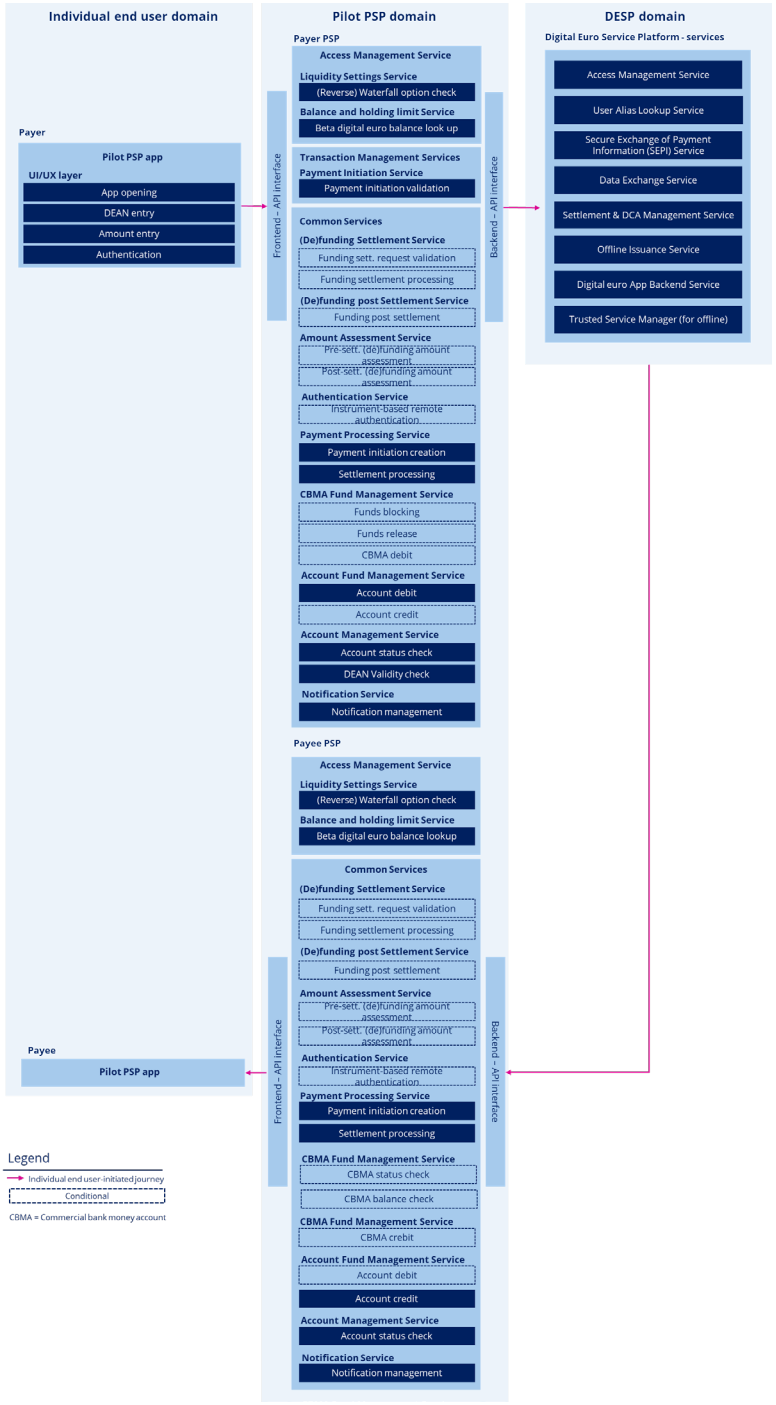

Figure 12 P2P transaction with DEAN

3.3.2. P2P transaction with alias – payer initiated

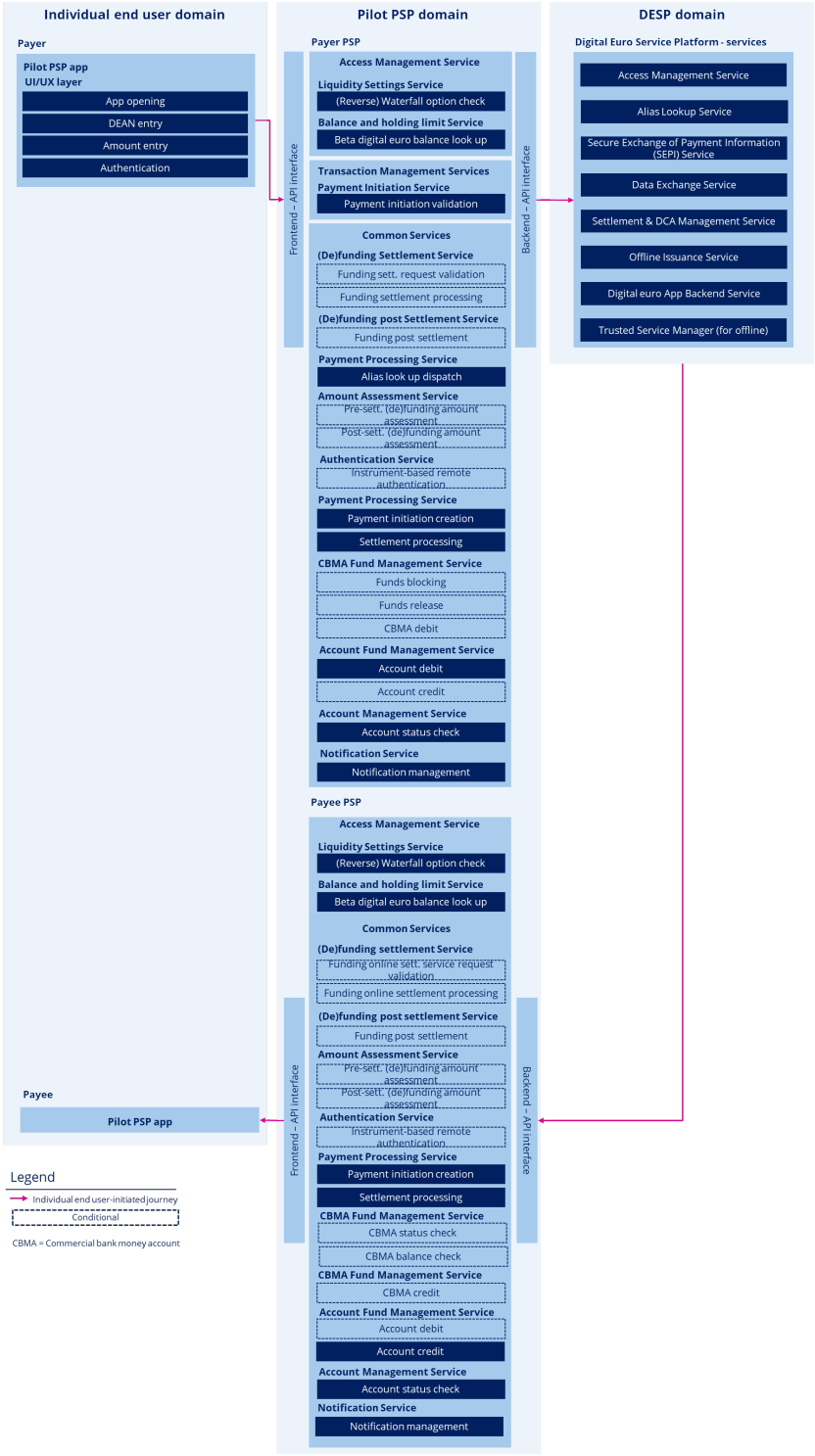

Figure 13 P2P transaction with alias - payer initiated

3.3.3. Transactions at (Soft)POS – NFC mobile payment

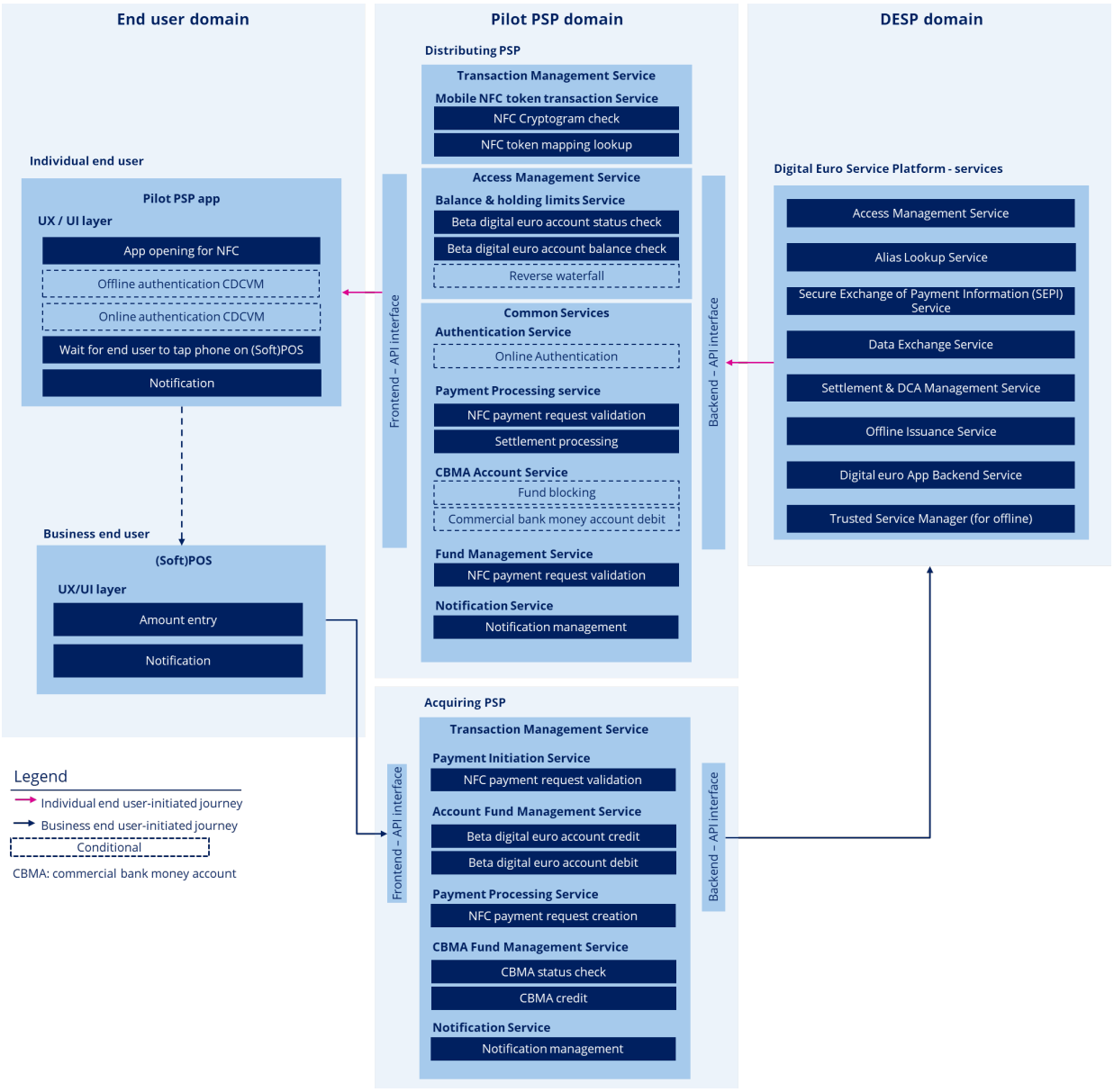

Figure 14 Transactions at (Soft)POS - NFC mobile payment

3.3.4. Balance enquiry

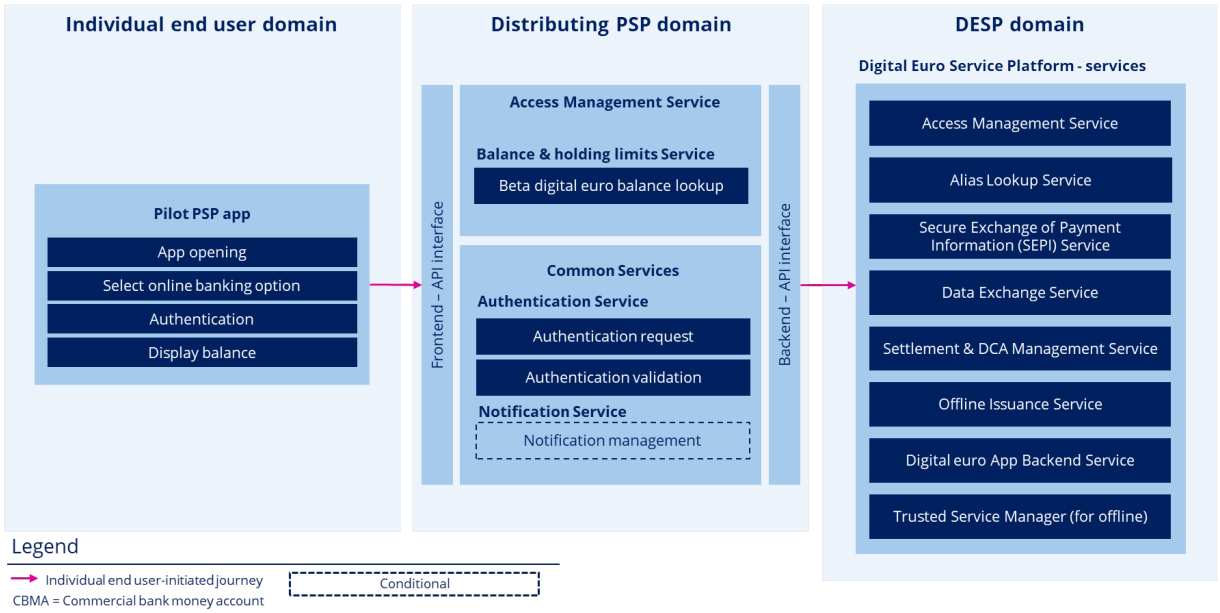

Figure 15 Balance enquiry

3.3.5. Transactions history

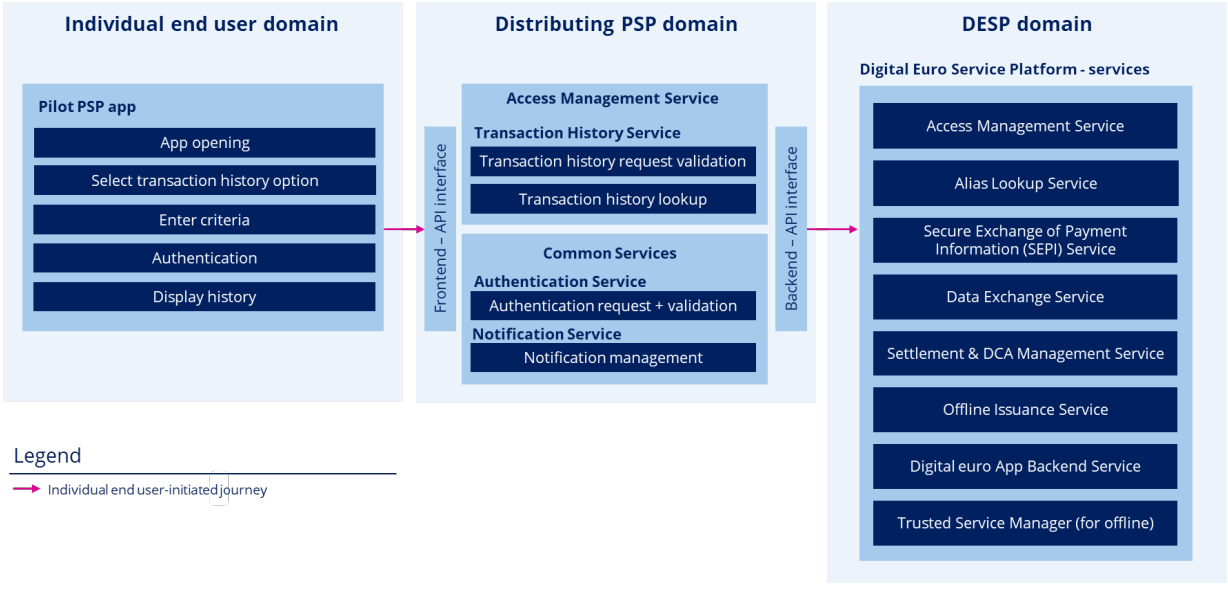

Figure 16 Transactions history

4. List of services

The graphical representations shown in the previous chapter illustrate the services and functions needed to support the use cases addressed in the current version of the specifications. These are listed and described in this section.

Service	Function	Description of function	Core Service
Alias registration service *Management of alias registration*	Alias registration request validation	The distributing PSP checks if the new alias registration is valid.	Access Management Service
	Alias ownership proof validation	The distributing PSP receives proof of alias ownership and validates it.	Access Management Service
	Alias registration for a DEAN	The distributing PSP registered the alias linked to a DEAN.	Access Management service
Authentication *Management of unique elements allowing the end user to confirm the ownership on beta digital euro holdings*	End user authentication	The pilot PSP servicing the beta digital euro account checks the ownership of the end user.	Common Service
Balance and holding limits service *Management of rules to validate that the conditions are met at beta digital euro account for a beta digital euro transaction proper execution.*	Beta digital euro account balance lookup	The pilot PSP servicing the beta digital euro account retrieves the available balance for Online holdings. It ensures there are enough funds to proceed with the transaction.	Access Management Service
	Beta digital euro account balance check	Function that verifies the current balance of a beta digital euro account before or after the settlement. This ensures the availability of funds and compliance with the holdings limit to trigger waterfall or reverse waterfall.	Access Management Service
Commercial bank money account funds management service *Management of rules to manage funds in commercial bank money*	Commercial bank money account debit	The pilot PSP servicing the commercial bank money account defunds commercial bank money account	Common Service
	Commercial bank money account credit	The pilot PSP servicing the commercial bank money account	Common Service

Service	Function	Description of function	Core Service
accounts (debit, credit, block, release)		funds commercial bank money account.	
	Funds blocking	The pilot PSP servicing the commercial bank money account blocks funds on commercial bank money account if reverse waterfall is needed for transaction proper execution.	Common Service
	Funds release	The pilot PSP servicing the commercial bank money account releases funds blocked on commercial bank money account if the transaction is discontinued.	Common Service
Commercial bank money account service *Management of the commercial bank money account involved in beta digital euro processes.*	Commercial bank money account balance lookup	The pilot PSP servicing commercial bank money account retrieves the available balance. It ensures there are enough funds to proceed with the transaction.	Common Service
	Commercial bank money account status check	The pilot PSP servicing the commercial bank money account verifies the operational status of a commercial bank money account. It ensures that the account is active, valid, and ready for transactions.	Common Service
	Commercial bank money account balance check	The pilot PSP servicing the commercial bank money account verifies the current balance of a commercial bank money account if a reverse waterfall is needed for transaction proper execution or a manual funding request is initiated.	Common Service
Beta digital euro account existence check service *Management of beta digital euro account existence check*	Beta digital euro account existence check – same pilot PSP	The distributing PSP checks if the existing customer already has a beta digital euro account.	Access Management Service
	Beta digital euro account existence check – other pilot PSP	The distributing PSP checks if the existing customer already has a beta digital euro account serviced by another pilot PSP.	Access Management Service
Beta digital euro account service	Beta digital euro account status check	The PSP checks the beta digital euro account status.	Common Service

Service	Function	Description of function	Core Service
Management of beta digital euro account settings and life cycle	Beta digital euro account (un)blocking request validation	The PSP checks if the beta digital euro account blocking or beta digital euro account unblocking request sent by the end user's device is valid.	Common Service
	Beta digital euro account blocking	The pilot PSP blocks the beta digital euro account.	Common Service
	Beta digital euro account unblocking	The PSP unblocks the beta digital euro account.	Common Service
	DEAN registration	The pilot PSP requests a DEAN to DESP and stores it.	Common Service
	DEAN validity check	The pilot PSP servicing the beta digital euro account checks if the DEAN provided is consistent.	Common Service
	DEAN deregistration and PSP mapping removal	The PSP requests to DESP the deregistration of the digital euro access number (DEAN) and the removal of its mapping to the pilot PSP.	Common Service
Beta digital euro funds management service *Management of rules to handle funds in beta digital euro account (debit, credit)*	Beta digital euro account debit	The PSP servicing the beta digital euro account debits the beta digital euro account.	Common Service
	Beta digital euro account credit	The PSP servicing the beta digital euro account credits the beta digital euro account.	Common Service
Funding/Defunding settlement processing service *Management of funding/defunding settlement request – in relation to DESP*	Funding online settlement request	The pilot PSP sends a funding online request to DESP.	Common Service
	Funding online settlement processing	The pilot PSP manages the interactions with DESP and the funding online transaction life cycle.	Common Service
	Defunding online settlement request	The pilot PSP sends a defunding online request to DESP.	Common Service
	Defunding online settlement processing	The pilot PSP manages the interactions with DESP and the defunding online transaction life cycle.	Common Service

Service	Function	Description of function	Core Service
Funding/Defunding post settlement processing service *Management of funding/defunding post settlement processing reception of the settlement notification sent by DESP*	Funding post settlement	The pilot PSP checks the funding settlement outcome received from DESP and orchestrates the next steps to close the funding process.	Common Service
	Defunding post settlement	The pilot PSP checks the defunding settlement outcome received from DESP and orchestrates the next steps to close the defunding process.	Common Service
Individual end user access management service *Management of requests initiated by an individual end user who:* - *wants to be onboarded as an end user* - *wants to close their beta digital euro account* - *wants to update their user data*	KYC registration	The distributing PSP requests KYC information and performs checks based on data provided by the individual end user.	Access Management Service
	New customer registration	The distributing PSP creates the individual end user as a new customer.	Access Management Service
	New individual end user registration request validation	The distributing PSP checks if the registration request sent by the end user is valid.	Access Management Service
	New individual end user registration	The distributing PSP registers the individual end user as a customer.	Access Management Service
	Individual end user amendment	The distributing PSP updates the individual end user data.	Access Management Service
	End user offboarding request validation	The distributing PSP checks if the offboarding request sent by the end user is valid.	Access Management Service
	End user deregistration	The distributing PSP orchestrates the final steps as part of the offboarding request.	Access Management Service
Linked account settings service *Management of the link between a commercial bank money account and the beta digital euro account*	Linked account settings request validation	The distributing PSP checks if the linked account settings request is valid.	Common Service
	Linked account settings storage	The distributing PSP stores the information related to linked commercial bank money account (new link, updated link, link removal)	Common Service

Service	Function	Description of function	Core Service
	Linked commercial bank money account ownership validation	The distributing PSP receives proof of commercial bank money account ownership and validates it.	Common Service
	Linked account settings removal	The distributing PSP removes the link between the commercial bank money account and the beta digital euro account as part of the individual end user offboarding.	Common Service
	Linked commercial bank money account lookup	The distributing PSP checks whether the end-user has linked a commercial bank money account to the beta digital euro account.	Common Service
	Linked commercial bank money account check-	During a payment transaction that requires a waterfall or reverse waterfall process, the pilot PSP checks whether the beta digital euro account is linked to a commercial bank money account and, if so, returns the IBAN of that linked account.	Common Service
Liquidity settings service *Management of parameters defined by the end-user to perform the beta digital euro account automated funding and defunding.*	Liquidity settings request validation	The distributing PSP validates if the liquidity settings request sent by the end-user's device is valid.	Access Management service
	Liquidity settings storage	The distributing PSP stores the parameters related to liquidity management.	Access Management service
	Liquidity settings removal	The distributing PSP removes the liquidity settings as part of the individual end user offboarding.	Access Management service
	Liquidity settings lookup	The distributing PSP retrieves the liquidity settings.	Access Management service
	Waterfall – Reverse Waterfall option check	The distributing PSP checks whether the end-user enables automatic: Reverse Waterfall or Waterfall options.	Access Management service

Service	Function	Description of function	Core Service
Manual (de)funding initiation service *Management of manual funding and manual defunding request*	Manual funding/defunding request validation	The pilot PSP servicing the beta digital euro account checks if the funding or defunding request in valid.	Liquidity Management Service
NFC service management *Management of rules handling NFC mobile payment enrollment and life cycle operations initiated via mobile app*	NFC enrolment management	The service offered by the distributing PSP allows its application to initiate an NFC enrollment preparation request. Once the user's eligibility is verified, the distributing PSP calls the HCE SDK backend API to request the enrollment preparation. Upon receiving a reference from the backend, the PSP returns this reference to its Pilot PSP app, enabling it to trigger the NFC enrollment process via the HCE SDK.	Access Management Service
	NFC enrolment confirmation	Once the NFC payment enrollment process is successfully completed, the HCE SDK backend sends a notification to the distributing PSP. This message confirms that the user's device is now enrolled and ready to use the NFC payment service. It also includes the NFC token generated by the SEPI TSP, allowing the PSP to update its records by associating the token with the corresponding DEAN and Pilot PSP app ID.	Access management service
	NFC Token Mapping registration	The distributing PSP performs the token mapping registration by linking the NFC token with the DEAN and the Pilot PSP app ID.	Access management service
	NFC termination by Pilot PSP app management	The service offered by the distributing PSP allows its application to initiate an NFC termination preparation request. The distributing PSP calls the HCE SDK backend API to request the termination preparation. Upon receiving a	Access management service

Service	Function	Description of function	Core Service
		reference from the backend, the PSP returns this reference to its Pilot PSP app, enabling it to trigger the NFC termination process via the HCE SDK.	
	NFC termination by Pilot PSP app confirmation	Once the NFC payment termination process is successfully completed, the HCE SDK backend sends a notification to the distributing PSP. This message confirms that the user's device is now not enrolled anymore to the NFC payment service.	Access Management Service
	NFC Termination by distributing PSP	The service provided by the distributing PSP enables the PSP to request the NFC termination service. The PSP provides the token to the HCE SDK Backend.	Access Management Service
	NFC Token Mapping deregistration	The distributing PSP performs the token mapping deregistration by removing the association between the NFC token, the DEAN, and the Pilot PSP app ID.	Access Management Service
Mobile NFC token transaction service *Upon receiving and validating the NFC payment request, the distributing PSP initiates a cryptogram check request to SEPI, then retrieves the DEAN and App ID.*	NFC cryptogram check	The distributing PSP sends an NFC cryptogram verification request to the DESP-SEPI, which responds with the outcome of the cryptogram check.	Transaction management service
	NFC token mapping lookup	The distributing PSP retrieves the DEAN and the Pilot PSP app ID thanks to the token. After the Distributing PSP can continue to validate the payment request with others common service like “balance check”, fraud check.	Transaction management service

Service	Function	Description of function	Core Service
Notification service *Management of rules to provide notifications (reflecting transaction final status) to end users through identified channel.*	End user's notification	The pilot PSP servicing the beta digital euro account provides notification to the end user confirming (or not) the proper execution of the transaction.	Common Service
Notification settings service *Management of parameters defined by the end user to receive notifications*	Notification settings request validation	The distributing PSP validates if the notification settings request sent by the end user's device is valid.	Access Management Service
	Notification settings storage	The distributing PSP stores the parameters.	Access Management Service
	Notification settings removal	The distributing PSP removes the notification settings as part of the individual end user offboarding.	Access Management service
Payment Initiation service *Management of payment requests initiated by an end user (covering all form factors)*	Alias lookup request	The pilot PSP servicing the beta digital euro account requests an alias resolution to DESP to retrieve payee's details.	Common Service
	Alias validity check	The pilot PSP checks the alias consistency and triggers the alias lookup dispatch function.	Common Service
	Pilot PSP ID lookup	The pilot PSP servicing the beta digital euro account requests the pilot PSP ID corresponding to the DEAN to DESP for routing purpose.	Common Service
	Individual end user payment instruction initiation validation	The pilot PSP servicing beta digital euro account checks the payment instruction sent by the end user's device is consistent and contains mandatory information.	Transaction Management Service
	Individual end user payment request initiation validation	The pilot PSP servicing beta digital euro account checks the payment request sent by the end user's device is consistent and contains mandatory information.	Transaction Management Service

Service	Function	Description of function	Core Service
Transactions history service *Management of rules handling transactions history requested via mobile app*	Transactions history request validation	The distributing PSP servicing the beta digital euro account checks if the transaction history request sent by the end user's device is consistent and contains mandatory information.	Transaction Management service
	Transactions history lookup	The distributing PSP servicing the beta digital euro account retrieves the transactions according to the criteria.	Transaction Management service

All the services, functions identified as Common Service are described **in Digital euro pilot – Frontend specifications - Common Services.**

5. Access Management Service

The access management core service is dedicated to functions needed for end users and account management.

5.1. Alias registration service

An end user may initiate a payment using an alias, as long as the alias is associated with their DEAN.

This service is triggered whenever an alias is created, updated, or removed.

Service	Function/sub-functions	Description
Alias registration service	Alias registration request validation	The distributing PSP checks if the alias registration request sent by the end user's device is valid. Registration request manages creation, update and removal of alias.
	Alias ownership proof validation	The distributing PSP receives proof of alias ownership and validates it.
	Alias registration for a DEAN	The distributing PSP requests a link between alias and DEAN to DESP.

5.1.1. Functions description

5.1.1.1. Alias registration request validation

5.1.1.1.1. Pre-requisite

The pilot PSP app is already connected to the distributing PSP through a Strong Customer Authentication process specific to that pilot PSP.

5.1.1.1.2. Requirements

An alias registration request is sent by the end user's device to the distributing PSP. The distributing PSP must validate the request is consistent enough to proceed properly with the registration.

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent.
2	M	The request must contain the type of request - CREATE ➔ Initial registration for an alias - UPDATE ➔ Update of an alias already registered - DELETE ➔ Removal of an alias already registered
3	M	The request must contain an alias type and an alias value

#	Mandatory Optional Conditional	Business rules description
4	M	The alias value must be consistent with the alias type
5	M	The request can be set to 'CREATE' only if the provided alias is not already registered
6	M	The request can be set to UPDATE" or "DELETE" only if the provided alias is already registered
7	M	Each end user is allowed to register only one alias, even if they use multiple apps
8	M	The function must provide a return code. In case of failure, a reason code must be provided

5.1.1.1.3. Interface description

A dedicated interface received from an end user's device (incoming message) triggers the request validation function. The function generates an outgoing message providing the result of the function execution.

5.1.1.1.3.1. Message structure

Incoming message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of incoming message	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "alias registration"	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	Datetime UTC	24	M	CreationDateTime
Type of request	Type of the request sent by the end user's device **CRED - CREATE** **UPDT - UPDATE** **DELT - DELETE**	SSET	4	M	RequestType
Type of alias	Classification defining a pseudonymous account identifier Refer to ISO external code ExternalProxyAccountType1code **EMAL** (email address) **MBNO** (Mobile phone number)	SSET	4	M	ProxyAccountType
Value of alias	Pseudonymous account identifier alias (EMAIL or MBNO) of the account identifier Note: MBNO alias shall be strict E.164 format with leading + and no other characters	STR	2048	M	ProxyAccount

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.1.1.1.3.2. Return code

#	Description
0	Successful
1	Failure

5.1.1.1.3.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service)**.

5.1.1.2. Alias ownership proof validation

5.1.1.2.1. Requirements

The individual end user submits alias details to the distributing PSP for registration. The pilot PSP then verifies that the alias is valid and belongs to the individual end user.

The current rules used by the distributing PSP to validate the ownership of a mobile phone number or an email address are reused.

#	Mandatory Optional Conditional	Business rules description
1	M	The function must provide a return code. In case of failure, a reason code must be provided.

5.1.1.2.2. Interface description

This function is triggered without any interface exchanged between the end user's device and the distributing PSP and is activated by the distributing PSP once the alias registration request is validated.

The result of the alias ownership proof validation function is sent through an outgoing message.

5.1.1.2.2.1. Message structure

Incoming message

The incoming message is the request received from the device and validated by the alias registration request validation function. Refer to **paragraph 5.1.1.1.3.1**.

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	Datetime UTC	24	M	CreationDateTime
Alias ownership indicator	Indicator validating the alias ownership • Set to 1 is the ownership is validated • Set to 0 is the ownership is not validated	BOOL	1	M	Ownership
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.1.1.2.2.2. Return code

#	Description
0	Successful
1	Failure

5.1.1.2.2.3. Functional error description (reason code)

The functional error descriptions are listed in Notification Service. Refer to **Digital euro pilot – Frontend specifications - Common Services (section 4.2 - Cross functional service / Notification).**

5.1.1.3. Alias registration for a DEAN

The DESP manages the link between an alias and a DEAN. For any action related to an alias—whether it's creation, update, or removal—the distributing PSP must submit a registration request to the DESP.

Refer to **Digital euro pilot - Backend implementation specifications ➔ "GET/ users /{userID} – Request" | "GET/ users /{userID}– Response"**

5.2. Individual end user access management service

The individual end user access management service is triggered during onboarding, offboarding, and throughout the end user life cycle. It offers a set of functions dedicated to end user management, which are executed at specific stages of various use cases.

Service	Function/sub-functions	Description
Individual end user access management service	KYC registration	The distributing PSP requests KYC information and performs checks based on data provided by the end user.
	New customer registration	The distributing PSP creates the individual end user as a new customer.
	New individual end user registration request validation	The distributing PSP checks if the registration request sent by the end user's device is valid.
	New individual end user registration	The distributing PSP registers the individual end user.
	Individual end user amendment request validation	The distributing PSP checks if the amendment request sent by the end user's device is valid.
	Individual end user amendment	The distributing PSP updates the individual end user data.

	Individual end user offboarding request validation	The distributing PSP checks if the deregistration request sent by the end user's device is valid.
	Individual end user deregistration	The distributing PSP deregisters the individual end user.

5.2.1. Functions description

5.2.1.1. KYC registration

The KYC process is performed before starting any business relationship between a customer and a pilot PSP. The process is already in place and is not modified to support the beta digital euro. It can be applied without any modification.

5.2.1.2. New customer registration

The customer who wants to open a beta digital euro account at a pilot PSP must be identified as the same pilot PSP's customer. If this is not the case, it must first be registered as a customer. The existing process is applied.

5.2.1.3. New individual end user registration request validation

5.2.1.3.1. Pre-requisite

The individual end user has been provided with a pilot PSP app either after completing a KYC process as a prospective customer, or as an existing customer of the distributing PSP.

The pilot PSP app is already connected to the distributing PSP through a Strong Customer Authentication process specific to that pilot PSP.

5.2.1.3.2. Requirements

An individual end user request is sent by the pilot PSP app to the distributing PSP to trigger the onboarding process. The distributing PSP must validate the request is consistent enough to proceed properly with the registration.

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent.
2	M	The following data must be provided: - Account type (DEUR) And the pilot PSP must directly trigger beta digital euro account creation and related steps.

#	Mandatory Optional Conditional	Business rules description
3	M	The function must provide a return code. In case of failure, a reason code must be provided.

5.2.1.3.3. Interface description

A dedicated interface received from an end user's device (incoming message) triggers the request validation function. The function generates an outgoing message providing the result of the function execution.

5.2.1.3.3.1. Message structure

Incoming message

Data element	Description	Format	Size	Presence indicator	Standardised name
Identifier of the message	Unique identifier of incoming message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "end user registration".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	Datetime UTC	24	M	CreationDateTime
Type of the new account	Type of account to be opened "DEUR" (TBD)	SSET	4	M	CashAccountType

Outgoing message

Data element	Description	Format	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	Datetime UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode

Data element	Description	Format	Length	Presence indicator	Standardised name
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.2.1.3.3.2. Return code

#	Description
0	Successful
1	Failure

5.2.1.3.3.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.2.1.4. New individual end user registration

5.2.1.4.1. Requirements

The distributing PSP creates a new individual end user and stores minimum set of data. The function is triggered when the "end user registration request validation" function is successfully executed.

#	Mandatory Optional Conditional	Business rules description
1	M	The distributing PSP must create the individual end user data envelop that is: - complemented during the onboarding process, - updated during amendment process and offboarding process
2	M	The end user data envelop must contain the following data - Individual end user Identifier - Technical proof - Entry Identifier - Individual end user **type** - Individual end user **creation date** - Individual end user **status** - Individual end user **country** - Individual end user **language**
3	M	The distributing PSP must link the end user identifier with the pilot PSP app ID associated with the app that initiated the request.
4	M	The end user identifier must be generated according to the following rules: (placeholder)
5	M	The technical proof must be generated according to the following rules: (placeholder)
6	M	The entry identifier must be generated according to the following rules: (placeholder)
7	M	The end user creation date must be equal to the current date.

#	Mandatory Optional Conditional	Business rules description
8	M	The end user status must be initiated according to the end user life cycle. - PENDING: the customer is registered but has not yet opened a beta digital euro account
9	M	The end user type must be “Individual” (INDI).
10	M	The function must provide a return code. In case of failure, a reason code must be provided.

5.2.1.4.2. Interface description

This function is triggered without any interface exchanged between the end user's device and the distributing PSP.

The result of the new individual end user registration function is sent through an outgoing message.

5.2.1.4.2.1. Message structure

Incoming message

The incoming message is the request received from the device and validated by the new individual end user request validation function. Refer to **paragraph 5.2.1.3.3.1.**

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message	STR	35	O	OriginalMessageIdentification
Date of the message	YYYY-MM-DDThh:hh:sssZ	Datetime UTC	24	M	CreationDateTime
Identifier of the end user	Unique identifier of the end user Populated if the process is successful	STR	TBD	O	***(New for beta digital euro)***
Technical proof	Cryptographic root key used to prove the ownership of holdings. Populated if the process is successful.	STR	TBD	O	***(New for beta digital euro)***
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode

Data element	Description	Type	Length	Presence indicator	Standardised name
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.2.1.4.2.2. Return code

#	Description
0	Successful
1	Failure

5.2.1.4.2.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.2.1.5. Individual end user amendment

The individual end user data that can be modified is the same as those currently editable. The already existing functions are reused without any modification.

5.2.1.6. Individual end user offboarding request validation

5.2.1.6.1. Requirements

The individual end user intends to close their beta digital euro account and has submitted a request to the distributing PSP servicing the beta digital euro account through their device.

The distributing PSP must validate the request is consistent and valid and that the pre-requisites are fulfilled.

#	Mandatory Optional Conditional	Description
1	M	The request format and content must be consistent.
2	M	The request must contain the User ID or the DEAN.
3	C	If the request contains the DEAN, the distributing PSP must find the corresponding end user identifier.
4	C	If the offboarding request is valid, the status of the end user must be updated to reflect the ongoing offboarding
5	M	The function must provide a return code. In case of failure, a reason code must be provided.

5.2.1.6.1. Interface description

A dedicated interface received from an end user's device (incoming message) triggers the request validation function. The function generates an outgoing message providing the result of the function execution.

5.2.1.6.1.1. Message structure

Incoming message

Data element	Description	Type	Length	Presence Indicator	Standardised name
Identifier of the message	Unique identifier of incoming message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "Individual end user offboarding".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation. YYYY-MM-DDThh:hh:sssZ	Datetime UTC	24	M	CreationDateTime
DEAN	Digital euro access number of the individual end user.	STR	18	O	***(New for beta digital euro)***
Identifier of the end user	Unique identifier of the individual end user	STR	TBD	O	***(New for beta digital euro)***

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation. YYYY-MM-DDThh:hh:sssZ	Datetime UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode

Data element	Description	Type	Length	Presence indicator	Standardised name
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.2.1.6.1.2. Return code

#	Description
0	Successful
1	Failure

5.2.1.6.1.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.2.1.7. Individual end user deregistration

5.2.1.7.1. Requirements

As part of individual end user offboarding process, the final steps must be orchestrated by the distributing PSP servicing the beta digital euro account. The individual end user deregistration function is activated once the DESP has confirmed the DEAN registration and pilot PSP mapping removal.

#	Mandatory Optional Conditional	Description
1	M	- The distributing PSP must check if a commercial bank money account is still linked to the beta digital euro account. Refer to **Digital euro pilot - Frontend specifications - Common Services (section 1.5 - Linked account settings service)**
2	C	- If a commercial bank money account is still linked to a beta digital euro account, the distributing PSP must remove the existing link. Refer to **Digital euro pilot - Frontend specifications - Common Services (section 1.5 - Linked account settings service)**
3	M	- The distributing PSP must remove the liquidity settings still defined for the individual end user. Refer to **Digital euro pilot – Frontend specifications - Distributing PSP (section 5.8 – Liquidity settings service)**
4	M	- The distributing PSP must remove the notification settings still defined for the individual end user. Refer to **Digital euro pilot - Frontend specifications - Distributing PSP (section 5.9– Notification settings service)**
5	M	- The distributing PSP must request the individual end user deactivation. Refer to **Digital euro pilot - Frontend specifications - Common Services (section 1.7 – End user deactivation)**
6	C	If the deactivation is successful, the following data must be updated: - End user status is updated and switched to “CLOSED” - End user closing date is populated with the current date

#	Mandatory Optional Conditional	Description
7	M	- The function must provide the return code. In case of failure, a reason code must be provided.

5.2.1.7.2. Interface description

The function is activated by the confirmation message sent by DESP after DEAN deregistration and pilot PSP mapping successful removal.

5.2.1.7.2.1. Message structure

Incoming message

Refer to **Digital euro pilot - Backend implementation specifications ➔ "PUT/ deans – Response".**

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	Datetime UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.2.1.7.2.2. Return code

#	Description
0	Successful
1	Failure

5.2.1.7.2.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.3. Beta digital euro account existence check service

5.3.1. Functions descriptions

5.3.1.1. Beta digital euro account existence check – same pilot PSP

Each individual end user is allowed to open only one beta digital euro account. The distributing PSP is responsible for ensuring that this condition is met before submitting a request to get a DEAN.

5.3.1.1.1. Requirements

The distributing PSP must first verify in its own system that no existing beta digital euro account is already associated with the individual end user.

#	Mandatory Optional Conditional	Business rules description
1	M	The internal checks must be performed only for an already known customer who wants to be onboarded as an end user.
2	M	A beta digital euro account shall be deemed to exist only if its status is other than “Closed”
3	C	If a “valid” beta digital euro account is associated with the individual end user, the onboarding process must be discontinued.
4	M	The function must provide a return code. In case of failure, a reason code must be provided.

5.3.1.1.2. Interface description

This function is triggered without any interface exchanged between the individual end user's device and the distributing PSP. The distributing PSP activates the function as part of the onboarding process.

The result of the beta digital euro account existence check function is sent through an outgoing message.

5.3.1.1.2.1. Message structure

Incoming message

The distributing PSP will generate the incoming message that triggers the function based on its own data to check for the existence of the beta digital euro account.

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Indicator of already existing DEAN	Indicator showing whether an already existing account is found: - Set to 0 if no existing account is found - Set to 1 if an existing account is found	BOOL	1	M	***(New for beta digital euro)***
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.3.1.1.2.2. Return code

#	Description
0	Successful
1	Failure

5.3.1.1.2.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.3.1.2. Beta digital euro account existence check – other pilot PSP

5.3.1.2.1. Requirements

After verifying in its own system that no account exists, the distributing PSP must also check if the individual end user is already registered in DESP.

Refer to **Back-end implementation specifications ➔ "GET/ users/{userID} – Request" | "GET/ users/{userID} – Response".**

If the individual end user is already registered in DESP, the individual end user must be requested to perform an account switching.

5.4. Balance and holding limits service

The balance and holding limits service is invoked during payment transactions and funding/defunding transactions processing. It offers the needed functions to validate that the conditions are met for a beta digital euro transaction proper execution. This service is also used in the balance enquiry use case.

Service	Function/sub-functions	Description
Balance and holding limits service	Beta digital euro account balance check	The distributing PSP servicing the beta digital euro account activates the function during the process when a beta digital euro account balance check is required to validate that the account balance has sufficient funds to proceed with the payment or if the holdings limit is not exceeded.
	Beta digital euro account balance lookup	This distributing PSP servicing the beta digital euro account activates this function to retrieve the available balance and the upcoming balance

5.4.1. Functions description

5.4.1.1. Beta digital euro account balance check

5.4.1.1.1. Requirement

The distributing PSP servicing the beta digital euro account activates the function during the process when a beta digital euro account balance check is required to validate that the account balance has sufficient funds to proceed with the payment or if the holdings limit is not exceeded.

#	Mandatory Optional Conditional	Business rules description
1	M	**Real-Time Balance Verification:** The balance check must be conducted in real time to ensure accuracy and avoid transaction failures due to insufficient funds.
2	M	**Minimum Balance Requirement**: Verify if the account balance meets any specified minimum threshold required for initiating a transaction.
3	M	**Balance Availability Check**: Verify if the account balance is available and not held or reserved for other pending transactions. Balance availability check must consider potential concurrent transactions.
4	M	The function must retrieve the holding limits set for the corresponding beta digital euro account and check if the transaction does not result in exceeding the holding limits. Refer to **Digital euro pilot – Frontend specifications - Distributing PSP (section 5.8 - Liquidity settings service).**
5	C	If the beta digital euro account balance check function is triggered during the online manual funding process, the distributing PSP must check if other incoming digital transactions are still being processed. If incoming transactions still being processed are found, the manual funding process must be discontinued.
6	C	If the balance check reveals insufficient funds, the transaction is rejected.
7	C	If the balance check reveals holding limits are exceeded, transaction is rejected.
8	M	The function must provide the return code. In case of failure, a reason for the rejection must be provided.

5.4.1.1.2. Interface description

Beta digital euro account balance check is triggered without any interface exchanged between the individual end user's device and the distributing PSP.

The result of the beta digital euro account balance check function is sent through an outgoing message.

5.4.1.1.2.1. Message structure

Incoming message

The incoming message is generated by the distributing PSP and contains the following data:

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "Balance check".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation. YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Digital euro access number	Identification of the beta digital euro account.	STR	18	M	***(New for beta digital euro)***

Data element	Description	Type	Length	Presence indicator	Standardised name
Transaction amount	Amount transaction.	NUM	18	M	Amount

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.4.1.1.2.2. Return code

#	Description
0	Successful
1	Failure

5.4.1.1.2.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.4.1.2. Beta digital euro account balance lookup

This function can be invoked either by the pilot PSP app through the payment or funding-defunding menus to display the available balance (beta digital euro account balance), the reserved balance (upcoming beta digital euro account balance), and the holding limits, or as part of the transaction processing.

5.4.1.2.1. Pre-requisite

If the balance lookup is requested from the pilot PSP app, the pilot PSP app is already connected to the distributing PSP through a Strong Customer Authentication process specific to that pilot PSP.

5.4.1.2.2. Requirement

The pilot PSP servicing the beta digital euro account must check that the beta digital euro balance lookup request is valid and contains the expected data and retrieves the requested balance.

#	Mandatory Optional Conditional	Description
1	M	The request format and content must be consistent.
2	C	If the request is valid, the pilot PSP must retrieve the available balance for the online holdings.
3	M	The available balance must represent the total amount of money in the beta digital euro account.
4	O	The pilot PSP should be able to provide the amount of money representing the upcoming balance. The upcoming balance must be in accordance with the calculation below: ***Available balance – reserved funds – upcoming standing orders – upcoming recurring payments*** • *Reserved funds represent funds temporarily unavailable (e.g. hotel or car rental pre-authorization)*
5	M	The function must provide the holding limits set for the corresponding beta digital euro account. Refer to **Digital euro pilot – Frontend specifications - Distributing PSP (section 5.8 – Liquidity settings service).**
6	M	The function must provide a return code.
7	C	If the available balance cannot be retrieved, the execution must be considered as “failed” a reason code must be provided.
8	C	If the function execution is requested by the pilot PSP app: If the available balance can be retrieved, and the upcoming balance and/or the holding limits cannot be provided, the execution must be considered as “successful”. A dedicated reason code must be provided to inform the user that only the available balance is provided.

5.4.1.2.1. Interface description

The pilot PSP servicing the beta digital euro account retrieves the available balance for online holdings. It ensures there are enough funds to proceed with the transaction.

5.4.1.2.1.1. Message structure

Incoming message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: “beta digital euro account balance”	STR	35	M	EventIdentification

Data element	Description	Type	Length	Presence indicator	Standardised name
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Digital euro access number of the individual end user.	STR	18	M	***(New for beta digital euro)***
Beta digital euro account balance type	Specifies the nature of the balance. AVLB = Available	SSTR	4	M	BalanceType
Beta digital euro account upcoming balance indicator	Indicates if the upcoming balance must be retrieved.	BOOL	1	O	Indicator

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason
Beta digital euro account balance type	Corresponds to the data provided in the incoming message.	SSTR	4	M	BalanceType
Beta digital euro account upcoming balance indicator	Corresponds to the data provided in the incoming message.	BOOL	1	O	Indicator
Beta digital euro account balance date	Date of the balance Populated if the function execution is successful.	DATE	8	O	Date
Beta digital euro account balance	Available balance Populated if the function execution is successful.	NUM	18	O	Balance
Beta digital euro account upcoming balance	Populated if beta digital euro account upcoming balance indicator is equal to “1” and the function execution is successful.	NUM	18	O	Balance

Data element	Description	Type	Length	Presence indicator	Standardised name
Holding limits	Populated if the function execution is successful. Maximum amount of beta digital euro that can be held by an individual end user.	NUM	18	O	***(New for beta digital euro)***

5.4.1.2.1.2. Return code

#	Description
0	Successful
1	Failure

5.4.1.2.1.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.5. Beta digital euro account service

The beta digital euro account service is triggered during the onboarding of an end user, throughout the beta digital euro account life cycle management. It also offers some functions to validate that the conditions are met for a beta digital euro transaction proper execution.

Service	Function/sub-functions	Description
Beta digital euro account service	Beta digital euro account status check	The pilot PSP checks the beta digital euro account status.
	Beta digital euro account (un)blocking request validation	The pilot PSP checks if the beta digital euro account blocking or beta digital euro account unblocking request sent by the end user's device is valid.
	Beta digital euro account blocking	The pilot PSP blocks the beta digital euro account.
	Beta digital euro account unblocking	The pilot PSP unblocks the beta digital euro account.
	DEAN registration	The pilot PSP requests a DEAN to DESP and stores it.
	DEAN Validity check	The pilot PSP servicing the beta digital euro account checks if the DEAN provided is consistent.
	DEAN registration and pilot PSP mapping removal	The pilot PSP requests to DESP the deregistration of the DEAN and the removal of its mapping to the pilot PSP.

Those functions are used by several processes performed by both distributing and acquiring PSPs and are described as a Common function.

Refer to **Digital euro pilot – Frontend specifications - Common Services**.

5.6. Commercial bank money account service

Commercial bank money account service is invoked during a payment transaction process by the pilot PSP servicing the commercial bank money account of the end user. It offers the needed functions to validate that the conditions are met for a beta digital euro transaction proper execution.

Service	Function	Description
Commercial bank money account service	Commercial bank money account balance lookup	The pilot PSP servicing commercial bank money account retrieves the available balance. It ensures there are enough funds to proceed with the transaction.
	Commercial bank money account status check	The pilot PSP servicing commercial bank money account verifies the operational status of a commercial bank account.
	Commercial bank money account balance check	The pilot PSP servicing commercial bank money account verifies the current balance of a commercial bank money account if either a reverse waterfall is needed for transaction execution, or a funding amount is needed for a liquidity management service.

Those functions are used by several processes performed by both distributing and acquiring PSPs and are described as a Common function.

Refer to **Digital euro pilot – Frontend specifications - Common Services**.

5.7. Linked account settings service

An end user who has successfully opened a beta digital euro account must have the ability to link a commercial bank money account to their beta digital euro account for funding and defunding purposes. The link between both accounts is a pre-requisite to waterfall and reverse waterfall processes. This service is activated to define, update or remove the link.

Service	Function/sub-functions	Description
Linked account settings service	Linked account settings request validation	The pilot PSP checks if the linked account settings request is valid.

Service	Function/sub-functions	Description
	Linked account settings storage	The pilot PSP stores the information related to linked commercial bank money account (new link, updated link, link removal) upon end user request.
	Linked account settings removal	The pilot PSP clears the linked account when the individual end user is offboarded.
	Linked commercial bank money account ownership validation	The pilot PSP servicing the commercial bank money account confirms the ownership of linked account.
	Linked commercial bank money account lookup	The pilot PSP checks whether the end user has linked a commercial bank money account to the beta digital euro account.
	Linked commercial bank money account check	During a payment transaction that requires a waterfall or reversewaterfall process, the pilot PSP checks whether the beta digital euro account is linked to a commercial bank money account and, if so, returns the IBAN of that linked account.

Those functions are used by several processes performed by both distributing and acquiring PSPs and are described as a Common function.

Refer to **Digital euro pilot – Frontend specifications - Common Services**.

5.8. Liquidity settings service

An individual end user who has successfully opened a beta digital euro account must have the ability to configure automated funding and defunding operations. This service is triggered to define, update or remove the configuration parameters.

Assumptions: Settings are managed at the level of the individual end user. The functions and the business rules are defined accordingly.

Service	Function/sub-functions	Description
Liquidity settings service	Liquidity settings request validation	The distributing PSP checks if the liquidity settings request sent by the individual end user's device is valid.
	Liquidity settings storage	The distributing PSP stores the new or updated parameters and confirms the settings.
	Liquidity settings removal	The distributing PSP clears all the parameters when the individual end user is offboarded
	Liquidity setting lookup	The distributing PSP retrieves the existing liquidity settings.

Service	Function/sub-functions	Description
	Waterfall/Reverse Waterfall option check	This function checks whether the beta digital euro account holder has activated the required Waterfall or Reverse Waterfall automatic transfer option during a combined payment transaction; if the relevant option is not enabled, the pilot PSP internally rejects the transaction and returns a status with a return code and, if needed, a reason code.

5.8.1. Functions description

5.8.1.1. Liquidity settings request validation

5.8.1.1.1. Pre-requisite

The individual end user has been provided with a pilot PSP app either after completing a KYC process as a prospective customer, or as an existing customer of the distributing PSP.

5.8.1.1.2. Requirements

The individual end user must be able to request dedicated liquidity settings through their own pilot PSP app. The corresponding request is received by the distributing PSP servicing the beta digital euro account, which must verify that the request is valid and consistent.

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent.
2	M	The request must contain the User ID or the DEAN.
3	M	The request must contain the type of request - CREATE ➔ Initial set-up - UPDATE ➔ Set-up amendment - DELETE ➔ Set-up removal
4	M	The request can be set to 'CREATE' only if no existing parameters are already defined.
5	M	The request can be set to UPDATE" or "DELETE" if a configuration already exists for the beta digital euro account.
6	M	The function must provide a return code. In case of failure, a reason code must be provided.

5.8.1.1.3. Interface description

A dedicated interface received from a user's device (incoming message) triggers the liquidity settings request validation. The function generates an outgoing message providing the result of the function execution.

5.8.1.1.3.1. Message structure

Incoming message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "Liquidity settings request"	STR	35	M	EventIdentification
Date of the message	Date time of the message creation. YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Type of request	Type of request sent by the end user' device. - **CRED - CREATE** - **UPDT - UPDATE** - **DELT - DELETE**	SSET	4	M	RequestType
Waterfall Indicator	Dedicated indicator defining if the user opts for waterfall.	BOOL	1	O	***(New for beta digital euro)***
Reverse waterfall indicator	Dedicated indicator defining if the user opts for reverse waterfall.	BOOL	1	O	***(New for beta digital euro)***

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.8.1.1.3.2. Return code

#	Description
0	Successful
1	Failure

5.8.1.1.3.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.8.1.2. Liquidity settings storage

5.8.1.2.1. Requirements

Once the distributing PSP validates the liquidity settings request initiated by the individual end user through device, the new or updated parameters are stored.

#	Mandatory Optional Conditional	Business rules description
1	M	The distributing PSP must store the liquidity parameters provided by the individual end user. These liquidity parameters include: • Waterfall indicator (0 = activated / 1 = not activated) • Reverse waterfall indicator (0 = activated / 1 = not activated)
2	M	Only new or updated information must be stored.
3	M	An amendment date must be stored. It must be in accordance with the current date.
4	M	The function must provide a return code to confirm or reject the liquidity set-up.

5.8.1.2.2. Interface description

This function is not triggered by an interface exchanged between the individual end user device and the distributing PSP. It is activated upon the proper execution of the liquidity settings request validation function. The result of the storage function is sent through an outgoing message.

5.8.1.2.2.1. Message structure

Incoming message

The incoming message is the request received from the device and validated by the liquidity settings request validation function. Refer to **section 5.8.1.1**.

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.8.1.2.2.2. Return code

#	Description
0	Successful
1	Failure

5.8.1.2.2.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.8.1.3. Liquidity settings removal

5.8.1.3.1. Requirements

When an individual end user initiates an offboarding request, any previous liquidity settings must be removed. This dedicated step of the offboarding process is executed under the coordination of the distributing PSP.

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent.
2	M	The request must contain the DEAN or the end user identification.
3	C	If the request contains the DEAN, the distributing PSP must retrieve the end user identifier.

#	Mandatory Optional Conditional	Business rules description
4	M	The distributing PSP must remove all the liquidity settings defined by the end user.
5	O	An amendment date could be stored. It must be in accordance with the current date.
6	M	The function must provide a return code. In case of failure, a reason code must be provided.

5.8.1.3.1.1. Message structure

Incoming message

The incoming message is generated by the distributing PSP and contains the following data:

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of the message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "Liquidity settings removal"	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Identification of the beta digital euro account.	STR	18	O	***(New for beta digital euro)***
Identifier of the end user	Unique identifier of the end user.	STR	TBD	O	***(New for beta digital euro)***

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode

Data element	Description	Type	Length	Presence indicator	Standardised name
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.8.1.3.1.2. Return code

#	Description
0	Successful
1	Failure

5.8.1.3.1.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.8.1.4. Liquidity settings lookup

This function can be called either to present the current settings to the end user or during transaction processing to ensure that the conditions required for proper execution are met.

5.8.1.4.1. Prerequisites

If the liquidity settings lookup is requested from the pilot PSP app, the pilot PSP app should already be connected to the distributing PSP through a Strong Customer Authentication process specific to that pilot PSP.

5.8.1.4.2. Requirement

The distributing PSP servicing the beta digital euro account, which must verify that the request is valid and consistent before providing the requested data.

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent.
2	C	If the request is valid, the pilot PSP must retrieve the already stored settings.
3	C	If the Waterfall Indicator is set to “1” in the incoming message, the function must retrieve the waterfall setting. Without existing setting, the function must assume the waterfall option is not selected by the individual end user.
4	C	If the Reverse Waterfall Indicator is set to “1” in the incoming message, the function must retrieve the reverse waterfall setting.

#	Mandatory Optional Conditional	Business rules description
		Without existing setting, the function must assume the reverse waterfall option is not selected by the individual end user.
5	C	If the Holdings limits indicator is set to "1" in the incoming message, the function must retrieve the corresponding setting.
6	M	The function must provide a return code.
7	C	If there aren't any existing settings, the function must be considered as successful.

5.8.1.4.3. Interface description

5.8.1.4.4. Message structure

Incoming message

The incoming message may either originate from the mobile application when requesting to view the existing settings or be generated directly by the pilot PSP if retrieving the settings (e.g., holdings limit) is required during the execution of a transaction.

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "Liquidity settings lookup".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Digital euro access number of the individual end user.	STR	18	O	***(New for beta digital euro)***
Waterfall Indicator	Dedicated indicator defining if the waterfall setting is requested.	BOOL	1	O	***(New for beta digital euro)***
Reverse waterfall indicator	Dedicated indicator defining if the reverse waterfall setting is requested.	BOOL	1	O	***(New for beta digital euro)***
Holdings limit indicator	Dedicated indicator defining if the holdings limit setting is requested.	BOOL	1	O	***(New for beta digital euro)***

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation. YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason
Waterfall Indicator	Dedicated indicator defining if the user opts for waterfall.	BOOL	1	O	***(New for beta digital euro)***
Reverse waterfall indicator	Dedicated indicator defining if the user opts for reverse waterfall.	BOOL	1	O	***(New for beta digital euro)***
Holdings limit	Maximum amount of beta digital euro that can be held by an individual end user.	NUM	18	O	***(New for beta digital euro)***

5.8.1.4.4.1. Return code

#	Description
0	Successful
1	Failure

5.8.1.4.4.2. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service)**.

5.8.1.5. Waterfall/Reverse Waterfall option check

This function ensures that, during a combined payment transaction involving funding or defunding, the distributing PSP verifies whether the beta digital euro account holder has enabled the appropriate automatic liquidity transfer option—Waterfall for defunding in case of excessive funds, or Reverse Waterfall for funding in case of insufficient funds. The function retrieves the relevant indicator (Waterfall or Reverse Waterfall) from the account's liquidity settings and rejects the transaction if the required option is not activated. It operates internally within the pilot PSP, without any interaction with the end user's device, and returns a result containing a return code and, when applicable, a reason code indicating the cause of rejection.

5.8.1.5.1. Pre-requisite

This function applies only when the funding amount is included in the incoming message (indicating a reverse waterfall combined transaction) or the defunding amount is included in the incoming message (indicating a waterfall combined transaction).

5.8.1.5.2. Requirements

The pilot PSP is the payer PSP

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent.
2	M	The function must retrieve the Reverse Waterfall indicator set for the corresponding beta digital euro account. Refer to **Digital euro pilot – Frontend specifications - Distributing PSP (section 5.8 - Liquidity settings service).**
3	M	If the Reverse Waterfall indicator is set to "NO" and a Funding amount is provided in the incoming message, the function rejects the transaction.
4	M	The function must provide a return code. In case of failure, a reason code must be provided.

The pilot PSP is the payee PSP

#	Mandatory Optional Conditional	Description
1	M	The request format and content must be consistent.
2	M	The function must retrieve the Waterfall indicator set for the corresponding beta digital euro account. Refer to **Digital euro pilot – Frontend specifications - Distributing PSP (section 5.8 - Liquidity settings service)**.
3	M	If the Waterfall indicator is set to "NO" and a defunding amount is provided in the incoming message, the function rejects the transaction.
4	M	The function must provide a return code. In case of failure, a reason code must be provided.

5.8.1.5.3. Interface description

This function is triggered without any interface exchanged between the end user's device and the distributing PSP. An internal request is generated by the pilot PSP servicing the beta digital euro account. The result of the Waterfall – Reverse Waterfall option lookup function is generated through an outgoing message.

5.8.1.5.3.1. Message structure

Incoming message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of message	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "Waterfall – Reverse Waterfall lookup"	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Identification of the beta digital euro account	STR	18	M	AccountIdentification
Defunding amount	Defunding amount in the case of a combined transaction on the payee side or a manual liquidity management defunding request.	NUM	18	O	Amount
Funding amount	Funding amount in the case of a combined transaction on the payer side or a manual liquidity management funding request.	NUM	18	O	Amount

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.8.1.5.3.2. Return code

#	Description
0	Successful
1	Failure

5.8.1.5.3.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.9. Notification settings service

Once an individual end user has successfully opened a beta digital euro account, they must be able to configure the events that trigger notifications. This service enables the definition, modification, or removal of notification configuration parameters.

Assumptions: Settings are managed at the level of the end user. The functions and the business rules are defined accordingly.

Service	Function/sub-functions	Description
Notification settings service	Notification settings request validation	The distributing PSP checks if the notification settings request sent by the individual end user's device is valid.
	Notification settings storage	The distributing PSP stores the new or updated parameters and confirms the settings.
	Notification settings removal	The distributing PSP clears all the parameters when the individual end user is offboarded.

5.9.1. Functions description

5.9.1.1. Notification settings request validation

5.9.1.1.1. Pre-requisite

The individual end user has been provided with a pilot PSP app either after completing a KYC process as a prospective customer, or as an existing customer of the distributing PSP.

5.9.1.1.2. Requirements

The individual end user must be able to request dedicated notification settings through their own device. The corresponding request is received by the distributing PSP servicing the beta digital euro account, which must verify that the request is valid and consistent.

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent.
2	M	The request must contain the User ID or the DEAN.
3	M	The request must contain the type of request: - CREATE ➔ Initial set-up - UPDATE ➔ Set-up amendment - DELETE ➔ Set-up removal
4	M	The request can be set to 'CREATE' only if no existing parameters are already defined.
5	M	The request can be set to UPDATE" or "DELETE" if a configuration already exists for the beta digital euro account.
6	C	If the request contains the User ID, the distributing PSP must find the beta digital euro account associated with the individual end user.
7	M	The function must provide a return code. In case of failure, a reason code must be provided.

5.9.1.1.3. Interface description

A dedicated interface received from an individual end user's device (incoming message) triggers the notification settings request validation. The function generates an outgoing message providing the result of the function execution.

5.9.1.1.3.1. Message structure

Incoming message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "Liquidity settings request"	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Type of request	Type of request sent by the individual end user' device. - **CRED - CREATE** - **UPDT - UPDATE** - **DELT - DELETE**	SSET	4	M	RequestType
Communication channel type	Method by which the notification is delivered: - **EMAL** (transmission by email) - **ONLI** (transmission online – e.g., in App) - **MBNO** (transmission by mobile phone) - **MULT** (Multichannel)	SSET	4	M	CommunicationMethodCode
Communication channel value	Detailed information of the method chosen to receive notification	STR	140	M	CommunicationMethodValue
Beta digital euro account credit event indicator	Specifies whether a notification is triggered when the beta digital euro account is credited.	BOOL	1	O	***(New for beta digital euro)***
Beta digital euro account debit event indicator	Specifies whether a notification is triggered when the beta digital euro account is debited.	BOOL	1	O	***(New for beta digital euro)***
Waterfall transaction event indicator	Specifies whether a notification is triggered when a waterfall process affects the beta digital euro account.	BOOL	1	O	***(New for beta digital euro)***
Reverse waterfall transaction event indicator	Specifies whether a notification is triggered when a reverse waterfall process affects the beta digital euro account.	BOOL	1	O	***(New for beta digital euro)***

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.9.1.1.3.2. Return code

#	Description
0	Successful
1	Failure

5.9.1.1.3.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.9.1.2. Notification settings storage

5.9.1.2.1. Requirements

Once the distributing PSP validates the notification settings request initiated by the individual end user through device, the new or updated parameters are stored.

#	Mandatory Optional Conditional	Business rules description
1	M	The distributing PSP must store the notification parameters provided by the individual end user.
2	M	Only new or updated information must be stored.
3	M	An amendment date must be stored. It must be in accordance with the current date.
4	C	If the event parameter is set to “False”, it should be interpreted as opting out.

5	C	If the event parameter is set to “True”, it should be interpreted as opting in.
6	M	The function must provide a return code to confirm or reject the notification set-up.

5.9.1.2.2. Interface description

This function is triggered without any interface exchanged between the payer and the distributing PSP. The result of the storage function is sent through an outgoing message.

5.9.1.2.2.1. Message structure

Incoming message

The incoming message is the request received from the device and validated by the notification settings request validation function. Refer to **section 5.9.1.1.3.1**.

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation. YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.9.1.2.2.2. Return code

#	Description
0	Successful
1	Failure

5.9.1.2.2.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.9.1.3. Notification settings removal

5.9.1.3.1. Requirements

When an individual end user initiates an offboarding request, any previous notification settings must be removed. This dedicated step of the offboarding process is executed under the coordination of the distributing PSP.

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent
2	M	The request must contain the DEAN or the end user identification
3	C	If the request contains the DEAN, the distributing PSP must retrieve the end user identifier
4	M	The distributing PSP must remove all the notification settings defined by the end user.
5	O	An amendment date could be stored. It must be in accordance with the current date
6	M	The function must provide a return code. In case of failure, a reason code must be provided.

5.9.1.3.1.1. Message structure

Incoming message

The incoming message is generated by the distributing PSP and contains the following data:

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of message	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: “Notification settings removal””	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Digital euro access number	Identification of the beta digital euro account	STR	18	O	***(New for beta digital euro)***
Identifier of the end user	Unique identifier of the end user	STR	TBD	O	***(New for beta digital euro)***

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation. YYYYMMDDThh:mm:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

5.9.1.3.1.2. Return code

#	Description
0	Successful
1	Failure

5.9.1.3.1.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.10. NFC CPACE management service

Since the end-to-end flows related to NFC services are defined in a generic manner and do not account for the integration of the two ECB components—the HCE SDK within the pilot PSP app and the HCE SDK backend in the PSP distribution environment—flow diagrams have been introduced to improve visualisation and understanding.

The NFC service management is triggered during initial individual end user enrolment to the NFC service and throughout the operational life cycle of the NFC functionality. It provides the necessary mechanisms to authorise enrolment by verifying individual end user eligibility and account status. It ensures that only valid beta digital euro accounts can access and maintain secure NFC capabilities. It manages various life cycle capabilities, including:

- NFC enrolment services

- NFC termination
- NFC key replenishment

Service	Function/sub-functions	Description
NFC CPACE service management *Management of rules handling NFC mobile payment enrollment and life cycle operations initiated via mobile app*	NFC CPACE enrolment management	The service offered by the distributing PSP allows its application to initiate an NFC enrolment preparation request. Once the individual end user's eligibility is verified, the distributing PSP calls the HCE SDK backend API to request the enrolment preparation. Upon receiving a reference from the backend, the pilot PSP returns this reference to its pilot PSP app, enabling it to trigger the NFC enrolment process via the HCE SDK.
	NFC CPACE enrolment confirmation	Once the NFC payment enrolment process is successfully completed, the HCE SDK backend sends a notification to the distributing PSP. This message confirms that the individual end user's device is now enrolled and ready to use the NFC payment service. It also includes the NFC token generated by the SEPI TSP, allowing the pilot PSP to update its records by associating the token with the corresponding DEAN and pilot PSP app ID.
	NFC CPACE Token mapping registration	The distributing PSP performs the token mapping registration by linking the NFC token with the DEAN and the pilot PSP app ID.
	NFC CPACE termination by Pilot PSP app management	The service offered by the distributing PSP allows its application to initiate an NFC termination preparation request. The distributing PSP calls the HCE SDK backend API to request the termination preparation. Upon receiving a reference from the backend, the pilot PSP returns this reference to its pilot PSP app, enabling it to trigger the NFC termination process via the HCE SDK.
	NFC CPACE termination by pilot PSP app confirmation	Once the NFC payment termination process is successfully completed, the HCE SDK backend sends a notification to the distributing PSP. This message confirms that the individual end user's device is now not enrolled anymore to the NFC payment service.
	NFC CPACE Termination by distributing PSP	The service provided by the distributing PSP enables the pilot PSP to request the NFC termination service. The pilot PSP provides the token to the HCE SDK Backend.
	NFC CPACE Token mapping deregistration	The distributing PSP performs the token mapping deregistration by removing the association between the NFC token, the DEAN, and the pilot PSP app ID.

5.10.1. Distributing PSP Onboarding pre-requisites

The distributing PSP has integrated the ECB SDK, which includes the HCE SDK, into its end user application, as well as the SDK backend, which includes the HCE SDK backend, into its backend environment. (Placeholder for reference to HCE SDK and HCE SDK backend ECB implementation guidelines).

The pilot PSP app is already connected to the distributing PSP through a Strong Customer Authentication process specific to that pilot PSP.

5.10.2. Functions description

5.10.2.1. NFC enrolment management

The service offered by the distributing PSP allows its application to initiate an NFC enrollment preparation request. Once the individual end user's eligibility is verified, the distributing PSP calls the HCE SDK backend API to request the enrollment preparation. Upon receiving a reference from the backend, the pilot PSP returns this reference to its pilot PSP app, enabling it to trigger the NFC enrollment process via the HCE SDK.

5.10.2.1.1. Full integration overview

This diagram provides a detailed view of the enrolment process for NFC mobile payments using a beta digital euro account.

Placeholder for E2E flow reference: Enrolment to NFC mobile payment with beta digital euro account

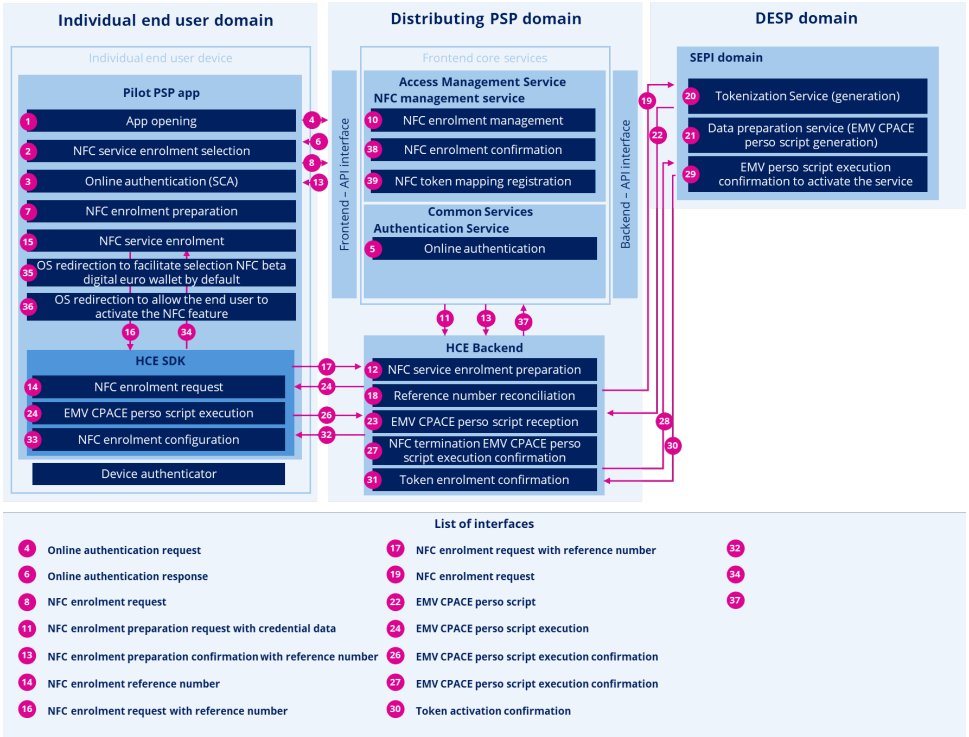

Figure 17 NFC CPACE enrolment

This section covers the following topics:

- 8 - NFC enrolment Request
- 9 – Local Fraud and sanction check (optional)
- 10 – NFC enrolment management
- 11 - NFC enrolment preparation request with credential data
- 13 - NFC enrolment preparation confirmation with ref number

5.10.2.1.2. Requirement

#	Mandatory Optional Conditional	Description
1	M	The request format and content must be consistent.
2	M	The distributing PSP checks the end user's beta digital account status to verify that the user is authorised to enroll in the NFC service.
3	O	The distributing PSP performs an internal fraud check.
4	M	The distributing PSP retrieves the last four digits of the DEAN.
5	M	The distributing PSP retrieves its PSP ID.
6	M	Based on its risk policies, the distributing PSP defines Key Counter Limits: • Maximum sessions allowed following a key replenishment • Trigger Replenishment Limit: When the number of remaining session keys falls below this threshold, the HCE SDK initiates a replenishment request to its backend (token requestor), provided there is internet connectivity. • Force Replenishment Limit: If internet connectivity remains unavailable and the number of remaining keys drops below this lower threshold, the HCE SDK triggers a notification prompting the user to connect their device to the internet in order to continue making NFC payments. Refer to **Key Replenishment Parameters in NFC HCE Payments for beta digital euro** below.
7	M	The distributing PSP calls the NFC enrolment preparation API of the HCE SDK backend, providing the pilot PSP ID, the last four digits of the DEAN, the maximum sessions allowed following a key replenishment, the Trigger Replenishment Limit and the Force Replenishment Limit. In return, it receives a reference. This API is defined by (Placeholder for HCE SDK and HCE SDK backend ECB implementation guidelines).
8	M	The distributing PSP returns the reference to its pilot PSP app, enabling it to trigger the NFC enrolment process via the HCE SDK.
9	M	The function provides a return code. In case of failure, a reason code must be provided.

Key Replenishment Parameters in NFC HCE Payments for beta digital euro

In the context of NFC HCE payments for the beta digital euro account, the Distributing PSP is responsible for configuring key replenishment parameters during the initial enrolment of the individual end user's device.

These parameters are critical for maintaining the cryptographic integrity of the payment process and must be set according to the pilot PSP's risk management policy.

Parameters defined by the pilot PSP

- **Maximum sessions allowed following a key replenishment**
  This defines the total number of NFC payment sessions that can be performed using a given set of session keys before a new key replenishment is required.
- **Trigger replenishment limit**
  When the number of remaining session keys drops below this threshold, the HCE SDK automatically initiates a replenishment request to its backend (token requestor), provided the device has internet connectivity.
- **Force replenishment limit**
  If the device remains offline and the number of remaining session keys drops below this critical threshold, the SDK triggers a notification to the pilot PSP app, prompting the individual end user to reconnect to the internet to continue making NFC payments.
  At this stage, the pilot PSP app must integrate the notification mechanism provided by the HCE SDK to ensure proper individual end user guidance.
- **Reference implementation parameters for key replenishment parameters**

Parameter	Minimum Value	Maximum Value
Maximum Sessions Allowed	8	12
Trigger Replenishment Limit	5	8
Force Replenishment Limit	2	4

Note: In NFC beta digital euro implementation, the replenishment mechanism is based on key usage counters, not on fixed time durations.

5.10.2.1.3. Interface description

5.10.2.1.3.1. Message structure

Incoming message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the event	Unique identifier of the event that triggers the message: "NFC enrolment request".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason
DEAN last four Digits	DEAN last four Digits to be printed on the payment receipt.	STR	4	O	***(New for beta digital euro)***
Reference	The reference is a unique identifier returned by the HCE SDK backend after a successful NFC enrolment preparation request. It serves as a secure handle that the pilot PSP app uses to initiate or continue the NFC enrolment process with the HCE SDK.	UUID	36	O	***(New for beta digital euro)***

5.10.2.1.3.2. Return code

#	Description
0	Successful
1	Failure

5.10.2.1.3.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.10.2.2. NFC enrolment confirmation

Once the NFC payment enrolment process is successfully completed, the HCE SDK backend sends a notification to the distributing PSP. This message confirms that the user's device is now enrolled and ready to use the NFC payment service. It also includes the NFC token generated by the SEPI TSP, allowing the pilot PSP to update its records by associating the token with the corresponding DEAN and Pilot PSP app ID.

5.10.2.2.1. Full integration overview

This diagram provides a detailed view of the enrolment process for NFC mobile payments using a beta digital euro account. It complements the end-to-end (E2E) flows that will be included in a future release.

Placeholder for E2E flow: Enrolment to NFC mobile payment with beta digital euro account.

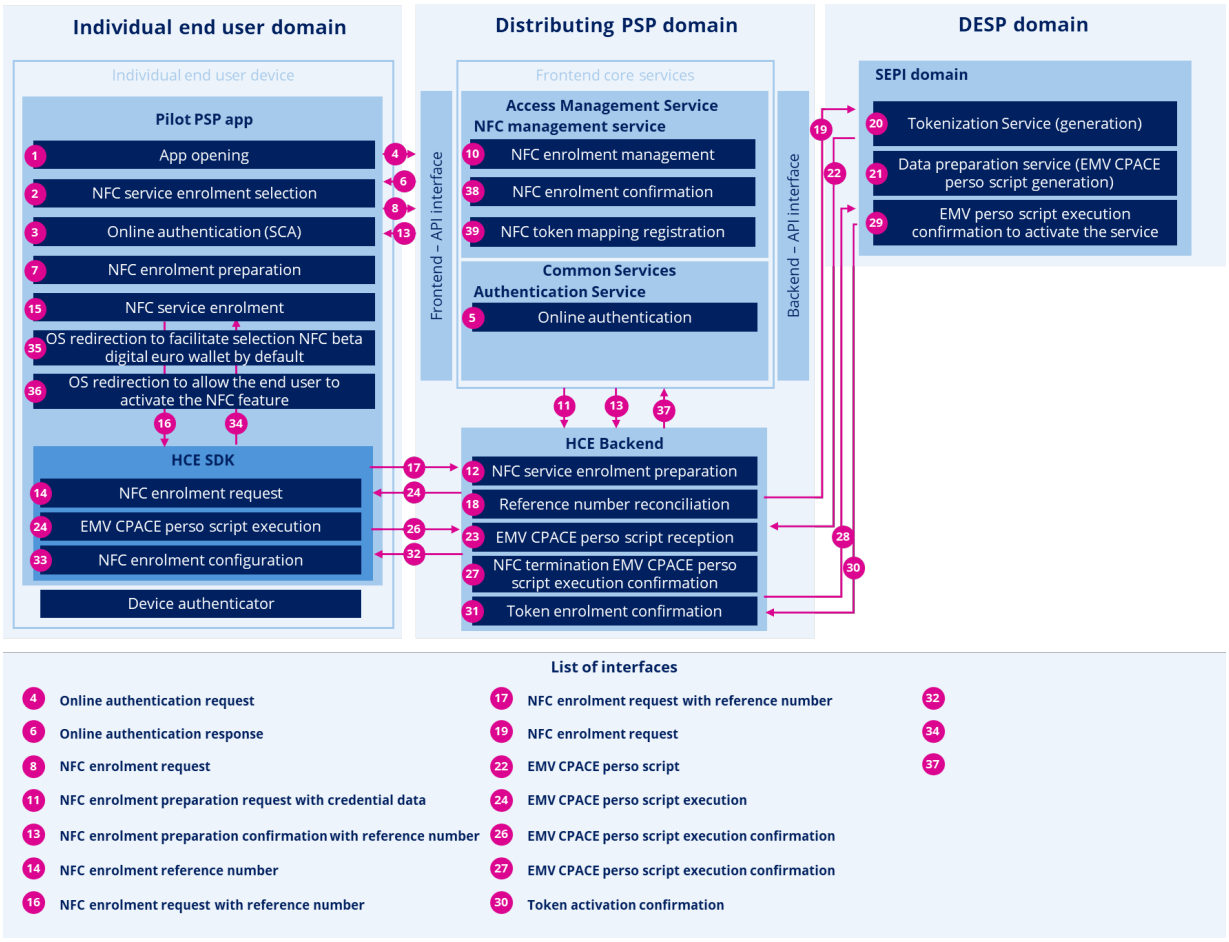

Figure 18 NFC CPACE enrolment

This section covers the following topics:

- 37 - NFC service enrolment confirmation
- 38 - NFC enrolment confirmation

5.10.2.2.2. Requirement

#	Mandatory Optional Conditional	Description
1	M	Once the enrollment is completed, the HCE SDK backend notifies the distributing PSP of the successful completion and provides the generated NFC token. This notification is defined by ECB (Placeholder for HCE SDK and HCE SDK backend ECB implementation guidelines).
2	M	The distributing PSP links the NFC token received from the HCE SDK backend to the DEAN and the pilot PSP app ID, enabling future detokenization during NFC transactions.

5.10.2.3. NFC Token mapping registration

The distributing PSP performs the token mapping registration by linking the NFC token with the DEAN and the pilot PSP app ID.

5.10.2.3.1.1. Full integration overview

This diagram provides a detailed view of the enrolment process for NFC mobile payments using a beta digital euro account.

Placeholder for E2E flow reference: Enrolment to NFC mobile payment with beta digital euro account

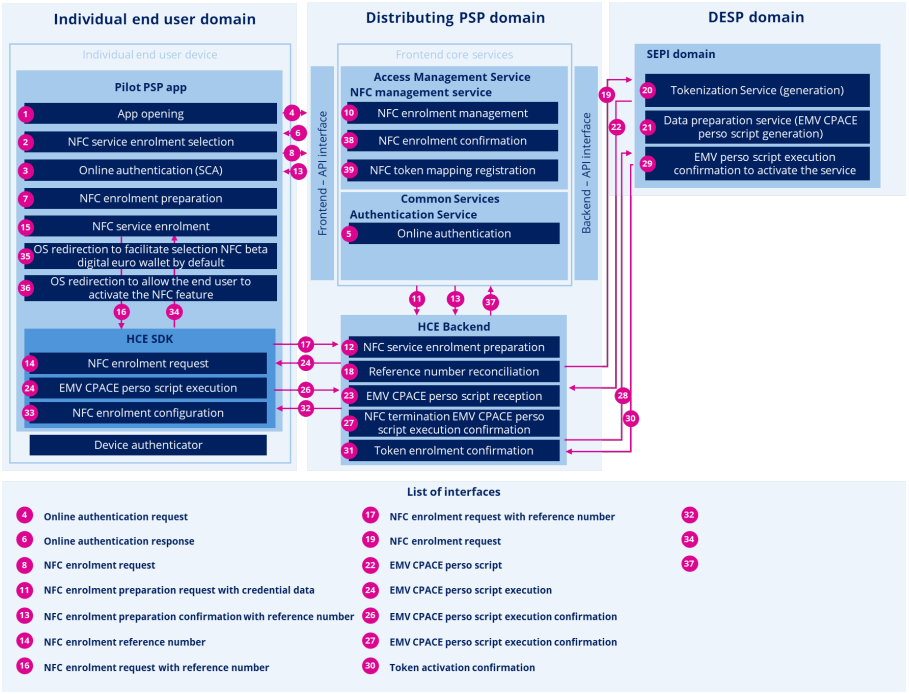

Figure 19 NFC CPACE enrolment

This section covers the following topics:

- 39 – NFC token mapping registration

5.10.2.3.1.2. Requirement

#	Mandatory Optional Conditional	Description
1	M	The distributing PSP performs the token mapping registration by linking the NFC token with the DEAN and the pilot PSP app ID.

5.10.2.4. NFC Termination by pilot PSP app management

The service offered by the distributing PSP allows its application to initiate an NFC termination preparation request. The distributing PSP calls the HCE SDK backend API to request the termination preparation. Upon receiving a reference from the backend, the pilot PSP returns this reference to its pilot PSP app, enabling it to trigger the NFC termination process via the HCE SDK.

5.10.2.4.1. Full integration overview

This diagram provides a detailed view of the termination process by pilot PSP for NFC mobile payments using a beta digital euro account. It complements the end-to-end (E2E) flows that will be included in a future release.

Placeholder for E2E flow reference: Termination of NFC mobile payment with beta digital euro account

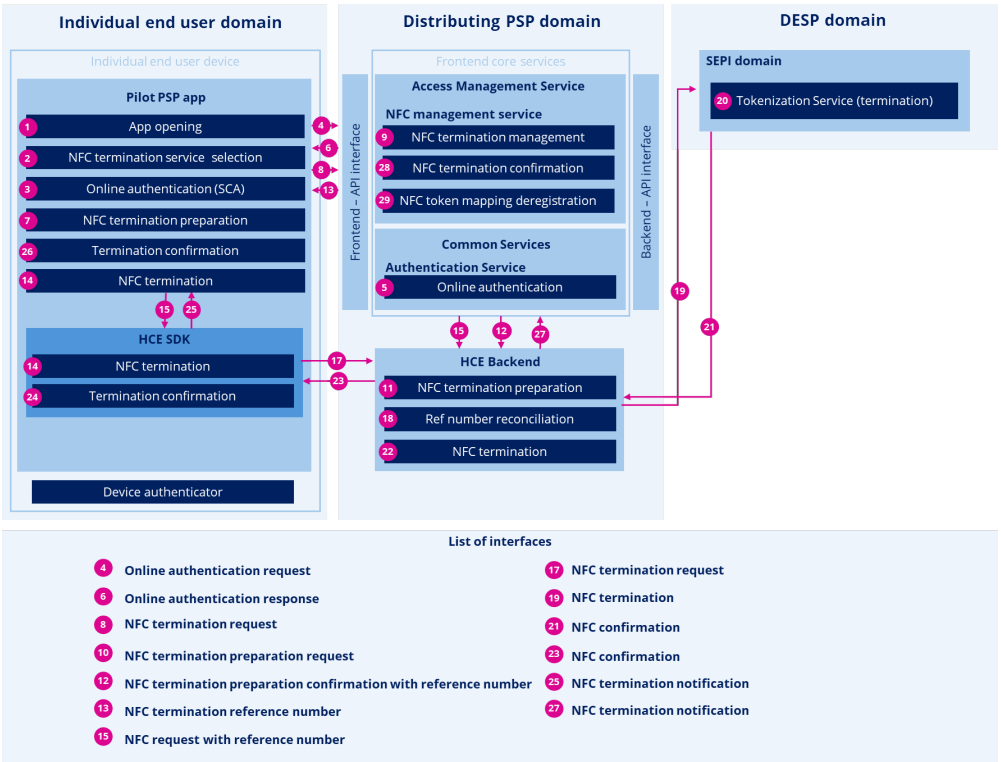

Figure 20 NFC CPACE termination by app

This section covers the following topics:

- 8 - NFC termination request
- 9 - NFC termination management
- 10 - NFC termination preparation request

- 12 - NFC termination preparation confirmation with Ref number
- 13 - NFC termination Ref number

5.10.2.4.2. Requirement

#	Mandatory Optional Conditional	Description
1	M	The request format and content must be consistent.
2	M	The distributing PSP checks the individual end user's beta digital account status to verify that the individual end user is enrolled in the NFC service.
3	M	The distributing PSP calls the NFC termination preparation API of the HCE SDK backend, providing the NFC token linked to the DEAN and the pilot PSP app ID. In return, it receives a reference. This API is defined by ECB (Placeholder for HCE SDK and HCE SDK backend ECB implementation guidelines).
4	M	The distributing PSP returns the reference to its pilot PSP app, enabling it to trigger the NFC termination process via the HCE SDK.
5	M	The function provides a return code. In case of failure, a reason code must be provided.

5.10.2.4.3. Interface description

5.10.2.4.3.1. Message structure

Incoming message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "NFC termination request"	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification

Data element	Description	Type	Length	Presence indicator	Standardised name
Date of the message	Date time of the message creation. YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason
Reference	The reference is a unique identifier returned by the HCE SDK backend after a successful NFC termination preparation request. It serves as a secure handle that the pilot PSP app uses to initiate or continue the NFC enrolment process with the HCE SDK.	UUID	36	O	***(New for beta digital euro)***

5.10.2.4.3.2. Return code

#	Description
0	Successful
1	Failure

5.10.2.4.3.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

5.10.2.5. NFC termination by pilot PSP app confirmation

Once the NFC payment termination process is successfully completed, the HCE SDK backend sends a notification to the distributing PSP. This message confirms that the individual end user's device is now not enrolled anymore to the NFC payment service.

5.10.2.5.1. Full integration overview

This diagram provides a detailed view of the termination process by pilot PSP for NFC mobile payments using a beta digital euro account. It complements the end-to-end (E2E) flows that will be included in a future release.

Placeholder for E2E flow reference: Termination of NFC mobile payment with beta digital euro account

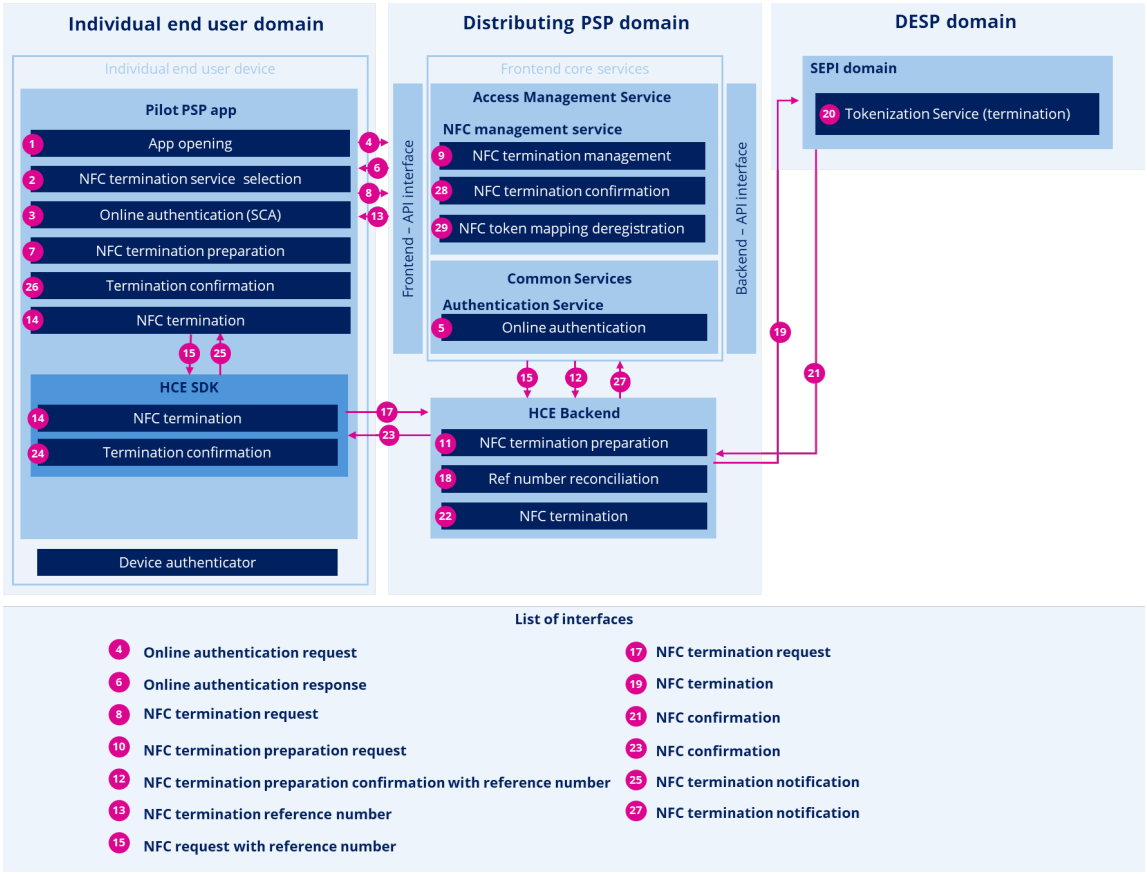

Figure 21 NFC CPACE termination by app

This section covers the following topics:

- 27 - NFC termination notification
- 28 - NFC termination confirmation

5.10.2.5.2. Requirement

#	Mandatory Optional Conditional	Description
1	M	Once the enrolment is completed, the HCE SDK backend notifies the distributing PSP of the successful termination. This notification is defined by ECB (Placeholder for HCE SDK and HCE SDK backend ECB implementation guidelines)
2	M	The distributing PSP deletes the mapping between the NFC Token and its linked DEAN and pilot PSP app ID.

5.10.2.6. NFC Termination by distributing PSP

The service provided by the distributing PSP enables the pilot PSP to request the NFC termination service. The pilot PSP provides the token to the HCE SDK Backend.

5.10.2.6.1. Full integration overview

This diagram provides a detailed view of the termination process by distributing PSP for NFC mobile payments using a beta digital euro account. It complements the end-to-end (E2E) flows that will be included in a future release.

Placeholder for E2E flow: Enrolment to NFC mobile payment with beta digital euro account

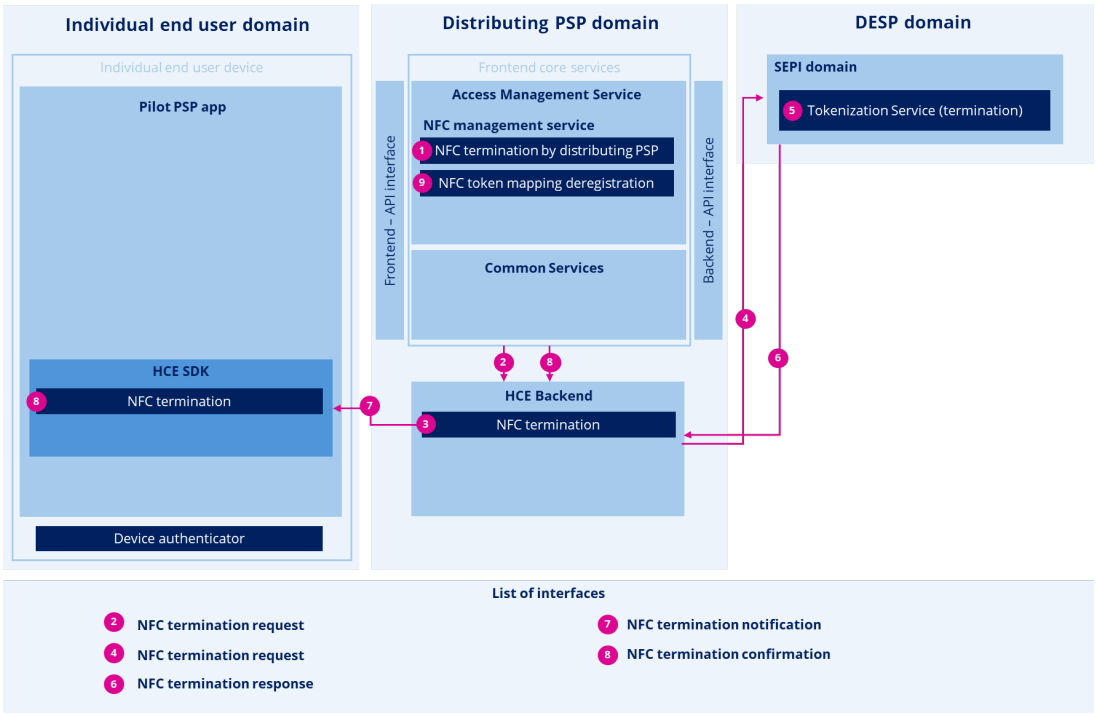

Figure 22 NFC termination by distributing PSP

This section covers the following topics:

- 1 - NFC termination by pilot PSP
- 2 - NFC termination request

5.10.2.6.2. Requirement

#	Mandatory Optional Conditional	Description
1	M	The request format and content must be consistent.
2	M	The distributing PSP retrieves the associated NFC token corresponding to the DEAN and pilot PSP app when the termination is targeted and call an API provided by the HCE app Backend. (Placeholder for HCE SDK and HCE SDK backend ECB implementation guidelines)

5.10.2.7. NFC Token mapping deregistration

The distributing PSP performs the token mapping deregistration by removing the association between the NFC token, the DEAN, and the pilot PSP app ID. This function is part of the termination process.

5.10.2.7.1.1. Full integration overview

This diagram provides a detailed view of the termination process by pilot PSP for NFC mobile payments using a beta digital euro account. It complements the end-to-end (E2E) flows that will be included in a future release.

Placeholder for E2E flow reference: Termination of NFC mobile payment with beta digital euro account

Termination by app

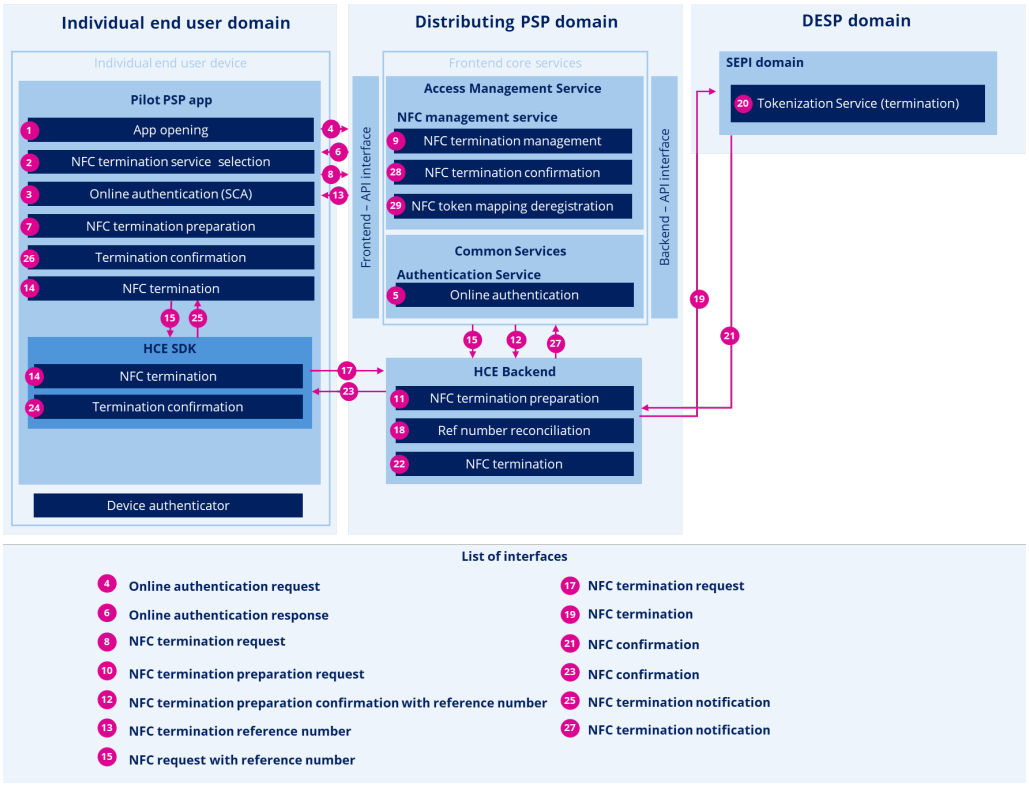

Figure 23 NFC termination by app

This section covers the following topics:

- 29 – NFC token mapping deregistration

Termination by distributing PSP

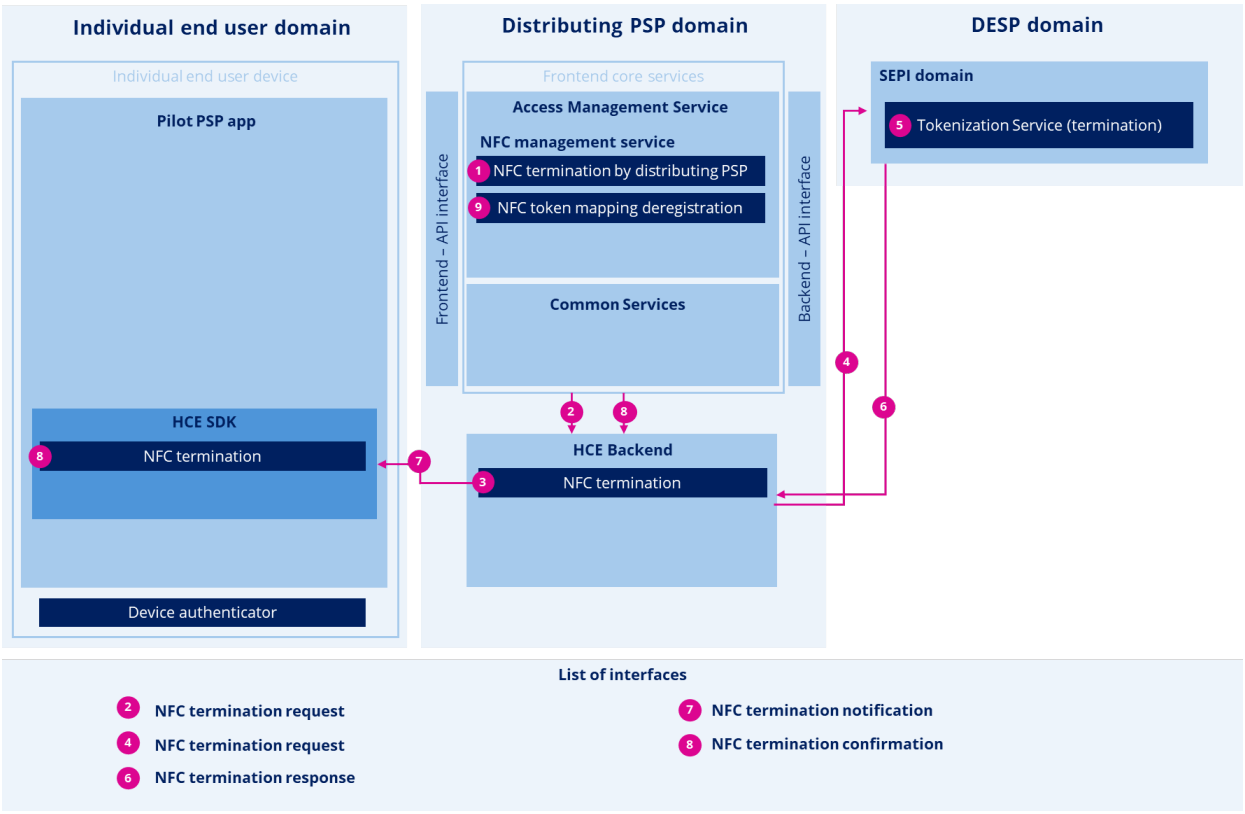

Figure 24 NFC termination by distributing PSP

This section covers the following topics:

- 9 – NFC token mapping deregistration

5.10.2.7.1.2. Requirement

#	Mandatory Optional Conditional	Description
1	M	The distributing PSP performs the token mapping deregistration by removing the association between the NFC token, the DEAN, and the pilot PSP app ID.

6. Liquidity Management Service

The liquidity management core service is dedicated to functions needed for funding and defunding beta digital euro accounts and commercial bank money accounts.
The following services are identified for the current scope of use cases:

- Manual (de)funding initiation service
- Commercial bank money account funds management service
- Beta digital euro account funds management service
- Funding - Defunding settlement processing service
- Funding - Defunding post settlement processing

6.1. Manual (de)funding initiation service

The manual service is invoked during a funding or defunding process by the distributing PSP servicing the beta digital euro account of the individual end user.

Service	Function	Description
Manual(de)funding initiation	Online manual funding/defunding request initiation validation	The distributing PSP checks if the funding request or the defunding request in valid.

6.1.1. Functions description

6.1.1.1. Manual funding/defunding request initiation validation

6.1.1.1.1. Requirements

A manual funding request is sent by the individual end user's device to the distributing PSP to trigger the funding or defunding processing. The distributing PSP must validate the request is consistent enough to proceed properly with the funding or defunding.

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent.
2	M	The type of request must be “ONLF” (Online funding) or “ONLD” (Online defunding)
3	C	If the type of request is “ONLF” (Online funding): - the identifier of the event must be “Online funding request” - the type of debited account must be “CBMA” (Commercial Bank Money Account) - the type of credited account must be “DEUR” (beta digital euro account) - the debited Commercial Bank Money account must be populated - the transaction type must be “FU01”
4	C	If the type of request is “ONLD” (Online defunding):

#	Mandatory Optional Conditional	Business rules description
		- the identifier of the event must be “Online defunding request” - the type of debited account must be “DEUR” (beta digital euro account) - the type of credited account must be “CBMA” (Commercial Bank Money Account) - the credited Commercial Bank Money Account must be populated - the transaction type must be “DF01”
5	C	If the function execution is successful, the pilot PSP must generate the Unique Identifier providing an end-to-end reference for the (de)funding transaction and store the transaction. This unique Identifier (UETR) format must follow the UUID type (xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx)
6	M	The function must provide the execution status. In case of failure, a reason code must be provided.

6.1.1.1.2. Interface description

6.1.1.1.2.1. Message structure

Incoming message

The same structure is used for funding and defunding requests:

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message.	STR	35	M	EventIdentification
Type of request	Type of request sent by the individual end user's device. - **ONLF** (Online funding request) - **ONLD** (Online defunding request)	SSET	4	M	RequestType
Transaction type[1]	Type of transaction initiated by the individual end user: - PP01: P2P payment	SSET	4	M	TransactionType

[1] The “transaction type” data element is defined as “payment type” in the back-end specifications. A mapping between these two data must be performed when transmitting a transaction to DESP. However, the coding of the different types remains the same.

Data element	Description	Type	Length	Presence indicator	Standardised name
	- PB01: POI payment – e-com / POS - BP01: B2P payment - FU01: Funding - DF01: Defunding				
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Digital euro access number of the individual end user.	STR	18	O	***(New for beta digital euro)***
Type of debited account	Type of account that is debited: - **CBMA** (Commercial bank money account - **DEUR** (beta digital euro account)	SSET	4	M	CashAccountType
Debited account identification	Identification of the account that is debited	STR	34	M	DebtorAccount
Debited Account pilot PSP identification	Identifier of the pilot PSP (BIC) servicing the debited account.	STR	11	O	DebtorAgent
Credited account identification	Identification of the account that is credited.	STR	34	M	CreditorAccount
Type of credited account	Type of account that is credited: - **CBMA** (Commercial bank money account - **DEUR** (beta digital euro account)	SSET	4	M	CashAccountType
Credited Account pilot PSP identification	Identifier of the pilot PSP (BIC) servicing the credited account.	STR	11	O	CreditorAgent
Amount	Amount that is transferred during the online funding process.	NUM	18	M	Amount
Unstructured remittance	Free-text or code/value provided for facilitating reconciliation.	STR	140	O	UnstructuredRemittanceInformation
Transaction Identification	End to end identification entered by the end user at the initiation of the transaction.	STR	35	O	EndToEndId

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

6.1.1.1.2.2. Return code

#	Description
0	Successful
1	Failure

6.1.1.1.2.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 – Notification Service).**

6.2. Commercial bank money account funds management service

Commercial bank money account funds management service is invoked during a payment or a funding-defunding transaction process by the pilot PSP servicing the commercial bank money account of the individual end user. It offers the needed functions to handle the commercial money account funds.

Service	Function	Description
Commercial bank money account funds management service	Commercial bank money account debit	The pilot PSP servicing commercial bank money account defunds commercial bank money account.
	Commercial bank money account credit	The pilot PSP servicing commercial bank money account funds commercial bank money account.

Service	Function	Description
	Funds blocking	The pilot PSP servicing commercial bank money account blocks funds on commercial bank money account if reverse waterfall is needed for transaction execution and in funding processes.
	Funds release	The pilot PSP servicing commercial bank money account releases funds blocked on commercial bank money account if the process is discontinued.

This service is used in several processes and is described as a Common function. Refer to **Digital euro pilot – Frontend specifications - Common Services**.

6.3. Beta digital euro account funds management service

Beta digital euro funds management service is invoked during a payment transaction process (to debit or credit the beta digital euro account) or a funding or defunding process by the distributing PSP servicing the beta digital euro account of the individual end user.

Service	Function	Description
Beta digital euro account funds management service	Beta digital euro account debit	The distributing PSP debits the beta digital euro account.
	Beta digital euro account credit	The distributing PSP credits the beta digital euro account.

This service is used in several processes and is described as a Common function. Refer to **Digital euro pilot – Frontend specifications - Common Services**.

6.4. Funding - Defunding settlement processing service

Funding / defunding settlement processing service is invoked in funding and defunding processes, in relation to DESP.

Service	Function	Description
Funding defunding settlement processing service	Funding online settlement request creation	The pilot PSP sends a funding online request to DESP.
	Funding online settlement request validation	The pilot PSP receives a funding online settlement request from DESP and validates it.
	Funding online settlement processing	The pilot PSP manages the interactions with DESP and the funding online transaction life cycle.

Service	Function	Description
	Defunding online settlement request creation	The pilot PSP sends a defunding online request to DESP.
	Defunding online settlement request validation	The pilot PSP receives a defunding online settlement request from DESP and validates it.
	Defunding online settlement processing	The pilot PSP manages the interactions with DESP and the defunding online transaction life cycle.

This service is used in several processes and is described as a Common function. Refer to **Digital euro pilot – Frontend specifications - Common Services**.

6.5. Funding – Defunding post settlement service

Funding – Defunding post settlement service is invoked during a funding or defunding process by the pilot PSP servicing the beta digital euro account of the individual end user to close the funding or defunding process post settlement.

Service	Function	Description
Funding – Defunding post settlement service	Funding post settlement	The distributing PSP checks the funding settlement outcome received from DESP and orchestrates the next steps to close the funding process.
	Defunding post settlement	The distributing PSP checks the defunding settlement outcome received from DESP and orchestrates the next steps to close the defunding process.

This service is used in several processes and is described as a Common function. Refer to **Digital euro pilot – Frontend specifications - Common Services**.

7. Transaction Management Service

Transaction management core service is dedicated to functions needed for paying or receiving payments in beta digital euro. All the payment methods are covered.

7.1. Payment initiation service

The payment initiation service is invoked when a payment transaction is initiated by a payer or a payee. It offers the needed functions to initiate the payment process regardless of the form factors.

Service	Function	Description
Payment initiation service	DEAN validity check	The distributing PSP servicing the beta digital euro account checks if the DEAN provided is consistent.
	Alias validity check	The distributing PSP servicing the beta digital euro account receives an alias validity check from a device, checks the consistency and triggers the alias lookup dispatch.
	Alias lookup dispatch	The distributing PSP servicing the beta digital euro account requests an alias resolution to DESP to retrieve payer's or payee's details (DEAN and pilot PSP Identifier).
	PSP ID lookup dispatch	The distributing PSP servicing the beta digital euro account requests the pilot PSP Identifier corresponding to a DEAN to DESP for routing purpose.
	Tokenization request	The distributing PSP servicing the beta digital euro receives a payment request through QR or Pay-by-link and requests a token to DESP.
	Pay-by-link creation	The distributing PSP servicing the beta digital euro account generates a payment link.
	Individual end user payment instruction initiation validation	The distributing PSP servicing the beta digital euro account of the payer checks the payment instruction initiation sent by the end user's device is consistent and contains mandatory information.
	Individual end user payment request initiation validation	The distributing PSP servicing the beta digital euro account of the payee checks the payment request initiation sent by the end user's device is consistent and contains mandatory information.

7.1.1. Functions descriptions

7.1.1.1. Individual end user payment instruction initiation validation

7.1.1.1.1. Requirements

A payment instruction initiation is sent by the payer's device to the distributing PSP to trigger the payment instruction processing. The distributing PSP must validate the payment instruction is consistent enough to proceed properly with the payment.

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent
2	M	Following data must always be provided - Message Identifier - Event Identifier - Message date time - Transaction initiation Method - Transaction type - Transaction Amount - Transaction Currency - Transaction date - Merchant Category Code
3	M	The transaction type must be populated with "PP01" for P2P payment
4	M	The Merchant Category Code must be populated with "4829" for P2P transaction
5	O	The following data could be entered by the end user and provided with the payment instruction: - Unstructured remittance - Transaction identification
7	C	If the Transaction initiation method is "alias" the following data must be provided: - Payee alias type - Payee alias value
8	C	If the Transaction initiation method is "alias" an alias resolution is already performed the following data must be provided: - Payee DEAN - Payee pilot PSP ID
9	C	If the function execution is successful, the distributing PSP must generate the Unique Identifier providing an end-to-end reference for the payment transaction and store the transaction. This unique Identifier (UETR) format must follow the UUID type (xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx)
10	M	The function must provide the return code. In case of rejection, a reason code must be provided.

7.1.1.1.2. Interface description

A dedicated interface received from an end user's device (incoming message) triggers the payment request validation function. The function generates an outgoing message providing the result of the function execution.

7.1.1.1.2.1. Message structure

Incoming message

A same incoming message structure is proposed for all the payment methods:

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of incoming message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "Individual end user payment initiation"	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Transaction initiation method	Method chosen by the individual end user to initiate a payment transaction. - Alias - DEAN Code set TBD	STR	4	M	PaymentMethod
Transaction type[2]	Type of transaction initiated by the individual end user: - PP01: P2P payment - PB01: POI payment – e-com / POS - BP01: B2P payment - FU01: Funding - DF01: Defunding	STR	4	M	TransactionType
Transaction Amount	Amount of the transaction	NUM	18	M	Amount
Transaction Currency	Currency of the transaction Code set defined in ISO 4217	SSET	3	M	Currency

[2] The "transaction type" data element is defined as "payment type" in the back-end specifications. A mapping between these two data must be performed when transmitting a transaction to DESP. However, the coding of the different types remains the same.

Data element	Description	Type	Length	Presence indicator	Standardised name
Transaction Date	Date of the transaction YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Payee Name	Name of the payee	STR	70	O	Name
Payee DEAN	DEAN of the payee	STR	18	O	***(New for beta digital euro)***
Payee pilot PSP ID	Pilot PSP Identifier of the payee	STR	11	O	CreditorAgent
Payee alias value	Alias of the payee	STR	2048	O	ProxyAccount
Payee alias Type	Alias type of the payee Code set defined in ExternalProxyAccountType1code	SSET	4	O	ProxyAccountType
Merchant Category Code	Four-digit code that classifies the business type of the merchant Code set defined in ISO 18245	NSET	4	M	MerchantCategoryCode
Unstructured remittance	Free text information provided to the payer regarding the payment, which may include transaction details or references.	STR	140	O	UnstructuredRemittanceInformation
Transaction Identification	End-to-end identification entered by the end user at the initiation of the transaction.	STR	35	O	EndToEndId

Outgoing message

Data	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode

Data	Description	Type	Length	Presence indicator	Standardised name
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

7.1.1.1.2.2. Return code

#	Description
0	Successful
1	Failure

7.1.1.1.2.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 - Notification Service).**

7.2. Transaction history service

This service is invoked when an individual end user requests to view their transactions history. It enables dynamic retrieval.

Service	Function	
Transaction history service	Transactions history request validation	The distributing PSP servicing the beta digital euro account checks if the transaction history request sent by the end user's device is consistent and contains mandatory information.
	Transactions history lookup	The distributing PSP servicing the beta digital euro account retrieves the transactions according to the criteria.

7.2.1. Functions description

7.2.1.1. Transaction history request validation

7.2.1.1.1. Pre-requisite

The individual end user has been provided with a pilot PSP app either after completing a KYC process as a prospective customer, or as an existing customer of the distributing PSP.

7.2.1.1.2. Requirements

A request is sent by the individual end user's pilot PSP app to the distributing PSP to retrieve transactions history according to criteria. The distributing PSP must validate that the request is consistent enough before proceeding with the transaction history lookup.

Assumptions ➔ The format and consistency checks performed by the Mobile App are not duplicated in the service.

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent.
2	M	The function must provide a return code. In case of rejection, a reason code must be provided.
3	C	If the function is successfully processed, the transactions history lookup function must be triggered.

7.2.1.1.3. Interface description

7.2.1.1.3.1. Message structure

Incoming message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "Transactions history request"	STR	35	M	EventIdentification
Date of the message	Date time of the message creation. YYYY-MM-DDThh:hh:sssZ	Datetime UTC	24	M	CreationDateTime
Transaction type[3]	Type of transaction initiated by the individual end user: - PP01: P2P payment - PB01: POI payment – e-com / POS - BP01: B2P payment - FU01: Funding - DF01: Defunding	SSET	4	M	TransactionType
Transaction Start Date	Start date of the search period (YYYY-MM-DD)	Date	10	O	DateFrom
Transaction End Date	End date of the search period (YYYY-MM-DD)	Date	10	O	DateTo
Transaction Identification	End-to-end identification entered by the end user at the initiation of the transaction.	STR	35	O	EndToEndId

[3] The "transaction type" data element is defined as "payment type" in the backend implementation specifications. A mapping between these two data must be performed when transmitting a transaction to DESP. However, the coding of the different types remains the same.

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	O	OriginalMessageIdentification
Date of the message	Date time of the message creation. YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Return Code	Exit code of function providing the status that the process returns when executed.	BOOL	1	M	ReturnCode
Reason code	Populated only in case of rejection and corresponds to the rejection root cause.	NSET	4	O	StatusReason

7.2.1.1.3.2. Return code

#	Description
0	Successful
1	Failure

7.2.1.1.3.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 - Notification Service).**

7.2.1.2. Transaction history lookup

7.2.1.2.1. Requirements

If the transactions history request sent by the individual end user's device is valid, the distributing PSP retrieves the transactions according to the criteria stipulated in the request.

#	Mandatory Optional Conditional	Business rules description
1	M	The selection of transactions must be consistent with the criteria received in the request.

#	Mandatory Optional Conditional	Business rules description
2	C	If the individual end user provides a transaction identification, only the corresponding transaction must be provided.
3	C	If there are no dates (start date – end-date) in the request, all the transactions must be considered. (The pilot PSP has the ability to determine how much history is included)
4	C	If there is only the start date in the request, only the transactions whose date is equal or greater than this date must be considered.
5	C	If there is only the end-date in the request, only the transactions whose date is lower or equal to this date must be considered.
6	C	If there are both the start date and the end-date in the request, all the transactions whose date is equal or greater than the start date and lower or equal to the end-date must be considered.
7	C	If one or many transaction types are in the request, all the transactions holding the requested transaction types must be considered.
8	C	If there are no transaction types in the request, all the transactions must be considered.
9	M	The function must provide a return code. In case of rejection, a reason code must be provided.
10	C	If the function is processed successfully but no transactions meet the criteria, a reason code must be provided to inform the individual end user.

7.2.1.2.2. Interface description

7.2.1.2.2.1. Message structure

Incoming message

The incoming message is the request received from the device and validated by the Transaction history request validation function (**section 7.2.1.1.3.1**).

Outgoing message

Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	35	M	MessageIdentification
Identifier of the incoming message	Unique identifier of the corresponding incoming message populated if the function is triggered by an incoming message.	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Transaction list	*	STR	*	M	*
Transaction type	Type of transaction initiated by the individual end user: - Purchase	STR	4	M	TransactionType

Data element	Description	Type	Length	Presence indicator	Standardised name
	- Refund - Account to Account (List to be confirmed and code set to be clarified)				
Transaction Amount	Amount of the transaction	NUM	18	M	Amount
Transaction Currency	Currency of the transaction (code set defined in ISO 4217)	STR	3	M	Currency
Transaction Date	Date of the transaction YYYY-MM-DDThh:hh:sssZ	DATETIME UTC	24	M	CreationDateTime
Payee Name / Merchant Name	Name of the payee or name of the business end user	STR	70	O	Name
DEAN	Identification of the beta digital euro account of the payee.	STR	18	M	***(New for beta digital euro)***
Transaction Identification	End-to-end identification entered by the end-user at the initiation of the transaction.	STR	35	O	EndToEndId

7.2.1.2.2.2. Return code

#	Description
0	Successful
1	Failure

7.2.1.2.2.3. Functional error description (reason code)

The functional error descriptions are listed in **Digital euro pilot – Frontend specifications – Common Services (section 4.2 - Notification Service)**.

7.3. Mobile NFC token transaction service

As the end-to-end flows related to NFC services are not yet defined, sequence diagrams are introduced to enhance visualisation and understanding. These diagrams are temporary and will be removed from the specifications once the corresponding end-to-end flows are provided.

Upon receiving and validating the NFC payment request, the distributing PSP initiates a cryptogram check request to SEPI, then retrieves the DEAN and app ID.

Service	Function/sub-functions	Description
Mobile NFC token	NFC cryptogram check	The distributing PSP sends an NFC cryptogram verification request to the DESP-SEPI, which responds with the outcome of the cryptogram check.

Service	Function/sub-functions	Description
transaction service	NFC token mapping look up	The distributing PSP retrieves the DEAN and the pilot PSP app ID thanks to the token. Once this is done, the distributing PSP can continue to validate the payment request with other common services like “balance check”.

7.3.1.1. Function description

7.3.1.1.1. NFC cryptogram check

7.3.1.1.1.1. Full integration diagram

This diagram provides a detailed view of the NFC mobile payments process using a beta digital euro account.

E2E flow reference: TM-1.6 Online contactless SoftPOS payment with mobile device - same pilot PSP

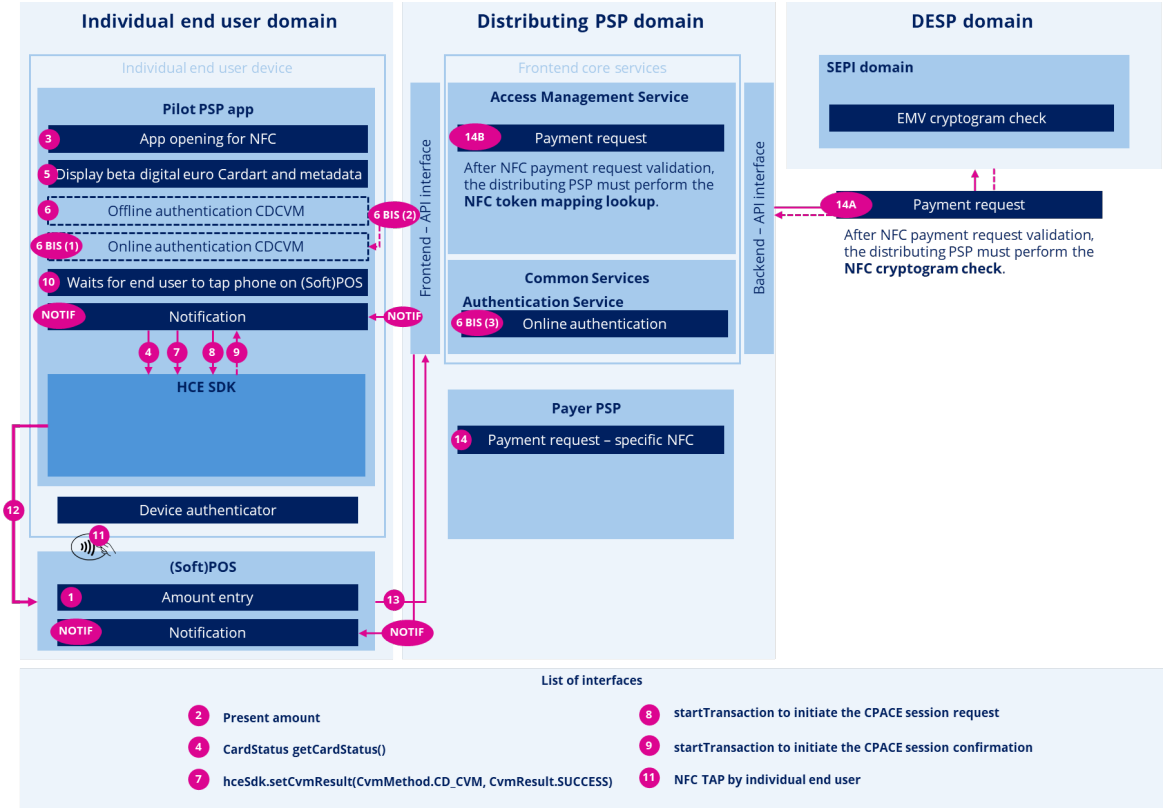

Figure 25 Online contactless (Soft)POS payment with mobile device - same pilot PSP

This section covers the following topics:

14-A – NFC cryptogram check

7.3.1.1.1.2. Requirement

#	Mandatory Optional Conditional	Description
1	M	The distributing PSP sends an NFC cryptogram verification request to the DESP-SEPI, which responds with the outcome of the cryptogram check. Refer to **Digital euro pilot – Backend specifications**.

7.3.1.1.2. NFC token mapping lookup

7.3.1.1.2.1. Full integration diagram

This diagram provides a detailed view of the NFC mobile payments process using a beta digital euro account.

E2E flow reference: TM-1.6 Online contactless SoftPOS payment with mobile device - same pilot PSP

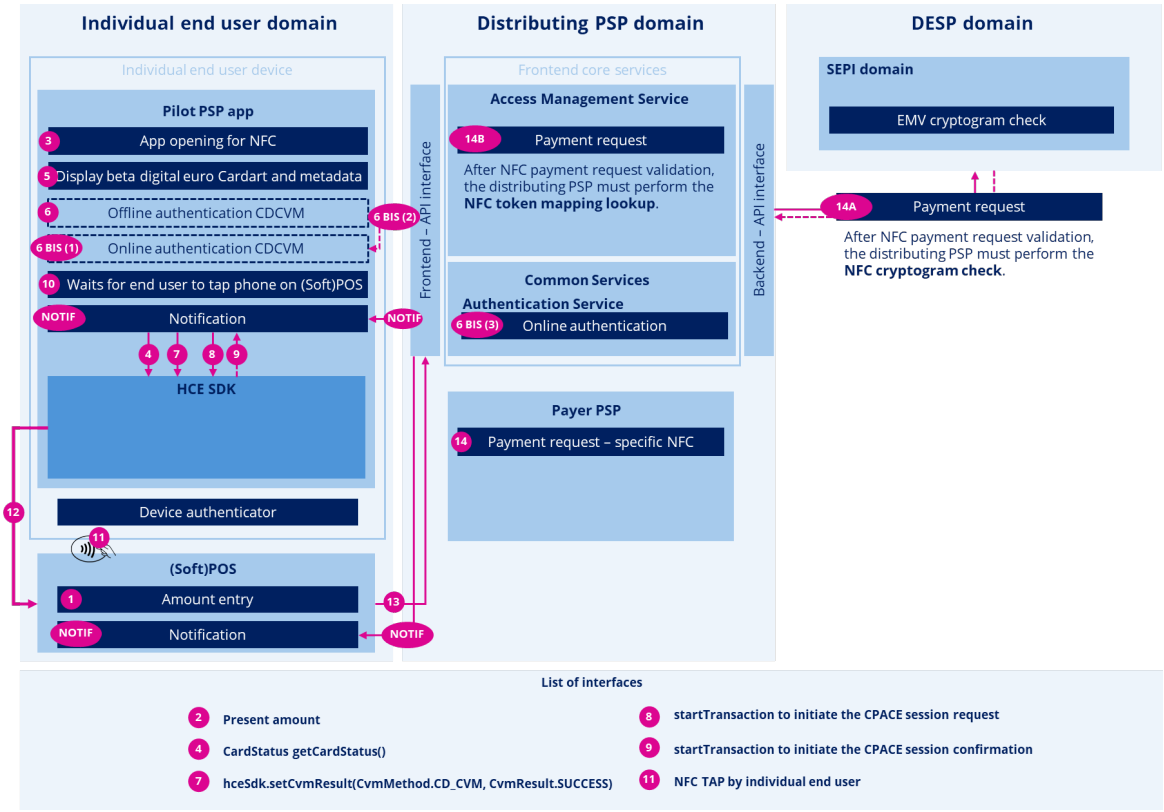

Figure 26 Online contactless (Soft)POS payment with mobile device - same pilot PSP

This section covers the following topics:

14-B – NFC token mapping lookup

7.3.1.1.2.2. Requirement

#	Mandatory Optional Conditional	Description
1	M	The distributing PSP must retrieve the DEAN and the pilot PSP app ID thanks to the token.
2	M	After retrieving the DEAN and the pilot PSP app ID, the distributing PSP must continue the process and orchestrate the other functions.