EUROPEAN CENTRAL BANK

EUROSYSTEM

Digital euro pilot

Back-end implementation specifications

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

EUROPEAN CENTRAL BANK

EUROSYSTEM

CONTENTS

1.	**Introduction**	**4**
1.1.	Definitions and annotations	5
1.2.	Exceptions management	6
1.2.1.	*Positive responses*	*6*
1.2.2.	*Functional errors*	*6*
1.3.	Error message header	7
1.4.	Response codes	7
1.4.1.	*Functional errors (Settlement and other business operations)*	*8*
1.4.2.	*Technical errors*	*8*
1.5.	Error message body	9
2.	**Structure of each service-specific section**	**10**
2.1.	Service description	10
2.2.	Function description	10
2.3.	Interface description	10
2.3.1.	*Sequence diagrams*	*10*
2.3.2.	*Overview of API endpoints*	*11*
2.3.3.	*Description of API endpoints*	*11*
2.3.4.	*Exception management*	*12*
3.	**Secure Exchange of Payment Information (SEPI) Service**	**14**
3.1.	Service description	14
3.2.	Function description	16
3.3.	Interface description	16
3.3.1.	*Overview of covered scenarios*	*16*
3.3.2.	*Overview of API endpoints*	*16*
3.4.	DESP API for pilot PSPs to send requests to DESP	17
3.4.1.	*POST /cryptogram-validations – Request*	*17*
3.4.2.	*POST /cryptogram-validations – Response*	*17*
4.	**Access Management Service**	**19**
4.1.	Service description	19
4.2.	Function Description	19
4.3.	Interface description	20
4.3.1.	*Sequence diagrams*	*20*

EUROPEAN CENTRAL BANK

EUROSYSTEM

4.3.2.	*Overview of API endpoints*	*25*
4.3.3.	*Exception management*	*36*
5.	**Alias lookup service**	**37**
5.1.	Service description	37
5.2.	Function description	39
5.3.	Interface description	39
5.3.1.	*Sequence diagrams*	*39*
5.3.2.	*Overview of API endpoints*	*40*
5.3.3.	*DESP API for pilot PSPs to send requests to DESP*	*40*
5.3.4.	*Error handling*	*42*
6.	**Settlement Service (SE)**	**43**
6.1.	Service description	43
6.2.	Functional description	44
6.3.	Interface description	45
6.3.1.	*Sequence diagrams*	*47*
6.3.2.	*Overview of API endpoints*	*59*
6.3.3.	*DESP API for pilot PSPs to send requests to DESP*	*61*
6.3.4.	*PSP API for pilot PSPs to receive requests from DESP*	*89*
7.	**Data dictionary**	**109**
7.1.	Complex data types	109
7.2.	Generic data types	114
7.3.	Code lists	114
7.4.	Other ISO-related basic types	116

EUROPEAN CENTRAL BANK
EUROSYSTEM

1. Introduction

This document provides detailed implementation specifications for interfaces that pilot PSPs can implement to connect with the Digital Euro Service Platform (DESP) domain, as presented in the figure below (marked in the pink frame).

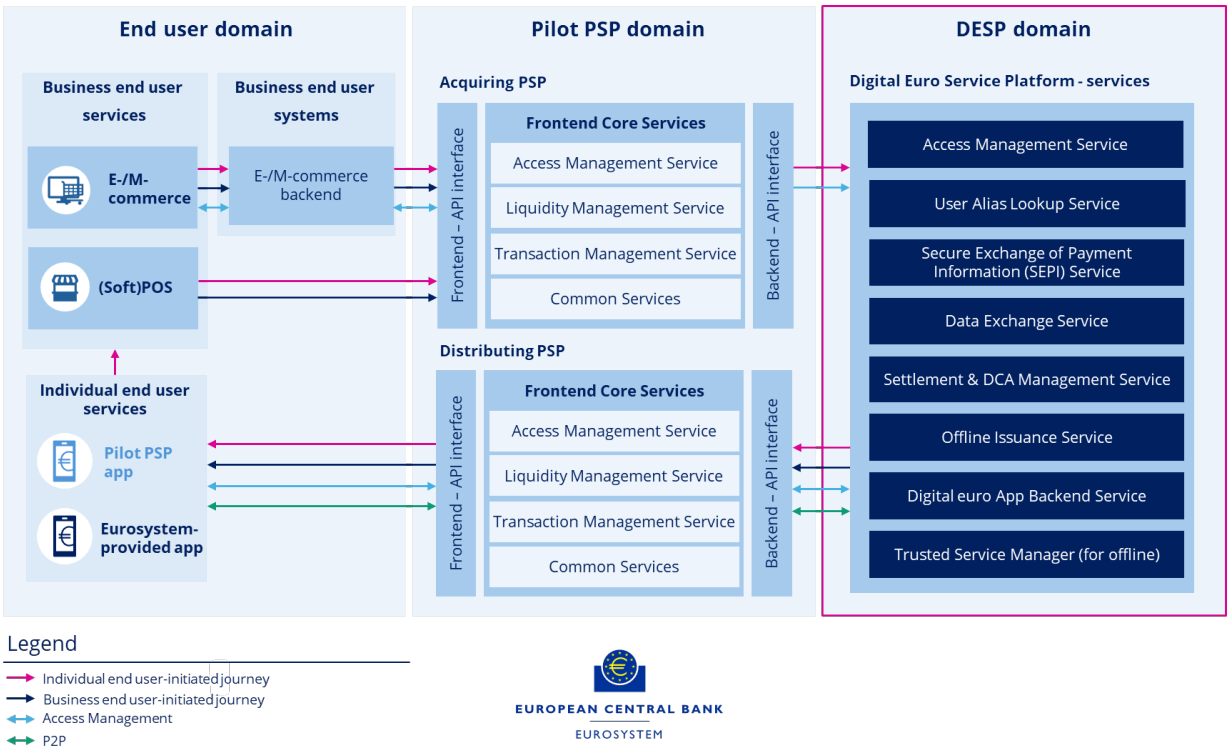


Figure 1 Digital euro pilot - Functional architecture

The specifications described in this document are accompanied by the corresponding YAML file. The YAML file contains also an API data model that covers functions, supported data elements, the relation between the data elements and its occurence. Additionally, in the ”Interface description” section, data elements with respective definitions and usage rules are included for each endpoint. The specifications are developed based on ISO20022 data dictionary where available, and reused in the design process of the applicable parts of other market standards such as Berlin Group’s Mobile P2P Interoperability Framework for the Alias Lookup Service, and CPACE for the SEPI Service.

The DESP supports a set of APIs provided centrally, which were grouped into services to reflect a specific business function.

Table 1-1 Overview of DESP backend services

EUROPEAN CENTRAL BANK

EUROSYSTEM

Service	Service scope
Access Management Service (AM)	Management of end user-centric and account-specific functions that are not payment related. This includes end user registration and management, DEAN creation and management, alias registration and management.
Alias Lookup Service (AL)	Alias lookup function. *Please note: alias registration and management is supported via Access Management Service.*
Secure Exchange of payment information (SEPI) Service	This service groups all interfaces related to the tokenization of payment information: • Contactless (NFC) surrogate value generation and detokenization.
Settlement Service (SE)	Functions supporting processing beta digital euro transactions, i.e. payment (incl. refunds), funding, defunding, combined (funding, defunding and payment).
Data exchange Service (DE)	The service supports access to pre-defined reports and queries.[1]

In the following sub-sections, general rules that are applicable to all services are described, starting with definitions and annotations (sub-section 1) and description of exceptions management, types of errors, error messages and response codes (sub-section 2-5). In chapter 2, an overview of the structure of service specific sections is introduced, providing an overview of the structure that is used in description of each service introduced in **Table 1-1**.

1.1. Definitions and annotations

Data element definition

The following elements are used to define data elements:

Element	Definition, annotation and example
Data element name	Name of data element, e.g. “Payee PSP ID”.
Description	Description of what a data element entails. For example: the town of the point of interaction. For remote payments, this refers to the town where the payee is registered. The valid town name shoulld be in the local language, e.g. “Frankfurt”.
Type	Either a complex data type described in the data dictionary or a basic data type (see section *Data types* for more details).
Presence indicator	Mandatory (M), conditional (C), optional (O), mandatory and repeatable (MR), optional and repeatable (OR), not present (NP).

Data types

Data types used in the back-end specifications can be either:

[1] DESP would only offer an API for generic exchange of encrypted data, for the participants to use to exchange messages (for e.g. AML compliance and reporting) without processing or validating the information shared through it.

- complex data types (based on ISO20022 where possible, e.g Address) or
- basic data types (e.g. Max140Text).

The complex data types are described in detail in the data dictionary (**section 6**), in addition to the list of applicable codes.

1.2. Exceptions management

This sub-section describes the approach to exceptions handling that applies to all services. In addition to a **positive result** (reflected by HTTP response codes 200, 201 and 202 for ok, created and accepted respectively), the following **exception types** are distinguished and described in the tables below: functional errors, technical errors, functional time out, and technical time out

1.2.1. Positive responses

HTTP Code	Class Code	Includes Problem+JSON	Description
200	OK	No	OK response for a GET, PUT or a POST.
201	Created	No	Response when a resource is created. The response body will usually contain a reference (ID) to the created resource.
202	Accepted	No	Response when a resource is created, but a response will be provided asynchronously.
204	No Content	No	Positive response without reponse body.

1.2.2. Functional errors

Functional errors occur when the request is formally and technically correct, but the requested operation cannot be performed due to business validation errors, such as insufficient funds. In such cases, an HTTP response code 200/201 and a reason code will be returned. When a functional error occurs during settlement, the fields listed below will be present in the response payload.

Data element	Description	Type	Presence indicator
Transaction status	Indication of the success or failure of the request. Based on ISO external code set: ExternalPaymentTransactionStatus1Code	Status Code	M
Status reason code	Code and text providing details about the negative business validation result.	Status Reason	C, mandatory in case of negative validation

Technical errors

Technical errors occur either because the request is incorrect or because the server is unable to process the request. When the request is incorrect, HTTP response code 4xx (where x stands for a possible range of digits, see table below) will be returned together with a problem+JSON, where applicable. In case the

server is unable to process the request, HTTP response code 5xx will be returned. The table with the technical error/HTTP codes (see **section 1.4.2 Technical errors**) describes in which cases a 'problem+JSON' will be added.

Functional time out

A functional time out occurs when the processing of a business operation exceeds the maximum processing time defined in the non-functional requirements (**Digital euro pilot – Pilot business architecture**). When one of the involved parties (pilot PSP or DESP) comes to the conclusion that the maximum processing time is or will be exceeded, it sends a negative response. The normal HTTP code 200/201 will be given and the status field will contain the status RJCT and the status reason code will be AB03 "AbortedSettlementTimeout". In addition, DESP sends a reject notification to all involved pilot PSPs.

Functional timeouts apply only to business operations. For query-type requests, timeouts are handled as technical timeouts (e.g. HTTP 504).

Data element	Description	Type	Presence indicator
Transaction status	Indication of the success or failure of the request. In case of a functional time out: "RJCT". Based on ISO external code set: ExternalPaymentTransactionStatus1Code	Status Code	M
Status reason code	Code and text providing details about the negative business validation result. In case of a functional time out: "AB03".	Status Reason	C, mandatory in case of negative validation

Technical time out

A technical time out occurs when the requester does not receive a response within a defined timeframe or when the server responds with HTTP code 504 (gateway timeout).

1.3. Error message header

Each header for request and responses must contain at least the following header fields:

Data element	Description	Type	Presence indicator
Request ID	Unique ID that identifies a request.	UUID	M
Response timestamp	Time and date of the response message.	ISODateTime	M

1.4. Response codes

The following tables provide an overview of applicable exception response codes. HTTP status reflects technical processing (e.g. resource creation), while the business outcome is conveyed via transaction status and reason codes.

1.4.1. Functional errors (Settlement and other business operations)

HTTP Code	Class Code	Includes Problem+JSON	Description
200	OK	No	In case the business operation cannot be processed due to a functional error (e.g. insufficient funds).
201	Created	No	In case the request cannot be processed due to functional error.

1.4.2. Technical errors

HTTP Code	Class Code	Includes Problem+JSON	Description
400	Bad Request	Yes	Request cannot be processed due to syntax error (wrong field parameter, etc).
401	Unauthorised	Yes	The required authentication for the requested resource is missing, not provided, or wrong.
403	Forbidden	Yes	The caller of the service is not authorized to use this resource. For an API, this can mean for instance that the request's token was valid, but was missing a scope for this endpoint. Or that some object-specific authorization failed.
404	Not Found	Yes	The requested resource could not be found.
405	Method Not Allowed	Yes	The request method is not supported for this resource.
406	Not Acceptable	Yes	The requested resource is capable of generating only content not acceptable according to the Accept headers sent in the request.
408	Request Timed Out	No	The server timed out waiting for the request. The client may repeat the request without modifications at any later time.
409	Conflict	Yes	Indicates that the request could not be processed because of conflict in the current state of the resource, such as an edit conflict between multiple simultaneous updates.
415	Unsupported Media Type	No	The client did not provide a supported content-type for the request body.
429	Too Many Requests	Yes	The caller of the service has sent too many requests in a given amount of time. Intended for use with rate-limiting schemes.This is defined at DESP API level; any NSP-level traffic management is infrastructure-specific and out of scope here unless explicitly specified. A 429 response must include a Retry-After header.
500	Internal Server Error	No	A generic error message, given when an unexpected condition was encountered and no more specific message is suitable.
502	Bad Gateway	Yes	The server, while acting as a gateway or proxy, received an invalid response from an inbound server attempting to fulfill the request. It may be used by a server to indicate that an inbound service is creating an unexpected result instead of 500. Clients should be careful with retrying on this response, since the nature of the problem is unknown and must be expected to continue.
503	Service Unavailable	No	The server cannot handle the request (because it is overloaded or down for maintenance).
504	Gateway Timeout	No	The server, while acting as a gateway or proxy, did not receive a timely response. May be used by servers to indicate that an inbound service cannot process the request fast enough. Client

EUROPEAN CENTRAL BANK

EUROSYSTEM

			may retry the request immediately exactly once, to check whether this was a temporary issue.

1.5. Error message body

Data elements for error responses (HTTP codes 400, 401, 403, 404, 405, 406, 409, 415, 422, 429, 500, 503)

Data element		Element description	Type	Presence indicator
Type		An URI reference [RFC3986] that identifies the problem type.	Max70Text	M
Status		The HTTP status code that is returned by the server. This must be the same code as given as an HTTP response code.	HTTP Status	O
Title		Short human readable description of error type.	Max70Text	O
Detail		Additional text to explain the error.	Max500Text	O
Instance		A string containing an URI reference that identifies the specific occurence of the problem.	Max70Text	O
Response (reason) code		A code which provides details about the result of the request in case of failure.	Max70Text	M
Additional errors		Array of additional error details containing the 3 4 fields below.	Array of fields	O
Title Detail	Title	Short human readable description of error type.	Max70Text	O
	Detail	Additional text to explain the error.	Max500Text	O
	pointer	This is a JSON-POINTER that uses the RFC 6901 syntax to refer to the location within the Json where the proble occures.	Max70Text	O
	Response (reason) code	A code which provides details about the result of the request in case of failure.	*tbd*	M
Links		Link to a page containing an explanation of the error.	Max256Text	O

EUROPEAN CENTRAL BANK

EUROSYSTEM

2. Structure of each service-specific section

For each service exposed by DESP, this document describes the business function addressed by the service. It then defines the requirements for pilot PSPs that serve individual end users, as well as those that serve business end users.

Each service section consists of a reference to an underlying data model of the service, followed by function description and interface description with details on covered scenarios, overview of API endpoints and respective messages.

N. **Service Name**

- N.1. Service description
- N.2. Function description
- N.3. Interface description
  - N.3.1. Sequence diagrams
  - N.3.2. Overview of API endpoints
  - N.3.3. DESP APIs for pilot PSPs to send requests to DESP

2.1. Service description

This subsection provides a high level introduction of a service and related processes that need to be supported in the pilot PSP to DESP domain.

2.2. Function description

This section offers an overview of the set functions that constitute a service. It provides an additional background information on the applicable functions and the context in what they are performed.

2.3. Interface description

This section provides the definition of API message types supporting functions introduced in the “Function description” section, including scenarios, API endpoints and their detailed description.

2.3.1. Sequence diagrams

This section provides an API view of the flow in the pilot PSP to DESP domain by means of sequence diagrams. For a complete view of related flows please refer to **Digital euro pilot – End-to-end process flows**.

EUROPEAN CENTRAL BANK

EUROSYSTEM

2.3.2. Overview of API endpoints

The section includes an overview of API endpoints applicable for the service. These endpoints are listed separately for endpoints exposed by DESP and those exposed by a pilot PSP.

2.3.3. Description of API endpoints

This section describes the endpoints applicable for a given service and includes data elements and details of the requests and responses for each endpoint. The section, describes separately:

- DESP APIs - endpoints exposed by DESP, and
- PSP APIs - exposed by a pilot PSP.

2.3.3.1. Technical header

The section below describes a part of the message that is common across all messages, including respective data elements. Any exceptions, as well as description of remaining message parts that are endpoint-specific, can be found in the “Interface description” sections.

2.3.3.1.1. Request header

The technical request header contains the following data elements that should be used in all requests described in this document:

Data element	Description	Type	Presence indicator
Request ID	Unique ID that identifies a request and correlates the response to the request.	UUID	M
Request Timestamp	Time and date of the request message.	ISODateTime	M
Requesting Agent	The BIC of the requesting pilot PSP	BICFI	M for API calls made by pilot PSP to DESP. DESP will not send this header to a pilot PSP.

2.3.3.1.2. Response header

The technical response header contains the following data elements that should be used in all responses described in this document:

Data element	Description	Type	Presence indicator
Request ID	Unique ID that identifies a request and correlates the response to the request.	UUID	M
Response Timestamp	Time and date of the response message.	ISODateTime	M

2.3.3.1.3. Idempotency header

All requests and responses shall include an unique Request-Id, (UUID). The Request-Id in the response shall equal the Request-Id in the related request.

The uniqueness implies that request B with the same Request-Id as request A should be responded as follows to achieve idempotency:

- If the business content of request B (e.g., path, query parameters, body, and headers as defined in this specification) is identical to the business content of request A, and if the time deviation is not excessive (implementation/request-specific), and if the internal state of the related resource has not changed in the meantime, then the server's response to request B should be identical to its response to request A.
- Otherwise, the requested message should be rejected with a 4xx response code.

2.3.4. Exception management

The table below contains a non-exhaustive list of errors, the codes used and some examples of when these errors may apply.

HTTP code	Error code	Type[2]	Title	Detail (example)
400	ALLOWED_VALUE_ERROR*	Value-not-allowed	Value provided is not allowed	Invalid user type (e.g. 'CONS') in registration request.
400	FORMAT_ERROR	Missing-field	Mandatory field missing	Pilot PSP ID missing in alias registration request.
		Data-format-invalid	Invalid data format	Alias value '1234' with alias type 'MBNO' in registration request.
		Data-type-invalid	Invalid data type	Alias type 'MAIL' in registration request.
		Format-incorrect	Incorrect format	Alias type 'EMAIL' in registration request.
404	NOT_FOUND	Resource-not-found	Resource not found	The DEAN does not exist.
409	CONFLICT	Request-conflicts-with current-state-of-resource	Request conflicts with the current state of resource	DEAN is already linked to another pilot PSP. Alias already exists.

[2] The type in the table is always preceded by a URL (e.g. https://urlplaceholder.net/missing-field). The URL has not been determined yet and for reasons of readability, this part has been omitted from the table.

EUROPEAN CENTRAL BANK

EUROSYSTEM

Please note that technical errors in settlement follow the same patterns as illustrated above, but functional errors in settlement have a different structure:

Status	HTTP code	Status reason code	Status reason description[3]	Detail (example)[4]
RJCT	200	AC05	ClosedDebtorAccountNumber	Debtor account EU123456789012 is closed.
		ARFR	AlreadyRefusedRTP	The debtor has already refused request 12345678910 on 20260610.
		D002	Unspecified business validation error	An unknown validation error occurred.
		D005	NotReachable	PSP ABRONL2X is not a participant in the scheme.
RJCT	201	AB03	AbortedSettlementTimeout	Processing time exceeded the scheme limit.
	201	AC06	BlockedAccount	Creditor account EU123456789012 is closed.

[3] This description is not part of the response. It has only been added as an explanation of the status reason code.
[4] This additional information is not prescribed by the implementation specifications. The pilot PSP can choose how to best explain what has caused an error exactly.

EUROPEAN CENTRAL BANK

EUROSYSTEM

3. Secure Exchange of Payment Information (SEPI) Service

3.1. Service description

The SEPI Service (SEPI) plays a key role in enabling online beta digital euro payments with NFC (Near Field Communication) method.

Individual end users that have NFC capable devices can enrol to the service to perform proximity payments using their online holdings.

To facilitate NFC payments, SEPI supports various functions:

- End user enrolment and lifecycle management that are in the domain of the frontend specifications (refer to **Digital euro pilot – Frontend specifications**).
- Application cryptogram validation that is in scope of the back-end specifications and is detailed out in this document.

The following section provides a high-level summary of SEPI's role in the end to end process, offering greater context for its overall capabilities, while only the application cryptogram validation is in scope of this document.

Enrolment and lifecycle management

Upon end user enrolment, SEPI generates the enrolment data including a surrogate value and the first session key set. The session keys are later going to be used for application cryptogram generation.

SEPI is also responsible for surrogate value activation and deactivation and subsequent session key set generation.

SEPI itself does not directly interact with end user mobile devices. The data is deployed to the mobile device via the relevant HCE components.

Application cryptogram validation

This function is in scope of the back-end specifications for the distributing PSPs. The NFC payment transaction's uniqueness and authenticity is validated by SEPI with the support of an application cryptogram. The application cryptogram is generated during each NFC payment transaction.

The application cryptogram is generated following a successful end user authentication and completion of data exchange between the payee and payer devices based on the CPACE standard.

EUROPEAN CENTRAL BANK
EUROSYSTEM

The HCE SDK generates the application cryptogram using a set of inputs, referred to as cryptogram metadata, that is defined in the form of standard CPACE tags. The list of the required tags can be found in the POST/cryptogram-validations request section.

The HCE SDK uses the metadata structured in CDOL1 value (see cryptogram validations endpoint) and one session key as input for application cryptogram generation. To verify the transaction, the distributing PSP sends an application cryptogram validation request to SEPI, including the token information, the application cryptogram, and the underlying data.

Upon receiving these inputs, SEPI recalculates the application cryptogram. If the recalculated application cryptogram matches the received application cryptogram, the transaction's authenticity is confirmed back to the distributing PSP.

NFC high-level payment flow

Below diagram is a simplified schematic flow of a NFC transaction. It is designed to emphasise the specific aspects of NFC functionality while abstracting away the complexities of various potential configurations in terms of actors, roles, and related processes.

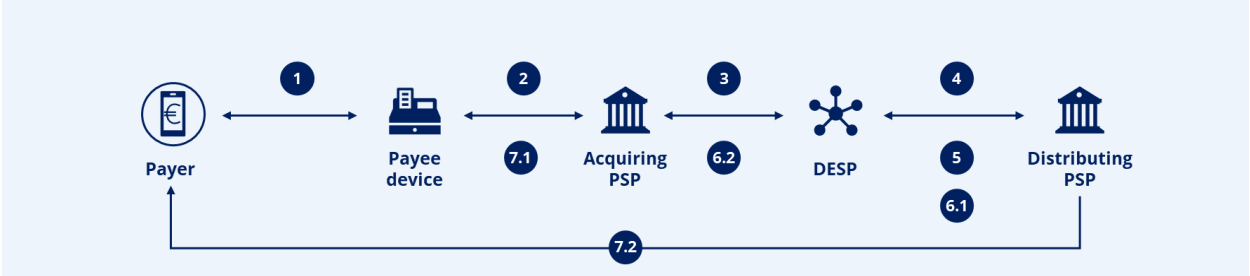


Figure 2 Schematic flow NFC transaction

Step	Description
1	• Payee enters amount to payee device. • Payer authenticates and activates the NFC payment function and taps mobile device to payee device. • Payer and payee device interaction happens based on CPACE protocol. • Mobile device generates application cryptogram (AC) and provides it to payee device.
2	• Payee device sends a payment initiation request to the acquiring PSP.
3	• Acquiring PSP sends a payment request to DESP.
4	• Based on the distributing PSP ID, DESP routes the request to the relevant pilot PSP.
5	• The distributing PSP sends an application cryptogram validation request to SEPI. • Following a successful application cryptogram validation response from SEPI (and other controls), the distributing PSP confirms the payment request to DESP.

EUROPEAN CENTRAL BANK

EUROSYSTEM

6.1 6.2	• DESP settles the payment transaction and sends status updates both to acquiring and the distributing PSPs.
7.1 7.2	• Acquiring PSP and distributing PSP inform the payee device and payer devices respectively on the successful transaction outcome.

3.2. Function description

Application cryptogram validation

The distributing PSP populates the necessary data in the application cryptogram validation request to SEPI from the inputs it previously received from the payment request. SEPI recalculates the application cryptogram and provides the synchronous response to the distributing PSP. A positive validation result confirms the transaction's authenticity.

3.3. Interface description

This chapter provides a detailed description of the interface designed to facilitate communication between pilot PSPs and the DESP for the SEPI service. The interface is based on RESTful API and the data format utilised is JSON (JavaScript Object Notation).

3.3.1. Overview of covered scenarios

Figure 3 Application cryptogram validation flow

API Call	Description	Covered in section
1	Payer PSP sends the application cryptogram validation request to DESP.	POST /cryptogram-validations – Request
2	Synchronous response from DESP to provide the application cryptogram validation outcome.	POST /cryptogram-validations – Response

3.3.2. Overview of API endpoints

DESP API endpoints

For pilot PSPs to submit requests to DESP:

API Nr	Description	Sending Party	Exposing Party	Covered in section
1	Endpoint for the payer PSP to send a request for application cryptogram validation.	Pilot PSP	DESP	DESP API for pilot PSPs to send requests to DESP

EUROPEAN CENTRAL BANK
EUROSYSTEM

PSP API endpoints

For DESP to send requests to pilot PSPs.

Not applicable.

3.4. DESP API for pilot PSPs to send requests to DESP

Method	Endpoint	Resource	Description
POST	/v1/cryptogram-validations	Cryptogram	Endpoint for the payer PSP to send a request for cryptogram validation.

3.4.1. POST /cryptogram-validations – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Request body data elements

Data element	Element description	Type	Length	Presence indicator
Token (surrogate value)	Token key of payer details. (5A)	NUM	16	M
Cryptogram envelope	Concatenated datapoints and tags encoded in base64 format in the below order (as received in the payment request). Preliminary list: 1. Token – surrogate value (5A) 2. Issuer application data (9F10) 3. Application transaction counter – ATC (9F36) 4. Cryptogram information data (9F27) 5. EMV cryptogram (9F26) 6. CDOL1 (Card data Object List 1) – data wrapped in 8C: • Transaction type 9C: • Transaction currency (5F24), • Application Interchange Profile (AIP – 82), • Terminal transaction date (9A), • Card verification method result (CVM – 9F34), • Amount, authorised (9F02), • Amount, other (9F03), • Unpredictable number (9F37)	STR	TBA	M

3.4.2. POST /cryptogram-validations – Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**

- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 200.

Data element	Element description	Type	Length	Presence indicator
Cryptogram validation status	Outcome of the application cryptogram validation.	SSET	4	M

EUROPEAN CENTRAL BANK

EUROSYSTEM

4. Access Management Service

4.1. Service description

The Access Management Service supports a set of end user-centric functions covering onboarding and lifecycle management of end users, their accounts and payment instruments.

The steps in the table that are out of scope of this specification are greyed out and are covered by **Digital euro pilot – Frontend specifications**).

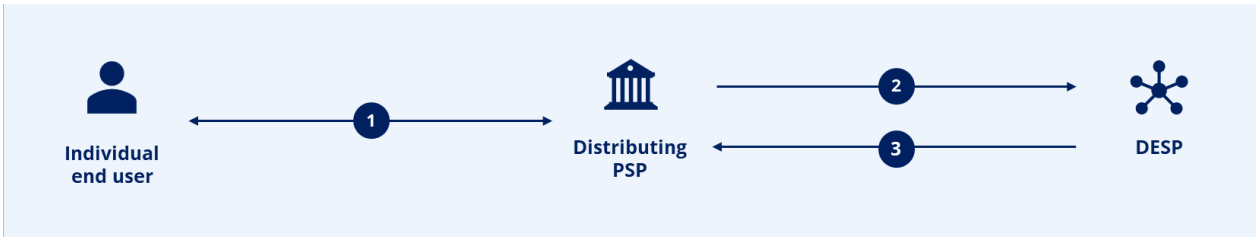


Figure 4 Access management high-level flow

Step	Description
1	Individual end user interacts with distributing PSP to set up or to manage the beta digital euro account.
2	The pilot PSP interacts with DESP to conduct necessary steps.
3	DESP returns onboarding or beta digital euro management responses.

4.2. Function Description

The end user onboarding process consists of a set of functions combined into the Access Management Service. The pilot PSP is allowed to manage only own data, i.e. data of own end users, their accounts and aliases.

End user registration and management (including offboarding)

Each natural or legal person being registered for pilot payment services is represented in DESP by a (hashed) end user identifier, generated by the pilot PSP.

The hashed end user identifier is the result of a defined algorithm based on unique end user‘s data as input.

Digital euro access number (DEAN) creation and management

The beta digital euro account opened by an end user with a pilot PSP is represented by a DEAN, generated by the DESP upon a request of the pilot PSP, and mandatory for each account. The DEAN allows end users and their pilot PSPs to be reachable in view of receiving payments. Each end user can have a defined

maximum number of accounts, which is limited to one for individual end users and is unlimited for business end users.

Proxy alias registration and management

In addition to the compulsory DEAN, the end user can define an optional proxy alias (e.g. MSISDN = a mobile phone number, hereafter an “alias”). A DEAN can have only one alias value per alias type, and any optional alias is assigned to only one DEAN.

4.3. Interface description

This section provides a detailed description of the interface designed to facilitate communication between pilot PSPs and the DESP for the Access Management Service. The interface is based on RESTful API principles and the data format utilised is JSON (JavaScript Object Notation).

4.3.1. Sequence diagrams

The section below includes examples of the scenarios supported by the Access Management Service, along with the corresponding interfaces. For each interface, a reference to the relevant section of this document, which contains details on the supported messages, is included.

4.3.1.1. Multiple (bulk) DEANs creation

A set of interfaces supporting the pilot PSP request for multiple (bulk) DEANs creation. This interface is optional for the pilot PSP. The pilot PSP can register those DEANs in its internal systems and assign a DEAN upon onboarding of an end user. The created DEANs are active and can be assigned by the pilot PSP to end users.

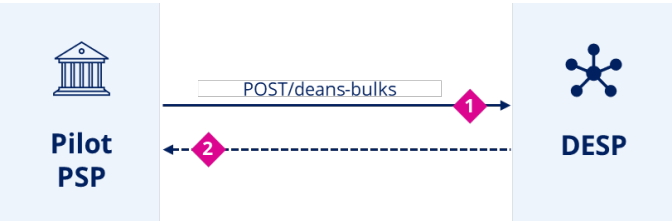


Figure 5 DEAN bulk creation[5]

API Call	Description	Covered in section
1	Pilot PSP sends a request to DESP to create multiple DEANs.	POST /deans-bulks - Request
2	Synchronous response from DESP containing a set of DEANs.	POST /deans-bulks - Response

[5] AM-1.4 Bulk DEAN request

EUROPEAN CENTRAL BANK

EUROSYSTEM

4.3.1.2. End user onboarding

A set of interfaces supporting onboarding end users to pilot payment services (with or without upfront bulk DEAN; with or without alias registration): (A) upfront or as part of the end user's onboarding and, in the latter case, whether the onboarding involves also alias registration (C) or not (B). In all cases, the assignment of the DEAN to the end user is done by the pilot PSP outside DESP.

A. The DEAN has been created upfront (see: *Multiple (bulk) DEANs creation*); the pilot PSP must register the end user in DESP and (optionally) registers alias for the end user's DEAN.

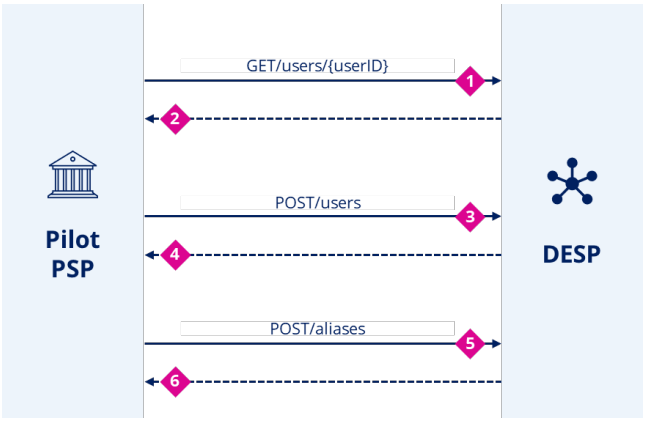


Figure 6 End user onboarding (DEAN created upfront)

API Call	Description	Covered in section
1	Pilot PSP sends a request to check whether a end user is registered in DESP or not.	GET /users/{userid} – Request
2	Synchronous response from DESP to confirm if an end user is registered in DESP.	GET /users/{userid} – Response
3	Pilot PSP sends a request to register the end user in DESP.	POST /users – Request
4	Synchronous response from DESP to the call in Step 3.	POST /users - Response
5	Pilot PSP sends a request to register an alias linked to an existing DEAN (if DEAN was created upfront in a bulk request – please see POST /deans-bulks).	POST /aliases - Request
6	Synchronous response from DESP to the call in Step 5.	POST /aliases - Response

EUROPEAN CENTRAL BANK

EUROSYSTEM

B. The DEAN has not been created upfront; the pilot PSP must register the end user in DESP and requests only a DEAN creation (alias is registered at a later point in time using POST /aliases).

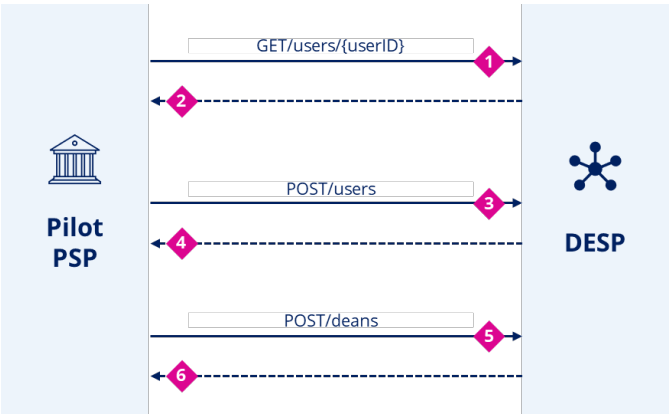


Figure 7 End user onboarding (DEAN not created upfront)[6]

API Call	Description	Covered in section
1	Pilot PSP sends a request to check whether an end user is registered in DESP or not.	GET /users/{userid} – Request
2	Synchronous response from DESP to confirm whether the end user is registered in DESP.	GET /users/{userid} – Response
3	Pilot PSP sends a request to register the end user in DESP.	POST /users – Request
4	Synchronous response from DESP to the call in Step 3.	POST /users - Response
5	Pilot PSP sends a request to generate a DEAN (no alias registration).	POST /deans - Request
6	Synchronous response from DESP to the call in Step 5.	POST /deans - Response

[6] Please see AM-1.1 Onboarding of an individual end user part I (online) and AM-1.2 Onboarding of a business end user (online and offline)

EUROPEAN CENTRAL BANK

EUROSYSTEM

C. The DEAN has not been created upfront; the pilot PSP must register the end user in DESP, pilot PSP requests a DEAN creation and alias registration.

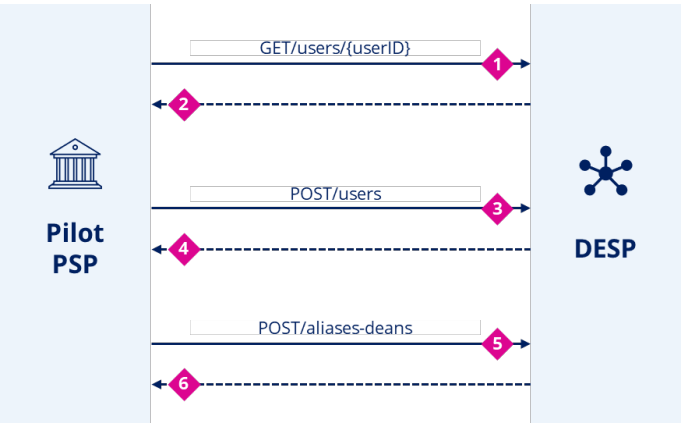


Figure 8 End user onboarding (DEAN not created upfront – request DEAN and alias)[7]

API Call	Description	Covered in section
1	Pilot PSP sends a request to check whether an end user is registered in DESP or not.	GET /users/{userid} – Request
2	Synchronous response from DESP to confirm whether the end user is registered in DESP.	GET /users/{userid} – Response
3	Pilot PSP sends a request to register the end user in DESP.	POST /users – Request
4	Synchronous response from DESP to the call in Step 3.	POST /users - Response
5	Pilot PSP sends a request to generate a DEAN and register an alias.	POST /aliases-deans - Request
6	Synchronous response from DESP to the call in Step 5.	POST /aliases-deans - Response

4.3.1.3. End user lifecycle management

A set of interfaces supporting the end user's lifecycle management (updates to end user's data registered in DESP), as well as an update of an optional alias. To note: for registering an alias for an end user that has already been onboarded, POST /aliases can be used, as described in previous section (point A).

[7] Please see AM-1.1 Onboarding of an individual end user part I (online) and AM-1.2 Onboarding of a business end user (online and offline).

Figure 9 End user lifecycle management – end user registration update

API Call	Description	Covered in section
1	Pilot PSP sends a request to update the end user registration data in DESP.	POST /users-updates – Request
2	Synchronous response from DESP to the call in Step 1.	POST /users-updates – Response

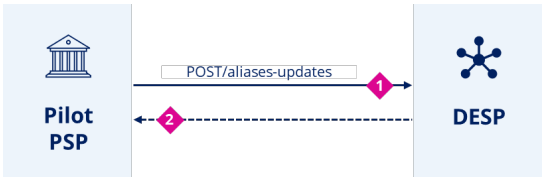


Figure 10 End user lifecycle management – Alias registration update[8]

API Call	Description	Covered in section
1	[OPTIONAL] Pilot PSP sends a request to update the end user's alias value.	POST /aliases-updates – Request
2	Synchronous response from DESP to the call in Step 1.	POST /aliases-updates – Response

4.3.1.4. End user offboarding

A set of interfaces supporting the offboarding of the end user from the pilot payment services.

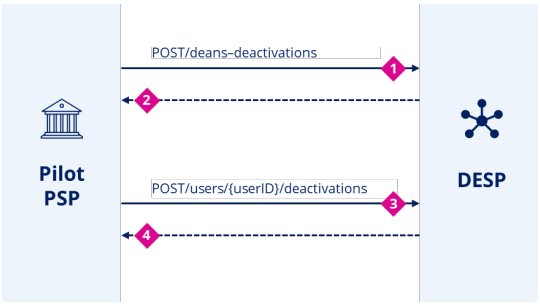


Figure 11 End user offboarding[9]

API Call	Description	Covered in section

[8] Please see AM-4.1.9 Individual end user amendments (alias registration).
[9] Please see AM-3.1 Offboarding of an individual end user (online and offline) and AM-3.2 Offboarding of a business end user (online and offline).

1	Pilot PSP sends a request to remove DEAN to pilot PSP mapping.	POST /deans-deactivations - Request
2	Synchronous response from DESP to the call in Step 1.	POST /deans-deactivations - Response
3	Pilot PSP sends a request to deregister user in DESP.	POST / users/{userid}/deactivations – Request
4	Synchronous response from DESP to the call in Step 3.	POST / users/{userid}/deactivations - Response

4.3.2. Overview of API endpoints

DESP API endpoints: For pilot PSPs to submit requests to pilot PSPs via DESP or to DESP.

API Nr	Description	Sending Party	Exposing Party	Covered in section
1	Verification whether the end user is already registered in DESP (end user lookup) or not.	Pilot PSP	DESP	GET /users/{userid}
2	Registration of the end user in DESP.	Pilot PSP	DESP	POST /users
3	Update of end user registration data in DESP.	Pilot PSP	DESP	POST / users-updates
4	Offboarding of an end user in DESP.	Pilot PSP	DESP	POST / users/{userid}/deactivations
5	Generation of a single DEAN.	Pilot PSP	DESP	POST /deans
6	Generation of multiple DEANs (bulk). *Note: DEANs are active and can be assigned by the pilot PSP to users directly.*	Pilot PSP	DESP	POST /deans-bulks
7	Deactivation of a DEAN in DESP. *Note: The alias linked to the DEAN is deactivated at the same time.*	Pilot PSP	DESP	POST /deans-deactivations
8	Generation of a single DEAN and registration of a new alias.	Pilot PSP	DESP	POST /aliases-deans
9	Registration of an alias for an existing DEAN.	Pilot PSP	DESP	POST /aliases
10	Update of an existing alias value.	Pilot PSP	DESP	POST /aliases-updates

4.3.2.1. DESP API for pilot PSPs to send requests to DESP

4.3.2.1.1. POST /users-lookups

Method	Endpoint	Resource	Description
GET	/v1/users-lookups	End user	Lookup of end user registration in DESP.

4.3.2.1.1.1. GET /users/{user-id} – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Data element	Element description	Type	Presence indicator
User-ID	Unique end user identifier generated by the pilot PSP based on a subset of mandatory attributes in the PID, according to the latest draft of the eIDAS Architecture Reference Framework (TBD).	User ID	M

Request Body

Not applicable.

4.3.2.1.1.2. POST /users-lookups – Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: HTTP 404. Please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 200.

Data element	Description	Type	Presence indicator
userId	Unique end user identifier. Generated by the pilot PSP based on a subset of mandatory attributes in the PID, according to the latest draft of the eIDAS Architecture Reference Framework (TBD).	User ID	M
userType	Indicates the type of end user as specified by the pilot PSP (individual end user or business end user).	User type	M
creationDateTime	End user creation date, current date.	ISODateTime	C
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN for the end user that is provided in the request.	Financial institution Identification1	M

4.3.2.1.2. POST /users

Method	Endpoint	Resource	Description
POST	/v1/users	End user	End user registration in DESP.

4.3.2.1.2.1. POST /users – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Request Body

Data element	Description	Type	Presence indicator
userId	Unique end user identifier generated by the pilot PSP based on a subset of mandatory attributes in the PID, according to the latest draft of the eIDAS Architecture Reference Framework (TBD).	User ID	M
userType	Indicates the type of end user as specified by the pilot PSP (individual end user or business end user).	User type	M
hashedTechnicalProof	The hashed technical proof is generated by the pilot PSP and submitted to DESP. The data is stored in DESP.	Hashed Technical Proof	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the end userId provided in the request.	Financial institution Identification1	M

4.3.2.1.2.2. POST /users - Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 201.

Data element	Description	Type	Presence indicator

userId	Unique end user identifier generated by the pilot PSP based on a subset of mandatory attributes in the PID, according to the latest draft of the eIDAS Architecture Reference Framework (TBD).	User ID	M
creationDateTime	End user creation date, current date.	ISODateTime	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the end userId provided in the request.	Financial institution Identification1	M

4.3.2.1.3. POST /users-updates

Method	Endpoint	Resource	Description
POST	/v1/users-updates	User	Update of a user identifier in DESP.

4.3.2.1.3.1. POST /users-updates – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Request Body

Data element	Description	Type	Presence indicator
oldUserId	Unique end user identifier generated by the pilot PSP based on a subset of mandatory attributes in the PID, according to the latest draft of the eIDAS Architecture Reference Framework (TBD).	User ID	M
newUserId	Unique end user identifier generated by the pilot PSP based on a subset of mandatory attributes in the PID, according to the latest draft of the eIDAS Architecture Reference Framework (TBD).	User ID	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M

4.3.2.1.3.2. POST /users-updates – Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.

- In case of positive technical validation: HTTP Code 200.

Data element	Description	Type	Presence indicator
userId	Unique end user identifier generated by the pilot PSP based on a subset of mandatory attributes in the PID, according to the latest draft of the eIDAS Architecture Reference Framework (TBD).	User ID	M
updateDateTime	End user update date, current date.	ISODateTime	M

4.3.2.1.4. POST /users-deactivations

Method	Endpoint	Resource	Description
POST	/v1/ users-deactivations	User	Offboarding an end user from a pilot PSP in DESP.

4.3.2.1.4.1. POST /users-deactivations – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header**:** not applicable.

Request Body

Data element	Description	Type	Presence indicator
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the user Id provided in the request.	Financial institution Identification1	M
userId	Unique end user identifier generated by the pilot PSP based on a subset of mandatory attributes in the PID, according to the latest draft of the eIDAS Architecture Reference Framework (TBD).	User ID	M

4.3.2.1.4.2. POST /users-deactivations - Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.

- In case of positive technical validation: HTTP Code 200.

Data element	Description	Type	Presence indicator
deactivationDateTime	End user deactivation date, current date.	ISODateTime	M

4.3.2.1.5. POST /deans

DEAN generation by DESP

Method	Endpoint	Resource	Description
POST	/v1/deans	DEAN	Requesting generation of a single DEAN in DESP.

4.3.2.1.5.1. POST /deans – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Request Body

Data element	Description	Type	Presence indicator
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M
indicatorDigit	An indicator specifying a type of end user a DEAN is requested for.	Indicator Digit	M

4.3.2.1.5.2. POST /deans - Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 201.

Data element	Description	Type	Presence indicator
account	Unique identifier of the beta digital euro account.	Account Reference	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M

creationDateTime	End user creation date, current date.	ISODateTime	M

4.3.2.1.6. POST /deans-bulks

Bulk DEAN creation

Method	Endpoint	Resource	Description
POST	/v1/deans-bulks	DEAN	Generation of multiple DEANs (bulk). *Note: DEANs are active and can be assigned by the pilot PSP to end users directly.*

4.3.2.1.6.1. POST /deans-bulks – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Request Body

Data element	Description	Type	Presence indicator
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN(s).	Financial institution Identification1	M
numberOfDeanss	Number of requested DEANs. Maximum numbers per request = 100.	Number	M
indicatorDigit	An indicator specifying a type of end user a DEAN is requested for.	Indicator Digit	M

4.3.2.1.6.2. POST /deans-bulks - Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 201.

Response body returns an array of data elements specified below:

Data element	Description	Type	Presence indicator
account	Unique identifier of the beta digital euro account.	Account Reference	M

digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M
creationDateTime	DEAN creation date, current date.	ISODateTime	M

4.3.2.1.7. POST /deans-deactivations

DEAN deactivation

Method	Endpoint	Resource	Description
POST	/v1/deans-deactivations	DEAN	Deactivation of DEAN(s) in DESP. *Note: The alias linked to the DEAN is deactivated at the same time.*

4.3.2.1.7.1. POST /deans-deactivations – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Request Body Data element	Description	Type	Presence indicator
account	Unique identifier of the beta digital euro account, associated with the alias (multiple values allowed).	Account Reference	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M

4.3.2.1.7.2. POST /deans-deactivations - Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 200.

Response Body (to be repeated for each DEAN request)

Data element	Description	Type	Presence indicator
account	Unique identifier of the beta digital euro account.	Account Reference	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M

deactivationDateTime	User update date, current date.	ISODateTime	M

4.3.2.1.8. POST /aliases

Alias registration

Method	Endpoint	Resource	Description
POST	/v1/aliases	Proxy alias	Registration of a new alias for an existing DEAN.

4.3.2.1.8.1. POST /aliases – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Request Body

Data element	Description	Type	Presence indicator
alias	The optional alias associated with the end user's DEAN.	Proxy Account Identification	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M
account	Unique identifier of the beta digital euro account, associated with the alias.	Account Reference	M

4.3.2.1.8.2. POST /aliases - Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 201.

Data element	Description	Type	Presence indicator
account	Unique identifier of the beta digital euro account, associated with the alias.	Account Reference	M
alias	The optional alias associated with the end user's DEAN.	Proxy Account Identification	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M
creationDateTime	Alias creation date, current date.	ISODateTime	M

EUROPEAN CENTRAL BANK

EUROSYSTEM

4.3.2.1.9. POST /aliases-deans

Alias registration and DEAN creation (simultaneous)

Method	Endpoint	Resource	Description
POST	/v1/aliases-deans	Proxy alias	Simultaneous registering of a new alias and requesting a DEAN creation in DESP.

4.3.2.1.9.1. POST /aliases-deans – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Request Body

Data element	Description	Type	Presence indicator
alias	The optional alias associated with the end user's DEAN.	Proxy Account Identification	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M
indicatorDigit	An indicator specifying a type of end user a DEAN is requested for.	Indicator Digit	M

4.3.2.1.9.2. POST /aliases-deans - Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 201.

Data element	Description	Type	Presence indicator
account	Unique identifier of the beta digital euro account, associated with the alias.	Account Reference	M
alias	Value of an alias, currently supported alias type is mobile phone number (MSISDN format).	Proxy Account Identification	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M

creationDateTime	Beta digital euro account creation date and alias registration date, current date.	ISODateTime	M

4.3.2.1.10. POST /aliases-updates

Alias management

Method	Endpoint	Resource	Description
POST	/v1/aliases-updates	Alias	Update of an existing alias value (and type).

4.3.2.1.10.1. POST /aliases-updates – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Request Body

Data element	Description	Type	Presence indicator
account	Unique identifier of the beta digital euro account, which alias is associated to.	Account Reference	M
currentAlias	Value of an alias, currently supported alias type is mobile phone number (MSISDN).	Proxy Account Identification	M
newAlias	Value of an alias, currently supported alias type is mobile phone number (MSISDN).	Proxy Account Identification	M

4.3.2.1.10.2. POST /aliases-updates – Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 200.

Data element	Description	Type	Presence indicator
newAlias	Value of an alias, currently supported alias type is mobile phone number (MSISDN).	Proxy Account Identification	O

updateDateTime	Alias update date, current date.	ISODateTime	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M

4.3.3. Exception management

The table below contains a non-exhaustive list of errors, the codes used and some examples of when these errors may apply.

Error code	HTTP code	Type	Title	Detail (example)
ALLOWED_VALUE_ERROR*	400	Value-not-allowed	Value provided is not allowed	Alias type 'EMAL' in alias registration request.
FORMAT_ERROR	400	Missing-field	Mandatory field missing	Alias type missing in alias registration request.
		Data-format-invalid	Invalid data format	Alias value '1234' with alias type 'MBNO'.
		Data-type-invalid	Invalid data type	Alias type 'MAIL' is not a correct value.
		Data-range-invalid	Invalid data range	
		Format-incorrect	Incorrect format	Alias type 'EMAIL' in alias registration request.
		Type-format-incorrect	Invalid type format	ISODateTime containing alphanumeric characters.
ALIAS_NOT_FOUND	404	Resource-not-found	Resource not found	The alias does not exist.
LIMIT_REACHED	400	Limit-reached	Maximum number of contractual arrangements for the user has been reached.	Request to register an individual end user that already has a beta digital euro account.
ALREADY_EXISTS	400	Resource-exists	Resource already exists	The alias to be registered already exists in DESP.

EUROPEAN CENTRAL BANK
EUROSYSTEM

5. Alias lookup service

5.1. Service description

The Alias Lookup Service supports different pilot payment services in scope of the digital euro pilot:

1. P2P payment (via DEAN)

Individual end user (payer) pays another individual end user using the DEAN of the payee. The pilot PSP ID of the payee needs to be retrieved in order to route the payment to the correct pilot PSP. The steps in the table that are out of scope of this specification are greyed out and are covered in **Digital euro pilot – Frontend specifications**.

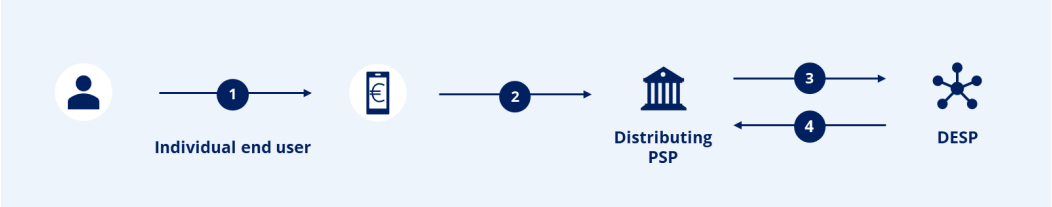


Figure 12 P2P payment (via DEAN)

Step	Description
1	Individual end user enters payee's DEAN.
2	DEAN is sent to payer's PSP.
3, 4	Payer's PSP requests and receives from the DESP a pilot PSP ID linked to the DEAN.

2. P2P payment (via an alias)

An individual end user (payer) pays other individual end user using the alias of the payee. The distributing PSP looks up the alias of the payee in the back-end to retrieve an associated DEAN and payee PSP ID, necessary for routing, before instructing the payment. The steps in the table that are out of scope of this specification are greyed out and are covered in **Digital euro pilot – Frontend specifications**.

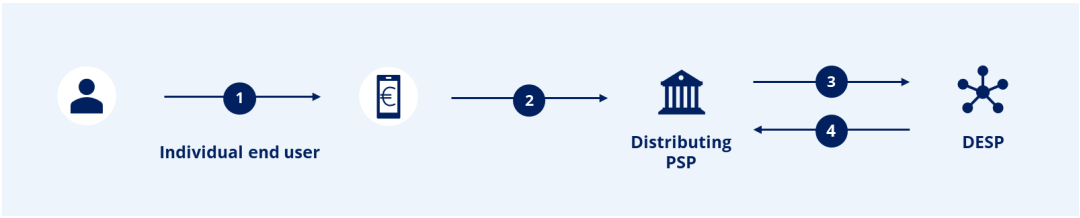


Figure 13 P2P payment (via alias)

EUROPEAN CENTRAL BANK

EUROSYSTEM

Step	Description
1	Individual end user enters payee's alias.
2	Individual end user submits an alias payment request to a pilot PSP.
3, 4	Pilot PSP requests and receives a resolved alias from DESP.

E-/m-commerce payment (via an alias)

3.

Business end user accepts beta digital euro payments via an alias and the acquiring PSP looks up the alias of the payer in the back-end to retrieve an associated DEAN and payer PSP ID, necessary for routing, before requesting the payment. The steps in the table that are out of scope of this specification are greyed out and are covered in **Digital euro pilot – Frontend specifications**.

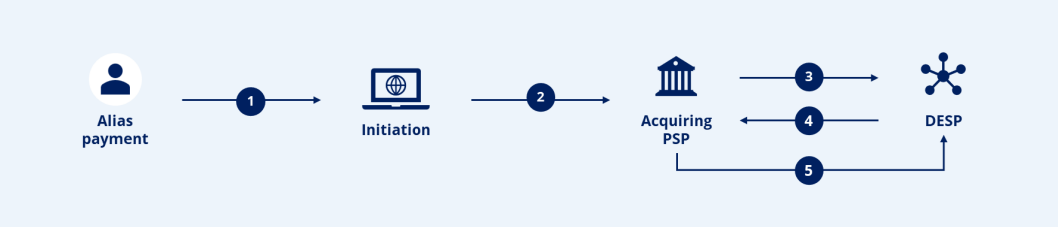


Figure 14 E-/m-commerce payment (via alias)

Step	Description
1	Individual end user enters their alias.
2	Business end user initiates payment request based on provided alias.
3, 4	Acquiring PSP requests alias resolution.
5	Without passing the resolved alias (DEAN and pilot PSP ID) back to business end user, the pilot PSP requests the payment.

4. E-/m- commerce payment (via DEAN)

Business end user accepts beta digital euro payments via DEAN and the acquiring PSP looks up the pilot PSP ID of the payer in the back-end to resolve it before requesting the payment. The steps in the table that are out of scope of this specification are greyed out and are covered in **Digital euro pilot – Frontend specifications**.

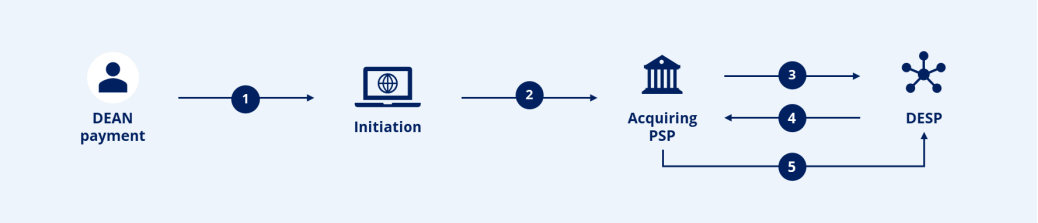


Figure 15 E-/m-commerce payment (via DEAN)

EUROPEAN CENTRAL BANK

EUROSYSTEM

Step	Description
1	Individual end user enters their DEAN.
2	Business end user initiates payment request based on the provided DEAN.
3, 4	Acquiring PSP requests and receives a pilot PSP ID linked to the DEAN from the DESP.
5	The acquiring PSP requests the payment.

5.2. Function description

Payer initiated payment with alias

- An end user can optionally use an additional alias for P2P and remote payments instead of a DEAN.
- An alias can be assigned to only one DEAN.
- Only one alias value is allowed per alias type.
- Only the mobile phone number (“MSISDN”) is allowed as additional alias type.

Payer-initiated payments with DEAN

- For payer-initiated payments using DEAN, the pilot PSP of the payer will use the Alias Lookup Service to retrieve the pilot PSP ID of the payee transaction to ensure the payment request can be routed to the right party.

5.3. Interface description

This section provides a detailed description of the interface designed to facilitate communication between pilot PSPs and the DESP for the Alias Lookup Service. The interface is based on RESTful API principles and the data format utilised is JSON (JavaScript Object Notation).

5.3.1. Sequence diagrams

5.3.1.1. Alias mapping lookup

Figure 16 Alias mapping flow[10]

API Call	Description	Covered in section

[10] Please see TM-2.2 E-commerce payment with alias or DEAN, TM-3.5 P2P payment with alias (payer-initiated)

EUROPEAN CENTRAL BANK

EUROSYSTEM

1	Pilot PSP sends a request to retrieve a DEAN and pilot PSP ID using an alias as a body parameter.	POST / lookups/aliases – Request
2	Synchronous response from DESP with requested details in call 1.	POST / lookups/aliases – Response

5.3.1.2. DEAN mapping lookup

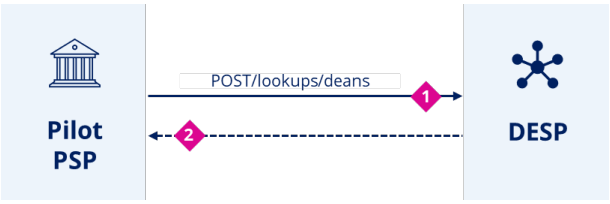


Figure 17 DEAN mapping flow[12]

API Call	Description	Covered in section
1	Pilot PSP sends a request to retrieve pilot PSP ID using DEAN as a body parameter.	POST /lookups/deans – Request
2	Synchronous response from DESP with requested details in call 1.	POST / lookups/deans – Response

5.3.2. Overview of API endpoints

DESP API endpoints: For pilot PSPs to submit requests to pilot PSPs via DESP or to DESP.

API Nr	Description	Sending Party	Exposing Party	Covered in section
1	A request to retrieve a DEAN and pilot PSP ID using an alias as a body parameter.	Pilot PSP	DESP	POST /lookups/alias
2	A request to retrieve pilot PSP ID using DEAN as a body parameter.	Pilot PSP	DESP	POST /lookups/dean

PSP API endpoints: For DESP to push or forward messages to pilot PSPs.

Not applicable.

5.3.3. DESP API for pilot PSPs to send requests to DESP

5.3.3.1. POST / lookups/aliases

Method	Endpoint	Resource	Description
POST	/v1/lookups/aliases	Alias	P2P Alias lookup request to retrieve a DEAN using an alias as a body parameter.

5.3.3.1.1. POST / lookups/aliases – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Request body

Data elements for requests supporting payer and payee initiated payment with an **alias**.

Data element	Description	Type	Presence indicator
alias	Value of an alias. Currently the only supported alias type is a mobile phone number.	Proxy Account Identification	M

5.3.3.1.2. POST / lookups/aliases – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 200.

Data element	Description	Type	Presence indicator
alias	Value of an alias, currently only supported alias type is a mobile phone number (MSISDN format).	Proxy Account Identification	M
account	Payer's (P2B or P2P payment) or payee's (P2P payment) DEAN.	Account Reference	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M

5.3.3.2. POST /lookups/deans

Method	Endpoint	Resource	Description
POST	/v1/lookups/deans	Alias	Pilot PSP ID or alias lookup using a DEAN as a body parameter.

5.3.3.2.1. POST /lookups/deans – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

EUROPEAN CENTRAL BANK
EUROSYSTEM

Request body

Data element	Description	Type	Presence indicator
account	The DEAN of the end user whose pilot PSP ID is requested.	Account Reference	M

5.3.3.2.2. POST / lookups/deans – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 200.

Data element	Description	Type	Presence indicator
account	The DEAN of the end user whose pilot PSP ID is requested.	Account Reference	M
digitalEuroAgent	The identifier (BIC) of the pilot PSP servicing the DEAN provided in the request.	Financial institution Identification1	M

5.3.4. Error handling

The table below contains a non-exhaustive list of errors, the codes used and some examples of when these errors may apply.

Error code	HTTP code	Type	Title	Detail (example)
ALLOWED_VALUE_ERROR*	400	Value-not-allowed	Value provided is not allowed	Alias type 'EMAL' in alias registration request.
FORMAT_ERROR	400	Missing-field	Mandatory field missing	Alias type missing in alias lookup request.
		Data-format-invalid	Invalid data format	Alias value '1234' with alias type 'MBNO'.
		Data-type-invalid	Invalid data type	Alias type 'MAIL' is not a correct value.
		Format-incorrect	Incorrect format	Alias type 'EMAIL' in alias lookup request.
ALIAS_NOT_FOUND	404	Resource-not-found	Resource not found	The alias does not exist.

EUROPEAN CENTRAL BANK

EUROSYSTEM

6. Settlement Service (SE)

6.1. Service description

The DESP Settlement Service (SE) is responsible for the settlement verification (i.e., checking that a certain amount can be transferred between online beta digital euro and/or dedicated cash account (DCA) holdings) and settlement recording (i.e., irrevocably and unconditionally transferring a certain amount of beta digital euro and/or DCA holdings). The SE in DESP therefore covers not only payments in beta digital euro but also the functionalities of funding and defunding. Based on the information recorded for the settlement of the types of transactions mentioned above, the SE provides also settlement reporting (i.e., preparing the necessary settlement confirmation and rejection messages).

The objective of this section is to outline the functionalities that can be invoked by pilot PSPs involved in beta digital euro payment transactions, including the encrypted communication between instructing and instructed parties authorising a transaction[11]. **Section 6.2 Functional description** describes the types of transactions processed by DESP. These include payment, funding, defunding, combined (a payment to be made together with funding and/or defunding) and refunds. The outcome of processing a transaction is therefore the transfer of funds from the payer, whose entry[12] is debited, to the payee, whose entry is credited.

A transaction may contain payment data[13], settlement data[14], or both. The payment process starts when the first API call[15] with the payment data (and settlement data, if initiated by the payer PSP), which associates an initial acceptance timestamp to the transaction, reaches DESP. If the first API call, which contains payment data, is sent by the payee PSP, then it is a payee-initiated payment. If the first API call containing payment data is sent by the payer PSP, then it is a payer-initiated payment, and it also contains settlement data.

Beta digital euro transactions must be settled atomically: either the whole transaction is settled, or nothing is settled. Consequently, a beta digital euro settlement instruction can only be settled by DESP once it contains data on applicable parts of the foreseen settlement recording (funding, defunding and payment). Since a beta digital euro payment transaction may involve more than one pilot PSP, the pilot PSP who initiates the payment transaction may lack some necessary data to complete it. In such scenario, the end-

[11] Please refer to digital euro glossary.
[12] Entry is either an end user entry, which represents the identifier of a beta digital euro holding in the settlement ledger, or a DCA entry, which represents the identifier of a dedicated cash account in the settlement ledger.
[13] Payment data is the data exchanged between the pilot PSPs, describing the payment (incl. amount, remittance information), accounts and parties involved, required by the pilot PSPs to process the transaction.
[14] Settlement data is the minimal data set required for DESP to process the transaction. It includes at least entry of both payer and payee, amount, acceptance timestamp and fee calculation related data.
[15] A list of the API calls for initiating a transaction is specified in section 6.3.3 **DESP API endpoints**.

to-end process flows foresee a role for DESP to facilitate the build-up of the settlement instruction without back-and-forth of messages among pilot PSPs, and no need for one of them to submit the completed instruction for all parties involved. Instead, the pilot PSP initiating the transaction sends the API request with partial settlement data to DESP that then takes care of compiling the overall settlement instruction. To do so, DESP forwards only the necessary information to other pilot PSPs involved in the transaction, in line with the data minimisation principle, and collects the parts of their responses that are necessary for settlement. Throughout the settlement process, other participants in the transaction chain, such as payee PSPs, contribute with additional settlement data. This combined information enables DESP to compile a comprehensive settlement instruction with complete settlement data that allows atomic settlement, ensuring all aspects of the transaction, including any (de)funding, are settled simultaneously. This process applies equally to refunds and funding or defunding transactions that involve multiple pilot PSPs.

Overview of pilot PSPs' roles in settlement

The settlement of a beta digital euro transaction requires pilot PSPs to perform two or four different roles:

(1) Payer's PSP,

(2) Payee's PSP,

(3) Commercial bank money payer's PSP (same as (1) during digital euro pilot)

(4) Commercial bank money payee's PSP (same as (2) during digital euro pilot)

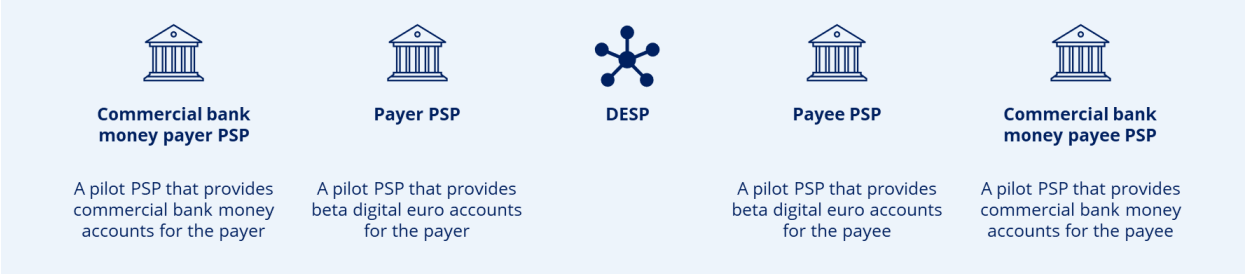


Figure 18 Pilot PSP role in settlement of beta digital euro

A pilot PSP needs to use different endpoints, each with a different set of data included in the corresponding requests and responses, depending on the role it plays. The introduction to the section **6.3 Interface description** outlines the structure of a settlement API request/response payload. The required payload parts, including their objects, depend on the transaction type and on the role of the pilot PSP.

6.2. Functional description

The section below shows the different types of transactions that pilot PSPs can support.

Funding transaction

Funding is the process of increasing an end user's holdings in their account or device through beta digital euro issuance, in combination with a reduction of another liquidity source from the end user (commercial bank money).

Defunding transaction

Defunding is the process of reducing an end user's holdings in their account or device through beta digital euro redemption, in combination with an increase of end user's commercial bank money.

Payment transaction

A beta digital euro payment transaction is initiated by either a payer or payee PSP and confirmed by the corresponding pilot PSP. DESP performs settlement once the settlement data and confirmation are received from both payer and payee PSPs.

Combined payment transaction

A combined transaction is a beta digital euro payment transaction involving payment with funding (*reverse waterfall* - if the end user does not have sufficient holdings) or payment with defunding (*waterfall* - if the end user's holding limit is reached).

Refund transaction

A refund is a reversal (in whole or in part) of an earlier payment initiated by a payee, usually at the payer's request. A refund always requires funding if the original payee's funds are not sufficient to settle the refund and may require defunding on the original payer's side. DESP performs settlement once the settlement data and confirmation are received from both payer and payee PSPs.

6.3. Interface description

This section provides a detailed description of the interface designed to facilitate communication between pilot PSPs and the DESP for the SE. The interface is based on RESTful API principles and the data format utilised is JSON (JavaScript Object Notation). The data dictionary, including complex data types, is based on ISO20022 where available.

The structure of the request and response for the settlement service can be summarised as follows:

- **Header** contains metadata about the request. It includes generic data elements that are used in all API endpoints, such as content-type, and specific values only used for certain endpoints.

- **Payload:**
  - **Root** contains top-level data elements of the JSON payload structure, such as UETR, acceptance timestamp and objects.
  - **Objects** contain collections of data elements and are nested within the root.

Payload structure summary for settlement requests

Element	Encryption	Condition to be included in the payload
Root		
UETR	Unencrypted or encrypted[16]	Mandatory for all transactions.
Acceptance Date Time	Unencrypted or encrypted[17]	Mandatory for all transactions.
Objects		
Payment Instruction object	Unencrypted	Mandatory in case the message concerns a payment initiated by the payer PSP.
	Encrypted	Included if payer and payee PSPs are different parties, as the content needs to be forwarded to the other pilot PSP.
Payment object	Encrypted	Mandatory in case the message concerns a payment request initiated by the payee PSP.
Refund object	Unencrypted	Mandatory in case the message concerns a refund transaction.
	Encrypted	Included if payer and payee PSPs are different parties and that the content needs to be forwarded to the other pilot PSP.
Funding object	Unencrypted	Only required if the payer's balance is lower than the transaction amount. Mandatory in case the message concerns a funding transaction.
Defunding object[18]	Unencrypted	Only required if the transaction results in the payee's digital euro balance exceeding payee's holding limit. Mandatory in case the message concerns a defunding transaction.

Similarly, which data elements are present in settlement notifications and responses received by a pilot PSP depends on the role of the pilot PSP and the context of the transaction.

Additional status elements for settlement notifications and responses

Element	Condition to be included in the payload
Payment Instruction ID and status	Mandatory in case the transaction request contains a payment instruction object. Payment instruction ID is present only if the validation result is positive.

[16] UETR is encrypted in endpoints for payee-initiated payments, i.e., POST /payments, POST /payments/{payment-id}/status and GET /payments/{payment-id}/status.

[17] Acceptance timestamp is encrypted in endpoints for payee-initiated payments, i.e., POST /payments, POST /payments/{payment-id}/status.

[18] A defunding object could be included in the response (from the pilot PSP to DESP only) to a payment instruction and refunds, if defunding is required.

	Only received by the payer's and payee's PSP.
Refund ID and status	Mandatory in case the transaction request contains a refund object. Refund ID is present only if the validation result is positive. Only received by the payer's and payee's PSP.
Funding ID and status	Mandatory in case the transaction request contains a funding object. Funding ID is present only if the validation result is positive. Only received by the requester and receiver of the funding request (the pilot PSP).
Defunding ID and status	Mandatory in case the transaction request contains a defunding object. Defunding ID is present only if the validation result is positive. Only received by the requester and receiver of the defunding request (the pilot PSP).

6.3.1. Sequence diagrams

This section presents a selection of baseline scenarios that cover the key flows in the beta digital euro settlement process, taking into account:

- the type of transaction ((de)funding, (combined) payment, refund);
- the party that initiates the process (PSP of payer or payee);
- the role a pilot PSP performs in the process which may vary depending on the end user type (e.g. individual end user or business end user) and the transaction setup (e.g. single PSP vs multiple pilot PSPs involved);

The description of API calls in sections (**7.3.3 DESP APIs endpoints for PSPs to send requests to DESP** and **7.4.4 PSP APIs endpoints for PSPs to receive requests to DESP**) includes information on whether the steps in the process are conditional (e.g. for funding and defunding).

6.3.1.1. Funding and defunding transactions

6.3.1.1.1. Funding transaction

The sequence diagram below presents a funding process, in which the pilot PSP and commercial bank money PSP are the same and all necessary settlement data can be shared with DESP in one API call.

EUROPEAN CENTRAL BANK

EUROSYSTEM

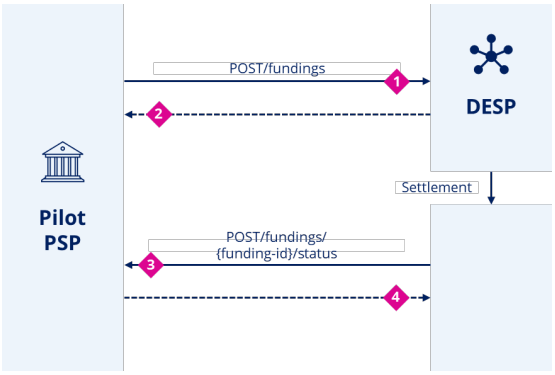
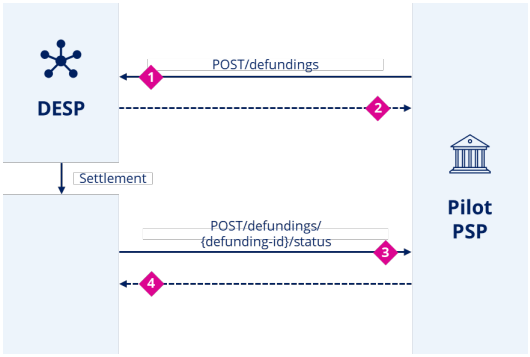


Figure 19 Funding transaction

API Call	Description	Covered in section
1	Pilot PSP sends the request with complete settlement data to DESP.	DESP API POST /fundings - Request
2	Synchronous[19] response from DESP to confirm receipt of the request.	DESP API POST /fundings - Response
3	DESP sends the status of the funding transaction back to pilot PSP.	PSP API POST /fundings/{funding-id}/status - Request
4	Synchronous response from the pilot PSP to confirm receipt of the status notification.	PSP API POST /fundings/{funding-id}/status - Response

6.3.1.1.2. Defunding transaction

The sequence diagram below presents a defunding process, when the pilot PSP and commercial bank money PSP are the same.



[19] Synchronous API means that the client sends a request to the server and waits until it receives the response. During this time, the client cannot perform other operations or send additional requests on the same transaction.

EUROPEAN CENTRAL BANK
EUROSYSTEM

Figure 20 Defunding transaction

API Call	Description	Covered in section
1	Pilot PSP sends the request with complete settlement data to DESP.	DESP API POST /defundings - Request
2	Synchronous response from DESP to confirm receipt of the request.	DESP API POST /defundings - Response
3	DESP sends status of the defunding transaction back to pilot PSP.	PSP API POST /defundings/{defunding-id}/status - Request
4	Synchronous response from pilot PSP to confirm receipt of the status notification.	PSP API POST /defundings/{defunding-id}/status - Response

6.3.1.2. Payment transactions

6.3.1.2.1. Payment transaction with same pilot PSP for payer and payee

The sequence diagram below presents a payment process in which the pilot PSP for a payee and a payer is the same, and no funding/defunding of beta digital euro is required.

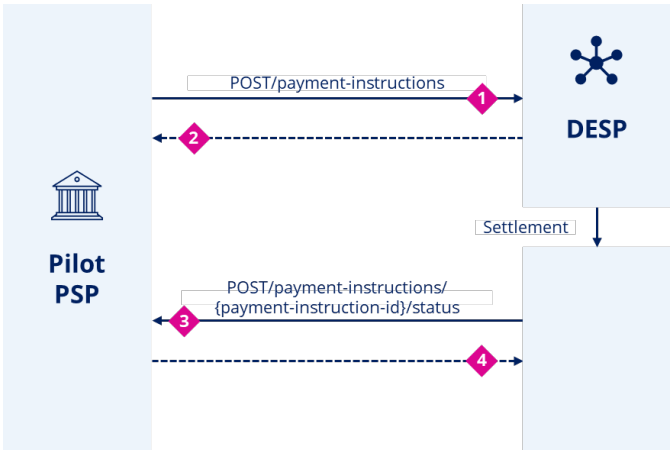


Figure 21 Payment transaction – same pilot PSP

API Call	Description	Covered in section
1	Pilot PSP sends the request with a payment object with complete settlement data.	DESP API POST /payment-instructions - Request
2	Synchronous response from DESP to confirm receipt of the request sent in API call 1.	DESP API POST /payment-instructions - Response

3	DESP sends status of the payment transaction back to pilot PSP.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Request
4	Synchronous response from pilot PSP to confirm receipt of the status notification.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Response

The sequence diagram below presents a negative scenario when DESP rejects the incoming request. As the request fails at the technical validation step, the status cannot be queried.[20]

Figure 22 Payment transaction rejection – same pilot PSP

API Call	Description	Covered in section
1	Pilot PSP sends the request with a payment object with complete settlement data.	DESP API POST /payment-instructions - Request
2	Synchronous response from DESP to reject receipt of the request sent in API call 1.	DESP API POST /payment-instructions - Response

6.3.1.2.2. Payer-initiated payment transaction

The sequence diagram below presents a payment process initiated by the payer PSP[21], in which the pilot PSPs for a payee and a payer are different, and no funding/defunding of digital euro is required.

[20] Functional errors that come in a status message can be queried, while any other error (e.g., technical errors with HTTP code 4xx) than status after a 20x message cannot be queried. In case of a technical error, the pilot PSP can submit the same message with a new UETR after the cause is identified.

[21] Examples of the payer-initiated flows: payer-initiated alias or DEAN payments. For details, please refer **to Digital euro pilot – End-to-end process flows**.

EUROPEAN CENTRAL BANK

EUROSYSTEM

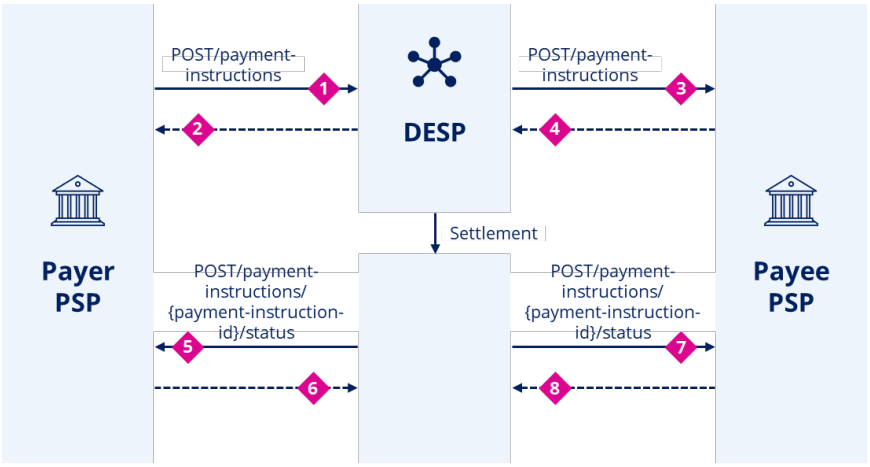


Figure 23 Payer-initiated payment transaction

API Call	Description	Covered in section
1	Payer PSP sends the request with a payment object with partial settlement data.	DESP API POST /payment-instructions - Request
2	Synchronous response from DESP to confirm receipt of the message sent in API call 1.	DESP API POST /payment-instructions - Response
3	DESP forwards the request that contains payment object from API call 1 to payee PSP.	PSP API POST /payment-instructions - Request
4	Synchronous response from payee PSP to send back the validation result and the complement settlement data (if the validation result is positive).	PSP API POST /payment-instructions - Response
5	DESP sends status of the payment transaction back to payer PSP, simultaneously with API call 7.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Request
6	Synchronous response from payer PSP to confirm receipt of the status notification.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Response
7	DESP sends status of the payment transaction back to payee PSP, simultaneously with API call 5.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Request
8	Synchronous response from payee PSP to confirm receipt of the status notification.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Response

The sequence diagram below presents a negative scenario when the payee PSP rejects the incoming request.

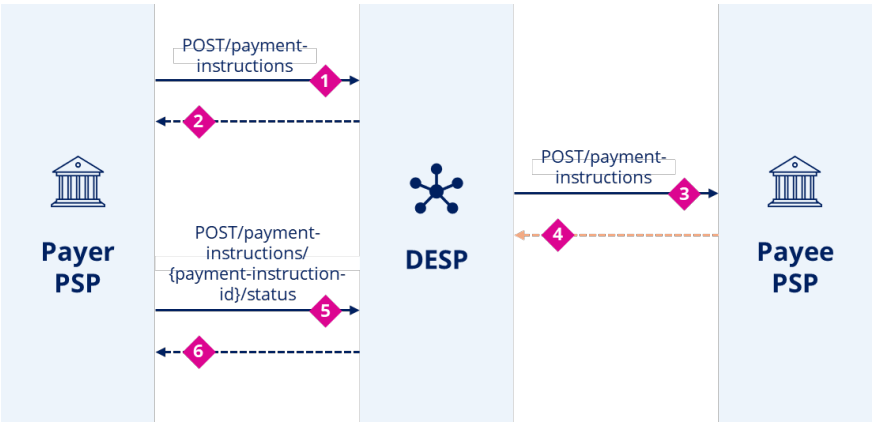


Figure 24 Payer-initiated payment transaction rejection by payee PSP

API Call	Description	Covered in section
1	Payer PSP sends the request with a payment object with partial settlement data.	DESP API POST /payment-instructions - Request
2	Synchronous response from DESP to confirm receipt of the message sent in API call 1.	DESP API POST /payment-instructions - Response
3	DESP forwards the request that contains payment object from API call 1 to payee PSP.	PSP API POST /payment-instructions - Request
4	Synchronous response from payee PSP to reject the incoming request API call 3.	PSP API POST /payment-instructions - Response
5	DESP sends status of the payment transaction back to the payer PSP.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Request
6	Synchronous response from payer PSP to confirm receipt of the status notification.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Response

6.3.1.2.3. Payee-initiated payment transaction

The sequence diagram below presents a payee-initiated payment transaction, as the first API call containing payment data is submitted by the payee PSP. Examples of the applicable use case are E-commerce payment with alias or DEAN [22], or NFC/chip-based payment via mobile device at the (Soft)POS[23].

[22] See E2E flows TM-2.2.
[23] See E2E flows TM-1.2, TM-1.3, TM-1.6.

EUROPEAN CENTRAL BANK

EUROSYSTEM

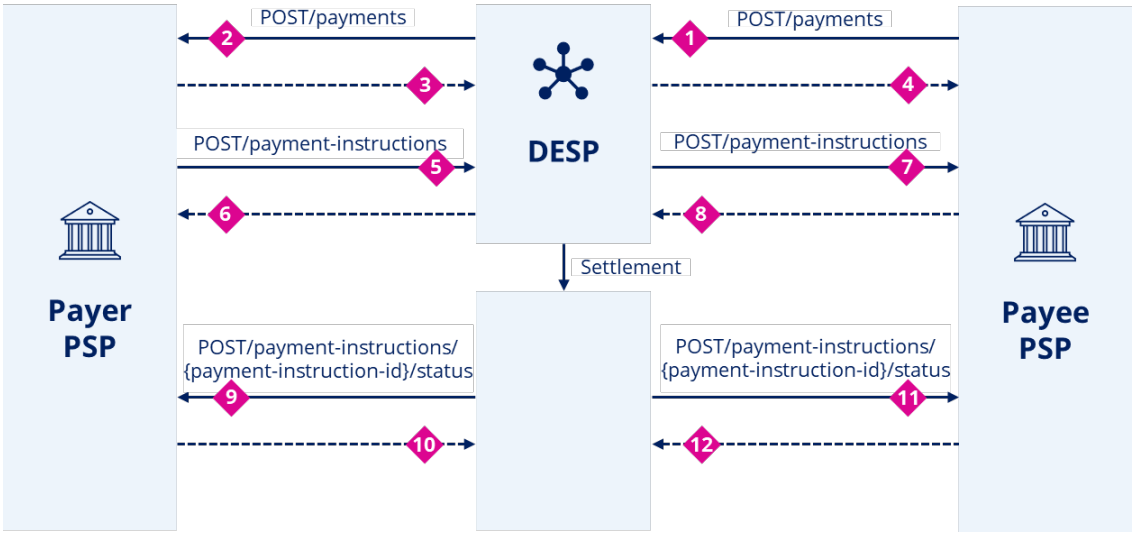


Figure 25 Payer-initiated payment transaction with original request stemming from the payee PSP

API Call	Description	Covered in section
1	Payee PSP sends the request for payment API call with encrypted payment data.	DESP API POST /payments - Request
2	DESP forwards the request sent in API call 2.	PSP API POST /payments – Request
3	Payer PSP performs technical validation and responds with acceptance or rejection.[24]	PSP API POST /payments - Response
4	DESP forwards the response sent in API call 3. In the case of a rejection, the flow stops at API call 4.	DESP API POST /payments - Response
5	Payer PSP sends the payment instruction request with a payment object with partial settlement data.	DESP API POST /payment-instructions - Request
6	Synchronous response from DESP to confirm receipt of the message sent in API call 5.	DESP API POST /payment-instructions - Response
7	DESP forwards the request that contains payment object from API call 6 to payee PSP.	PSP API POST /payment-instructions - Request
8	Synchronous response from payee PSP to send back the validation result and the complement settlement data (if the validation result is positive).	PSP API POST /payment-instructions - Response
9	DESP sends status of the payment transaction back to the payer PSP, simultaneously with API call 11.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Request

[24] Depending on the payment use cases, other API calls can happen between API call 3 and 5. For details please refer to **Digital euro pilot – End-to-end process flows**. For example, cryptogram check is required for NFC/chip-based payment with mobile device at the (Soft)POS; payer authorization of the payment is required in the app for E-commerce payment with alias or DEAN.

EUROPEAN CENTRAL BANK

EUROSYSTEM

10	Synchronous response from the payer PSP to confirm receipt of the status notification.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Response
11	DESP sends status of the payment transaction back to payee PSP, simultaneously with API call 9.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Request
12	Synchronous response from payee PSP to confirm receipt of the status notification.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Response

The sequence diagram below presents a negative scenario where the request for payment is rejected asynchronously by the payer PSP (i.e. the rejection is not sent directly to payment request in step 3, but just after some time in step 5).[25]

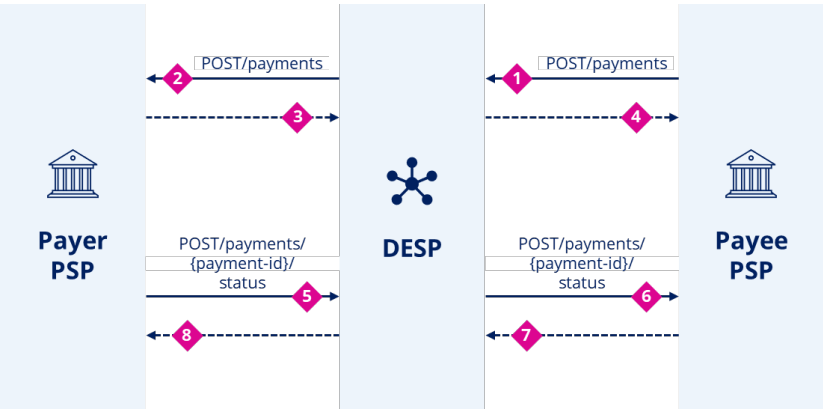


Figure 26 Request for payments asynchronously rejected by the payer PSP[26]

Step 1 to step 4 is the same as in **Figure 25**.

API Call	Description	Covered in section
5	In case of a functional rejection, the payer PSP sends the status of the request for payment.	DESP API POST /payments/{payment-id}/status - Request
6	DESP forwards the request in API call 5 to the payee PSP.	PSP API POST /payments/{payment-id}/status - Request
7	Synchronous response from the payee PSP to confirm receipt of the status notification.	PSP API

[25] For example, the payer rejects or does not respond in time to the incoming request in the app in E-commerce payment with alias or DEAN or in payee initiated P2P payment with alias or DEAN.

[26] Please see TM-2.2 E-commerce payment with alias or DEAN, TM-1.6 Online contactless SoftPOS payment with mobile device - same pilot PSP, TM-1.2 Online contact and contactless (Soft)POS payment with mobile device - same pilot PSP.

EUROPEAN CENTRAL BANK
EUROSYSTEM

		POST /payments/{payment-id}/status - Response
8	DESP forwards the response in API call 7 to the payer PSP.	DESP API POST /payments/{payment-id}/status - Response

6.3.1.3. Combined payment transactions

6.3.1.3.1. Payment transaction and (de)funding with same PSP for payer and payee

The sequence diagram below presents the payment process in which the pilot PSP for a payee and a payer is the same, as well as for funding/defunding.

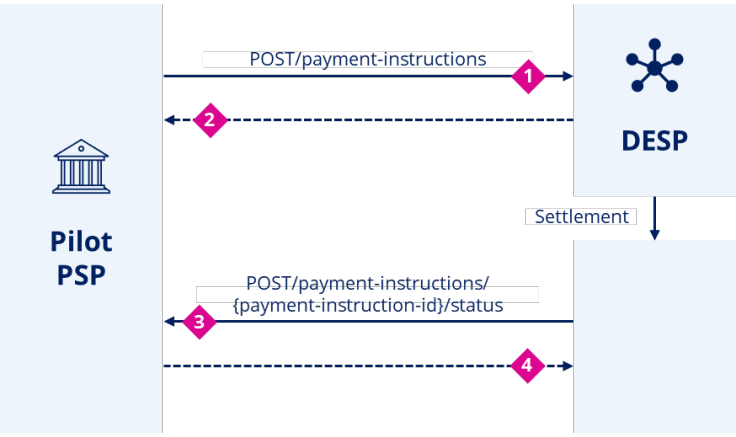


Figure 27 Same payer, payee and (de)funding pilot PSP

API Call	Description	Covered in section
1	Pilot PSP sends the request with (1) payment object with complete settlement data and (2) with funding and/or defunding object(s) with complete settlement data.	DESP API POST /payment-instructions - Request
2	Synchronous response from DESP to confirm the receipt of the message sent in API call 1.	DESP API POST /payment-instructions - Response
3	DESP sends status of the payment transaction back to the pilot PSP.	PSP API POST /payment-instructions/{payment-instruction-id}/status – Request
4	Synchronous response from the pilot PSP to confirm the receipt of the status notification.	PSP API POST /payment-instructions/{payment-instruction-id}/status - Response

EUROPEAN CENTRAL BANK

EUROSYSTEM

6.3.1.4. Refunds

6.3.1.4.1. Refund for E-m-commerce

The sequence diagram below presents an e-/m-commerce refund process, in which the pilot PSPs for a payee and a payer are different, no defunding of beta digital euro is required.

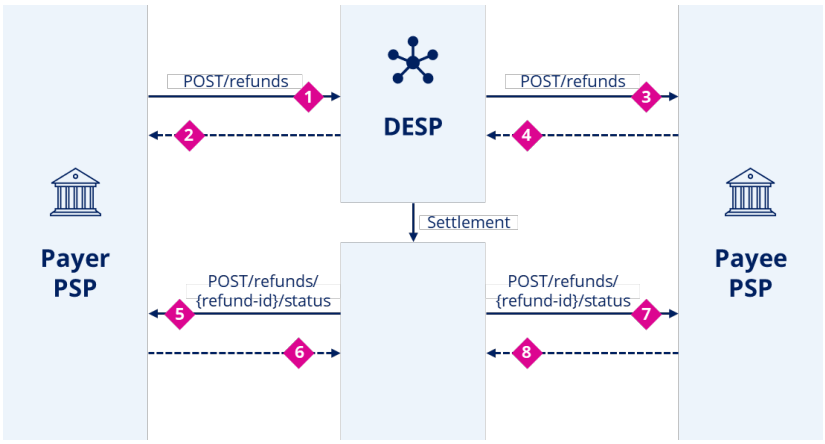


Figure 28 Refund transaction[27]

API Call	Description	Covered in section
1	The payer PSP of the refund (i.e. acquiring PSP, which is the payee PSP of the original payment) sends a request with refund object and funding object with partial settlement data to DESP.	DESP API POST /refunds – Request
2	Synchronous response from DESP to confirm receipt of the request sent in API call 1.	DESP API POST /refunds – Response
3	DESP forwards the request which contains the refund object from API call 1 to the payee (the original payer PSP).	PSP API POST /refunds - Request
4	Synchronous response from the payee PSP of the refund to send back the validation result and the complement settlement data (if the validation result is positive).	PSP API POST /refunds – Response
5	DESP sends status of the refund transaction back to the payer PSP, simultaneously with API call 7.	PSP API POST /refunds/{refund-id}/status - Request
6	Synchronous response from the payer PSP to confirm receipt of the status notification.	PSP API POST /refunds/{refund-id}/status - Response
7	DESP sends status of the refund transaction back to the payee PSP, simultaneously with API call 5.	PSP API POST /refunds/{refund-id}/status - Request
8	Synchronous response from the payee PSP to confirm receipt of the status notification.	PSP API

[27] Please see TM-7.2 Refund (E-commerce)

EUROPEAN CENTRAL BANK

EUROSYSTEM

		POST /refunds/{refund-id}/status - Response

6.3.1.5. Status query

The sequence diagram below presents a query which can be sent by the payee PSP to the payer PSP.

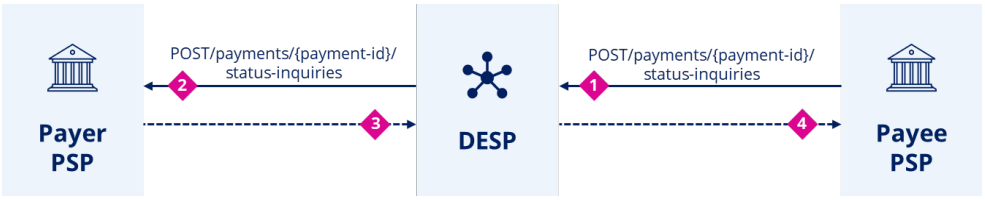


Figure 29 Request for payments query initiated by the payee

API Call	Description	Covered in section
1	The payee PSP sends the query to the payer PSP via DESP.	DESP API GET / payments/{payment-id}/status - Request
2	DESP forwards the request in API call 2 to the payer PSP.	PSP API POST /payments/{payment-id}/status-inquiries - Request
3	Synchronous response from the payer PSP to confirm the status of the request.	PSP API POST /payments/{payment-id}/status-inquiries - Response
4	DESP forwards the response in API call 3 to the payer PSP.	DESP API POST /payments/{payment-id}/status-inquiries - Response

6.3.1.6. Transaction report

The transactions report enables pilot PSPs to retrieve detailed information on successfully settled transactions initiated or received by instructing parties serviced by the pilot PSP. It provides transaction-level data, including identifiers, amounts, timestamps, and originating pilot PSP information.

The report is generated by the DESP based on the pilot PSP request parameters and made available according to the defined reporting frequency. Pilot PSPs retrieve the report via dedicated API endpoint once it is available. DESP notifies the pilot PSP when the report is ready to be retrieved.

For recovery scenarios, where a report ready notification is not received by the pilot PSP, or cannot be processed, the pilot PSP may query the DESP for the list of available reports. The DESP in this case returns a list of available reports together with their metadata, allowing the pilot PSP to identify the relevant transactions report and retrieve it via the dedicated transaction report retrieval endpoint.

6.3.1.6.1. Report attributes

Recipient	Mode	View	Frequency

EUROPEAN CENTRAL BANK

EUROSYSTEM

Pilot PSP	Full or Delta	Detailed	As per subscription

The Transactions report shall include the following information:

- the period (e.g. the calendar day) for which the information is retrieved
- currency
- identifier of the pilot PSP (BIC) originating the transaction
- amount
- transaction reference
- settlement timestamp

6.3.1.6.2. Transaction report retrieval process

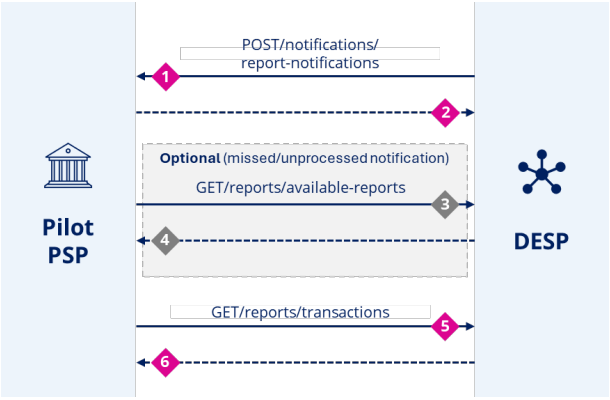


Figure 30 Report retrieval process

API Call	Description	Covered in section
1	DESP sends report-ready notification to the pilot PSP.	POST /notifications/report-notifications - Request
2	The pilot PSP confirms the receipt of the report-ready notification.	POST /notifications/report-notifications - Response
3(Optional)	The pilot PSP retrieves the available reports.	GET/reports/available-reports - Request
4(Optional)	DESP provides the list of available reports to the pilot PSP.	GET/reports/available-reports - Response
5	The pilot PSP retrieves the transaction report by specifying report parameters.	GET/reports/transactions - Request
6	DESP provides the requested transaction report to the pilot PSP.	GET/reports/transactions - Response

Steps 3 and 4 are optional and shall be executed in case of missed or unprocessed notifications.

EUROPEAN CENTRAL BANK

EUROSYSTEM

6.3.2. Overview of API endpoints

6.3.2.1. Pilot PSPs sending requests to DESP – API endpoints

API Nr	Description	Sending Party	Exposing Party*	Covered in section
1	Pilot PSP submits a funding request to DESP.	Pilot PSP	DESP*	POST /fundings
2	Pilot PSP submits a defunding request to DESP.	Pilot PSP	DESP*	POST /defundings
3	Payer PSP submits payment instruction to payee PSP to start the settlement process. If both payer and payee have the same pilot PSP, the payment instruction is to be submitted directly at this endpoint with complete settlement data to DESP.	Payer PSP	Payee PSP (via DESP) or DESP*	POST /payment-instructions
4	Payee PSP submits payment request to payer PSP to start the payment process.	Payee PSP	Payer PSP (via DESP)	POST /payments
5	Payer PSP sends status update of a payment request initiated by the payee PSP.	Payer PSP	Payee PSP (via DESP)	POST /payments/{payment-id}/status
6	The payee PSP of the original payment (payer PSP in refund transaction) sends the refund payment instruction to the payer PSP of the original payment (payee PSP in refund transaction)	Payer PSP	Payee PSP (via DESP)	POST /refunds
7	Pilot PSP queries status of a funding transaction.	Pilot PSP	DESP	GET /fundings/{funding-id}/status
8	Pilot PSP queries status of a defunding transaction.	Pilot PSP	DESP	GET /defundings/{defunding-id}/status
9	Payer PSP OR payee PSP queries status of a payment transaction.	Payer PSP or payee PSP	DESP	GET /payment-instructions/{payment-instruction-id}/status
10	Payee PSP queries status of a payment request with payer PSP.	Payee PSP	Payer PSP (via DESP)	POST /payments/{payment-id}/status-inquiries
11	Payer PSP OR payee PSP queries status of a refund transaction.	Payer PSP or payee PSP	DESP	GET /refunds/{refund-id}/status
12	Pilot PSP retrieves the transactions report.	Pilot PSP	DESP	GET/reports/transactions
13	Pilot PSP initiates end user entry(-ies) query.	Pilot PSP	DESP	GET/ queries/enduser-entries

14	Pilot PSP retrieves the available reports overview.	Pilot PSP	DESP	GET /reports/available-reports/?report-type=<report-type>

*When the instructing PSP is also the instructed party for a payment transaction, funding transaction or defunding transaction, then the exposing party will be DESP.

6.3.2.2. Pilot PSPs receiving requests from DESP – API endpoints

API Nr	Description	Sending Party	Exposing Party	Covered in section
14	DESP forwards funding request to pilot PSP.	DESP	Pilot PSP	POST /fundings
15	DESP sends status of a funding transaction to pilot PSP.	DESP	Pilot PSP	POST /fundings/{funding-id}/status
16	DESP forwards defunding request to pilot PSP.	DESP	Pilot PSP	POST /defundings
17	DESP sends status of a defunding transaction to pilot PSP.	DESP	Pilot PSP	POST /defundings/{defunding-id}/status
18	DESP forwards payment instruction to payee PSP.	DESP	Payee PSP	POST /payment-instructions
19	DESP sends status of a payment transaction (initiated at POST/payment-instructions endpoint) to payer and payee PSP.	DESP	Payer and payee PSP	POST /payment-instructions/{payment-instruction-id}/status
20	DESP forwards payment request from payee PSP to payer PSP.	DESP	Payer PSP	POST /payments
21	DESP forwards status update of a payment request from payer PSP to payee PSP.	DESP	Payee PSP	POST /payments/{payment-id}/status
22	DESP forwards query on status of a payment request from digital euro payee PSP to payer PSP.	DESP	Payer PSP	POST /payments/{payment-id}/status-inquiries
23	DESP forwards refund payment instruction to the payee PSP.	DESP	Payee PSP	POST /refunds
24	DESP sends status of a refund to payer and payee PSP.	DESP	Payer and payee PSP	POST /refunds/{refund-id}/status
25	DESP sends a notification to the pilot PSP when the transaction report is ready to be pulled.	DESP	Pilot PSP	POST /notifications/report-notifications

EUROPEAN CENTRAL BANK

EUROSYSTEM

6.3.3. DESP API for pilot PSPs to send requests to DESP

This section provides a detailed description of all API endpoints used by pilot PSPs to submit requests to DESP (hereafter DESP API).

6.3.3.1. POST /fundings

Method	Endpoint	Resource	Description
POST	/v1/fundings	Funding	Endpoint for funding transaction initiation

6.3.3.1.1. POST /fundings – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Request body data elements

Root

Data element	Element description	Type	Presence indicator
acceptanceDateTime	Date and time. This timestamp is used for determination of maximum processing time. This is set by the first pilot PSP which starts the settlement process, and should not be altered by other pilot PSPs.	ISODateTime	M
paymentIdentification	Unique end-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the payment chain. Set by the instructing pilot PSP.	Payment Identification2	M
fundingObject	See the next tables.	-	M

Funding object data elements – unencrypted part

Data element	Element description	Type	Presence indicator
instructingAgent	Unique identification of the end user's pilot PSP (BIC).	Financial institution Identification1	M
instructedAgent	In the context of the pilot, the same value as instructing pilot PSP must be filled in.	Financial institution Identification1	M

amount	The amount of beta digital euro to be issued on the beta digital euro account.	Amount	M
userEntry	Identification of the end user's holding in the settlement ledger.	Entry	M
encryptedFundingObject	Encrypted data fields, see next table.		C

Funding object data elements – encrypted part

In the context of the pilot, this request body does not contain any encrypted part.

6.3.3.1.2. POST /fundings – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors**.
- In case of positive technical validation: HTTP Code 201.

Data element	Description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	Payment Identification2	M
fundingId	Resource ID generated by DESP to identify the funding transaction.	UUID	M
fundingTransactionStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACTC -ACCP	TransactionStatus	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'fundingTransactionStatus'

6.3.3.2. POST /defundings

Method	Endpoint	Resource	Description

POST	/v1/defundings	Defunding	Endpoint for defunding transaction initiation

6.3.3.2.1. POST /defundings – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Request body data elements

Root

Data element	Element description	Type	Presence indicator
acceptanceDateTime	Date and time. This timestamp is used for determination of maximum processing time. This is set by the first pilot PSP which starts the settlement process and should not be altered by other pilot PSPs.	ISODateTime	M
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	Payment Identification2	M
defundingObject	See the next table.	Defunding object	M

Defunding object data elements – unencrypted part

Data element	Element description	Type	Presence indicator
instructingAgent	Unique identification of the user's pilot PSP (BIC).	Financial institution Identification1	M
instructedAgent	In the context of the pilot, the same value as instructing PSP must be filled in.	Financial institution Identification1	M
amount	The amount of beta digital euro to be issued on the beta digital euro account.	Amount	M
userEntry	Identification of the end user's holding in the settlement ledger.	Entry	M
encryptedDefundingObject	Encrypted data fields, see next table.		C, mandatory in case of open defunding

EUROPEAN CENTRAL BANK
EUROSYSTEM

Defunding object data elements – encrypted part

In the context of the pilot, this request body does not contain any encrypted part.

6.3.3.2.2. POST /defundings – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 201.

Data element	Description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	Payment Identification2	M
defundingId	Resource ID generated by the DESP.	UUID	M
defundingTransactionStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACTC -ACCP	TransactionStatus	M
statusReason	Reason for the negative validation result.	statusReason	C, mandatory in case of failure

6.3.3.3. POST /payment-instructions

Method	Endpoint	Resource	Description
POST	/v1/payment-instructions	Payment instruction	Endpoint for a payer PSP to submit a payment instruction to start the settlement process of a payment transaction.

6.3.3.3.1. POST /payment-instructions – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Request body data elements

Root

Data element	Element description	Type	Presence indicator
acceptanceDateTime	Date and time. This timestamp will be used for determination of maximum processing time. This is set by the first pilot PSP which starts the settlement process, and should not be altered by other pilot PSPs.	ISODateTime	M
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	Payment Identification2	M
paymentInstructionObject	See the next table.	-	M
fundingObject	See the next table. This object is mandatory to be included if funding is needed.	-	O
defundingObject	See the next table. This object is mandatory to be included if defunding is needed	-	O

Payment instruction object data elements – unencrypted part

Data element	Element description	Type	Presence indicator
debtorAgent	Unique identification of the payer's PSP (BIC).	Financial institution Identification1	M
creditorAgent	Unique identification of the payee's PSP (BIC).	Financial institution Identification1	M
amount	The amount to be transferred from the payer to the payee. This should be the total amount reflected in the encrypted part of the payment object, if additional amount field is used.	Amount	M
debtorEntry	Unique identification of the payer's holding in the ledger.	Entry	M
creditorEntry	Unique identification of the payee's holding in the ledger. This field must be included if payer and payee have the same pilot PSP.	Entry	C, mandatory in case digital euro payer and payee PSPs are the same
encryptedPaymentInstructionObject	Encrypted data fields, see next table.		C, mandatory

EUROPEAN CENTRAL BANK

EUROSYSTEM

			in case digital euro payer and payee PSPs are different

Payment instruction object data elements – encrypted part

The encrypted part is only required if payer and payee have different pilot PSPs.

Data element	Element description	Type	Presence indicator
creditorAccount	Unique identification of the payee's beta digital euro account.	Account Reference	M
creditor	Name of the payee.	PartyDescription1	
ultimateCreditor	Payee party details	PartyDescription	O
debtorAccount	Unique identification of the payer's beta digital euro account.	Account Reference	M
debtor	Name of the payer.	PartyDescription1	M
remittanceInformationUnstructured	The field provides additional information about (the purpose of) the transaction in free format text. It is mandatory to include this field if the end user provides the remittance.	Max140Text	O
remittanceInformationStructured	The fields provide additional information about (the purpose of) the transaction in a standardised and structured format. It may include a type, a reference to a standard and a value. It is mandatory to include this field if the user provides the remittance.	Remittance	O
purposeCode	Predefined code to indicate the purpose of the payment.	Purpose Code	O
paymentIdentification	Resource ID generated by the payee PSP to identify a payment request.	PaymentIdentification1	O

transactionAmount	The transaction amount to be transferred from the payer to the payee.	Amount	M
additionalAmount	Amount paid in addition to the transaction amount, e.g. in case of cashback or tip.	Amount	O
totalAmount	Sum of transaction amount and additional amount.	Amount	C, mandatory if 'additionalAmount' field is present.
originalPaymentId	The resource identification of the payment request that is initiated by the payee PSP at POST /payments endpoint.	UUID	C, mandatory if it is a payee-initiated payment
originalUetr	The unique end-to-end Transaction Reference of the payment request initiated by the payee PSP at POST /payments endpoint. OriginalUetr refers to the UETR of the original payment.	UUID	C, mandatory if it is a payee-initiated payment

Funding object data elements - unencrypted part

Data element	Element description	Type	Presence indicator
instructedAgent	In the context of the pilot, the same value as instructing PSP must be filled in.	Financial institution Identification1	M
amount	The amount of beta digital euro to be issued on the beta digital euro account.	Amount	M
encryptedFundingObject	Encrypted data fields, see next section.		C

Funding object data elements - encrypted part

In the context of the pilot, this request body does not contain any encrypted part. The same applies to 'Funding object data elements – encrypted part' in DESP API POST /fundings.

Defunding object data elements - unencrypted part

Defunding object is required only when defunding is required and when payer and payee have the same pilot PSP.

EUROPEAN CENTRAL BANK
EUROSYSTEM

Data element	Element description	Type	Presence indicator
instructedAgent	In the context of the pilot, the same value as instructing PSP must be filled in.	Financial institution Identification1	M
amount	The amount of beta digital euro to be issued on the beta digital euro account.		M
encryptedDefundingObject	Encrypted data fields, see next section.		C

Defunding object data elements – encrypted part

In the context of the pilot, this request body does not contain any encrypted part. The same applies to 'Defunding object data elements – encrypted part' in DESP API POST /defundings.

6.3.3.3.2. POST /payment-instructions – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 201.

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	paymentIdentification2	M
paymentInstructionId	Resource ID generated by DESP to identify a payment transaction.	UUID	M
paymentTransactionStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACTC	TransactionStatus	M
fundingId	Newly created resource id for the funding part of the payment transaction.	UUID	C, only required if funding object is present in the request and if validation result is positive as indicated in the field 'paymentTransactionStatus'

fundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACTC	TransactionStatus	C, only required if funding object is present in the request
defundingId	Newly created resource id for the defunding part of the payment transaction.	UUID	C, only required if defunding object is present in the request and if validation result is positive as indicated in the field 'paymentTransactionStatus'
defundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACTC	TransactionStatus	C, only required if defunding object is present in the request
statusReason	Reason for the negative validation result.	statusReason	C, mandatory in case of failure as indicated in the field 'paymentTransactionStatus'

6.3.3.4. POST /payments

Method	Endpoint	Resource	Description
POST	/v1/payments	payment	Endpoint for the payee PSP to send a request for payment.

6.3.3.4.1. POST /payments – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Request body data elements

Root

Data element	Element description	Type	Presence indicator
acceptanceDateTime	Date and time. This timestamp is used for determination of maximum processing time. This is set by the pilot PSP which starts the settlement process, and should not be altered by other pilot PSPs.	ISODateTime	M
debtorAgent	Unique identification of the payer's PSP (BIC).	FinancialInstitutionIdentification1	M

creditorAgent	Unique identification of the payee's PSP (BIC).	FinancialInstitutionIdentification1	M
encryptedPaymentObject	See the next table.	-	M

Payment object data elements – encrypted part

The below data elements are subject to further changes.

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	PaymentIdentification2	M
paymentId	Resource ID generated by the payee PSP to identify a payment request.	UUID	M
expiryDateTime	Date and time at which the payment request expires.	ISODateTime	M
transactionAmount	The amount to be transferred from the payer to the payee.	Amount	M
additionalAmount	Amount paid in addition to the transaction amount, e.g. in case of cashback or tip.	Amount	O
totalAmount	Sum of transaction amount and additional amount.	Amount	C, mandatory if 'additionalAmount' field is present.
creditorAccount	Unique identification of the payee's beta digital euro account.	Account Reference	M
creditor	Information of the payee.	Party Description	M
ultimateCreditor	Information of the party. Ultimate party to which an amount of money is due.	Party Description	O
debtorAccount	Unique identification of the payer's beta digital euro account.	Account Reference	M
debtor	Information of the payer.	Party Description	M
remittanceInformationUnstructured	The field provides additional information	Max140Text	O

	about (the purpose of) the transaction in free format text. It is mandatory to include this field if the user provides the remittance.		
remittanceInformationStructured	The fields provide additional information about (the purpose of) the transaction in a standardised and structured format. It may include a type, a reference to a standard and a value. It is mandatory to include this field if the user provides the remittance.	Remittance	O
purposeCode	Predefined code to indicate the purpose of the payment.	Purpose Code	O
paymentIdentification1	Unique identification, as assigned by the initiating user, to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain. In case no reference was given by the user, "NOTPROVIDED" must be used.	PaymentIdentification1	M

6.3.3.4.2. POST /payments – Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 201.

Root

The full payload is encrypted. The payload is the same as in PSP API POST /payments.

Data element	Element description	Type	Presence indicator

paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a request to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	PaymentIdentification2	M
paymentStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACTC	TransactionStatus	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'payment Status'

6.3.3.5. POST /payments/{payment-id}/status

Method	Endpoint	Resource	Description
POST	/v1/payments/{payment-id}/status-inquiries	Payment	Endpoint for the payer PSP to asynchronously confirm status of a payment request.

6.3.3.5.1. POST /payments/{payment-id}/status – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header:

Data element	Element description	Type	Presence indicator	Path/Header/ Query
paymentId	Resource ID generated by the payee PSP to identify a request for payment.	UUID	M	Path

Request body data elements

Root

Data element	Element description	Type	Presence indicator
debtorAgent	Unique identification of the payer's PSP (BIC).	FinancialInstitutionIdentification1	M
creditorAgent	Unique identification of the payee's PSP (BIC).	FinancialInstitutionIdentification1	M
encryptedPaymentStatusObject	See the next table.	-	M

EUROPEAN CENTRAL BANK

EUROSYSTEM

Payment Status Object Encrypted part

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a request to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	paymentIdentification2	M
paymentStatus	Indication of the success or failure of the request. Allowed values: -RJCT	StatusCode	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'payment Status'

6.3.3.5.2. POST / payments/{payment-id}/status – Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive validation: HTTP Code 204, without response body.

6.3.3.6. GET /payment-instructions/{payment-instruction-id}/status

Method	Endpoint	Resource	Description
GET	/v1/payments/{payment-id}/status-inquiries	Payment	Endpoint for payer PSP or payee PSP to query the status of a payment transaction.

6.3.3.6.1. GET /payment-instructions/{payment-instruction-id}/status – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**

EUROPEAN CENTRAL BANK
EUROSYSTEM

- API specific header:

Data element	Element description	Type	Presence indicator	Path/Header/ Query
payment-instruction-Id	Resource identification generated by DESP to identify a payment transaction.	UUID	M	Path

6.3.3.6.2. GET /payment-instructions/{payment-instruction-id}/status – Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 200.

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	Payment Identification2	M
settlementDateTime	Time and date of the moment the settlement was recorded in DESP.	ISODateTime	C, mandatory in case of positive request result as indicated at the field 'payment transaction status'
settlementBusinessDay	Business day of the settlement which was recorded in Dedicated Cash Account (DCA) in DESP.	ISODate	C, mandatory in case of positive validation as indicated at the field 'payment transaction status' and in case funding or/and defunding object was/were included in the original request
paymentTransactionStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	TransactionStatus	M
fundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT	TransactionStatus	C, mandatory if funding object is present in the request

	-ACSC		
fundingId	Resource ID generated by the DESP for the funding part of the payment transaction.	UUID	C, mandatory if funding object is present in the request and in case of positive validation as indicated at the field 'payment transaction status'
defundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	TransactionStatus	C, mandatory if defunding object is present in the request
defundingId	Resource ID generated by the DESP for the defunding part of the payment transaction.	UUID	C, mandatory if defunding object is present in the request and in case of positive validation as indicated at the field 'payment transaction status'
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'payment transaction status'

6.3.3.7. GET /fundings/{funding-id}/status

This endpoint is used for pilot PSPs to query statuses of fundings.

Method	Endpoint	Resource	Description
GET	/v1/fundings/{funding-id}/status	Funding	This endpoint is for commercial bank money payer or payee PSP to query the status of a funding transaction.

6.3.3.7.1. GET /fundings/{funding-id}/status – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header:

Data element	Element description	Type	Presence indicator	Path/Header/ Query
funding-id	Unique identification of the funding transaction for which the status is requested.	UUID	M	path

6.3.3.7.2. GET /fundings/{funding-id}/status – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 200.

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	paymentIdentification2	M
fundingTransactionStatus	Indication of the success or failure of the request Allowed values: -RJCT -ACSC	TransactionStatus	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'funding transaction status'
settlementDateTime	Time and date of the moment the settlement was recorded in DESP.	ISODateTime	C, in case of a positive validation result as indicated in the field 'funding transaction status'
settlementBusinessDay	Business day of the settlement which was recorded in Dedicated Cash Account (DCA) in DESP.	ISODate	C, in case of a positive validation result as indicated in the field 'funding transaction status'

6.3.3.8. GET /defundings/{defunding-id}/status

Method	Endpoint	Resource	Description
GET	/v1/defundings/{defunding-id}/status	Defunding	This endpoint for commercial bank money payer or payee PSP to query the status of a defunding transaction.

6.3.3.8.1. GET /defundings/{defunding-id}/status – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header:

Data element	Element description	Type	Presence indicator	Path/Header/ Query
defunding-id	Unique identification of the defunding transaction for which the status is requested.	UUID	M	path

6.3.3.8.2. GET /defundings/{defunding-id}/status – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 200.

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	paymentIdentification2	M
defundingTransactionStatus	Indication of the success or failure of the request Allowed values: -RJCT -ACSC	TransactionStatus	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'defunding transaction status'
settlementDateTime	Time and date of the moment the settlement was recorded in DESP.	ISODateTime	C, in case of a positive validation result as indicated in the field

			'defunding transaction status'
settlementBusinessDay	Business day of the settlement which was recorded in Dedicated Cash Account (DCA) in DESP.	ISODate	C, in case of a positive validation result as indicated in the field 'defunding transaction status'

6.3.3.9. POST /payments/{payment-id}/status-inquiries

Method	Endpoint	Resource	Description
GET	/v1/payments/{payment-id}/status-inquiries	payment	Endpoint for the payee PSP to query status of a payment request with the payer PSP.

6.3.3.9.1. POST /payments/{payment-id}/status-inquiries – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header:

Data element	Element description	Type	Presence indicator	Path/Header/ Query
payment-id	Resource identification generated by the payee PSP to identify a payment request.	UUID	M	path

Request body data elements

Data element	Element description	Type	Presence indicator
debtorAgent	Unique identification of the payer's PSP (BIC).	FinancialInstitutionIdentification1	M

6.3.3.9.2. POST /payments/{payment-id}/status-inquiries – Response

Response Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 200.

Root
The full payload is encrypted.

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a request to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	paymentIdentification2	M
paymentStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	StatusCode	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'payment Status'

6.3.3.10. POST /refunds

Method	Endpoint	Resource	Description
POST	/v1/refunds	Refund	Endpoint to initiate a refund transaction.

6.3.3.10.1. POST /refunds – Request

Request Header

- Technical header: please refer to **section 2.3.3.1 Technical header.**
- API specific header: not applicable.

Request body data elements

Root

Data element	Element description	Type	Presence indicator
acceptanceDateTime	Date and time. This timestamp is used for determination of maximum processing time. This is set by the first pilot PSP which starts the settlement process, and should not be altered by other pilot PSPs.	ISODateTime	M
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be	PaymentIdentification2	O

	located at any time, by any of the parties in the chain. Set by the instructing PSP.		
refundObject	See the table below.	-	M
fundingObject	See the table below. This object is mandatory to be included if funding is needed.	-	O
DefundingObject	See the table below. This object is mandatory to be included if defunding is needed.	-	O

Refund object data elements – unencrypted part

Data element	Element description	Type	Presence indicator
debtorAgent	Unique identification of the payer's PSP (the payee PSP of the original payment that is being refunded).	Financial institution Identification1	M
creditorAgent	Unique identification of the payee's PSP (the payer PSP of the original payment that is being refunded).	Financial institution Identification1	M
amount	The amount to be transferred from the payer to the payee (from the original payee to the original payer). This could be the full amount of the original payment or a lower amount.	Amount	M
debtorEntry	Unique identification of the payer's holding in the ledger.	Entry	M
creditorEntry	Unique identification of the payee's holding in the ledger. This field must be included if payer and payee have the same pilot PSP.	Entry	O
encryptedRefundObject	Encrypted data fields, see next table.		C, mandatory in case digital euro payer and payee PSPs are different

Refund object data elements – encrypted part

Data element	Element description	Type	Presence indicator
creditorAccount	Unique identification of the payee's (original payee's) beta digital euro account.	Account Reference	M
creditor	Name of the payee (original payer).	PartyDescription1	M
debtorAccount	Unique identification of the payer's (original payee's) beta digital euro account.	Account Reference	M
debtor	Name of the payer (original payee).	PartyDescription1	M

remittanceInformationUnstructured	The field provides additional information about (the purpose of) the transaction in free format text. It is mandatory to include this field if the user provides the remittance.	Max140Text	O
remittanceInformationStructured	The fields provide additional information about (the purpose of) the transaction in a standardised and structured format. It may include a type, a reference to a standard and a value. It is mandatory to include this field if the user provides the remittance.	Remittance	O
purposeCode	Predefined code to indicate the purpose of the payment.	Purpose Code	O
paymentIdentification	Unique identification, as assigned by the initiating user, to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain. In case no reference was given by the user, "NOTPROVIDED" must be used.	PaymentIdentification1	M
originalPaymentInstructionId	The resource identification of the payment transaction that is being refunded.	UUID	M
originalUetr	The unique end-to-end Transaction Reference of the payment transaction that is being refunded. OriginalUetr refers to the UETR of the original payment.	PaymentIdentification2	M

Funding object data elements - non-encrypted part

Same as 'Funding object data elements – non-encrypted part' in DESP API POST /payment-instructions.

Funding object data elements - encrypted part

In the context of the pilot, this request body does not contain any encrypted part. The same applies to **'Funding object data elements – encrypted part'** in DESP API POST /fundings.

6.3.3.10.2. POST /refunds – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 201.

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	PaymentIdentification1	O
refundId	Resource identification generated by DESP to identify a refund payment transaction.	UUID	M
refundTransactionStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACTC	TransactionStatus	M
fundingId	Newly created resource id for the funding part of the payment transaction.	UUID	C, only required if funding object is present in the request and if validation result is positive as indicated in the field 'refund transaction status'
fundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACTC	TransactionStatus	C, only required if funding object is present in the request
defundingId	Newly created resource id for the defunding part of the payment transaction.	UUID	C, mandatory if defunding object is present in the request and in case of positive validation as indicated at the field 'refund transaction status'
defundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	TransactionStatus	C, mandatory if defunding object is present in the request
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'refund transaction status'

6.3.3.11. GET /refunds/{refund-id}/status

Method	Endpoint	Resource	Description
GET	/v1/refunds/{refund-id}/status	Refund	Endpoint for payer PSP or payee PSP to query the status of a refund transaction.

6.3.3.11.1. GET /refunds/{refund-id}/status – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header:

Data element	Element description	Type	Presence indicator	Path/Header/ Query
refund-id	Resource identification generated by DESP to identify a refund transaction.	UUID	M	path

6.3.3.11.2. GET /refunds/{refund-id}/status – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 200.

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	paymentIdentification2	M
settlementDateTime	Time and date of the moment the settlement was recorded in DESP.	ISODateTime	C, mandatory in case of positive request result as indicated at the field 'refundTransactionStatus'
refundTransactionStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	TransactionStatus	M

fundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	TransactionStatus	C, mandatory if funding object is present in the request
fundingId	Resource ID generated by the DESP for the funding part of the payment transaction.	UUID	C, mandatory if funding object is present in the request and in case of positive validation as indicated at the field 'refundTransactionStatus'
defundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	TransactionStatus	C, mandatory if defunding object is present in the request
defundingId	Resource ID generated by the DESP for the defunding part of the payment transaction.	UUID	C, mandatory if defunding object is present in the request and in case of positive validation as indicated at the field 'refundTransactionStatus'
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'refundTransactionStatus'

6.3.3.12. GET /queries/end-user-entries

Method	Endpoint	Resource	Description
GET	/v1/queries/end-user-entries ?entryId=<entryId>	QUERIES	User entry(ies) query

6.3.3.12.1. GET /queries/end-user-entries – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header:

Data element	Element description	Type	Presence indicator	Path/Header/ Query
entryId	The Identifier that identifies the entry.	Entry	C, either entry identifier or pilot PSP identifier to be filled	query

6.3.3.12.2. GET /queries/end-user-entries – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 200.

Data element	Element description	Type	Presence indicator
dateTime	Timestamp of the generation of the response.	ISODateTime	M
digitalEuroAgent	The BIC of the pilot PSP that manages the entries.	financialInstitutionIdentification1	M
entryId	The identifier that identifies the entry.	Entry	M
availableAmount	The amount available related to the entry.	Amount	M
reservedAmount	The amount reserved on the entry.	Amount	M
totalAmount	The sum of available and reserved amount related to the entry.	Amount	M

6.3.3.13. GET /reports/transactions

Method	Endpoint	Resource	Description
GET	/v1/reports/transactions/?reportId=<reportId>	REPORTS	Transactions report

6.3.3.13.1. GET /reports/transactions – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header

Data element	Element description	Type	Presence indicator	Path/Header/ Query
reportId	The id of the report as pushed by the notification, or retrieved from the list of available reports	UUID	M	query

6.3.3.13.2. GET /reports/transactions – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 200.

Data element	Element description	Type	Presence indicator
reportedDate	The date for which the transactions are included.	ISODate	M
currency	The currency in which the transaction amount is expressed. ISO 4217 three-letter code.	Currency Code	M
digitalEuroAgent	The BIC of the pilot PSP for which the report is generated.	financialInstitutionIdentification1	M
reportStartDateTime	Start timestamp for which the account statement is issued.	ISODatetime	M
reportEndDateTime	End timestamp for which the account statement is issued.	ISODatetime	M
The following fields repeat per each transaction			
originatingDigitalEuroAgent	The BIC of the pilot PSP originating the transaction.	financialInstitutionIdentification1	M
amount	The amount of the transaction.	Amount	M
paymentInstructionId	Resource ID generated by DESP to identify a payment instruction.	UUID	M
settlementBusinessDay	RTGS Business day on which the payment instruction is settled	IsoDate	C, mandatory in case of successful settlement
settlementDateTime	The timestamp when the settlement took place.	ISODateTime	C, mandatory in case of successful settlement

6.3.3.14. GET /available-reports

Method	Endpoint	Resource	Description
GET	/v1/ reports/available-reports?report-type=<report-type>	REPORTING	Endpoint for recipient to retrieve the statement of account turnover.

6.3.3.14.1. GET /available-reports

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header:

Data element	Element description	Type	Presence indicator	Path/Header/ Query
reportType	Indicates the type of report. Values: - DEUR: Digital euro report - DETR: Digital euro transactions report - FECR: Fee calculation report - REDR: Repository data report	Report Type	O	Query

6.3.3.14.2. GET /available-reports – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 200.

The report body consist of an array of the following attributes:

EUROPEAN CENTRAL BANK

EUROSYSTEM

Data element	Element description	Type	Presence indicator
reportType	Indicates the type of report. Values: - TRRE; Transactions Report	Report Type	M
The below data elements repeat per available report instance			
reportId	Unique identification of the report that is generated by DESP and allows the PSP to GET the report.	UUID	M
reportMode	Indicates whether the report is requested in full mode or delta mode. Values: - FULL - DELTA	Report Mode	M
reportRtgsDate	The RTGS date for which the report is available.	ISODate	O
reportCalendarDate	The calendar date for which the report is available.	ISODate	O
reportStartDateTime	Start timestamp for which the report is issued.	ISODateTime	C, in case of delta mode only
reportEndDateTime	End timestamp for which the report is issued.	ISODateTime	C, in case of delta mode only

EUROPEAN CENTRAL BANK

EUROSYSTEM

6.3.4. PSP API for pilot PSPs to receive requests from DESP

This section describes in detail all API endpoints for PSPs to receive requests from DESP (hereafter PSP API).

6.3.4.1. POST /fundings

Method	Endpoint	Resource	Description
POST	/v1/fundings	Funding	This endpoint is used for the commercial bank money PSP to receive and validate funding requests.

6.3.4.1.1. POST /fundings – Request

Funding requests might result from other endpoints in open funding case, such as DESP API:

- POST /payment-instructions
- POST /refunds

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Request body data elements

Root

Data element	Element description	Type	Presence indicator
acceptanceDateTime	Date and time. This timestamp will be used for determination of maximum processing time. This is set by the first pilot PSP which starts the settlement process, and should not be altered by other pilot PSPs.	ISODateTime	M
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	paymentIdentification2	M
fundingObject	See the next table.	Funding Object	M

Funding object data elements – unencrypted section

The user entry that was specified by the instructing PSP will not be forwarded to Instructed PSP.

Data element	Element description	Type	Presence indicator

fundingId	Resource ID generated by DESP to identify the funding transaction or the funding leg of a payment/reservation transaction.	UUID	M
instructingAgent	Unique identification of the end user's pilot PSP (BIC).	Financial institution Identification1	M
InstructedAgent	In the context of the pilot, the same value as instructing PSP must be filled in.	Financial institution Identification1	M
amount	The amount of beta digital euro to be issued on the beta digital euro account.	Amount	M
encryptedFundingObject	Encrypted data fields, see next section.		M

Funding object data elements – encrypted section

Same as 'Funding object data elements – encrypted part' in DESP API POST /fundings.

6.3.4.1.2. POST /fundings – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 201.

Root

Data element	Description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	paymentIdentification2	M
fundingTransactionStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACCP	TransactionStatus	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the

EUROPEAN CENTRAL BANK

EUROSYSTEM

			field 'funding transaction status'

6.3.4.2. POST /fundings/{funding-id}/status

Method	Endpoint	Resource	Description
POST	/v1/fundings/{funding-id}/status	Funding	Endpoints for DESP to provide status of a funding transaction

6.3.4.2.1. POST /fundings/{funding-id}/status – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header:

Data element	Element description	Type	Presence indicator	Path/Header/ Query
funding-id	Resource ID generated by DESP to identify the funding.	UUID	M	path

Request body data elements

Root

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	paymentIdentification2	M
fundingTransactionStatus	Indication of the success or failure of the request Allowed values: -RJCT -ACSC	TransactionStatus	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'funding transaction status'
settlementDateTime	Time and date of the moment the settlement was recorded in DESP.	ISODateTime	C, mandatory in case of successful settlement

settlementBusinessDay	Business day of the settlement which was recorded in Dedicated Cash Account (DCA) in DESP.	ISODate	C, mandatory in case of successful settlement

6.3.4.2.2. POST /fundings/{funding-id}/status – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 204, without response body.

6.3.4.3. POST /defundings

Method	Endpoint	Resource	Description
POST	/v1/defundings	Defunding	This endpoint is for commercial bank money PSP to receive and validate defunding requests.

6.3.4.3.1. POST /defundings – Request

Defunding requests might result from other endpoints from open defunding, such as DESP API:

- POST /payment-instructions
- POST /refunds

or response to PSP API:

- POST /payment-instructions
- POST /refunds

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Request body data elements

Root

Data element	Element description	Type	Presence indicator
acceptanceDateTime	Date and time. This timestamp will be used for determination of maximum processing time. This is set by the first pilot PSP which starts the settlement process, and should not be altered by other pilot PSPs.	ISODateTime	M
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	paymentIdentification2	M
defundingObject	See the next table.	-	M

Defunding object data elements – unencrypted section

The end user entry, that was specified by the instructing PSP will not be forwarded to instructed PSP.

Data element	Element description	Type	Presence indicator
defundingId	Resource ID generated by DESP to identify the defunding transaction or the defunding leg of a payment transaction.	UUID	M
instructingAgent	Unique identification of the user's pilot PSP (BIC).	Financial institution Identification1	M
InstructedAgent	In the context of the pilot, the same value as instructing PSP must be filled in.	Financial institution Identification1	M
amount	The amount of beta digital euro to be issued on the beta digital euro account.	Amount	M
encryptedDefundingObject	Encrypted data fields, see next section.		M

Defunding object data elements – encrypted section

Same as 'Defunding object data elements – encrypted part' in DESP API POST /defundings.

6.3.4.3.2. POST /defundings – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 201.

EUROPEAN CENTRAL BANK
EUROSYSTEM

Root

Data element	Description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	paymentIdentification2	M
defundingTransactionStatus	Indication of the success or failure of the request Allowed values: -RJCT -ACCP	TransactionStatus	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'funding transaction status'

6.3.4.4. POST /defundings/{defunding-id}/status

Method	Endpoint	Resource	Description
POST	/v1/defundings/{defunding-id}/status	Defunding	Endpoints for DESP to provide status of a defunding transaction

6.3.4.4.1. POST /defundings/{defunding-id}/status – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header:

Data element	Element description	Type	Presence indicator	Path/Header/ Query
defunding-id	Resource ID generated by DESP to identify the defunding.	UUID	M	path

Request body data elements

Root

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which	paymentIdentification2	M

	allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.		
defundingTransactionStatus	Indication of the success or failure of the request Allowed values: -RJCT -ACSC	TransactionStatus	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'funding transaction status'
settlementDateTime	Time and date of the moment the settlement was recorded in DESP.	ISODateTime	C, mandatory in case of successful settlement
settlementBusinessDay	Business day of the settlement which was recorded in Dedicated Cash Account (DCA) in DESP.	ISODate	C, mandatory in case of successful settlement

6.3.4.4.2. POST /defundings/{defunding-id}/status – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 204, without response body.

6.3.4.5. POST /payment-instructions

Method	Endpoint	Resource	Description
POST	/v1/payment-instructions	Payment instruction	This endpoint is for DESP to forward payment instruction from payer PSP to payee PSP.

6.3.4.5.1. POST /payment-instructions – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Request body data elements

Root

Data element	Element description	Type	Presence indicator
acceptanceDateTime	Date and time. This timestamp will be used for determination of maximum processing time. This is set by the first pilot PSP which starts the settlement process, and should not be altered by other pilot PSPs.	ISODateTime	M
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	Payment Identification2	M
paymentInstructionObject	See the next table.	-	M

Payment instruction object data elements - unencrypted part

The payer entry that was specified by the payer PSP will not be forwarded to the payee PSP.

Data element	Element description	Type	Presence indicator
paymentInstructionId	Resource ID generated by DESP to identify a payment transaction.	UUID	M
debtorAgent	Unique identification of the payer's PSP (BIC).	Financial institution Identification1	M
creditorAgent	Unique identification of the payee's PSP (BIC).	Financial institution Identification1	M
amount	The amount to be transferred from the payer to the payee. This should be the total amount reflected in the encrypted part of the payment object, if additional amount field is used.	Amount	M
encryptedPaymentInstructionObject	Encrypted data fields, see next section.		M

Payment instruction object data elements - encrypted part

Same as 'Payment instruction object data elements – encrypted part' in DESP API POST/payment-instructions.

6.3.4.5.2. POST /payment-instructions – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 201.

Root

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	PaymentIdentification2	M
paymentTransactionStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACCP	TransactionStatus	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'payment transaction status'
paymentInstructionObject	See the next table.	-	C, mandatory in case of positive validation as indicated at the field 'payment transaction status'
paymentDefundingObject	See the next table. This object is mandatory to be included if defunding is needed.	-	O

Payment Instruction object data elements – unencrypted part

Data element	Element description	Type	Presence indicator
creditorEntry	Unique identification of the end user's holding in the ledger.	Entry	M

Payment defunding object data elements – unencrypted part

Same as 'Defunding object data elements – unencrypted part' in DESP API POST /payment-instructions.

EUROPEAN CENTRAL BANK

EUROSYSTEM

Payment defunding object data elements – encrypted part

In the context of the pilot, this request body does not contain any encrypted part. The same applies to 'Defunding object data elements – encrypted part' in DESP API POST /defundings.

6.3.4.6. POST /payment-instructions/{payment-instruction-id}/status

Method	Endpoint	Resource	Description
POST	/v1/payment-instructions/{payment-instruction-id}/status	Payment	Endpoint for digital euro payer and/or payee PSP to receive the status of a payment instruction from DESP

6.3.4.6.1. POST /payment-instructions/{payment-instruction-id}/status – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header

Data element	Element description	Type	Presence indicator	Path/Header/ Query
payment-instruction-id	Resource ID generated by DESP to identify a payment transaction.	UUID	M	path

Request body data elements

Root

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	PaymentIdentification2	M
settlementDateTime	Time and date of the moment the settlement was recorded in DESP.	ISODateTime	C, mandatory in case of positive validation as indicated at the field 'payment transaction status'
settlementBusinessDay	Business day of the settlement which was recorded in Dedicated Cash Account (DCA) in DESP.	ISODate	C, mandatory in case of successful settlement
paymentTransactionStatus	Indication of the success or failure of the request. Allowed values: -RJCT	TransactionStatus	M

	-ACSC		
fundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	TransactionStatus	C, mandatory if funding object is present in the request
fundingId	Resource ID generated by the DESP for the funding part of the payment transaction.	UUID	C, mandatory if funding object is present in the request and in case of positive validation as indicated at the field 'payment transaction status'
defundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	TransactionStatus	C, mandatory if defunding object is present in the request
defundingId	Resource ID generated by the DESP for the defunding part of the payment transaction.	UUID	C, mandatory if defunding object is present in the request and in case of positive validation as indicated at the field 'payment transaction status'
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'payment transaction status'

6.3.4.6.2. POST /payment-instructions/{payment-instruction-id}/status – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 204, without response body.

6.3.4.7. POST /payments

Method	Endpoint	Resource	Description

POST	/v1/payments	Payment	Endpoint for the payer PSP to receive a payment request.

6.3.4.7.1. POST /payments – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: Not applicable

Request body data elements

Payload is the same as in 'Request body data elements' in DESP API POST /payments

6.3.4.7.2. POST /payments – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 201.

The full payload must be encrypted.

Root

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a request to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	PaymentIdentification2	M
paymentStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACTC	TransactionStatus	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'paymentStatus'

EUROPEAN CENTRAL BANK

EUROSYSTEM

6.3.4.8. POST / payments/{payment-id}/status

Method	Endpoint	Resource	Description
POST	/v1/payments/{payment-id}/status-inquiries	Payment	Endpoint for the payee PSP to receive status confirmation of a payment request from the payer PSP.

6.3.4.8.1. POST /payments/{payment-id}/status – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header

Data element	Element description	Type	Presence indicator	Path/Header/ Query
payment-id	Resource ID generated by the payee PSP to identify a payment request.	PaymentIdentification	M	path

Request body data elements

Payload is the same as in 'Request body data elements' in DESP API POST /payments/{payment-id}/status

6.3.4.8.2. POST /payments/{payment-id}/status – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 204, without response body.

6.3.4.9. POST /payments/{payment-id}/status-inquiries

Method	Endpoint	Resource	Description
GET	/v1/payments/{payment-id}/status-inquiries	payment	Endpoint for the payer PSP to receive query from the payee PSP for the status of a payment request.

6.3.4.9.1. POST /payments/{payment-id}/status-inquiries – Request

EUROPEAN CENTRAL BANK
EUROSYSTEM

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header.

Data element	Element description	Type	Presence indicator	Path/Header/ Query
payment-id	Resource identification generated by the payee PSP to identify a payment request.	UUID	M	path

Request body data elements

Data element	Element description	Type	Presence indicator
debtorAgent	Unique identification of the payer's PSP (BIC).	FinancialInstitutionIdentification1	M

6.3.4.9.2. POST /payments/{payment-id}/status-inquiries – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.** In case of positive technical validation: HTTP Code 200.

Root
The full payload is encrypted.

Data element	Element description	Type	Presence indicator
paymentIdentification	Includes uetr: Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a request to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	PaymentIdentification2	M
paymentStatus	Indication of the success or failure of the request. Allowed values: -RJCT -RCVD -ACCP -ACSC	StatusCode	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation

EUROPEAN CENTRAL BANK

EUROSYSTEM

			as indicated at the field 'payment Status'

6.3.4.10. POST /refunds

Method	Endpoint	Resource	Description
POST	/v1/refunds	Refund	Endpoint for DESP to forward refund instruction from the payer (original payee) PSP to payee (original payer) PSP.

6.3.4.10.1. POST /refunds – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header:

Request body data elements

Root

Data element	Element description	Type	Presence indicator
acceptanceDateTime	Date and time. This timestamp will be used for determination of maximum processing time. This is set by the first pilot PSP which starts the settlement process, and should not be altered by other pilot PSPs.	ISODateTime	M
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the	PaymentIdentification2	M
refundObject	See the table below.	-	M

Refund object data elements – unencrypted part

The payer entry that was specified by the payer PSP (the payee PSP of the original payment that is being refunded) will not be forwarded to the payee PSP (the payer PSP of the original payment that is being refunded).

Data element	Element description	Type	Presence indicator
refundId	Resource identification generated by DESP to identify a refund payment transaction.	UUID	M
debtorAgent	Unique identification of the payer's PSP (the payee PSP of the original payment that is being refunded).	Financial institution Identification1	M
creditorAgent	Unique identification of the payee's PSP (the payer PSP of the original payment that is being refunded).	Financial institution Identification1	M

EUROPEAN CENTRAL BANK

EUROSYSTEM

amount	The amount to be transferred from the payer to the payee (from the original payee to the original payer). This could be the full amount of the original payment or a lower amount.	Amount	M
EncryptedRefundObject	Encrypted data fields, see next section.		M

Refund object data elements – encrypted part

Same as the content 'Refund object data elements – encrypted part' included in the request DESP API POST /refunds.

6.3.4.10.2. POST /refunds – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 201.

Root

Data element	Element description	Type	Presence indicator
paymentIdentification	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	PaymentIdentification2	M
refundTransactionStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACCP	TransactionStatus	M
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'refundTransactionStatus'
refundObject	See the table below.	-	M
defundingObject	See the next table. This object is mandatory to be included if defunding is needed.	-	O

Refund object data elements – unencrypted part

Data element	Element description	Type	Presence indicator
creditorEntry	Unique identification of the payee's holding in the ledger. This field must be included if payer and payee have the same pilot PSP.	Entry	O

Defunding object data elements – unencrypted part

Same as 'Defunding object data elements – encrypted part' in DESP API POST /payment-instructions.

Defunding object data elements – encrypted part

In the context of the pilot, this request body does not contain any encrypted part. The same applies to 'Defunding object data elements – encrypted part' in DESP API POST /defundings.

6.3.4.11. POST /refunds/{refund-id}/status

Method	Endpoint	Resource	Description
POST	/v1/refunds/{refund-id}/status	Refund	The endpoint is for the payee and/or payer PSP to receive the status of a refund transaction from DESP.

6.3.4.11.1. POST /refunds/{refund-id}/status – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header.

Data element	Element description	Type	Presence indicator	Path/Header/ Query
refund-id	Resource identification generated by DESP to identify a refund transaction.	UUID	M	path

Request body data elements

Root

Data element	Element description	Type	Presence indicator
paymentIdentification	Includes uetr: Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.	PaymentIdentification2	M
settlementDateTime	Time and date of the moment the settlement was recorded in DESP.	ISODateTime	C, in case of a positive validation result as

			indicated in the field 'payment transaction status'
settlementBusinessDay	Business day of the settlement which was recorded in Dedicated Cash Account (DCA) in DESP.	ISODate	C, mandatory in case of successful settlement
refundTransactionStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	TransactionStatus	M
fundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	TransactionStatus	M
fundingId	Resource ID generated by the DESP for the funding part of the payment transaction.	UUID	C, mandatory if funding object is present in the request and in case of positive validation as indicated at the field 'refundTransactionStatus'
defundingStatus	Indication of the success or failure of the request. Allowed values: -RJCT -ACSC	TransactionStatus	C, mandatory if defunding object is present in the request
defundingId	Resource ID generated by the DESP for the defunding part of the payment transaction.	UUID	C, mandatory if defunding object is present in the request and in case of positive validation as indicated at the field 'refundTransactionStatus'
statusReason	Reason for the negative business validation result.	statusReason	C, mandatory in case of negative validation as indicated at the field 'refundTransactionStatus'

6.3.4.11.2. POST /refunds/{refund-id}/status - Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**

- In case of positive technical validation: HTTP Code 204, without response body.

6.3.4.12. POST /notifications/report-notifications

Method	Endpoint	Resource	Description
POST	/v1/notifications/report-notifications	NOTIFICATION	Report ready notification to pilot PSP.

6.3.4.12.1. POST /notifications/report-notifications – Request

Request Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**
- API specific header: no applicable.

Request body data elements

Data element	Element description	Type	Presence indicator
Agent	The BIC of the pilot PSP for which the report will be generated.	financialInstitutionIdentification1	M
reportType	Please see the relevant reference data table.	Report Type	M
reportId	Unique Identification of the generated report	UUID	M
reportMode	Indicates whether the report is requested in full mode or delta mode. Values: - FULL - DELTA	Report Mode	M
reportRtgsDate	The RTGS date for which the report is available.	ISODate	O
reportCalendarDate	The calendar date for which the report is available.	ISODate	O
reportStartDateTime	Start timestamp for which the report is issued.	ISODateTime	C, in case of delta mode only
reportEndDateTime	End timestamp for which the report is issued.	ISODateTime	C, in case of delta mode only

6.3.4.12.2. POST /notifications/report-notifications – Response

Response Header

- Technical header: please refer to **section 2.3.3.1. Technical header.**

- API specific header: not applicable.

Response body data elements

- In case of negative technical validation error: please refer to **section 1.4.2 Technical errors.**
- In case of positive technical validation: HTTP Code 204.

EUROPEAN CENTRAL BANK

EUROSYSTEM

7. Data dictionary

The data dictionary describes and defines the data elements that are used in the interface specifications. Future updates of the back-end implementation specifications, including additional services not covered in this release, will rely on this data dictionary; however, they may also introduce additional data types.

This section covers complex data types[28] in Section 7.1, generic data types[29] in Section 7.2, code lists in section 7.3 and finally other ISO related basic types in section 7.4. The data types used are based on ISO20022 whenever possible, with additional data types created for cases where no corresponding data type was found ISO20022.

For more information on data elements definitions and annotation, please refer to **section 1.**

7.1. Complex data types

Account Reference

Attribute	Type	Condition	Description
dean	DEAN	{Or - Optional	To be used for beta digital euro accounts. Reference to an account using either a DEAN or an IBAN. Two values are mutually exclusive.
iban	IBAN	Or - Optional}	To be used for commercial bank money accounts. Reference to an account using either a DEAN or an IBAN. Two values are mutually exclusive.

Account owner

Attribute	Type	Condition	Description
accountOwner	Max140Text	Mandatory	The name of the owner of the account.

Additional Party Information

Attribute	Type	Condition	Description
tradeName	Trade Name	Optional	Trade name of the related party.
merchantCategoryCode	Merchant Category Code	Optional	Merchant Category Code as assigned by ISO.
logoUrl	Max2048Text	Optional	A hyperlink to the logo of the party.
geoLocation	GEO Location	Optional	The geo location of the related party.

Address

[28] A complex data type (in ISO20022 often referred to as a Complex Business Component or Message Component) is a structured, hierarchical data element composed of multiple sub-elements. It allows for the grouping of related data, such as a full postal address or detailed party information, and can contain both primitive types and other nested complex types.

[29] Generic types in ISO20022 hold a single value like a string or a number.

Attribute	Type	Condition	Description
streetName	Max70Text	Optional	Name of the street.
buildingNumber	String	Optional	The number identifying the building on the street.
townName	String	Optional	Name of the town or city. This data attribute might be mandated by ASPSP (Account Servicing Payment Service Providers) for certain payment products, e.g. SEPA products, following regulatory requirements.
postCode	String	Optional	Postal code of the address.
country	Country Code	Mandatory	Country of the address, expressed as an ISO country code.

Agent Description 1

Attribute	Type	Condition	Description
financialInstitutionId	Financial Institution Identification	Mandatory	

Amount

Attribute	Type	Condition	Description
currency	Currency Code	Mandatory	ISO 4217 Alpha 3 currency code.
amount	String	Mandatory	The amount given with fractional digits, where fractions must be compliant to the currency definition. Up to 14 significant figures. Negative amounts are signed by minus. The decimal separator is a dot. Example: Valid representations for EUR with up to two decimals are: 1056 5768.2 -1.50 5877.78

DEAN

Attribute	Type	Condition	Description
dean	Max18Text	Mandatory	Unique identifier of a beta digital euro account.

EUROPEAN CENTRAL BANK

EUROSYSTEM

Entry

Attribute	Type	Condition	Description
Entry ID	TBD	Mandatory	Unique identification of the end user's holding in the ledger.

Financial institution Identification1

Attribute	Type	Condition	Description
bicfi	BICFI	Mandatory	Code allocated to a financial institution. BIC11 should be used.

Hashed Technical Proof

Attribute	Type	Condition	Description
hashedTechnicalProof	STR (tbd)	Mandatory	The hashed technical proof is generated by the pilot PSP and submitted to DESP.

Indicator Digit

Attribute	Type	Condition	Description
indicatorDigit	IndicatorDigit	Mandatory	Indicates whether the requested DEAN is intended for personal/household or commercial/professional use.

Payment Identification 1

Attribute	Type	Condition	Description
endToEndId	Max35Text	Optional	Unique identification, as assigned by the initiating user, to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain.

Payment Identification 2

Attribute	Type	Condition	Description
uetr	uetr	Optional	Unique End-to-end Transaction Reference, a unique, unalterable reference which allows a payment to be located at any time, by any of the parties in the chain. Set by the instructing PSP.

Party Description

Attribute	Type	Condition	Description
name	Max140Text	Optional	Name of the party.
postalAddress	Postal Address	Optional	Postal Address of the party.
identification	Party Identification	Optional	Unique and unambiguous identification of a party
additionalPartyInformation	Additional Party Information	Optional	

EUROPEAN CENTRAL BANK

EUROSYSTEM

Party Description 1

Attribute	Type	Condition	Description
name	Max140Text	Optional	Name of the party.

Party description 2

Attribute	Type	Condition	Description
name	Max140Text	Optional	Name of the party.
postalAddress	Postal Address	Optional	Postal Address of the party.

Party identification 1

Attribute	Type	Condition	Description
privateId	Private Identification	Mandatory	A scheme name defined in a proprietary way.

Person Identification

Attribute	Type	Condition	Description
identification	Max35Text	Mandatory	Unique and unambiguous identification of an oersib[30].
schemeNameCode	Person Identification Code	{Or - Optional	An entry provided by an external ISO code list.
schemeNameProprietary	Max35Text	Or – Optional}	A scheme name defined in a proprietary way.
issuer	Max35Text	Optional	Issuer of the identification.

Postal Address

Attribute	Type	Condition	Description
addressLine	Array of Max140Text	{Or - Optional	At most seven entries are permitted. May only be used, if none of the structured address elements "streetName", "buildingNumber", "postcode" or "townName" is used.
department	Max70Text	Or -Optional}	Identification of a division of a large organisation or building.
subDepartment	Max70Text	Or -Optional}	Identification of a sub-division of a large organisation or building.
streetName	Max70Text	Or -Optional}	Name of a street or thoroughfare.
buildingNumber	Max16Text	Or -Optional}	Number that identifies the position of a building on a street.
buildingName	Max35Text	Or -Optional}	Name of the building or house.
floor	Max70Text	Or -Optional}	Floor or storey within a building.
postBox	Max16Text	Or -Optional}	Numbered box in a post office, assigned to a person or organisation, where letters are kept until called for.
room	Max70Text	Or -Optional}	Building room number.

[30] Definition, including the interesting word oersib, taken literally from Berlin Group Definition

postCode	Max16Text	Or -Optional}	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail.
townName	Max35Text	Or -Optional}	Name of a built-up area, with defined boundaries, and a local government. Usage rule: If address lines are not used, this attribute is mandatory.
townLocationName	Max35Text	Or -Optional}	Specific location name within the town.
districtName	Max35Text	Or -Optional}	Identifies a subdivision within a country sub-division.
countrySubDivision	Max35Text	Or -Optional}	Identifies a subdivision of a country such as state, region, county.
country	Country Code	Optional	Nation with its own government. Usage rule: If address lines are not used, this attribute is mandatory.

Private identification

Attribute	Type	Condition	Description
birthDate	IsoDate	Optional	Date of birth
provinceOfBirth	Max35Text	Optional	Province of birth
countryOfBirth	Country Code	Optional	Country of birth
others	array of Person Identification	Optional	Other information

Proxy Account Identification

Attribute	Type	Condition	Description
aliasType	ExternalProxyAccountType1Code	Mandatory	Type of proxy used to identify an account (e.g. phone number, email address).
identification	Max2048Text	Mandatory	Proxy value used to identify the account.

Remittance

Attribute	Type	Condition	Description
reference	Max35Text	Mandatory	The actual remittance reference.
referenceType	Max35Text	Optional	Type of the remittance reference.
referenceIssuer	Max35Text	Optional	Issuer of the remittance reference.

Status Reason

Attribute	Type	Condition	Description
statusReasonCode	ExternalStatusReasonCode1	Mandatory	Machine readable reason code

statusReasonAdditionalInformation	Max140Text	Optional	Human readable explanation

Terminal ID

Attribute	Type	Condition	Description
terminalId	Max8Text	Mandatory	Unique identifier of the terminal used to initiate the transaction.

Transaction Status

Attribute	Type	Condition	Description
status	StatusCode	Mandatory	Indicates the outcome of the request.
statusReason	StatusReason	Optional	Provides detailed reason information.

UETR

Attribute	Type	Condition	Description
uetr	UUID (v7)	Mandatory	Universally unique identifier to provide an end-to-end reference of a payment transaction.

User Identification

Attribute	Type	Condition	Description
userId	UUID (v7)	Mandatory	Unique end user ID generated by the pilot PSP based on a subset of mandatory attributes (tbd).

7.2. Generic data types

This specification is only using the basic data elements "String", "Boolean", "ISODateTime", "ISODate", "UUID" and "Integer" (with a byte length of 32 bits) and ISO based code lists.

Max35Text, Max70Text, Max140Text Max500Text, Max1000Text are defining strings with a maximum length of 35, 70, 140, 500 and 1000 characters respectively.

7.3. Code lists

Credit Transfer Payment Method Code

This code set corresponds to ISO 20022 PaymentMethod3Code:

Code	Description
TRF	Credit Transfer

Indicator Digit

Code	Description
0	DEAN used for personal or household capacity.
1	DEAN used for commercial or professional purposes.

Payment Type

This code set can be considered exhaustive

Code	Description
BP01	B2P payment
DF01	Defunding
FU01	Funding
PB01	POI payment (POS / eCOM)
PP01	P2P payment

Report Mode

Code	Description
DELT	Delta Report
FULL	Full Report

Report Type

Code	Description
ADMR	Account data maintenance Report
DIRE	Dispute report
FECR	Fee calculation report
FRMA	Fraud Management - Situation awareness and intelligence report
REDR	Repository data report
RDMR	Repository data maintenance report
TRRE	Transactions report

Status Code

Code	Code name	Code definition
ACCP	AcceptedCustomerProfile	Preceding the check of technical validation was successful. The customer profile check was also successful.
ACSC	AcceptedSettlementCompletedDebitorAccount	Settlement is completed. Usage: this can be used by a Market Infrastructure reporting to Infrastructure Participant or an Account Servicer to Account Owner to report that the transaction account entry has been completed.
ACTC	AcceptedTechnicalValidation	Authentication and syntactical and semantical validation are successful
RCVD	Received	Payment instruction has been received.

RJCT	Rejected	Payment instruction has been rejected.

Status Reason Code

Code	Description
ExternalStatusReasonCode1	Status reason code.

User Type

Code	Description
BUSI	Business end user.
INDI	Individual end user.

7.4. Other ISO-related basic types

The following codes and definitions are used from **ISO 20022**:

- Purpose Code: ExternalPurpose1Code
- Category Purpose Code: ExternalCategoryPurpose1Code
- Cash Account Type: ExternalCashAccountType1Code
- BICFI: BICFIIdentifier (BIC11)
- IBAN: IBAN2007Identifier
  Pattern: [A-Z]{2}[0-9]{2}[A-Z0-9]{1-30}
- Phone Number: PhoneNumber
- Merchant Category Code: Category code conform to ISO 18245
- Service Level Code: ExternalServiceLevel1Code
- Status Reason Code: ExternalStatusReasonCode1
- Alias Type: ExternalProxyAccountTypeCode1

The following code is a concatenated code from ISO20022:

- BankTransactionCode: This code type is concatenating the three ISO20022 Codes Domain Code, Family Code and Subfamily Code by hyphens, resulting in “DomainCode”-“FamilyCode”-“SubFamilyCode”.

For all codes used in JSON structures, not the abbreviation defined for XML encoding, but the name of the code is used as a value.

The following Codes are used from other ISO standards:

- Currency Code: Codes following ISO 4217 Alpha 3
- Country Code: Two characters as defined by ISO 3166

Further basic ISO data types

EUROPEAN CENTRAL BANK

EUROSYSTEM

- ISODateTime: A particular point in the progression of time defined by a mandatory date and a mandatory time component, expressed in either UTC time format (YYYY-MM-DThh:mm:ss.sssZ), local time with UTC offset format (YYYY-MMDDThh:mm:ss.sss+/-hh:mm), or local time format (YYYY-MM-DDThh:mm:ss.sss).

  These representations are defined in "XML Schema Part 2: Datatypes Second Edition - W3C Recommendation 28 October 2004" which is aligned with ISO 8601. Strong Recommendation: Only use UTC or UTC offset format.
- ISODate: A particular point in the progression of time in a calendar year expressed in the YYYY-MM-DD format.