



EUROPEAN CENTRAL BANK
EUROSYSTEM

Digital euro pilot

Frontend implementation specifications

Common Services



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.



EUROPEAN CENTRAL BANK

EUROSYSTEM

Table of Contents

Introduction	9
Structure of the document	9
1. Access Management Service	11
1.1. Alias validity check	11
1.1.1. <i>Requirements</i>	11
1.1.2. <i>Interface description</i>	11
1.1.2.1. <i>Message structure</i>	11
1.1.2.2. <i>Return code</i>	12
1.1.2.3. <i>Functional error description (reason code)</i>	13
1.2. Alias look-up dispatch	13
1.3. Pilot PSP identifier look-up dispatch	13
1.4. Beta digital euro account service	13
1.4.1. <i>Functions description</i>	14
1.4.1.1. <i>Beta digital euro account status check</i>	14
1.4.1.1.1. <i>Requirements</i>	14
1.4.1.1.2. <i>Interface description</i>	14
1.4.1.1.2.1. <i>Message structure</i>	14
1.4.1.1.2.2. <i>Return code</i>	15
1.4.1.1.2.3. <i>Functional error description (reason code)</i>	16
1.4.1.2. <i>Beta digital euro account (un)blocking request validation</i>	16
1.4.1.2.1. <i>Pre-requisite</i>	16
1.4.1.2.2. <i>Requirements</i>	16
1.4.1.2.3. <i>Interface description</i>	16
1.4.1.2.3.1. <i>Message structure</i>	16
1.4.1.2.3.2. <i>Return code</i>	17
1.4.1.2.3.3. <i>Functional error description (reason code)</i>	18
1.4.1.3. <i>Beta digital euro account blocking</i>	18



EUROPEAN CENTRAL BANK

EUROSYSTEM

1.4.1.3.1.	<i>Requirements</i>	18
1.4.1.3.2.	<i>Interface description</i>	18
1.4.1.3.2.1.	<i>Message structure</i>	18
1.4.1.3.2.2.	<i>Return code</i>	19
1.4.1.3.2.3.	<i>Functional error description (reason code)</i>	19
1.4.1.4.	<i>Beta digital euro account unblocking</i>	19
1.4.1.4.1.	<i>Requirements</i>	19
1.4.1.4.2.	<i>Interface description</i>	19
1.4.1.4.2.1.	<i>Message structure</i>	19
1.4.1.4.2.2.	<i>Return code</i>	20
1.4.1.4.2.3.	<i>Functional error description (reason code)</i>	20
1.4.1.5.	<i>DEAN validity check</i>	20
1.4.1.5.1.	<i>Requirements</i>	20
1.4.1.5.2.	<i>Interface description</i>	21
1.4.1.5.2.1.	<i>Message structure</i>	21
1.4.1.5.2.2.	<i>Return code</i>	22
1.4.1.5.2.3.	<i>Functional error description (reason code)</i>	22
1.4.1.6.	<i>DEAN registration</i>	22
1.4.1.7.	<i>DEAN deregistration and pilot PSP mapping removal</i>	23
1.5.	<i>Linked account settings service</i>	24
1.5.1.	<i>Functions description</i>	24
1.5.1.1.	<i>Linked account settings request validation</i>	24
1.5.1.1.1.	<i>Pre-requisite</i>	24
1.5.1.1.2.	<i>Requirements</i>	24
1.5.1.1.3.	<i>Interface description</i>	25
1.5.1.1.3.1.	<i>Message structure</i>	25
1.5.1.1.3.2.	<i>Return code</i>	26
1.5.1.1.3.3.	<i>Functional error description (reason code)</i>	26
1.5.1.2.	<i>Linked account settings storage</i>	26



EUROPEAN CENTRAL BANK

EUROSYSTEM

1.5.1.2.1.	<i>Requirements</i>	26
1.5.1.2.2.	<i>Interface description</i>	27
1.5.1.2.2.1.	<i>Message structure</i>	27
1.5.1.2.2.2.	<i>Return code</i>	27
1.5.1.2.2.3.	<i>Functional error description (reason code)</i>	28
1.5.1.3.	<i>Linked commercial bank money account ownership validation</i>	28
1.5.1.4.	<i>Linked account settings removal</i>	28
1.5.1.4.1.	<i>Requirements</i>	28
1.5.1.4.1.1.	<i>Message structure</i>	28
1.5.1.4.1.2.	<i>Return code</i>	29
1.5.1.4.1.3.	<i>Functional error description (reason code)</i>	29
1.5.1.5.	<i>Linked commercial bank money account look-up</i>	30
1.5.1.5.1.	<i>Requirements</i>	30
1.5.1.5.2.	<i>Interface description</i>	30
1.5.1.5.2.1.	<i>Message structure</i>	30
1.5.1.5.2.2.	<i>Return code</i>	31
1.5.1.5.2.3.	<i>Functional error description (reason code)</i>	31
1.5.1.6.	<i>Linked commercial bank money account check</i>	31
1.5.1.6.1.	<i>Pre-requisite</i>	32
1.5.1.6.2.	<i>Requirements</i>	32
1.5.1.6.3.	<i>Interface description</i>	32
1.5.1.6.3.1.	<i>Message structure</i>	32
1.5.1.6.3.2.	<i>Return code</i>	33
1.5.1.6.3.3.	<i>Functional error description (reason code)</i>	33
1.6.	<i>Commercial bank money account service</i>	34
1.6.1.	<i>Functions description</i>	34
1.6.1.1.	<i>Commercial bank money account balance look-up</i>	34
1.6.1.1.1.	<i>Requirements</i>	34
1.6.1.1.2.	<i>Interface description</i>	34



EUROPEAN CENTRAL BANK

EUROSYSTEM

1.6.1.2.	<i>Commercial bank money account status check</i>	35
1.6.1.2.1.	<i>Interface description</i>	35
1.6.1.3.	<i>Commercial bank money account balance check</i>	35
1.6.1.3.1.	<i>Requirements</i>	35
1.6.1.3.2.	<i>Interface description</i>	35
1.7.	<i>End user deactivation</i>	35
2.	Liquidity Management Service	37
2.1.	<i>Beta digital euro account funds management service</i>	37
2.1.1.1.	<i>Beta digital euro account debit</i>	37
2.1.1.1.1.	<i>Requirements</i>	37
2.1.1.1.2.	<i>Interface description</i>	37
2.1.1.1.2.1.	<i>Message structure</i>	37
2.1.1.1.2.2.	<i>Return code</i>	38
2.1.1.1.2.3.	<i>Functional error description (reason code)</i>	38
2.1.1.2.	<i>Beta digital euro account credit</i>	39
2.1.1.2.1.	<i>Requirements</i>	39
2.1.1.2.2.	<i>Interface description</i>	39
2.1.1.2.2.1.	<i>Message structure</i>	39
2.1.1.2.2.2.	<i>Return code</i>	40
2.1.1.2.2.3.	<i>Functional error description (reason code)</i>	40
2.2.	<i>Commercial bank money account funds management service</i>	40
2.2.1.	<i>Functions description</i>	41
2.2.1.1.	<i>Commercial bank money account debit</i>	41
2.2.1.1.1.	<i>Requirements</i>	41
2.2.1.1.2.	<i>Interface description</i>	41
2.2.1.2.	<i>Commercial bank money account credit</i>	41
2.2.1.2.1.	<i>Requirements</i>	41
2.2.1.2.2.	<i>Interface description</i>	42
2.2.1.3.	<i>Funds blocking</i>	42



EUROPEAN CENTRAL BANK

EUROSYSTEM

2.2.1.3.1.	<i>Requirements</i>	42
2.2.1.3.2.	<i>Interface description</i>	42
2.2.1.4.	<i>Funds release</i>	42
2.2.1.4.1.	<i>Requirements</i>	42
2.2.1.4.2.	<i>Interface description</i>	42
2.3.	<i>Funding - Defunding settlement processing service</i>	43
2.3.1.	<i>Functions description</i>	43
2.3.1.1.	<i>Funding online settlement request creation</i>	43
2.3.1.1.1.	<i>Requirements</i>	43
2.3.1.2.	<i>Funding online settlement processing</i>	44
2.3.1.2.1.	<i>Requirements</i>	44
2.3.1.3.	<i>Defunding online settlement request creation</i>	45
2.3.1.3.1.	<i>Requirements</i>	45
2.3.1.4.	<i>Defunding online settlement processing</i>	46
2.3.1.4.1.	<i>Requirements</i>	46
2.4.	<i>Funding – Defunding post settlement service</i>	47
2.4.1.	<i>Functions description</i>	47
2.4.1.1.	<i>Funding post settlement</i>	47
2.4.1.1.1.	<i>Requirements</i>	47
2.4.1.1.2.	<i>Interface description</i>	48
2.4.1.1.2.1.	<i>Message structure</i>	48
2.4.1.1.2.2.	<i>Return code</i>	50
2.4.1.1.2.3.	<i>Functional error description (reason code)</i>	50
2.4.1.2.	<i>Defunding post settlement</i>	50
2.4.1.2.1.	<i>Requirements</i>	50
2.4.1.2.2.	<i>Interface description</i>	51
2.4.1.2.2.1.	<i>Message structure</i>	51
2.4.1.2.2.2.	<i>Return code</i>	53
2.4.1.2.2.3.	<i>Functional error description (reason code)</i>	53



EUROPEAN CENTRAL BANK

EUROSYSTEM

3.	Transaction Management Service	54
3.1.	Amount assessment service	54
3.1.1.	<i>Functions description</i>	54
3.1.1.1.	<i>Pre-settlement payment - funding defunding amount assessment</i>	54
3.1.1.1.1.	<i>Requirements</i>	54
3.1.1.1.2.	<i>Interface description</i>	56
3.1.1.1.2.1.	<i>Message structure</i>	56
3.1.1.1.2.2.	<i>Return code</i>	58
3.1.1.1.2.3.	<i>Functional error description (reason code)</i>	58
3.1.1.2.	<i>Post settlement payment - funding defunding amount assessment</i>	58
3.1.1.2.1.	<i>Requirements</i>	58
3.1.1.2.2.	<i>Interface description</i>	59
3.1.1.2.2.1.	<i>Message structure</i>	59
3.1.1.2.2.2.	<i>Return code</i>	61
3.1.1.2.2.3.	<i>Functional error description (reason code)</i>	61
3.1.1.3.	<i>Pre-settlement payment refund – funding defunding amount assessment</i>	61
3.1.1.3.1.	<i>Requirements</i>	61
3.1.1.3.2.	<i>Interface description</i>	62
3.1.1.3.2.1.	<i>Message structure</i>	62
3.1.1.3.2.2.	<i>Return code</i>	64
3.1.1.3.2.3.	<i>Functional error description (reason code)</i>	64
3.1.1.4.	<i>Post settlement refund – funding defunding amount assessment</i>	64
3.1.1.4.1.	<i>Requirements</i>	64
3.1.1.4.2.	<i>Interface description</i>	66
3.1.1.4.2.1.	<i>Message structure</i>	66
3.1.1.4.2.2.	<i>Return code</i>	67
3.1.1.4.2.3.	<i>Functional error description (reason code)</i>	67
3.2.	Payment processing service	68
3.2.1.	<i>Functions description</i>	68



EUROPEAN CENTRAL BANK

EUROSYSTEM

3.2.1.1.	<i>Payment request creation</i>	68
3.2.1.1.1.	<i>Requirements</i>	69
3.2.1.2.	<i>Payment request validation</i>	69
3.2.1.2.1.	<i>Requirements</i>	69
3.2.1.3.	<i>Payment instruction creation</i>	70
3.2.1.3.1.	<i>Requirements</i>	70
3.2.1.4.	<i>Payment instruction validation</i>	70
3.2.1.4.1.	<i>Requirements</i>	71
3.2.1.5.	<i>Payment settlement processing</i>	71
3.2.1.5.1.	<i>Requirements</i>	71
3.3.	<i>Refund processing service</i>	73
3.3.1.	<i>Functions description</i>	73
3.3.1.1.	<i>Refund request creation</i>	73
3.3.1.1.1.	<i>Pre-requisite</i>	74
3.3.1.1.2.	<i>Requirements</i>	74
3.3.1.2.	<i>Refund request validation</i>	74
3.3.1.2.1.	<i>Requirements</i>	74
3.3.1.3.	<i>Refund settlement processing</i>	75
3.3.1.4.	<i>Requirements</i>	75
4.	Cross-functional Services	77
4.1.	Authentication Service	77
4.2.	Notification Service	77
4.2.1.	<i>Requirements</i>	77
4.2.2.	<i>Interface description</i>	79
4.2.3.	<i>Message structure</i>	79
4.2.4.	<i>Notification classification</i>	80
4.2.5.	<i>Functional error description</i>	81

Introduction

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

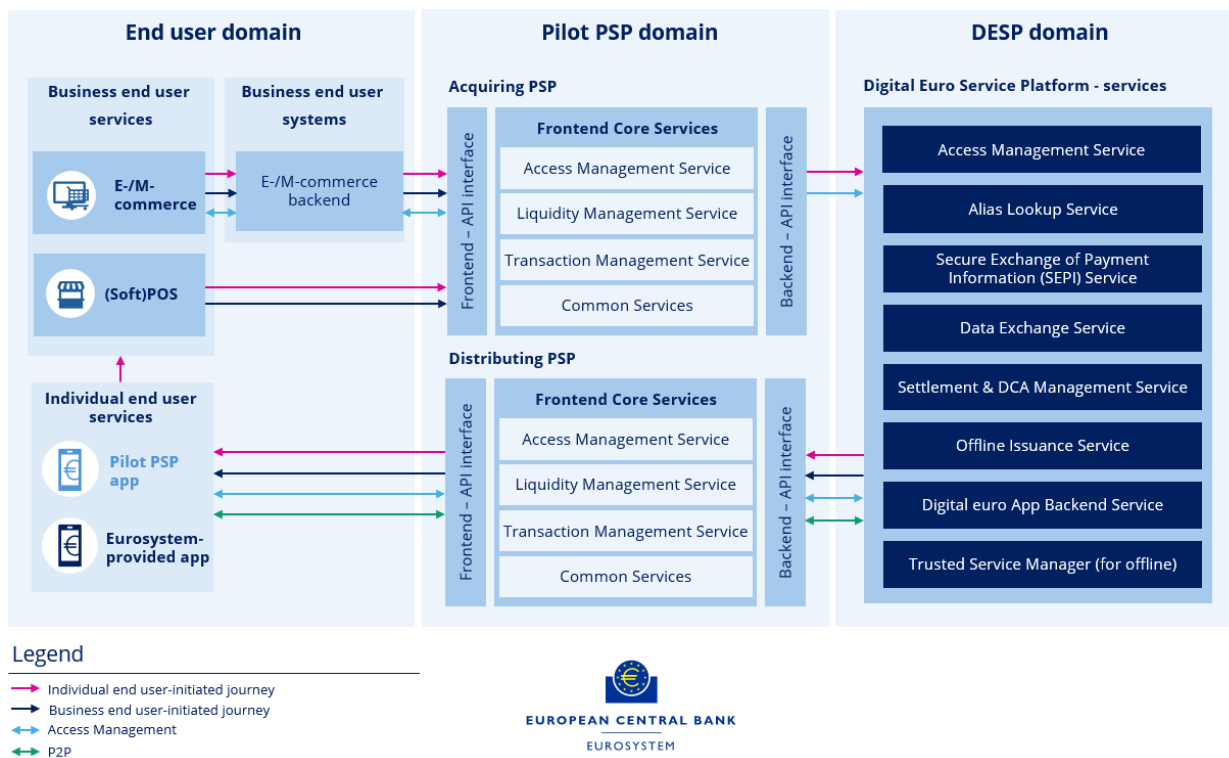


Figure 1 Digital euro pilot - Functional architecture

The current document is dedicated to common services. The common services are used by several processes handled by distributing PSP and acquiring PSP and to avoid having them described in several documents, they are specified here. A service can contain several functions, and each function consists of a description, applicable business rules and supported message types and data element definition.

The current document version considers only online beta digital euro payment transactions.

Structure of the document

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

- Access Management Service
- Liquidity Management Service
- Transaction Management Service



EUROPEAN CENTRAL BANK

EUROSYSTEM

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.

Certain services do not fall exclusively under a core service. They are detailed in a dedicated section entitled “Cross-functional Service”. 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 <ul style="list-style-type: none"> - M: Mandatory - O: Optional - C: Conditional
Standardised name	Name formatted according to a predefined conventions to ensure consistency and uniformity (ISO 20022). Note: No standardised 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 HH:MM:SS.sss
Datetime UTC	Date and time format in UTC YYYY-MM-DDThh:mm:ss.sssZ
Binary (BIN)	Method of encoding data using sequences of bits
UUID	Universally Unique Identifier xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

1. Access Management Service

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

1.1. Alias validity check

In a peer-to-peer transaction, a payment via an alias can be initiated by the payer.

- In the payer-initiated scenario, the payer uses their device to start the transaction and provides the payee's alias,

In an e-commerce transaction, a payment via an alias can be requested by the payer who provides their alias.

The end user device or the digital point of interaction sends an alias validity check request to the pilot PSP which checks its consistency and triggers the alias look-up dispatch function. This approach ensures flexibility and secure alias verification during the transaction initiation process.

1.1.1. Requirements

The pilot PSP must validate the consistency of the alias validity check request that is provided.

#	Mandatory Optional Conditional	Business rules description
1	M	The function triggering message format and content must be consistent
2	M	The alias value format must be consistent with the alias type.
3	M	The function must provide the execution status. In case of rejection, a reason code must be provided.

1.1.2. Interface description

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

1.1.2.1. Message structure

Incoming message

Data	Description	Type	Length	Presence indicator	Standardised name
Identifier of the message	Unique identifier of incoming message	STR	36	M	MessageIdentification



Data	Description	Type	Length	Presence indicator	Standardised name
Identifier of the event	Unique identifier of the event that triggers the message: "Alias Validity check".	STR	36	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Alias type	A classification defining a pseudonymous account identifier of the payer or the payee. Refer to ISO external Code (ExternalProxyAccountType1code) <ul style="list-style-type: none"> - EMAL (e-mail address) - MBNO (Mobile phone number) 	SSET	4	M	ProxyAccountType
Alias value	A pseudonymous account identifier of the payer or payee used in a transaction.	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	36	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	36	O	OriginalMessageIdentification
Date of the message	Date time of the message creation. YYYY-MM-DDThh:mm:ss.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

1.1.2.2. Return code

#	Description
0	Successful
1	Failure

1.1.2.3. Functional error description (reason code)

The functional error descriptions are listed in **section 4.2 - Notification Service**.

1.2. Alias look-up dispatch

When a business end user accepts beta digital euro payments via an alias or when an individual end user (payer) pays another individual end user using the alias of the payee, the alias must be resolved by the pilot PSP to proceed with the payment.

Alias resolution is performed by DESP, and the pilot PSP must invoke the corresponding service.

Digital euro pilot - Backend implementation specifications → "POST/ lookups/aliases – Request" | "POST/ lookups/aliases – Response"

1.3. Pilot PSP identifier look-up dispatch

When a business end user accepts beta digital euro payments with DEAN or when an individual end user (payer) pays another individual end user using the DEAN of the payee, the PSP ID of the DEAN must be identified to ensure the payment request can be routed to the right party.

PSP ID look-up is performed by DESP, and the pilot PSP must invoke the corresponding service.

Digital euro pilot - Backend implementation specifications → "POST/lookups/deans – Request" | "POST/lookups/deans – Response"

1.4. Beta digital euro account service

The beta digital euro account service is triggered during the onboarding of an individual end user and throughout the beta digital euro account lifecycle management.

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.



Service	Function/sub-functions	Description
	DEAN validity check	The pilot PSP servicing the beta digital euro account checks if the DEAN provided is consistent.
	DEAN registration	The pilot PSP requests a DEAN to DESP and stores it.
	DEAN deregistration 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.

1.4.1. Functions description

1.4.1.1. Beta digital euro account status check

1.4.1.1.1. Requirements

The pilot PSP servicing the beta digital euro account must check the account status before processing any transaction.

Exception: in the case of an NFC transaction performed while the payer device is not connected, this verification cannot be carried out. In such cases, the transaction may proceed without the account status check.

#	Mandatory Optional Conditional	Business rules description
1	M	The system must verify that the provided account is a beta digital euro account before proceeding with any further checks.
2	M	If the provided DEAN is not linked to a beta digital euro account in status active, the process must be discontinued.
3	M	The function must provide a return code. In case of failure, a reason code must be provided.

1.4.1.1.2. Interface description

This function is triggered without any interface exchanged between the end user's device and the distributing PSP. The distributing PSP activates the function during any process when beta digital euro account status check is required.

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

1.4.1.1.2.1. 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	36	M	MessageIdentification
Identifier of the event	Unique identifier of the event that triggers the message: "beta digital euro account status check"	STR	36	M	EventIdentification
Date of the message	Date time of the message creation. YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Digital euro access number	Identification of the digital euro account.	STR	18	M	<i>(New for beta digital euro)</i>

Outgoing message

Data element	Description	Type	Length	Presence Indicator	Standardised name
Identifier of the message	Unique identifier of outgoing message.	STR	36	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	36	O	OriginalMessageIdentification
Date of the message	Date time for the message creation. YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Digital euro account status	A representation of the state of the beta digital euro account according to its life cycle.	SSET	4	M	AccountStatusCode
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

1.4.1.1.2.2. *Return code*

#	Description
0	Successful
1	Failure



1.4.1.1.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 Notification Service**.

1.4.1.2. *Beta digital euro account (un)blocking request validation*

1.4.1.2.1. *Pre-requisite*

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

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

1.4.1.2.2. *Requirements*

The end user must be able to block and unblock their beta digital euro account directly from their own pilot PSP app. The corresponding request is received by the pilot PSP servicing the beta digital euro account, which must verify that the request is valid and consistent with the beta digital euro account status.

#	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	C	If the request contains the User ID, the distributing PSP must find the beta digital euro account associated with the individual end user.
4	M	The request must contain the type of request - UNBL → Unblocking - BLCK → Blocking
5	M	An unblocking request is valid only if the status of the beta digital euro account is "BLOCKED"
6	M	A blocking request is valid only if the status of the beta digital euro account is "OPEN"
7	M	The function must provide a return code. In case of failure, a reason code must be provided.

1.4.1.2.3. *Interface description*

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

1.4.1.2.3.1. *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: "DEAN (un)blocking request".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Type of request	Type of request sent by the end-user' device. <ul style="list-style-type: none"> - BLCK (Account blocking) - UNBL (Account unblocking) 	STR	4	M	RequestType
Digital euro access number	Identification of the beta digital euro account.	STR	18	O	<i>(New for beta digital euro)</i>
Identifier of the end user	Unique identifier of the end user. <i>Populated if the process is successful</i>	STR	TBD	O	<i>(New for beta digital euro)</i>

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:mm:ss.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

1.4.1.2.3.2. Return code

#	Description
0	Successful



#	Description
1	Failure

1.4.1.2.3.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

1.4.1.3. *Beta digital euro account blocking*

1.4.1.3.1. *Requirements*

Once the pilot PSP validates the blocking request initiated by the end user through device, the beta digital euro account is effectively blocked.

#	Mandatory Optional Conditional	Business rules description
1	M	The beta digital euro account status must be switched to "Blocked".
2	M	An amendment date must be provided. It must be in accordance with the current date.
3	O	An additional information can be added to track the account status change.
4	M	The function must provide a return code. In case of failure, a reason code must be provided.

1.4.1.3.2. *Interface description*

This function is triggered without any interface exchanged between the end user's device and the pilot PSP. The result of the beta digital euro account blocking function is sent through an outgoing message.

1.4.1.3.2.1. *Message structure*

Incoming message

The incoming message is the request received from the device and validated by the beta digital euro account (un)blocking request validation function.

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	DATETIME UTC	24	M	CreationDateTime



Data element	Description	Type	Length	Presence indicator	Standardised name
	YYYY-MM-DDThh:mm:ss.sssZ				
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

1.4.1.3.2.2. *Return code*

#	Description
0	Successful
1	Failure

1.4.1.3.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

1.4.1.4. *Beta digital euro account unblocking*

1.4.1.4.1. *Requirements*

Once the pilot PSP validates the unblocking request initiated by the end user through device, the beta digital euro account is effectively unblocked.

#	Mandatory Optional Conditional	Business rules description
1	M	The beta digital euro account status must be switched to "Open".
2	M	An amendment date must be provided. It must be in accordance with the current date.
3	O	An additional information can be added to track the account status change.
4	M	The function must provide a return code. In case of failure, a reason code must be provided.

1.4.1.4.2. *Interface description*

This function is triggered without any interface exchanged between the end user's device and the pilot PSP. The result of the beta digital euro account unblocking function is sent through an outgoing message.

1.4.1.4.2.1. *Message structure*

Incoming message



The incoming message is the request received from the device and validated by the beta digital euro account (un)blocking request validation function.

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:mm:ss.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

1.4.1.4.2.2. *Return code*

#	Description
0	Successful
1	Failure

1.4.1.4.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

1.4.1.5. *DEAN validity check*

1.4.1.5.1. *Requirements*

In payment instruction or payment request use cases, either the payer's PSP or the payee's PSP must confirm that the DEAN provided by the counterparty is valid.

#	Mandatory Optional Conditional	Business rules description
1	M	The function triggering message format and content must be consistent.
2	M	The DEAN length must be equal to 18 characters.
3	M	The DEAN format must respect the following structure.



#	Mandatory Optional Conditional	Business rules description
		<ul style="list-style-type: none"> EU code + Check Digits + Basic European Account Number.
4	M	EU code is 2 characters, must be Latin alphabetic characters and equal to "EU". Countries where Latin characters are not used must comply with this rule too.
5	M	Check digit is 2 characters and is used to ensure the validation of the DEAN. ISO/IEC 7064, MOD 97-10 algorithm must be used to confirm the validity of the DEAN.
6	M	Basic European Account Number is 14 characters and must be numeric characters. 0 to 9 digits are allowed.
7	M	The first digit of the Basic European Account Number is an indicator digit used to verify that the DEAN is consistent with the type of account holder it is associated with. The indicator digit must be equal to "0" or "1".
8	M	The function must provide the execution status. In case of rejection, a reason code must be provided.

1.4.1.5.2. Interface description

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

1.4.1.5.2.1. 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: "DEAN validity check"	STR	35	M	EventIdentification
Date of the message	Date time of the message creation. YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Identification of the beta digital euro account.	STR	18	M	(New for beta digital euro)

Outgoing message

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



Data	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:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
PSP identification	Unique identifier of the pilot PSP servicing the beta digital euro account (BIC). Populated if the pilot PSP requests a PSP ID look-up to DESP after DEAN validity check processing.	STR	11	O	BICFI
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

1.4.1.5.2.2. *Return code*

#	Description
0	Successful
1	Failure

1.4.1.5.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

1.4.1.6. *DEAN registration*

The beta digital euro account is created only if the following conditions are met:

- For a new client: The client has been registered in the system, and the initial mandatory checks have been successfully completed. The client is also registered as a beta digital euro end user (end user), and no beta digital euro account has already been opened.
- For an already known client: The client is also registered as an end user, and no beta digital euro account has already been opened.

Once these steps are completed, the beta digital euro account can be opened and activated. The DEAN registration is performed through a request sent to DESP.



Digital euro pilot - Backend implementation specifications → "POST/ deans – Request" | "POST/ deans – Response"

When a DEAN is generated by DESP, data must be stored in the pilot PSP's system.

#	Mandatory Optional Conditional	Business rules description
1	M	The following data must be created: <ul style="list-style-type: none"> - DEAN - DEAN creation date - DEAN status
2	M	DEAN creation date must be equal to the current date.
3	M	The status of the digital euro access number must be equal to "OPEN".
4	M	The beta digital euro account must be associated with the end user.
5	M	The end user status must be updated and switched from "PENDING" to "ACTIVE".

1.4.1.7. DEAN deregistration and pilot PSP mapping removal

As part of the beta digital euro account closure and end user offboarding process, the mapping between the DEAN and the pilot PSP must be removed. The DEAN registration and pilot PSP mapping are performed through a request sent to DESP.

Digital euro pilot - Backend implementation specifications → "POST/ deans-deactivations – Request" | "POST/ deans-deactivations – Response"

Once the DESP has confirmed the mapping removal, the pilot PSP must update the beta digital euro account status.

#	Mandatory Optional Conditional	Business rules description
1	M	The following data must be updated: <ul style="list-style-type: none"> - DEAN closing date - DEAN status
2	M	DEAN closing date must be equal to the current date.
3	M	The status of the digital euro access number must be equal to "CLOSED".
4	M	The transactions carried out on the beta digital euro account must be retained and archived in compliance with applicable legislation.



1.5. 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.
	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 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 look-up	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 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.

1.5.1. Functions description

1.5.1.1. *Linked account settings request validation*

1.5.1.1.1. *Pre-requisite*

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

1.5.1.1.2. *Requirements*

The end user must be able to manage the link between a commercial bank money account and a beta digital euro account directly from their own device. The corresponding request is received by the pilot PSP servicing the beta digital euro account, which must verify that the request is valid and consistent.



#	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 User ID, the distributing PSP must find the beta digital euro account associated with the end user.
4	M	The request must contain the type of request: <ul style="list-style-type: none"> - CREATE → Initial set-up - UPDATE → Set-up amendment - DELETE → Set-up removal
5	M	A "CREATE" request is valid only if there isn't any existing link between the beta digital euro account and a commercial bank money account.
6	M	An "UPDATE" request is valid only if there is already an existing link between the beta digital euro account and a commercial bank money account.
7	M	A "DELETE" request is valid only if there is already an existing link between the beta digital euro account and a commercial bank money account.
8	C	If the type of request is equal to "CREATE" or "UPDATE", the request must contain a valid IBAN. The IBAN can be held at the pilot PSP servicing the beta digital euro account or at another pilot PSP.
9	C	If the type of request is equal to "UPDATE" the IBAN or IBAN identifier provided in the request must be different from the one already linked to the beta digital euro account.
10	M	The function must provide a return code. In case of failure, a reason code must be provided.

1.5.1.1.3. Interface description

A dedicated interface received from an end user's device (incoming message) triggers the commercial bank money account link request validation. The function generates an outgoing message providing the result of the function execution. The same incoming message is used for linking, updating and removing a commercial bank money account.

1.5.1.1.3.1. 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 "Account Link settings request".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Type of request	Type of request sent by the end user device.	SSET	4	M	RequestType



Data element	Description	Type	Length	Presence indicator	Standardised name
	<ul style="list-style-type: none"> - CRED - CREATE - UPDT - UPDATE - DELT - DELETE 				
Commercial bank money account	Identification of the Commercial bank money account (IBAN).	STR	34	O	AccountIdentification

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:mm:ss.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

1.5.1.1.3.2. *Return code*

#	Description
0	Successful
1	Failure

1.5.1.1.3.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

1.5.1.2. *Linked account settings storage*

1.5.1.2.1. *Requirements*

Once the pilot PSP validates the linked account settings request initiated by the end user through device, the new or updated parameters are stored.



#	Mandatory Optional Conditional	Business rules description
1	M	The pilot PSP must store the linked account details provided by the 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 a deletion request is made, the linked account must be removed.
5	M	The function must provide a return code to confirm or reject the linked account set-up.

1.5.1.2.2. Interface description

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

1.5.1.2.2.1. Message structure

Incoming message

The incoming message is the request received from the device and validated by the linked account settings request validation function.

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:mm:ss.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

1.5.1.2.2.2. Return code

#	Description
0	Successful



1	Failure
---	---------

1.5.1.2.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

1.5.1.3. *Linked commercial bank money account ownership validation*

An end user who has successfully opened a beta digital euro account must have the ability link a commercial bank money account. The end user's pilot PSP servicing the commercial bank money account must verify and confirm the ownership of the commercial bank money account.

#	Mandatory Optional Conditional	Business rules description
1	M	Verifying the ownership of a commercial bank money account is an existing pilot PSP capability, so no changes are required to support the beta digital euro.
2	M	The function must provide a return code. In case of failure, a reason code must be provided.

1.5.1.4. *Linked account settings removal*

1.5.1.4.1. *Requirements*

When an end user initiates an offboarding request, any previously linked account must be removed. This dedicated step of the offboarding process is executed under the coordination of the pilot 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 identification of the commercial bank money account linked to the DEAN.
3	C	If the request contains the DEAN, the pilot PSP must identify the linked account.
4	M	The pilot PSP must remove the link between the DEAN and the commercial bank money account.
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.

1.5.1.4.1.1. *Message structure*

Incoming message

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



EUROPEAN CENTRAL BANK

EUROSYSTEM

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: "Linked account removal".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Commercial bank money account	Identification of the Commercial bank money account (IBAN).	STR	34	O	AccountIdentification
Digital euro access number	Identification of the beta digital euro account.	STR	18	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:mm:ss.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

1.5.1.4.1.2. *Return code*

#	Description
0	Successful
1	Failure

1.5.1.4.1.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.



1.5.1.5. Linked commercial bank money account look-up

During the execution of a payment transaction, the payer’s PSP or the payee’s PSP may be required to verify whether a current account is properly linked to the beta digital euro account. This check is required in case of waterfall, reverse waterfall mechanism and end user offboarding.

1.5.1.5.1. Requirements

The pilot PSP servicing the beta digital euro account must check that a commercial bank money account is linked to the beta digital euro account.

#	Mandatory Optional Conditional	Business rules description
1	M	The request format and content must be consistent.
2	M	The request must contain the DEAN.
3	M	The linked account indicator must be provided in the output message.
4	C	If a commercial bank money account is linked to the beta digital euro account, the linked account indicator must be set to ‘Yes’ (1) and the commercial bank money account identifier must be provided.
5	C	If there’s no commercial bank money account linked to the beta digital euro account, the linked account indicator must be set to “No” (0).
6	M	The function must provide a return code. In case of failure, a reason code must be provided.

1.5.1.5.2. Interface description

This function is triggered without any interface exchanged between the end user’s device and the pilot PSP. An internal request is generated by the pilot PSP servicing the beta digital euro account. The result of the linked commercial bank money account look-up function is generated through an outgoing message.

1.5.1.5.2.1. 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: "Linked account look-up"	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.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)



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:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Linked account indicator	Dedicated indicator defining if a commercial bank money account is linked to the beta digital euro account: <ul style="list-style-type: none"> ▪ 1 indicates that a link exists ▪ 0 indicates no link 	BOOL	1	M	<i>(New for beta digital euro)</i>
Commercial bank money account	Identification of the Commercial bank money account (IBAN).	STR	34	O	AccountIdentification

1.5.1.5.2.2. *Return code*

#	Description
0	Successful
1	Failure

1.5.1.5.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

1.5.1.6. *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.



1.5.1.6.1. 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).

1.5.1.6.2. Requirements

#	Mandatory Optional Conditional	Description
1	M	The function must check a commercial bank money account is linked to the beta digital euro account (section 1.5.1.5).
2	C	If a commercial bank money account is linked to the beta digital euro account, the function approves the transaction and returns the commercial bank money account identifier (IBAN).
3	C	If either a defunding amount or a funding amount is provided in the incoming message and if no commercial bank money account is linked to the beta digital euro account, the transaction must be discontinued.
4	M	The function must provide a return code. In case of failure, a reason code must be provided.

1.5.1.6.3. Interface description

This function is triggered without any interface exchanged between the end user's device and the pilot PSP. An internal request is generated by the pilot PSP servicing the beta digital euro account. The result of the linked commercial bank money account look-up function is generated through an outgoing message.

1.5.1.6.3.1. 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: "linked account check".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Identification of the beta digital euro account.	STR	18	M	(New for beta digital euro)
Funding amount	Funding transaction amount in the case of a waterfall combined transaction on	NUM	18	O	Amount



Data element	Description	Type	Length	Presence indicator	Standardised name
	the payer side or a manual liquidity management funding request.				
Defunding amount	Defunding transaction amount in the case of a reverse waterfall combined transaction on the payee side or a manual liquidity management defunding 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:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Commercial bank money account	Identification of the Commercial bank money account (IBAN).	STR	34	O	AccountIdentification
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

1.5.1.6.3.2. *Return code*

#	Description
0	Successful
1	Failure

1.5.1.6.3.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.



1.6. Commercial bank money account service

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

Service	Function	Description
Commercial bank money account service	Commercial bank money account balance look-up	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.

1.6.1. Functions description

1.6.1.1. Commercial bank money account balance look-up

1.6.1.1.1. Requirements

The pilot PSP retrieves the available balance on the commercial bank money account. It ensures there are enough funds to proceed with the transaction.

#	Mandatory Optional Conditional	Business rules description
1	M	Retrieving the balance of a commercial bank money account and verifying sufficient funds to process the transaction is an existing pilot PSP capability. No changes are needed to support the beta digital euro.
2	M	The function must provide a return code. In case of failure, a reason code must be provided.

1.6.1.1.2. Interface description

The pilot PSP retrieves the balance of a commercial bank money account and verifying sufficient funds to process the transaction. is an existing pilot PSP capability. No changes are needed to support the beta digital euro.



1.6.1.2. Commercial bank money account status check

The pilot PSP verifies the operational status of a commercial bank account. It ensures that the account is active, valid, and ready for combined transactions with a funding or defunding amount or a liquidity management transaction with a funding or defunding amount.

#	Mandatory Optional Conditional	Description
1	M	The pilot PSP servicing commercial bank money account verifies the operational status of a commercial bank account.
2	M	The function must provide a return code. In case of failure, a reason code must be provided.

1.6.1.2.1. Interface description

Verifying the status of a commercial bank money account to process the transaction is an existing pilot PSP capability, so no changes are required to support the beta digital euro.

1.6.1.3. Commercial bank money account balance check

1.6.1.3.1. Requirements

The pilot PSP 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.

#	Mandatory Optional Conditional	Business rules description
1	M	The pilot PSP servicing commercial bank money account verifies the current balance of a commercial bank money account.
2	M	The function must provide a return code. In case of failure, a reason code must be provided.

1.6.1.3.2. Interface description

Checking the commercial bank money account limits and checking the commercial bank money account balance is an existing capability within pilot PSP systems. No changes are required to support the beta digital euro.

1.7. End user deactivation

When an end user is offboarded, the end user identifier must be deactivated. The end user deactivation is performed by DESP, and the pilot PSP must invoke the corresponding service.



EUROPEAN CENTRAL BANK

EUROSYSTEM

Digital euro pilot - Backend implementation specifications → " POST /users-deactivations – request " | "POST/ users-deactivations – Response"



2. Liquidity Management Service

2.1. 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 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.

2.1.1.1. Beta digital euro account debit

2.1.1.1.1. Requirements

Once the defunding request sent to the DESP has been successfully validated - whether initiated through a manual defunding request, or a payment instruction - the pilot PSP must debit the beta digital euro account.

#	Mandatory Optional Conditional	Business rules description
1	M	The online holdings must be debited up to the requested amount.
2	M	The date of the debit must be stored to reflect the new balance date.
3	M	The pilot PSP sends the debited amount from the beta digital euro account and the operation result (successful / failed) to the notification service (section 4.2 – Notification Service). Note: The notification service is responsible for deciding whether to send notifications to the end user based on the end user's notification settings.
4	M	The function must provide the execution status. In case of failure, a reason code must be provided.

2.1.1.1.2. Interface description

2.1.1.1.2.1. Message structure

Incoming message

The pilot PSP must generate the incoming message containing the following data:



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 debit".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Identification of the beta digital euro account.	STR	18	O	(New for beta digital euro)
Beta digital euro account debited amount	Amount to be debited from the beta digital euro account.	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:mm:ss.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

2.1.1.1.2.2. *Return code*

#	Description
0	Successful
1	Failure

2.1.1.1.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.



2.1.1.2. Beta digital euro account credit

2.1.1.2.1. Requirements

Once the funding request sent to the DESP has been successfully validated—whether initiated through a manual funding request or a payment instruction response—the pilot PSP must credit the beta digital euro account.

#	Mandatory Optional Conditional	Business rules description
1	M	The online holdings must be credited up to the requested amount.
2	M	The date of the credit must be stored to reflect the new balance date.
3	M	The pilot PSP sends the credited amount on the beta digital euro account and the operation result (successful / failed) to the notification service (section 4.2 – Notification Service). Note: The notification service is responsible for deciding whether to send notifications to the end user based on the end user’s notification settings.
4	M	The function must provide the execution status. In case of failure, a reason code must be provided.

2.1.1.2.2. Interface description

2.1.1.2.2.1. Message structure

Incoming message

The pilot PSP must generate the incoming message containing the following data:

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 credit”.	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.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)
Beta digital euro account credited amount	Amount to be credited to the beta digital euro account.	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:mm:ss.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

2.1.1.2.2.2. *Return code*

#	Description
0	Successful
1	Failure

2.1.1.2.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

2.2. 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 end user's pilot PSP. It offers the needed functions to handle the funds located in the commercial bank money account.

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.
	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.



Service	Function	Description
	Funds release	The pilot PSP servicing commercial bank money account releases funds blocked on commercial bank money account if the process is discontinued.

2.2.1. Functions description

2.2.1.1. Commercial bank money account debit

2.2.1.1.1. Requirements

The pilot PSP debits the commercial bank money account upon pilot PSP request and sends the debited amount to the notification service.

#	Mandatory Optional Conditional	Business rules description
1	M	The pilot PSP debits the commercial bank money account
2	M	The payee PSP sends the amount credited on the commercial bank money account and the operation result (successful / failed) to the notification service (section 4.2 – Notification Service). Note: The notification service is responsible for deciding whether to send notifications to the end user based on the end user's notification settings.

2.2.1.1.2. Interface description

Debiting a commercial bank money account is an existing capability within pilot PSP systems. No changes are required to support the beta digital euro.

2.2.1.2. Commercial bank money account credit

2.2.1.2.1. Requirements

The pilot PSP credits the commercial bank money account and sends the debited amount to the notification service.

#	Mandatory Optional Conditional	Business rules description
1	M	The Pilot PSP credits the commercial bank money account.
2	M	The payee PSP sends the amount credited on the commercial bank money account and the operation result (successful / failed) to the notification service (section 4.2 – Notification Service). Note: The notification service is responsible for deciding whether to send notifications to the end user based on the end user's notification settings.



2.2.1.2.2. *Interface description*

Crediting a commercial bank money account is an existing capability within pilot PSP systems. No changes are required to support the beta digital euro.

2.2.1.3. *Funds blocking*

2.2.1.3.1. *Requirements*

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.

#	Mandatory Optional Conditional	Business rules description
1	M	Funds blocking on a commercial bank money account is an existing capability within pilot PSP systems. No changes are required to support the digital euro.
2	M	The function must provide a return code. In case of failure, a reason code must be provided.

2.2.1.3.2. *Interface description*

Funds blocking on a commercial bank money account is an existing capability within pilot PSP systems. No changes are required to support the beta digital euro.

2.2.1.4. *Funds release*

2.2.1.4.1. *Requirements*

The pilot PSP releases funds blocked on commercial bank money account if the process is discontinued.

#	Mandatory Optional Conditional	Business rules description
1	M	The pilot PSP servicing commercial bank money account releases funds blocked on commercial bank money account.
2	M	The function must provide a return code. In case of failure, a reason code must be provided.

2.2.1.4.2. *Interface description*

Funds release from a commercial bank money account is an existing capability within pilot PSP systems. No changes are required to support the beta digital euro.



2.3. Funding - Defunding settlement processing service

Funding / defunding settlement processing service is invoked in manual 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.
	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.

2.3.1. Functions description

2.3.1.1. Funding online settlement request creation

The pilot PSP sends a request with all funding settlement data. DESP will confirm the reception of the request.

2.3.1.1.1. Requirements

#	Mandatory Optional Conditional	Business rules description
1	M	The distributing PSP sends a request with funding settlement data (Funding object). The funding online settlement request format and content must be consistent with the DESP requirements. Digital euro pilot - Backend implementation specifications → POST /fundings – Request
2	M	DESP must confirm the reception of the request and provides the associated {id}. Digital euro pilot - Backend implementation specifications → POST /fundings – Response The {id} must be kept until the status is received following settlement by DESP. The {id} must be associated with the corresponding funding object.



#	Mandatory Optional Conditional	Business rules description
3	C	If the DESP doesn't confirm the reception of the request within a predefined timeframe the process must be discontinued. Digital euro pilot - Backend implementation specifications
4	M	The function must provide the execution status. In case of failure, a reason code must be provided.

2.3.1.2. Funding online settlement processing

After the funding settlement has been processed, DESP sends the status of the settlement to the distributing PSP, and the distributing PSP confirms the receipt of status.

The “funding online settlement processing” function is triggered when the pilot PSP receives a funding settlement notification in the following case:

- after a funding confirmation notification
- after a funding rejection notification

This function is also triggered by the pilot PSP when the funding settlement notification is not received (time out management)

2.3.1.2.1. Requirements

DESP provides a settlement notification to the pilot PSP

#	Mandatory Optional Conditional	Description
1	M	The pilot PSP receives from DESP a notification of funding settlement together with the ID of the corresponding initial online funding request. The funding online settlement request format and content must be consistent with the DESP requirements Digital euro pilot - Backend implementation specifications → POST /fundings/{funding-id}/status - Request The pilot PSP must use this {id} to retrieve the initial data of the funding object.
2	M	The pilot PSP must confirm the reception of the request. Digital euro pilot - Backend implementation specifications → POST /fundings/{funding-id}/status - Response



3	C	If the format is incorrect, the process is discontinued. The pilot PSP must then submit a dysfunction report to DESP and wait for an update from DESP before manually taking over the process.
4	C	In case of successful settlement, the pilot PSP triggers the Funding post settlement process with a settlement confirmation status = successful (section 2.4.1.1).
5	C	In case of failure the pilot PSP triggers the Funding post settlement process with a settlement confirmation status = failed (section 2.4.1.1).
6	M	The function must provide the execution status. In case of failure, a reason code must be provided.

Settlement notification is not received (time out management)

If DESP does not provide a settlement confirmation within a predefined timeline the process is discontinued.

Digital euro pilot - Backend implementation specifications. The pilot PSP must then submit a dysfunction report to DESP and wait for an update from DESP before manually taking over the process.

2.3.1.3. Defunding online settlement request creation

The distributing PSP sends a request with all defunding settlement data. DESP will confirm the reception of the request.

2.3.1.3.1. Requirements

#	Mandatory Optional Conditional	Business rules description
1	M	The distributing PSP sends a request with defunding settlement data (defunding object). The defunding online settlement request format and content must be consistent with the DESP requirements. Digital euro pilot - Backend implementation specifications → POST /defundings – Request
2	M	DESP must confirm the reception of the request by providing associated {id}. Digital euro pilot - Backend implementation specifications → POST /defundings – Response The {id} must be kept until the status is received following settlement by DESP. The {id} must be associated with the corresponding defunding object.
3	C	If the DESP doesn't confirm the reception of the request within a predefined timeframe the process must be discontinued. Digital euro pilot - Backend implementation specifications
4	M	The function must provide the execution status. In case of failure, a reason code must be provided.



2.3.1.4. Defunding online settlement processing

DESP sends the defunding settlement status to the distributing PSP, and the distributing PSP acknowledges receipt of this status.

The “defunding online settlement processing” function is triggered when the PSP receives a DESP settlement notification in the following case:

- after a defunding confirmation notification
- after a defunding rejection notification

This function is also triggered by the pilot PSP when the defunding settlement notification is not received (time out management).

2.3.1.4.1. Requirements

DESP provides a settlement notification to the pilot PSP

#	Mandatory Optional Conditional	Description
1	M	The pilot PSP receives from DESP a notification of defunding settlement together with the ID of the corresponding initial online defunding request. The defunding online settlement request format and content must be consistent with the DESP requirements. Digital euro pilot - Backend implementation specifications → POST /defundings/{defunding-id}/status - Request The pilot PSP must use this {id} to retrieve the initial data of the defunding object.
2	M	The pilot PSP must confirm the reception of the request (notification). Digital euro pilot- Backend implementation specifications → POST /defundings/{defunding-id}/status - Response
3	C	If the format is incorrect, the process is discontinued. The pilot PSP must then submit a dysfunction report to DESP and wait for an update from DESP before manually taking over the process.
4	C	In case of successful settlement, the pilot PSP triggers the Defunding post settlement process with a settlement confirmation status = successful (section 2.4.1.2).
5	C	In case of failure, the pilot PSP triggers the Defunding post settlement process with a settlement confirmation status = failed (section 2.4.1.2).
6	M	The function must provide the execution status. In case of failure, a reason code must be provided.

Settlement notification is not received (time out management)

If DESP does not provide a settlement confirmation within a predefined timeline the process is discontinued. **Digital euro pilot - Backend implementation specifications**. The pilot PSP must then submit a dysfunction report to DESP and wait for an update from DESP before manually taking over the process.



2.4. Funding – Defunding post settlement service

Funding – Defunding post settlement service is invoked during a funding or defunding process by the distributing PSP of the individual end user to close the funding or defunding process post settlement.

The service can also be invoked in the context of a combined payment instruction transaction that includes the data required for the funding or the defunding.

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.

2.4.1. Functions description

2.4.1.1. Funding post settlement

2.4.1.1.1. Requirements

This function analyses the outcome of the funding settlement process sent by DESP and determines the steps required to close the funding process.

Pilot PSP offering liquidity management services (beta digital euro account)

#	Mandatory Optional Conditional	Business rules description
1	C	<p>If the funding settlement is successful, the distributing PSP servicing the beta digital euro account must retrieve the transaction amount (or funding amount provided in a payment transaction) in order to determine the exact amount to credit to the beta digital euro account.</p> <p>The funding amount must be used to populate the “beta digital euro account credited amount” data in the outgoing message.</p> <p>The next step for the pilot PSP to finalise the transaction lifecycle is:</p> <ul style="list-style-type: none"> Beta digital euro account funds management service / Beta digital euro account credit (section 2.1.1.2).
2	C	If the funding settlement is failed, the funding request is discontinued.



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

Pilot PSP offering liquidity management services (commercial bank money)

#	Mandatory Optional Conditional	Description
1	C	<p>If the funding settlement is successful, the pilot PSP servicing the commercial account must retrieve the transaction amount in order to determine the exact amount to release the funds previously blocked and debit the commercial bank money account.</p> <p>Note: In the event that the funding amount has already been recovered, this step is not necessary.</p> <p>The funding amount must be used to populate the “Funds release” and the “Commercial bank money account debited amount” data in the outgoing message.</p> <p>The next steps for the pilot PSP to finalise the transaction life cycle are:</p> <ul style="list-style-type: none">• Commercial bank money account funds release (section 2.2.1.4)• Commercial bank money account debit (section 2.2.1.1).
2	C	<p>If the funding settlement is failed, the pilot PSP servicing the commercial account must retrieve the transaction amount in order to determine the exact amount to release the funds previously blocked.</p> <p>Note: In the event that the funding amount has already been recovered, this step shall not be necessary.</p> <p>The funding amount must be used to populate the “Funds release” data in the outgoing message. The next step for the pilot PSP to finalise the transaction lifecycle is:</p> <ul style="list-style-type: none">• Commercial bank money account funds release (section 2.2.1.4).
3	M	The function must provide the execution status. In case of failure, a reason code must be provided.

2.4.1.1.2. *Interface description*

2.4.1.1.2.1. *Message structure*

Incoming message

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



EUROPEAN CENTRAL BANK

EUROSYSTEM

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: "Funding post settlement".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Settlement confirmation status	Indicator specifying whether the DESP settlement confirmation status is successful or failed. 0 – successful 1 - failed	BOOL	1	M	TransactionStatus
Funding amount	Amount to credit.	NUM	18	M	Amount
Digital euro access number	Identification of the beta digital euro account.	STR	18	M	(New for beta digital euro)
Commercial bank money account	Identification of the commercial bank money account (IBAN).	STR	34	M	AccountIdentification

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:mm:ss.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)
Commercial bank money account	Identification of the commercial bank money account (IBAN).	STR	34	M	AccountIdentification
Beta digital euro account credited amount	Amount to be credited on the beta digital euro account amount. Populated only if the funding settlement is successful.	NUM	18	O	Amount



Data element	Description	Type	Length	Presence indicator	Standardised name
Funds release	Amount to be released from the commercial bank money account.	NUM	18	M	Amount
Commercial bank money account debited amount	Amount to be debited from the commercial bank money account. Populated only if the funding settlement is successful.	NUM	18	O	Amount
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

2.4.1.1.2.2. *Return code*

#	Description
0	Successful
1	Failure

2.4.1.1.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

2.4.1.2. *Defunding post settlement*

2.4.1.2.1. *Requirements*

This function analyses the outcome of the defunding settlement process sent by DESP and determines the steps required to close the defunding process.

Pilot PSP offering liquidity management services (beta digital euro account)

#	Mandatory Optional Conditional	Business rules description
1	C	If the defunding settlement is successful, the distributing PSP servicing the beta digital euro account must retrieve the transaction amount (or defunding amount provided in a payment transaction) in order to determine the exact amount to debit from the beta digital euro account. The defunding amount must be used to populate the "beta digital euro account debited " data in the outgoing message.



#	Mandatory Optional Conditional	Business rules description
		The next step for the pilot PSP to finalise the transaction lifecycle is: <ul style="list-style-type: none">Beta digital euro account funds management service / Beta digital euro account debit (section 2.1.1.1).
2	C	If the defunding settlement is failed, the defunding request is discontinued.
3	M	The function must provide the execution status. In case of failure, a reason code must be provided.

Pilot PSP offering liquidity management services (commercial bank money account)

#	Mandatory Optional Conditional	Description
1	C	If the defunding settlement is successful, the pilot PSP servicing the commercial account must retrieve the transaction amount in order to determine the exact amount to credit to the commercial bank money account. Note: In the event that the defunding amount has already been recovered, this step shall not be necessary. The defunding amount must be used to populate the “commercial bank money account credited amount” data in the outgoing message. The next step for the pilot PSP to finalise the transaction lifecycle is: <ul style="list-style-type: none">Commercial bank money account fund management service / Commercial bank money account credit.
2	C	If the defunding settlement is failed, the defunding request is discontinued.
3	M	The function must provide the execution status. In case of failure, a reason code must be provided.

2.4.1.2.2. *Interface description*

2.4.1.2.2.1. *Message structure*

Incoming message

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



EUROPEAN CENTRAL BANK

EUROSYSTEM

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: "Defunding post settlement".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Settlement confirmation status	Indicator specifying whether the DESP settlement confirmation status is successful or failed. 0 – successful 1 - failed	BOOL	1	M	TransactionStatus
Funding amount	Amount to credit.	NUM	18	M	Amount
DEAN	Identification of the beta digital euro account.	STR	18	M	(New for beta digital euro)
Commercial bank money account	Identification of the commercial bank money account (IBAN).	STR	34	M	AccountIdentification

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:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Identification of the beta digital euro account.	STR	18	M	(New for beta digital euro)
Commercial bank money account	Identification of the commercial bank money account (IBAN).	STR	34	M	AccountIdentification
Beta digital euro account debited amount	Amount to be debited from the beta digital euro account amount. Populated only if the defunding settlement is successful.	NUM	18	O	Amount
Commercial bank money	Amount to be credited to the commercial bank money	NUM	18	O	Amount



Data element	Description	Type	Length	Presence indicator	Standardised name
account credited amount	account. Populated only if the defunding settlement is successful.				
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

2.4.1.2.2.2. *Return code*

#	Description
0	Successful
1	Failure

2.4.1.2.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.



3. 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 in scope for the pilot are covered.

3.1. Amount assessment service

Service	Function/sub-functions	Description
Amount assessment service	Pre-settlement payment Funding-defunding amount assessment	This function is invoked prior to settlement to check that the conditions for execution are met – and to determine whether it is a simple transaction or a combined transaction.
	Post settlement payment Funding-defunding amount assessment	This function is invoked after the settlement and calculates the amounts to be credited or debited on the beta digital euro account, as well as any incremental amount to be defunded, which would require initiating a new defunding transaction on the beta digital euro account.
	Pre-settlement payment refund - defunding amount assessment	This function is invoked in case of refund transaction prior to settlement to check that the conditions for execution are met – and to determine whether it is a simple transaction or a combined transaction.
	Post settlement refund - defunding amount assessment	This function is invoked after the settlement of refund transaction and calculates the amounts to be credited or debited on the beta digital euro account, as well as any incremental amount to be defunded, which would require initiating a new defunding transaction on the beta digital euro account.

3.1.1. Functions description

3.1.1.1. Pre-settlement payment - funding defunding amount assessment

3.1.1.1.1. Requirements

Executing a payment transaction requires verifying, prior to settlement, that the conditions for execution are met - particularly determining whether it is a simple transaction or one that involves handling a waterfall and/or reverse waterfall.

This feature serves in a payment use case involving a payer's beta digital euro account and a payee's beta digital euro account.

This function is used regardless whether the end user is an individual or a business end user.

**The pilot PSP is the payer PSP**

#	Mandatory Optional Conditional	Business rules description
1	M	The pilot PSP must determine if a reverse waterfall is required.
2	M	The pilot PSP must calculate the difference between the available balance and the transaction amount. The available balance and the transaction amount are provided in the incoming message.
3	C	If the available balance covers the transaction amount (beta digital euro account available balance - transaction amount $\geq 0\text{€}$): <ul style="list-style-type: none"> • The transaction is a payment transaction that does not require a reverse waterfall. • The funding amount data is not provided in the outgoing message. • “Beta digital euro account amount debit” data is populated in the outgoing message with the transaction amount. <p>The next step for the PSP to finalise the transaction lifecycle is the debit of the beta digital euro account (section 2.1.1.1).</p>
4	C	If the available balance does not cover the transaction amount (beta digital euro account available balance - transaction amount $< 0\text{€}$): <ul style="list-style-type: none"> • The transaction is a payment transaction that can only be performed with a reverse waterfall. • The payer PSP must calculate the funding amount to be provided in the outgoing message: <ul style="list-style-type: none"> ➔ Funding amount = transaction amount – beta digital euro account available balance) • “Beta digital euro account amount debit” data is populated in the outgoing with the beta digital euro account available balance. <p>The next step for the pilot PSP is the debit of the beta digital euro account (section 2.1.1.1).</p>
5	M	The function must provide the return code. In case of failure, a reason code must be provided.

The pilot PSP is the payee PSP

#	Mandatory Optional Conditional	Description
1	C	If the payee is a business end user, the transaction is always a payment transaction that requires a waterfall. The defunding amount must be populated in the outgoing message. ➔ defunding amount = transaction amount
2	C	If the payee is an individual end user, the pilot PSP must determine if a waterfall is required by checking if the holding limit is exceeded (beta digital euro account available balance + transaction amount is compared to holding limit). The available balance, the transaction amount and the holding limit are provided in the incoming message.



#	Mandatory Optional Conditional	Description
3	C	If the holding limit of the individual end user is exceeded (beta digital euro account available balance + transaction amount > holding limit). <ul style="list-style-type: none"> The transaction is a payment transaction that can only be performed with a waterfall <ul style="list-style-type: none"> The payee PSP must calculate the defunding amount and populate the outgoing message as follows: <ul style="list-style-type: none"> → defunding amount = beta digital euro account available balance + transaction amount – holding limit
4	C	If the holding limit of the individual end user is not exceeded (beta digital euro account available balance + transaction amount ≤ holding limit) and there aren't any potential concurrent transactions. <ul style="list-style-type: none"> The transaction is a single transaction on payee side. The defunding amount is not provided in the outgoing message.
5	C	If the holding limit of the individual end user is not exceeded (beta digital euro account available balance + transaction amount ≤ holding limit) and at least one potential concurrent transaction awaiting for settlement confirmation and making the balance exceed the holding limits: <ul style="list-style-type: none"> The transaction is a payment transaction that can only be performed with a waterfall (the holding limit is already exceeded because at least one potential concurrent transaction awaiting settlement confirmation has already pushed the balance beyond the limit): <ul style="list-style-type: none"> → defunding amount = transaction amount
6	M	The function must provide the return code. In case of failure, a reason code must be provided.

3.1.1.1.2. Interface description

3.1.1.1.2.1. Message structure

Incoming message

The incoming message is generated by the pilot 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: "Pre-settlement amount assessment".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	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 of the transaction.	NUM	18	M	Amount
Beta digital euro account balance	Beta digital euro account balance.	NUM	18	O	Balance
Holding limit	Maximum amount of beta digital euro that an individual end user is authorised to hold. Required only for an individual end user.	NUM	18	O	<i>(New for beta digital euro)</i>

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:mm:ss.sssZ	Datetime UTC	24	M	CreationDateTime
Beta digital euro account amount debit	Amount to be debited from the beta digital euro account.	NUM	18	O	Amount
Funding amount	Funding amount in case of reverse waterfall combined transaction on payer side.	NUM	18	O	Amount
Defunding amount	Defunding amount in case of waterfall combined transaction on payee side.	NUM	18	O	Amount
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



3.1.1.1.2.2. *Return code*

#	Description
0	Successful
1	Failure

3.1.1.1.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

3.1.1.2. *Post settlement payment - funding defunding amount assessment*

Following the reception of the notification by the DESP upon settlement, it is necessary to identify the steps required to complete the transaction processing. This function is invoked by the pilot PSP and calculates the amounts to be credited or debited on the beta digital euro account, as well as any incremental amount to be defunded, which would require initiating a new defunding transaction on the beta digital euro account.

3.1.1.2.1. *Requirements*

The pilot PSP is the payer PSP

#	Mandatory Optional Conditional	Business rules description
1	M	The incoming message generated by the pilot PSP must be consistent and contain the funding amount.
2	C	If the settlement is successful, (settlement confirmation status = 0 in the incoming message) and if "Funding amount" is provided in the incoming message, the outgoing message is populated as followed: <ul style="list-style-type: none"> Beta digital euro account debited amount = Funding amount The next step for the pilot PSP to finalise the transaction lifecycle is the debit of the beta digital euro account (section 2.1.1.1).
3	C	If the settlement is failed, (settlement confirmation status = 1 in the incoming message) the beta digital euro account must be credited. The pilot PSP must populate the outgoing message as followed: <ul style="list-style-type: none"> Beta digital euro account amount credit = Transaction amount – Funding amount (if populated in the incoming message) The next step for the pilot PSP to finalise the transaction lifecycle is the credit of the beta digital euro account (section 2.1.1.2).
4	M	The function must provide the 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 incoming message generated by the pilot PSP must be consistent and contain the defunding amount.
2	C	If the settlement is successful, (settlement confirmation status = 0 in the incoming message), the outgoing message is populated as follows: <ul style="list-style-type: none"> Beta digital euro account credited amount = Transaction amount The next step for the pilot PSP to finalise the transaction life cycle is the credit of the digital euro account (section 2.1.1.2).
3	C	If the settlement is successful (settlement confirmation status = 0 in the incoming message) and if there is at least one potential concurrent transaction. The payee PSP calculates the incremental amount to be defunded: <ul style="list-style-type: none"> Incremental amount to be defunded = $\max(\text{Balance} + \sum(\text{each concurrent transaction amount}) - \text{holding limit} - \sum(\text{each concurrent transaction's already defunded amount}); 0)$ If incremental amount to be defunded = 0€, incremental amount is not provided in the response. Note: Other concurrent transactions are not included in the incoming message; therefore, the pilot PSP must consult the transaction history and identify any unsettled transactions. The next step for the pilot PSP must trigger the funding / Defunding settlement process for the incremental amount to be defunded (section 2.3.1.1).
4	M	The function must provide the return code. In case of failure, a reason code must be provided.

3.1.1.2.2. Interface description

This function is activated after settlement confirmation processing (**section 3.2.1.5**).

3.1.1.2.2.1. Message structure

Incoming message

The incoming message is generated by the payee 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: "Post Settlement amount assessment"	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Identification of the beta digital euro account.	STR	18	M	AccountIdentification



Data element	Description	Type	Length	Presence indicator	Standardised name
Settlement confirmation status	Indicator specifying whether the DESP settlement confirmation status is successful or failed. 0 – successful 1 - failed	BOOL	1	M	TransactionStatus
Transaction amount	Amount of the transaction.	NUM	18	M	Amount
Funding amount	Funding amount in case of reverse waterfall combined transaction on payer side.	NUM	18	O	Amount
Defunding amount	Defunding amount in case of waterfall combined transaction on payee side.	NUM	18	O	Amount
Digital euro account balance	Populated if the function execution is successful.	NUM	18	M	Balance
Holding limit	Maximum amount of beta digital euro that can be held by an individual end user.	NUM	18	M	<i>(New for beta digital euro)</i>

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:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Beta digital euro account debited amount	The amount to be debited from the beta digital euro account.	NUM	18	O	Amount
Beta digital euro account credited amount	The amount to be credited to the beta digital euro account.	NUM	18	O	Amount
Incremental amount to	The incremental amount to be defunded from the beta digital euro account.	NUM	18	O	Amount



Data element	Description	Type	Length	Presence indicator	Standardised name
be defunded					
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

3.1.1.2.2.2. *Return code*

#	Description
0	Successful
1	Failure

3.1.1.2.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

3.1.1.3. *Pre-settlement payment refund – funding defunding amount assessment*

3.1.1.3.1. *Requirements*

Executing a refund transaction requires verifying, prior to settlement, that the conditions for execution are met—particularly determining whether it is a single transaction or one that involves handling a waterfall for the former payer that is a new payee in the process of refund.

This feature applies to payment scenarios where a former payer’s beta digital euro account is refunded and subsequently becomes a new payee.

The payer PSP (new payee)

#	Mandatory Optional Conditional	Business rules description
1	M	The new payee (former payer) is an individual end user.
2	M	The pilot PSP must determine if a waterfall is required by checking if the holding limit is exceeded (beta digital euro account available balance + refund amount is compared to holding limit). The available balance, the refund amount and the holding limit are provided in the incoming message.



#	Mandatory Optional Conditional	Business rules description
3	C	<p>If the holding limit of the individual end user is exceeded (beta digital euro account available balance + transaction amount > holding limit):</p> <ul style="list-style-type: none"> • The refund transaction is a refund transaction that requires a waterfall mechanism. • The former payer PSP must calculate the defunding amount → defunding amount = beta digital euro account available balance + refund amount – holding limit • The defunding amount thus determined is incorporated in the defunding amount of the outgoing message.
4	C	<p>If the holding limit of the individual end user is not exceeded (beta digital euro account available balance + transaction amount ≤ holding limit) and there aren't any potential concurrent credit transactions exceeding the holding limit.</p> <ul style="list-style-type: none"> • The refund transaction is a single refund transaction on payee side. • The defunding amount is not provided in the outgoing message.
5	C	<p>If the holding limit is not exceeded (beta digital euro account available balance + transaction amount ≤ holding limit) and at least one potential concurrent credit transaction exceeding holding limit is waiting for settlement confirmation:</p> <ul style="list-style-type: none"> • The refund transaction requires a waterfall (the holding limit is already exceeded because, at least, one potential concurrent transaction awaiting settlement confirmation has already pushed the balance beyond the limit). • The defunding amount is incorporated in the outgoing message and is equal to transaction amount.
6	M	The function must provide the return code. In case of failure, a reason code must be provided.

The payee PSP (new payer)

It is not necessary to assess the amount for the refund as it has been provided in the refund request sent by the business end user to the pilot PSP.

A credit corresponding to the full refund amount must be posted to the business account to ensure the sufficient balance (reverse waterfall mechanism).

3.1.1.3.2. Interface description

3.1.1.3.2.1. Message structure

Incoming message

The incoming message is generated by the pilot 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



Data element	Description	Type	Length	Presence indicator	Standardised name
Identifier of the event	Unique identifier of the event that triggers the message: "Refund Pre-settlement assessment".	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Identification of the beta digital euro account.	STR	18	M	<i>(New for beta digital euro)</i>
Refund Amount	Amount of the refund transaction.	NUM	18	M	Amount
Beta digital euro account balance	Available balance. Required only for an individual end user.	NUM	18	O	Balance
Holding limit	Maximum amount of beta digital euro that can be held by an individual end user. Required only for an individual end user.	NUM	18	O	<i>(New for beta digital euro)</i>

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:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Defunding amount	Defunding amount in case of waterfall combined transaction on payee side.	NUM	18	O	Amount



Data element	Description	Type	Length	Presence indicator	Standardised name
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

3.1.1.3.2.2. *Return code*

#	Description
0	Successful
1	Failure

3.1.1.3.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.

3.1.1.4. *Post settlement refund – funding defunding amount assessment*

Following the reception of the notification by the DESP upon settlement, it is necessary to identify the steps required to complete the transaction processing.

This function is invoked by the pilot PSP servicing the beta digital euro account and calculates the amounts to be credited or debited on the beta digital euro account, as well as any incremental amount to be defunded, which would require initiating a new defunding transaction on the beta digital euro account.

This feature applies to refund scenarios in which:

- a former payer’s beta digital euro account is refunded and subsequently becomes a new payee;
- a former payee’s beta digital euro account refunds a payment transaction (in full or in part) and subsequently becomes a new payer.

3.1.1.4.1. *Requirements*

The payee PSP (new role as a payer)

#	Mandatory	Business rules description
	Optional	
	Conditional	



1	C	<p>If the pilot PSP receives a settlement confirmation notification (settlement confirmation status = 0 in the incoming message), the pilot PSP must define the amount to be debited from the digital euro account and generate the outgoing message accordingly:</p> <ul style="list-style-type: none"> • “Beta digital euro (online) account debited amount” is populated with the refund amount specified in the incoming message. <p>The next step for the pilot PSP to finalise the transaction life cycle is the debit of the beta digital euro account (section 2.1.1.1).</p>
2	C	<p>If the pilot PSP receives a settlement rejection notification (settlement confirmation status = 1 in the incoming message), the process must be discontinued.</p>
3	M	<p>The function must provide the return code. In case of failure, a reason code must be provided.</p>

The payer PSP (new role as a payee)

#	Mandatory Optional Conditional	Description
1	C	<p>If the settlement confirmation is successful (settlement confirmation status = 0 – successful in the incoming message), the pilot PSP must define the amount to be credited to the beta digital euro account and generate the outgoing message accordingly.</p> <ul style="list-style-type: none"> ▪ “Beta digital euro account credited amount” is populated with the refund amount specified in the incoming message. <p>The next step for the pilot PSP is to finalise the transaction life cycle is the credit of the digital euro account (section 2.1.1.2).</p>
2	C	<p>If the pilot PSP receives a settlement confirmation notification (settlement confirmation status = 0 in the incoming message), and if there is at least one potential concurrent credit transaction (others refund or funds received in P2P transactions).</p> <p>The payee PSP calculates the incremental amount to be defunded and generate the outgoing message as followed:</p> <ul style="list-style-type: none"> • “Beta digital euro account credited amount” of the incoming message populates the “Refund amount” of the outgoing message. • “Incremental amount to be defunded” is determined according to the following formula: $\max(\text{Balance} + \sum(\text{each concurrent transaction amount}) - \text{holding limit} - \sum(\text{each concurrent transaction's already defunded amount}); 0)$. • If the result > 0€, “Incremental amount to be defunded” is populated with the result • If incremental amount to be defunded = 0€, Incremental amount to be defunded is not provided in the outgoing message. <p>Note: Other concurrent transactions are not included in the incoming message; therefore, the pilot PSP must consult the transaction history and identify any unsettled transactions.</p>
3	M	<p>The function must provide the return code. In case of failure, a reason code must be provided.</p>



3.1.1.4.2. *Interface description*

This function is activated after settlement confirmation processing (**section 3.3.1.3**).

3.1.1.4.2.1. *Message structure*

Incoming message

The incoming message is generated by the former payer 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 : “Refund Post Settlement assessment”.	STR	35	M	EventIdentification
Date of the message	Date time of the message creation YYYY-MM-DDThh:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
DEAN	Identification of the beta digital euro account.	String	18	M	(New for beta digital euro)
Settlement confirmation status	Indicator specifying whether the DESP settlement confirmation status is successful or failed. 0 – successful 1 - failed	Boolean	1	M	TransactionStatus
Refund amount	Amount of the refund transaction.	Number	18	M	Amount
Defunding amount	Defunding amount in case of waterfall combined transaction on payee side.	Number	18	O	Amount
Digital euro account balance	Populated if the function execution is successful.	Number	18	M	Balance
Holding limit	Maximum amount of beta digital euro that can be held by an individual end user	Number	18	M	(New for beta digital euro)

Outgoing message



EUROPEAN CENTRAL BANK

EUROSYSTEM

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:mm:ss.sssZ	DATETIME UTC	24	M	CreationDateTime
Beta Digital euro account debited amount	The amount to be debited from the beta digital euro account.	NUM	18	O	Amount
Beta digital euro account credited amount	The amount to be credited to the beta digital euro account.	NUM	18	O	Amount
Incremental amount to be defunded	The incremental amount to be defunded from the beta digital euro account.	NUM	18	O	Amount
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

3.1.1.4.2.2. *Return code*

#	Description
0	Successful
1	Failure

3.1.1.4.2.3. *Functional error description (reason code)*

The functional error descriptions are listed in **section 4.2 - Notification Service**.



3.2. Payment processing service

Payment processing service is invoked during a payment transaction process, after payment initiation, in coordination with DESP. Two main types of interactions with DESP can be identified:

- Payment requests sent the payee's PSP as initial request sent to the payer's PSP.
- Payment instructions sent by the payer's PSP as a first step of the settlement process.

All interactions with the DESP adhere to this categorisation.

In addition to this categorisation, the interactions to DESP manages specificities of "On Us" payments.

Service	Function/sub-functions	Description
Payment processing service	Payment request creation	The payee's PSP servicing the beta digital euro account generates a payment request containing only payment data to DESP.
	Payment request validation	The payer's PSP receives a payment request from DESP and validates its consistency.
	Payment instruction creation	The payer's PSP sends the request with partial settlement data and funding data if a reverse waterfall mechanism is required.
	Payment instruction validation	The payee's PSP receives a payment instruction from DESP, validates its consistency and provides additional settlement data and defunding data if a waterfall mechanism is required.
	Payment settlement processing	The pilot PSP manages the lifecycle of a payment instruction in alignment with the settlement phase status.

3.2.1. Functions description

3.2.1.1. *Payment request creation*

The payee's PSP sends the payment request to DESP with only a payment object related to the authorisation details of the payment transaction.

The payment request format and content must be consistent with the DESP requirements.



3.2.1.1.1. Requirements

#	Mandatory Optional Conditional	Business rules description
1	M	The pilot PSP sends a payment request containing all payment data (payment object).
2	M	The payment request format and content must be consistent with the DESP requirements Digital euro pilot - Backend implementation specifications POST / payments – Request
3	M	DESP must confirm the reception of the payment request by providing associated {id} in the payment request response. Digital euro pilot - Backend implementation specifications → POST / payments – Response The {id} must be kept until the status is received following settlement by DESP. The {id} must be associated with the corresponding payment object and, if applicable, the funding settlement data (funding object) and/or defunding settlement data (defunding object).
4	C	If the DESP doesn't confirm the reception of the request within a predefined timeline the process must be discontinued. Digital euro pilot - Backend implementation specifications
5	M	The function must provide the execution status. In case of failure, a reason code must be provided.

3.2.1.2. Payment request validation

Payment requests initiated by a pilot PSP are forwarded to the other party by DESP. The pilot PSP receiving the request validates it.

This feature applies only if payer and payee are served by different pilot PSPs.

3.2.1.2.1. Requirements

#	Mandatory Optional Conditional	Business rules description
1	M	The pilot PSP receives a payment request with payment settlement data (payment object). The payment request format and content must be consistent with the DESP requirements Digital euro pilot - Backend implementation specifications → POST / payments – Request The {id} must be kept until the status is received following settlement by DESP. The {id} must be associated with the corresponding payment object.
2	M	The pilot PSP must confirm the consistency of the payment request. Digital euro pilot - Backend implementation specifications → POST / payments – Response
3	C	If the format is incorrect, the process is discontinued.
4	M	The function must provide the execution status. In case of failure, a reason code must be provided.



3.2.1.3. Payment instruction creation

The payer PSP sends the payment request to DESP with either as a single transaction (only payment object) or a combined transaction (payment object and funding object).

The payment instruction format and content must be consistent with the DESP requirements.

3.2.1.3.1. Requirements

#	Mandatory Optional Conditional	Business rules description
1	M	The pilot PSP sends a payment instruction to DESP containing all payment data (payment object).
2	C	If the payment transaction is a combined “on us” transaction, the pilot PSP must include, if applicable, the funding settlement data (funding object – if reverse waterfall ID required) and/or defunding settlement data (defunding object – if waterfall is required).
3	C	If the payment transaction is a combined transaction, but not a “on us” transaction, the pilot PSP must include if applicable, the funding settlement data (funding object – if reverse waterfall is required).
4	M	The payment instruction request format, and content must be consistent with the DESP requirements. Digital euro pilot - Backend implementation specifications → POST / payment-instructions – Request
5	M	DESP must confirm the reception of the payment instruction request by providing associated {id} in the payment instruction response. Digital euro pilot - Backend implementation specifications → POST / payment-instructions – Response The {id} must be kept until the status is received following settlement by DESP. The {id} must be associated with the corresponding payment object and, if applicable, the funding settlement data (funding object) and/or defunding settlement data (defunding object).
6	C	If the DESP doesn’t confirm the reception of the request within a predefined timeline the process must be discontinued. Digital euro pilot - Backend implementation specifications
7	M	The function must provide the execution status. In case of failure, a reason code must be provided.

3.2.1.4. Payment instruction validation

DESP forwards the payment instruction request to the payee PSP with only the payment object (even if the original request included a funding object in a reversewaterfall combined transaction on payer side). Before responding, the payee PSP must validate the request and perform payee side checks, determining whether the response should include a defunding object for a payee side waterfall combined transaction.

This feature applies only if payer and payee are served by different pilot PSPs.



3.2.1.4.1. *Requirements*

#	Mandatory Optional Conditional	Business rules description
1	M	The pilot PSP receives a payment instruction with payment settlement data (payment object). The payment instruction request format, and content must be consistent with the DESP requirements. Digital euro pilot - Backend implementation specifications → POST / payment-instructions – Request The request, including the payment object and also contains an associated ID. The {id} must be kept until the status is received following settlement by DESP. The {id} must be associated with the corresponding payment object.
2	M	The pilot PSP must confirm the validation of the payment instruction request and, provide additional settlement data and if applicable, the defunding settlement data (defunding object if waterfall is required). Digital euro pilot - Backend implementation specifications → POST / payment-instructions – Response
3	M	In case of failure, the distributing PSP rejects the transaction to DESP.
4	M	The function must provide the execution status. In case of failure, a reason code must be provided.

3.2.1.5. *Payment settlement processing*

A settlement service, hosted at DESP, covers online payment-instruction and online payment request related functions and flows.

This function is triggered when the pilot PSP receives a DESP settlement notification in the following cases:

- after a payment settlement confirmation notification
- after a payment settlement rejection notification

This function is also triggered by the pilot PSP if the payment settlement notification is not received (time out management).

3.2.1.5.1. *Requirements*

DESP provides a settlement notification to the payer PSP

#	Mandatory Optional Conditional	Business rules description
1	M	The payment settlement confirmation notification or the payment settlement rejection notification format and content must be consistent with the DESP requirements. Digital euro pilot - Backend implementation specifications → POST / payment-instructions / {payment-instructions-id}/status - Request



EUROPEAN CENTRAL BANK

EUROSYSTEM

#	Mandatory Optional Conditional	Business rules description
		The notification contains the {id} of the payment, and the pilot PSP must use it to retrieve the initial data of the payment (payment object) and, if applicable, the initial data of the funding settlement data (funding object).
2	M	The pilot PSP must confirm the reception of the request. Digital euro pilot - Backend implementation specifications → POST / payment-instructions /{payment-instructions-id}/status - Response
3	M	To prevent negative amounts on the beta digital euro account, requirements 4 and 5 must be executed in the specified order.
4	C	If the pilot PSP receives a notification related to a funding object (where the {id} status refers to an initial transaction containing both a payment object and a funding object, it must then trigger the Funding Post-Settlement process after retrieving the funding amount and the DEAN from the original transaction. The pilot PSP performs the funding post settlement process with a settlement notification status (section 2.4.1.1).
5	M	When receiving a notification related to a payment object, the pilot PSP must trigger the Payment post-settlement (amounts assessment) after retrieving the amount transaction, the DEAN, the funding amount (if any), from the original transaction. The pilot PSP performs the post settlement payment Funding-defunding amount assessment process with a settlement notification status (section 3.1.1.2).
6	M	The function must provide the execution status. In case of failure, a reason code must be provided.

DESP provides a settlement notification to the payee PSP

#	Mandatory Optional Conditional	Description
1	M	The payment settlement confirmation notification or the payment settlement rejection notification format and content must be consistent with the DESP requirements. Digital euro pilot - Backend implementation specifications → POST / payment-instructions /{payment-instructions-id}/status - Request The pilot PSP must use this {id} to retrieve the initial data of the payment (payment object) and, if applicable, the initial data of the funding settlement data (funding object) and/or the initial data of the defunding settlement data (defunding object).
2	M	The pilot PSP must confirm the reception of the request. Digital euro pilot - Back-end implementation specifications → POST / payment-instructions /{payment-instructions-id}/status - Response
3	C	If a settlement confirmation notification is received, the pilot PSP must recover the beta digital euro account available balance and holding limit before performing the next requirements. Refer to Digital euro pilot – Frontend specifications – Distributing PSP (section 5.4 - Balance and holding limits service / Digital euro account Balance look-up) .
4	M	To prevent negative amounts on the beta digital euro account, requirements 6, and 7 must be executed in the specified order.
5	M	The pilot PSP receives a confirmation related to a payment object → The pilot PSP must trigger the Payment post-settlement (amounts assessment) after retrieving the amount transaction, the DEAN, the defunding transaction (if any) from the original transaction.



#	Mandatory Optional Conditional	Description
		The pilot PSP performs the post settlement payment Funding-defunding amount assessment process with a settlement notification status (section 3.1.1.2).
6	C	If the pilot PSP receives a notification related to a defunding object (where the {id} status refers to an initial transaction containing both a payment object and a defunding object), it must then trigger the Defunding Post-Settlement process after retrieving the defunding amount and the DEAN from the original transaction. The PSP performs the defunding post settlement process with a settlement notification status (section 2.4.1.2).
7	M	The function must provide the execution status. In case of failure, a reason code must be provided.

Settlement notification is not received (time out management)

If DESP does not provide a settlement confirmation after a predefined delay, the process is discontinued.

Digital euro pilot - Backend implementation specifications. The pilot PSP must then submit a dysfunction report to DESP and wait for an update from DESP before manually taking over the process.

3.3. Refund processing service

A refund is a reversal (in whole or in part) of an earlier payment initiated by a business end user, usually at the payer's request. A refund always requires funding (as business end users have a zero holding limit) and may require defunding on the payer's side.

Service	Function	Description
Refund processing service	Refund request creation	The pilot PSP sends a refund request to DESP.
	Refund request creation validation	The pilot PSP receives a refund request from DESP and validates it.
	Refund confirmation creation	After receiving the refund request from the DESP and completing its internal checks, the payer PSP sends a dedicated refund confirmation.
	Refund settlement processing	The pilot PSP manages the interactions with DESP and the refund transaction lifecycle.

3.3.1. Functions description

3.3.1.1. Refund request creation

The pilot PSP sends a request with all refunding data. DESP will confirm the reception of the request.



3.3.1.1.1. *Pre-requisite*

Having retrieved the refund amount and all original data using the Original Transaction Identification. Refer to **Digital euro pilot – Frontend specifications – Acquiring PSP**. The pilot PSP can then generate the refund object, which is sent to DESP through this function.

3.3.1.1.2. *Requirements*

#	Mandatory Optional Conditional	Business rules description
1	M	The acquiring PSP sends to DESP a request with all refunding data (Refund object) and all funding data (funding object), the refund request format and content must be consistent with the DESP requirements. Digital euro pilot - Backend implementation specifications → POST /refunds – Request
2	M	DESP must confirm the reception of the request and provides the associated {id} Digital euro pilot - Backend implementation specifications → POST /refunds – Response The {id} must be kept until the status is received following settlement by DESP. The {id} must be associated with the corresponding refund object.
3	C	If the DESP doesn't confirm the reception of the request within a predefined timeline the process must be discontinued. Digital euro pilot - Backend implementation specification.
4	M	The function must provide the execution status. In case of failure, a reason code must be provided.

3.3.1.2. *Refund request validation*

When the commercial bank money account is held by a different distributing PSP, DESP forwards the refund request received from the pilot PSP servicing the beta digital euro account.

The pilot PSP servicing the commercial bank money account confirms the reception of the request and validates it.

3.3.1.2.1. *Requirements*

#	Mandatory Optional Conditional	Business rules description
1	M	The pilot PSP receives a request with refund data (Refund object). The refund request format and content must be consistent with the DESP requirements. Digital euro pilot - Backend implementation specifications → POST /refunds – Request The request also contains an associated ID. The {id} must be kept until the status is received following settlement by DESP. The {id} must be associated with the corresponding refund object.



2	M	The pilot PSP must confirm the request status (after performing the mandatory checks ensuring the funding proper execution) and send a response to DESP. The response contains refunding data (Refund object) and optionally defunding data if a waterfall mechanism is required on payer side (Defunding object). The refund response format and content must be consistent with the DESP requirements. Digital euro pilot - Back-end implementation specifications → POST /refunds – Response
3	M	In case of failure, the pilot PSP rejects the transaction to DESP.
4	M	The function must provide the execution status. In case of failure, a reason code must be provided.

3.3.1.3. Refund settlement processing

After refund settlement processing, DESP sends the status of the settlement to the pilot PSP, and the pilot PSP confirms the receipt of status.

A settlement service, hosted at DESP, covers refund request related functions and flows.

This function is triggered when the pilot PSP receives a DESP settlement notification in the following case:

- after a refund confirmation notification
- after a refund rejection notification

This function is also triggered by the pilot PSP if the refund settlement notification is not received (time out management).

3.3.1.4. Requirements

DESP provides a refund notification to the payee PSP (new role as a payer)

#	Mandatory Optional Conditional	Business rules description
1	M	The pilot PSP receives from DESP a confirmation of refunding settlement together with the ID of the corresponding initial refund request. The refund settlement request format and content must be consistent with the DESP requirements Digital euro pilot - Backend implementation specifications → POST /refunds/{refund-id}/status - Request The pilot PSP must use this {id} to retrieve the initial data of the refund object.
2	M	The pilot PSP must confirm the reception of the request. Digital euro pilot - Backend implementation specifications → POST /refunds/{refund-id}/status - Response
3	C	If the format is incorrect, the process is discontinued. The pilot PSP must then submit a dysfunction report to DESP and wait for an update from DESP before manually taking over the process.
4	M	To prevent negative amounts on the beta digital euro account, requirements 5 and 6 must be executed in the specified order.



EUROPEAN CENTRAL BANK

EUROSYSTEM

5	M	The pilot PSP has received a notification related to a funding object (where the {id} status refers to an initial transaction containing both a payment object and a funding object), it must then trigger the Funding Post-Settlement process after retrieving the funding amount and the DEAN from the original transaction. The pilot PSP performs the funding post settlement process with a settlement confirmation status (section 2.4.1.1).
6	M	The pilot PSP performs the Refunding post settlement process with a settlement confirmation status (section 3.1.1.4).
7	M	The function must provide the execution status. In case of failure, a reason code must be provided.

DESP provides a refund notification to the payer PSP (new role as a payee)

#	Mandatory Optional Conditional	Description
1	M	The pilot PSP receives from DESP a confirmation of refunding settlement together with the ID of the corresponding initial refund request. The refund settlement request format and content must be consistent with the DESP requirements. Digital euro pilot - Backend implementation specifications → POST /refunds/{refund-id}/status - Request The pilot PSP must use this {id} to retrieve the initial data of the refund object.
2	M	The pilot PSP must confirm the reception of the request. Digital euro pilot - Backend implementation specifications → POST /refunds/{refund-id}/status - Response
3	C	If the format is incorrect, the process is discontinued. The pilot PSP must then submit a dysfunction report to DESP and wait for an update from DESP before manually taking over the process.
4	M	To prevent negative amounts on the beta digital euro account, requirements 5, and 6 must be executed in the specified order.
5	M	The pilot PSP performs the Refunding post settlement process with a settlement confirmation status (section 3.1.1.4).
6	C	If the pilot PSP receives a confirmation related to a defunding object (where the {id} status refers to an initial transaction containing both a payment object and a defunding object), it must then trigger the Defunding Post-Settlement process after retrieving the defunding amount and the DEAN from the original transaction. The pilot PSP performs the defunding post settlement process with a settlement confirmation status (section 2.4.1.2).
7	M	The function must provide the execution status. In case of failure, a reason code must be provided.

Settlement notification is not received (time out management)

If DESP does not provide a settlement confirmation after a predefined delay the process is discontinued.

Digital euro pilot - Backend implementation specifications. The pilot PSP must then submit a dysfunction report to DESP and wait for an update from DESP before manually taking over the process.



4. Cross-functional Services

4.1. Authentication Service

Placeholder for a future release.

4.2. Notification Service

Notification Service is invoked during the processes and enables the end user to be informed about the result of any request through an identified channel.

4.2.1. Requirements

The pilot PSP must identify and check all the notification pre-conditions and parameters to provide the expected information to the end user.

#	Mandatory Optional Conditional	Description
1	O	Notification Triggers: Determine when a notification should be sent, such as: <ul style="list-style-type: none"> - <u>Event-based Triggers</u>: Send a notification when a specific event occurs (e.g., a new order is placed, an appointment is confirmed, a shipment is delivered). - <u>Time-based Triggers</u>: Send notifications based on a schedule or specific time (e.g., reminders for upcoming meetings or deadlines). - <u>Threshold-based Triggers</u>: Notifications triggered when certain thresholds are met (e.g., low account balance, a certain number of failed login attempts). - <u>System Health Alerts</u>: Notifications sent when system components (e.g., servers, databases) experience issues like downtime or performance degradation.
2	O	Notification Channels: Define which channels should be used for sending notifications, such as: <ul style="list-style-type: none"> - <u>Email</u>: Notifications sent via email, suitable for formal or detailed messages. - <u>SMS/Push Notifications</u>: Immediate notifications delivered to mobile devices for urgent or high-priority information. - <u>In-app Notifications</u>: Messages that appear within a specific application or platform. - <u>Web Push</u>: Browser-based notifications for real-time alerts on desktop or mobile web. - <u>Multi-channel Notifications</u>: Using multiple channels for critical notifications to ensure the message is received.
3	M	User Preferences and Opt-in: Govern end user consent and preferences, including: <ul style="list-style-type: none"> - <u>Opt-in/Opt-out</u>: End users must explicitly opt-in to receive notifications and should have the ability to easily opt-out. - <u>Preference Management</u>: Allow end users to customise which types of notifications they receive (e.g., marketing updates, order status, security alerts). - <u>Frequency Control</u>: Define how frequently notifications should be sent to avoid spamming end users (e.g., no more than one notification per hour). - <u>Do Not Disturb (DND)</u>: Block or delay notifications during end user-specified quiet hours.
4	O	Content Customisation: Define rules about what information should be included in the notification:



#	Mandatory Optional Conditional	Description
		<ul style="list-style-type: none"> - Personalisation: Tailor notifications to include end user-specific data (e.g., their name, order number, account details). - Localisation: Send notifications in the end user's preferred language or with region-specific details (e.g., currency, time zone). - Urgency Level: Define different templates for high, medium, or low-priority notifications (e.g., critical alerts may include bold text or colours to emphasise importance).
5	M	<p>Delivery and Retry Policies: Define rules related to ensuring notifications are delivered reliably:</p> <ul style="list-style-type: none"> - Retry Attempts: Define how many times the system should retry sending a notification in case of failure (e.g., failed SMS or email delivery). - Escalation Policies: Escalate the notification to higher-level alerts or additional channels if a critical notification is not acknowledged or delivered. - Delivery Confirmation: Track and confirm whether the end user has received and opened the notification.
6	M	<p>Security and Compliance: Define rules that ensure the notification service adheres to security and privacy standards:</p> <ul style="list-style-type: none"> - Data Privacy: Ensure notifications don't expose sensitive information (e.g., no account numbers or passwords in SMS messages). - Encryption: Use encrypted communication channels (e.g., SSL/TLS) to protect notification data in transit. - Regulatory Compliance: Comply with laws such as GDPR or Privacy directive ensuring user consent and clear unsubscribed options in all notifications.
7	M	<p>Priority and Queuing: Define rules for managing the priority of notifications.</p> <ul style="list-style-type: none"> - Priority Levels: Assign priority to notifications (e.g., critical system alerts may take precedence over promotional messages). - Queue Management: Define how notifications are queued and processed if there is a backlog (e.g., critical notifications bypass less important ones). - Load Balancing: Ensure the notification system can handle high volumes of notifications without overwhelming servers or causing delays.
8	M	<p>Notification expiration: Specify rules that define the lifespan of a notification:</p> <ul style="list-style-type: none"> - Expiry Time: Set an expiration for time-sensitive notifications (e.g., a sale reminder may expire after the sale ends). - Auto-archiving: Notifications that remain unread for a certain period may be automatically archived or deleted.
9	M	<p>Failure and Error Handling: Define rules for handling errors in the notification process:</p> <ul style="list-style-type: none"> - Fallback Mechanisms: Use alternative delivery methods if the primary channel fails (e.g., sending an email if the SMS fails). - Error Logging: Log notification failures and errors for auditing and troubleshooting. - End user Feedback: Notify end users if a critical notification fails to deliver (e.g., via an in-app message indicating an error occurred).
10	M	<p>Analytic and Reporting Rules: Define rules that govern tracking and analysing notifications:</p> <ul style="list-style-type: none"> - Open and Click Tracking: Track whether the end user opens or interacts with the notification. <p>Requirement: The system shall track notification open rates and end user interactions (clicks, dismissals).</p> <p>Scope: Applies to all notification channels (email, SMS, in-app).</p> <p>Standards:</p> <ul style="list-style-type: none"> • Tracking must use unique identifiers per message. • Privacy compliance (e.g., GDPR) must be ensured.



#	Mandatory Optional Conditional	Description
		<ul style="list-style-type: none"> - A/B Testing: Test different versions of notification templates or delivery methods to determine effectiveness. Requirement: A/B testing is required before launching any new notification template. Goal: The winning version must show at least 10% better performance in open or click rate. Duration: Testing should run for at least 3 business days, unless early results clearly show a winner. Review: Results must be shared with the Product Owner and documented before rollout. - Reporting: Provide insights into notification delivery success rates, user engagement, and failure trends. Requirement: The system must generate weekly dashboards and monthly reports for the following metrics: <ul style="list-style-type: none"> o Delivery success rate per channel o Open and click rates o Notification failure types (e.g., soft bounce, hard bounce) o A/B testing outcomes (if applicable) Distribution: Reports must be auto shared with Product, Marketing, and Compliance teams.

4.2.2. Interface description

The notifications are triggered without any interface exchanged between the end user and the end user's pilot PSP.

The notification service is automatically triggered during the process when a notification to the end user is required in case of rejection and in case of completion.

4.2.3. Message structure

Incoming message

The incoming message is generated by the pilot PSP according to the transaction context and the existing notification capabilities.

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 event	Unique identifier of the event that triggers the message.	STR	35	M	EventIdentification
Type of channel	Delivery channel type ID: • EMAL (transmission by email)	SSET	4	M	ChannelType



Data element	Description	Type	Length	Presence indicator	Standardised name
	<ul style="list-style-type: none"> • ONLI (transmission online – e.g., in-app) • MBNO (transmission by mobile phone) ▪ MULT (Multichannel) ▪ FILE (transmission by file transfer) <p>Code set to be confirmed</p>				
Notification Message	Notification message.	STR	*	M	
Delivery Address	Clients phone number or E-mail address. Type depends on Delivery channel type.	STR	100	M	DeliveryAddress
Language	Dictionary language name of notification message. (ISO 639 language code)	SSET	2	O	Language
Delivery Date	Date of notification needed to be delivered. Format DD.MM.YYYY	DATE	10	M	DeliveryDate
Urgency Level	Message urgency level (0 - high, 1- normal, 2 - low)	NUM	1	O	UrgencyLevel

* The length of the notification message is defined by the pilot PSP according to the technical architecture.

4.2.4. Notification classification

Below is a proposed classification of operational notifications, structured to guide policy framing for the digital euro pilot.

Each category includes its functional purpose and practical examples to assist pilot PSPs in implementation mapping.

Category	Purpose	Examples
Security & Fraud	Safeguard end user accounts and detect abnormal behaviour	<ul style="list-style-type: none"> ▪ Suspicious login attempt ▪ Password reset alert ▪ 2FA verification
Transaction Status	Ensure transparency and usability of payment operations.	<ul style="list-style-type: none"> ▪ Payment sent/received ▪ Payment failed or rejected ▪ Refund issued
Account Constraints	Notify restrictions that prevent usage.	<ul style="list-style-type: none"> ▪ Blocked or frozen account ▪ Limits reached ▪ Expired credentials
Service Availability	Inform end users about service disruptions or maintenance.	<ul style="list-style-type: none"> ▪ Scheduled downtime ▪ Service outages ▪ Restored services



Regulatory Compliance	Fulfil legal obligations or end user rights.	<ul style="list-style-type: none"> ▪ Strong Customer Authentication (SCA) request ▪ Data usage confirmation
Critical Updates	Deliver essential digital euro pilot or PSP infrastructure updates.	<ul style="list-style-type: none"> ▪ Terms of service update (if mandatory) ▪ Major security policy changes

4.2.5. Functional error description

Function	Error	Managed by
Alias resolution	Invalid alias format	DESP / Pilot PSP
	Alias not reachable	DESP
	Alias resolution failed	DESP
	Invalid alias resolution request	DESP
	Invalid Alias registration request	Pilot PSP
	Invalid alias ownership proof	Pilot PSP
	Alias registration failed	DESP
Individual end user registration	Invalid individual end user registration request	Pilot PSP
	Individual end user registration failed	Pilot PSP
Individual end user deregistration	Invalid individual end user offboarding request	Pilot PSP
	Individual end user deregistration failed	DESP
Business end user registration	Invalid business end user registration request	Pilot PSP
	Business end user registration failed	Pilot PSP
Business end user deregistration	Invalid business end user offboarding request	Pilot PSP
	Business end user deregistration failed	DESP
DEAN validity check	Invalid DEAN validity check request	Pilot PSP
	Invalid DEAN	Pilot PSP
Beta digital euro account	Insufficient beta digital euro balance	Pilot PSP
	Reverse waterfall option not activated	Pilot PSP
	Unsuccessful reverse waterfall	Pilot PSP
	Holding limit exceeded	Pilot PSP
	Waterfall option not activated	Pilot PSP
	Unsuccessful waterfall	Pilot PSP
	Invalid beta digital euro account blocking request	Pilot PSP



EUROPEAN CENTRAL BANK

EUROSYSTEM

Function	Error	Managed by
	Beta digital euro account blocking failed	Pilot PSP
	Invalid beta digital euro account unblocking request	Pilot PSP
	Beta digital euro account unblocking failed	Pilot PSP
	Beta digital euro account registration failed	DESP
	Beta digital euro account deregistration failed	DESP
	Beta digital euro balance look-up failed	Pilot PSP
	Beta digital euro account already exists	Pilot PSP
	Beta digital euro existence checks failed	DESP / Pilot PSP
Authentication	Invalid authentication details	Pilot PSP
Commercial bank money account funds management	Unsuccessful commercial bank money account credit	Pilot PSP
	Unsuccessful commercial bank money account debit	Pilot PSP
	Unsuccessful commercial bank money account funds blocking	Pilot PSP
	Unsuccessful commercial bank money account funds release	Pilot PSP
Beta digital euro account funds management	Unsuccessful beta digital euro account credit	Pilot PSP
	Unsuccessful beta digital euro account debit	Pilot PSP
	Invalid request for manual funding online	Pilot PSP
	Invalid request for manual defunding online	Pilot PSP
Balance online	Beta digital euro balance look-up failed	Pilot PSP
	Invalid beta digital euro account blocking request	Pilot PSP
Transactions history	Transactions history invalid request	Pilot PSP
	No transaction found for the requested criteria	Pilot PSP
Linked account management	Invalid linked account settings request	Pilot PSP
	Linked account settings storage failed	Pilot PSP
	Linked account settings removal failed	Pilot PSP
	Linked account settings look-up failed	Pilot PSP
	Linked account ownership validation failed	Pilot PSP
Liquidity settings management	Invalid liquidity settings request	Pilot PSP
	Liquidity settings storage failed	Pilot PSP



Function	Error	Managed by
	Liquidity settings removal failed	Pilot PSP
	Waterfall / Reverse waterfall option look-up failed	Pilot PSP
Notification settings management	Invalid notification settings request	Pilot PSP
	Notification settings storage failed	Pilot PSP
	Notification settings removal failed	Pilot PSP
NFC enrolment management	Invalid enrolment management request	Pilot PSP
	Invalid DEAN last Four Digits	Pilot PSP
	Others functional errors (placeholder HCE Backend specification)	HCE Backend
NFC Termination by pilot PSP app management	Invalid enrolment management request	Pilot PSP
	Invalid DEAN last Four Digits	Pilot PSP
	Others functional errors (placeholder HCE Backend specification)	HCE Backend
NFC Mobile Payment request validation	Invalid merchant	DESP
	Do not honour	DESP
	Error	DESP
	Invalid transaction	DESP
	Invalid amount	DESP
	Invalid Token	DESP
	No such issuer	DESP
	Format error	DESP
	Insufficient funds	DESP
	No checking account	DESP
	Transaction not permitted to cardholder	DESP
	Transaction not permitted to terminal	DESP
	Suspected fraud	DESP
	Security violation	DESP
	Issuer or switch inoperative	DESP
	Routing error	DESP
Violation of law	DESP	



EUROPEAN CENTRAL BANK

EUROSYSTEM

Function	Error	Managed by
	Duplicate transmission	DESP
	System malfunction	DESP
Commercial bank money account	Insufficient commercial bank money account balance	Pilot PSP
	Invalid commercial bank money account status	Pilot PSP
	Commercial bank money account balance look-up failed	Pilot PSP
PSP ID look-up	Invalid PSP ID	DESP
	PSP ID not found	DESP
	Invalid PSP ID look-up request	DESP
Payment initiation	Invalid individual end user payment instruction request	Pilot PSP
	Invalid individual end user payment request	Pilot PSP
	Invalid business end user payment request	Pilot PSP
	Invalid refund request	Pilot PSP
Settlement	Settlement failed	DESP
	Invalid payment request	DESP & Pilot PSP
	Invalid payment instruction	DESP & Pilot PSP
	Invalid refund request	DESP & Pilot PSP
	Invalid NFC CSPACE payment request	DESP & Pilot PSP
	Invalid funding online settlement request	DESP
	Invalid defunding online settlement request	DESP
	Timeout in receiving settlement notifications	Pilot PSP
	Timeout in receiving the response from DESP	Pilot PSP
Payment amount assessment	Pre-settlement funding amount assessment failed	Pilot PSP
	Pre-settlement funding amount assessment failed	Pilot PSP
	Post-settlement funding amount assessment failed	Pilot PSP
	Post-settlement funding amount assessment failed	Pilot PSP
Refund amount assessment	Pre-settlement refund defunding amount assessment failed	Pilot PSP
	Post-settlement refund defunding amount assessment failed	Pilot PSP