Central liquidity management

User detailed functional specifications

Author	4CB
Version	0.3
Date	16 July 2018

Table of contents

List of figures

List of tables

Introduction (completed)

This document describes the central liquidity management (CLM) as a component of the TARGET services and CLM participant’s interactions with other components and services. CLM is a business component of the TARGET service providing information on central bank liquidity and managing credit lines and central bank operations. In addition, CLM is the central component for funding the RTGS component, T2S and TIPS.

The document is intended to guide CLM participants to the proper understanding of the component and to offer all the information needed for the implementation of software interface on their side.

The UDFS document focuses on the provision of information to CLM participants to design and build the interface of their business application with CLM (A2A/U2A). The UDFS CLM is available for the whole community: in order to ensure the same level of knowledge for all CLM participants, including central banks, and participants.

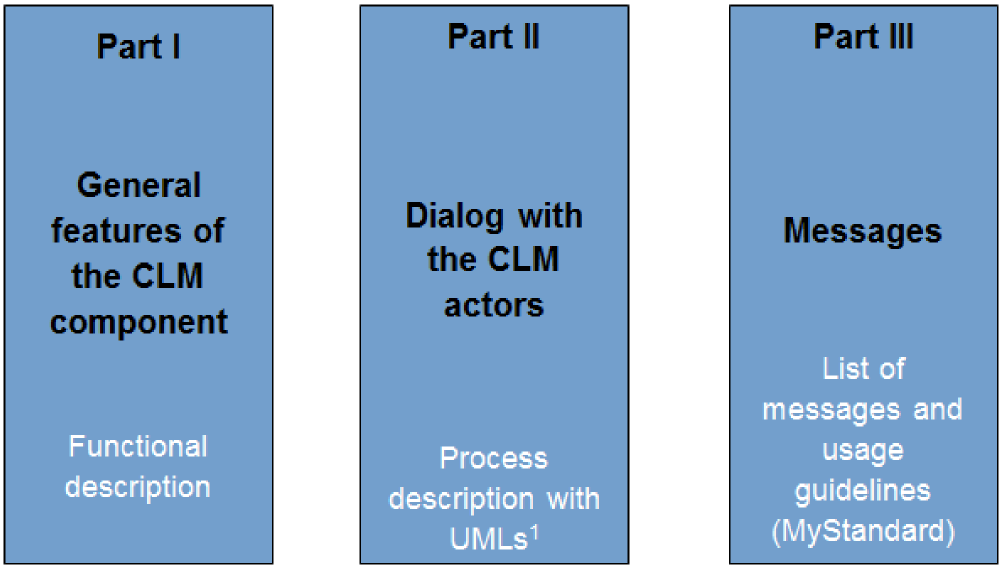
The document is divided into three parts.

- The first part provides a full description of all the CLM features and the related accounts and application processes, functional details concerning access to CLM and connectivity, dependencies and interactions with other services/components, operations and support features. The background information provided in chapter Overview of CLM component [▸ 25] supports the understanding of the CLM component with its applications and the interaction of the common components described in the following chapters. Afterwards, it guides the reader through the CLM core functionalities (i.e. participation and accounts) as well as an overview of common components used by CLM (e.g. CRDM, data warehouse). The contingency services are explained in chapter Contingency services [▸ 175] and central bank specific information is provided in chapter Additional information for central banks [▸ 177].
- The second part provides process descriptions, which allows CLM participants to interact with CLM via A2A as well as a functional overview of the U2A processes. This part aims at providing a comprehensive description of all processes being available in CLM and which the user may instruct. Moreover, the related settlement processes are explained in detail. Furthermore, the chapter Dialogue between CRDM and CRDM actors describes the dialogue between common reference data management (CRDM) and participants via A2A. Subsequently, also the interaction with ESMIG is outlined in chapter Dialogues and processes [▸ 200].
- The third part provides a detailed description of all XML messages CLM participants may use to interact in A2A mode with CLM. The description of the messages includes all required elements according to the schema. Wherever a message or its fields are referenced throughout the document, only the reference name is used.

This document has been submitted for market consultation on XX.XX.2018 and was finally published on the ECB website on XX.XX.2018.

Reader's guide (completed)

The document is structured as to guide the readers through the steps of the whole (A2A) interaction and processing details focusing on different user needs, i.e. business experts, IT experts and message experts.


Part I
General features of the CLM component
Functional description
Part II
Dialog with the CLM actors
Process description with UMLs[1]
Part III
Messages
List of messages and usage guidelines (MyStandard)

[1] UML = Unified Modelling Language

Figure 1 - Structure of the CLM UDFS

Different readers may have different needs and priorities and may not need to read the whole book.

For instance, business readers, interested mainly in organisational issues, may not wish to enter into the full details of each message description, but they might prefer going through a description of the business processes and the information flows between their own business application(s) and CLM. On the other hand, technical readers involved in the specification and development of technical interfaces to CLM may not be interested in the complete description of the features CLM offers. They would probably search the necessary information to design and build the interface of the CLM participants' business application with CLM. The following paragraphs show - with a couple of examples - how business and technical readers may follow different reading patterns in order to fulfil their needs.

All readers, whether business or technical, are invited to read the following UDFS sections, which are providing a minimum functional and technical background to the understanding of any other UDFS chapter.

- Overview of CLM component [▸ 25], which summarizes the CLM features and functionalities;
- Access to CLM [▸ 27], which focuses on how to connect to CLM including authentication and authorisation processes and explains the envisaged usage of access rights depending on the respective roles;
- Parties and accounts [▸ 29], which provides a general description of the main reference data needed for CLM and the accounts maintained in CLM, specifying how they are used for the settlement of a liquidity transfer (e.g. which parties and related accounts are involved and how to set up groups for monitoring the liquidity transfer activities);

- Contingency services [▸ 175], which informs how to act in case of a defined contingency situation.

Business oriented perspective

In addition, a business reader may be interested in the way information is structured in CLM. This user may want to follow the reading plan described below to find further details about the operations possible in CLM:

- Business day [▸ 49], where the business reader finds an overview of respective processes and schedules;
- Business and features description [▸ 50], which informs about the settlement process of payments as well as the liquidity-, reserve- and information-management;
- Overview of used common components in CLM component [▸ 126] completes the view on the message transfers used in CLM;
- Payment order processing [▸ 185] to find a description of the processing of a payment order and useful information in order to understand the management of liquidity
- Dialogue between CRDM and CRDM actors wherein query information may be of relevance;
- Index of business rules and error codes [▸ 284] including the relevant codes to perform functional checks.

Technical oriented perspective

- Processes with CLM components [▸ 185] respectively Dialogue between CRDM and CRDM actors, where an overview of the possible A2A dialogue with CLM is defined. Each sub-chapter of this chapter describes the flows within and to and from CLM. The reader can focus on the functionality of note, analysing the procedures and main scenarios;
- Part III Catalogue of messages [▸ 208], where a detailed description of the content of a given XML message is provided;
- Index of status value and codes [▸ 286] with further details on the checks to be performed and codes used in the messages.

This chapter is subject to further review depending on the subsequent maintenance of the UDFS document in the future.

I General features of the CLM component

1 Overview of CLM component (completed)

CCLM ensures the adequate provisioning and clear allocation of liquidity for the different settlement purposes across all TARGET services and accounts in a currency.

The primary aim of CLM is to offer a centralised mechanism for the steering, monitoring and management of payment capacity. All credit institution´s transactions with its central bank are managed in CLM (including the ones related to central bank operations such as reserve management and standing facilities). The interaction with central banks is segregated from the real-time interbank/customer payments as well as the ancillary system transactions in RTGS.

CLM provides:

- instruments for the management of liquidity such as immediate/standing or automatic liquidity transfer orders and floor/ceiling definitions;
- information tools, queries and reporting for the status monitoring of liquidity and processing results.

In order to reach these objectives, the CLM holds the main cash accounts (MCA) as the central source of liquidity; the main cash account is used for all central bank operations as well as the management of the credit line. The available liquidity is transferred to the dedicated cash accounts (dedicated cash accounts) of RTGS, TIPS and T2S. The minimum reserve calculation and automated standing facilities take all balances on relevant accounts (main cash account, dedicated cash account) into account. A main cash account holder is responsible for its own liquidity management and for monitoring the settlement process or grant access to another party to perform these tasks on its behalf. The A2A communication between credit institutions and all TARGET services and common components is based on the ISO 20022 compliant messages.

CLM makes use of the following Eurosystem common components.

- The Eurosystem single market infrastructure gateway (ESMIG) provides the central authentication, authorisation and user management features. It is network provider agnostic and thus offer participants the access to all TARGET services through the connection with a single certified network service provider. All network service providers require compliance with the same communication interface specifications in application-to-application (A2A) mode (in store-and-forward and real-time communication protocol) and user-to-application (U2A) mode via GUI.
- The common reference data management (CRDM) component offers features that allow authorised users to set up, maintain and query all reference data that TARGET services share for their processing activities. CRDM ensures the consistency and integrity of all reference data, processing and relationships across services. Furthermore, it avoids duplication of reference data or redundant implementation of the same functions in multiple services. Service-specific reference data objects (or functions) are set up and managed (or implemented) in the respective service. The access to all collected data allows to use a billing component as well as queries and reports.
- The data warehouse (DWH) component provides the data from T2 (i.e. CLM and, RTGS) or T2S for historical, statistical and regulatory reporting. It offers predefined reports and templates for database que-

ries. The data warehouse participants may access the data warehouse in A2A and U2A (via) mode and subscribe for respective reports and templates. The data of previous business days are available in data warehouse as of the next business day.

- The business day management offers the common scheduler and calendar for all services and components. A common scheduler defines the structure of the business day in the TARGET services as well as the events per currency for which participants may configure event-based standing orders and regular reports. The common calendar defines the days when a TARGET service or a common component is opened and follows the defined business day schedule or contrary is closed. Each TARGET service may have a different calendar per currency.
- The billing component ensures the preparation and processing of invoices for the different TARGET services and common components. To do so, relevant information for each cash account has to be defined in CRDM (e.g. to whom the invoice is addressed to, which main cash account is debited, etc.) and this information is then taken into account during the billing process. Further information on billing and the respective fees is defined in a pricing guide.
- The legal archiving component collects all information, which is subject to legal archiving requirements such as all incoming and outgoing business transactions from and to participants as well as relevant reports such as account statements. The information from TARGET services and common components is stored in legal archiving in its original content and format after thirty calendar days and is accessible within its retention period of ten years.

2 Access to CLM

2.1 Connectivity (U2A/A2A) (to be completed in iteration 4)

2.2 Authentication and authorisation process (to be completed in iteration 4)

2.3 Security (completed)

This section aims at describing the main processes performed by CLM in terms of principles applied to ensure CLM actors can securely exchange information with CLM.

It means that the following security conditions are met:

- **Confidentiality:** ensuring that information is accessible only to authenticated and authorised CLM Actors
- **Integrity:** safeguarding the accuracy and completeness of information
- **Availability:** ensuring that authorised users have access to information and associated assets when required
- **Monitoring:** detecting operational and technical problems and recording appropriate information for crisis management scenarios and future investigations
- **Auditability:** ensuring the possibility to establish whether a system is functioning properly and that it has worked properly

2.3.1 Confidentiality

The confidentiality of data is ensured by the possibility to grant specific access rights for any given set of data, as detailed in section “Access rights”. In conjunction with mechanisms of authentication and authorisation applied to all requests received by CLM in both A2A and U2A mode, this guarantees that each CLM actor’s data is treated confidentially and is not accessible to non-authorised actors.

2.3.2 Integrity

Within CLM, various business validations ensure the integrity of information. If a business validation fails, CLM has a concept of error handling in place. The requested action is not processed and CLM provides the user with detailed information regarding the nature of the error.

In U2A mode, CLM offers users in addition the possibility to further ensure the data integrity via usage of a dual authorisation concept, the 4-eyes principle. In case this option is chosen for a specified set of CLM operations, a second independent verification and confirmation is required before an operation becomes active in CLM. If, for example, a critical set of data should be modified and the person requesting the change is only allowed to do so under the 4-eyes principle, then a second person of the same party has to confirm the correctness of the request. Otherwise, the requested change is not implemented.

2.3.3 Availability

The overall availability of the CLM services is ensured by the functional design and a centralised technical architecture. This, together with a high level of inherent infrastructure redundancy and dedicated IT resources ensure the maximum availability for the CLM services.

2.3.4 Monitoring

CLM operational monitoring provides tools to the T2 operator for the detection in real-time of functional or operational problems. Technical monitoring allows for the detection of hardware and software problems via real-time monitoring of the technical components involved in the processing, including the network connections.

2.3.5 Auditability

CLM provides an audit trail with which it is possible to reconstruct user activities, exceptions and information security events. More in detail, the following data are collected:

- payment transaction and liquidity transfer records;
- authentication successes and failures of normal and privileged users;
- security related messages (e.g. changes of access rights, alerts and exceptional events).

2.4 Graphical user interface (to be completed in iteration 4)

3 Parties and accounts

3.1 Parties and CLM actors (completed)

Entities that interact with CLM are generally known as CLM actors. The CLM participation model envisions different types of actors, with different roles and responsibilities, as outlined in chapter Concept of party in CLM [▸ 30] CLM actors are defined as different entities in CRDM.

This chapter provides a detailed description of all the reference data CRDM stores and CLM uses for all CLM actors. More in detail, chapter Setup of CLM actors [▸ 29] identifies the reference data related to the setup of actors for CLM and it provides detailed information as to who is responsible for the setup of these reference data. Chapter 3.1.1.2 defines the concept of party in CRDM and the way this concept relates with the different types of legal entities that can interact with CLM. Chapter Hierarchical party model [▸ 31] describes the so-called hierarchical party model, i.e. the organisational structure of parties in the CRDM repository. The chapetrs Party identification [▸ 31] and Reference data for parties in CLM [▸ 31] illustrate the reference data required by CLM for each actor, i.e. the way a party can be identified in CLM and which attributes have to be stored for each actor.

3.1.1 Setup of CLM actors (completed)

The setup of CLM actors takes place in CRDM.

The T2 operator is responsible for setting up and maintaining party reference data for all central banks in CLM. Central banks are responsible for setting up and maintaining party reference data for the parties of their national community.

The following table summarises, for each reference data object related to the setup of CLM actors, the actor responsible for its configuration and it specifies which mode the actor can use for the configuration.

Table 1 - Setup of parties for CLM

Reference data object	Responsible actor	Mode
Party (central bank)	T2 operator	U2A
Party (CLM participant)	Central bank	U2A
Banking group	Central bank	U2A
Monetary financial institution (MFI)	Central bank	U2A

The CLM actor “authorised account user” will be described in iteration 4.

3.1.2 Concept of party in CLM (completed)

Any CLM actor, meaning any legal entity or organisation participating in and interacting with CLM, is defined as an entity in the CRDM repository. Depending on their role in CLM, CLM actors may be defined as a party in CRDM. Each party belongs to one of the following party types:

- T2 operator
- central bank
- CLM participant

The **T2 operator** is the organisational entity that operates CLM. They are responsible for the initial setup and day-to-day operations of CLM and act as single point of contact for central banks in case of technical issues. They are responsible for monitoring the system and carrying out corrective actions in case of incidents or in the event of service/component unavailability. The T2 operator is also responsible for setting up and maintaining central banks reference data in the CRDM repository and, if required, they may operate CLM functions on behalf of any CLM actor, upon request of the respective central bank. They have full access to all live and all archived reference data and transactional data in CLM.

Central banks are responsible for setting up and maintaining reference data in the CRDM repository for all CLM actors belonging to their community. Central banks can also act as CLM participants (see below) themselves. In addition they can act on behalf of one of their CLM participants in case of need.

In its central bank role, it may only own central bank accounts (see Glossary [▸ 287] for the definition of a central bank account); all other account types need to be owned under its CLM participant role.

CLM participants represent entities that own main cash accounts in CLM and are identified by a BIC11. CLM participants are responsible for their own liquidity management and have to make sure that sufficient liquidity is available in the different settlement services that they use. They are responsible for setting up their own main cash accounts, instructing payments and monitoring the liquidity usage. However, the creation and maintenance of the main cash accounts is done by central banks.

The role of **banking group** allows a number of parties (belonging to one or multiple central banks) to be viewed collectively for certain business purposes, such as oversight and regulation. Banking group is not defined as a party, but as a banking group identifier that central banks can define.

The role of **monetary financial institution (MFI)** allows a pool for management of minimum reserves. MFI is not defined as a party, but as a code that central banks can define.

Each legal entity may play different roles in CLM. Any legal entity playing multiple business roles in CLM results in the definition of multiple parties.

Conversely, a (non-central-bank) legal entity owning two main cash accounts within the books of a central bank would be defined as two different CLM participants, each identified by a different BIC11.

Similarly, a (non-central-bank) legal entity owning two main cash accounts within the books of two central banks would also be two separate CLM participants, each identified by a different BIC11.

3.1.3 Hierarchical party model (completed)

The party model of CLM is based on a hierarchical three-level structure. The T2 operator is the only party on the top level of the hierarchy and is responsible for the setup of each party of the second level, i.e. each central bank in CLM. Similarly, each party belonging to the second level (i.e. a central bank) is responsible for the setup of all parties of its community (i.e. CLM participants), represented by parties of the third level.

The hierarchical model also determines the reference data scope, i.e. the area of responsibility of each central bank and of the T2 operator. More into detail:

- The reference data scope of the T2 operator includes all reference data, i.e. all the reference data in scope of all central bank plus CLM operational specific reference data.
- The reference data scope of a central bank includes its own reference data, plus the reference data of all its CLM participants.
- The reference data scope of a CLM participant includes only its own reference data.

Each central bank and the T2 operator are responsible for their own reference data scopes, i.e. each of them is responsible for the input and maintenance of all information included in its reference data scope. The T2 operator may also act, upon request, on the reference data scope of a central bank and its CLM participants.

3.1.4 Party identification (completed)

CLM imposes a constraint in the assignment of BICs related to its parties, due to the fact that the settlement process must be able to determine the accounts to be debited and credited by a payment based on the BICs of the CLM participant and the central bank. This implies the need to ensure that any given BIC can only be assigned to one CLM participant and Central Bank and that two different CLM participants or Central Banks must have assigned two different BICs.

For this reason, CRDM prevents two different parties to be defined as CLM participants if they are identified by the same 11-character BIC. Therefore, in order to allow a given legal entity to be defined as two different CLM participants (by the same central bank or by two different central banks), the same legal entity must be defined in the CRDM repository as two parties identified by two different 11-character BICs.

3.1.5 Reference data for parties in CLM (completed)

The following table gives an overview on the party reference data attributes in CLM.

Table 2 - Party reference data attributes

Attribute	Description
Party identifier	It specifies the unique technical identifier of the party.
Party long name	It specifies the full name of the party.
Party short name	It specifies the short name of the party.
Jurisdiction	It specifies the country of jurisdiction for the party. This attribute shall be mandatory for a legal address. It shall be the same country as in the legal address, except for supranational institutions.
Street	It specifies the name of the street for the address.
House number	It specifies the house number for the address.
City	It specifies the name of the city for the address.
Postal code	It specifies the postal code for the address.
State or province	It specifies the state or province for the address. Its use shall depend on the country code of the address.
Country code	It specifies the two-character ISO country code (ISO3166-1) identifying the country code of the address.
Party BIC code	It specifies the BIC11 to uniquely identify the party in CLM.
Parent BIC code	It specifies the BIC11 code of the parent responsible for the party. Where the party is a parent and there is no other party having responsibility over it, then parent BIC code is the same as the party BIC code
Institutional sector code	It identifies the financial corporation's sector classification to which the party belongs with respect to the nature of its business.
Party type	Type of party. The exhaustive list of party types is as follows: - T2 operator - central bank - CLM participant
Party status	It specifies the business status of a party for processing in the system (e.g. active).
Intraday credit indicator	It specifies the intraday credit indicator by either allowing or not allowing intraday credit.
Intraday credit limitation	It specifies the maximum intraday credit authorised to a Party.
Standing facilities indicator	It specifies the standing facilities indicator by either allowing or not allowing standing facilities.

Attribute	Description
Minimum reserve obligation	It specifies whether or not the party is subject to/exempted from minimum reserve requirement.
Banking group identifier	It specifies the unique technical identifier of the banking group to which the party belongs to.
LEI	It specifies the unique identifier of the legal entity in accordance with the ISO 17442 standard.
Monetary financial institution (MFI)	It specifies the monetary financial institution (MFI) with which the party is associated for the calculation of minimum reserves via a pool.
MFI leader BIC	It specifies the BIC of the party designated as the MFI leader where minimum reserves are managed in a pool.
Account for minimum reserves	It identifies the account used by the MFI leader for minimum reserves.
Marginal lending account	It specifies the account number of the marginal lending account managed within CLM and maintained by a central bank to settle all marginal lending transactions.
Overnight deposit account	It specifies the account number of the overnight deposit account managed within CLM and maintained by a central bank to settle all overnight deposit transactions.
First activation date	It specifies the date of the first activation of the party.
Modification date	It specifies the activation date of the displayed party status.
Currency code	It specifies the national currency associated with the party.
VAT1	It specifies the national rate of value added tax associated with a Central Bank.
VAT 2	It specifies the additional national rate of value added tax associated with a central bank.
Account to be credited (for central banks only)	It specifies the cash account to be credited within the billing process. Different accounts may be specified for each different service.
Direct invoicing flag (for central banks only)	This flag indicates whether invoices are sent directly to the CLM participants or whether they are sent via the central bank.

The following table gives an overview on the party contact reference data attributes in CLM.

Table 3 - Party contact reference data attributes

Attribute	Description
Contact name	It specifies the name of the contact for the party.
Contact position	It specifies the position or role of the contact for the party.
Valid from date	It specifies the date from which the party contact is valid.
Office telephone number	It specifies the office telephone number for the party contact.
Mobile number	It specifies the mobile number for the party contact.
Email address	It specifies the email address for the party contact.
Valid to date	It specifies the date until when the party contact is valid.

The following table gives an overview on the banking group reference data attributes in CLM.

Table 4 - Banking group reference data attributes

Attribute	Description
Banking group identifier	It specifies the identifier of the banking group.
Banking group name	It specifies the name of the banking group.

The following table gives an overview on the monetary financial institution reference data attributes in CLM.

Table 5 - The monetary financial institution reference data attributes

Attribute	Description
MFI Code	It specifies the unique identifier of the monetary financial institution.
Current maintenance period from	It specifies the date range of the current maintenance period.
Current maintenance period to	It specifies the date range of the current maintenance period.
Minimum Reserves (EUR)	It specifies the minimum reserve requirement of the monetary financial institution.

As soon as a party is subject to minimum reserve, a main cash account is to be opened. If a party wants to participate in settlement in RTGS, T2S and/or TIPS, then it must hold a corresponding dedicated cash account (dedicated cash account).

3.2 Accounts structure and organisation (completed)

Accounts are opened in CLM for the provision of liquidity and the settlement of central banks operations.

This chapter provides a detailed description of the reference data CRDM stores and CLM uses for all its accounts.

The T2 operator and central banks input and maintain in the CRDM repository the following categories of accounts, depending on their role:

- main cash accounts
- dedicated transit accounts
- central bank accounts
- overnight deposit accounts
- marginal lending accounts
- central bank's ECB accounts
- ECB mirror accounts

Furthermore, CLM participants may define:

- liquidity transfer groups
- account monitoring groups
- direct debit mandate(s)
- linked dedicated cash account(s)
- floor/ceiling information
- current reservations(s)
- standing liquidity transfer orders
- standing orders for reservation
- notification message subscription
- report configuration

Even if defined by the CLM participant, the input and maintenance are in some cases done by central banks. It is however up to central banks to define the default dedicated cash accounts in CLM. This is described in the following chapters.

The following sections define the above mentioned reference data objects, whereas chapter "Reference data for accounts in CLM" provides a detailed description of the reference data required by CLM for the same reference data objects.

3.2.1 Main cash accounts (completed)

A main cash account is an account used for the settlement of central bank operations and liquidity transfers, as well as the management of the credit line (cash side).

A CLM actor may own several main cash accounts. However, the credit line can only be assigned to one of them.

A main cash account in CLM is identified by a BIC11 (that must be unique in CLM) and also by a unique account ID (that must be unique across all settlement services). In the case of settlement of credit transfers (pacs.009) and direct debits (pacs.010), the CLM participant's main cash account is identified by a unique "BIC11" code. In the case of liquidity transfers (camt.050), the CLM participant´s main cash account is identified by the account ID.

It is up to central banks to create and maintain main cash accounts for their CLM participants.

3.2.2 Dedicated transit accounts (completed)

Dedicated transit accounts in CLM are accounts that are owned by central banks which may have either zero or positive balance as they reflect any movement of liquidity from/to the various settlement services (i.e. RTGS, T2S and TIPS). They are technical accounts involved in the liquidity transfer process and cannot be involved in the settlement of central bank operations.

There is only one dedicated transit account per settlement service/settlement currency combination in CLM. The dedicated transit accounts for euro belong to the European Central Bank.

It is up to the T2 operator to create and maintain dedicated transit accounts for the central banks.

3.2.3 Central bank accounts (completed)

A central bank account in CLM is a cash account owned by a central bank of issue, is allowed to have negative balance and cannot be restricted or limited in its use.

A central bank account in CLM is identified by a BIC11 (that must be unique in CLM) and by a unique account ID (that must be unique across all settlement services).

It is up to the T2 operator to create and maintain the central bank accounts.

3.2.4 Overnight deposit accounts (completed)

An overnight deposit account is an account that is used in the context of overnight deposits. This account is owned by the relevant central bank but is opened in the name of the CLM participant.

There is one overnight deposit account for each CLM participant subject to standing facilities.

An overnight deposit account in CLM is identified by a unique account ID (that must be unique across all settlement services).

It is up to the central bank to create and maintain the overnight deposit accounts of its CLM participants.

3.2.5 Marginal lending accounts (completed)

A marginal lending account is an account that is used in the context of the marginal lending facility. This account is owned by the relevant central bank but is opened in the name of the CLM participant.

There is one marginal lending account for each CLM participant subject to standing facilities.

A marginal lending account in CLM is identified by a unique account ID (that must be unique across all settlement services).

It is up to the central bank to create and maintain the marginal lending accounts of its CLM participants.

3.2.6 Central bank’s ECB accounts (completed)

A central bank’s ECB account is an account that records the central bank´s asset/liability position towards the ECB in respect of cross-border transactions. This account is owned by the relevant central bank and is identified by a unique BIC11.

It is up to the T2 operator to create and maintain the central bank’s ECB accounts.

3.2.6.1 ECB mirror accounts (completed)

A ECB mirror account is an account owned by the ECB for each central bank on which the bookings done on the central bank’s ECB accounts are “mirrored“. This account is owned by the ECB and is identified by a unique BIC11.

It is up to the T2 operator to create and maintain the ECB mirror accounts.

3.2.7 Liquidity transfer groups (completed)

A liquidity transfer group refers to an optional grouping of main cash accounts for the purpose of arranging intra-CLM liquidity transfers between them. It is possible for an account to participate to one or multiple liquidity transfer groups.

The liquidity transfer group is identified by a specific ID.

It is up to central banks to create and maintain the liquidity transfer groups and define the main cash accounts linked to each liquidity transfer group.

3.2.8 Account monitoring groups (completed)

An account monitoring group is an optional grouping of accounts (main cash account(s) and dedicated cash account(s)) for liquidity monitoring purposes. It is possible for an account to participate to one or multiple account monitoring groups.

The account monitoring group is identified by a specific ID.

It is up to CLM participants to create and maintain their account monitoring groups and define the accounts linked to each account monitoring group.

3.2.9 Direct debit mandate (completed)

For each CLM participant CRDM manages the information about the direct debit(s) this participant has authorised and the related attributes (e.g. maximum amounts).

It is up to central banks to create and maintain the direct debit mandate(s) of a CLM participant in CRDM.

3.2.10 Default dedicated cash account (completed)

In case of automatic liquidity transfer orders, there is a need to define the default RTGS dedicated cash account to pull the amount of liquidity missing to settle the central bank operation.

It is up to central banks to define the default RTGS dedicated cash account of a CLM participant in CRDM.

3.2.11 Linked dedicated cash account (completed)

In the event the floor or ceiling on a main cash account is breached (after the settlement of a payment) and if the CLM participant has opted for the automated liquidity transfer order generation, CLM generates automatically an inter-service liquidity transfer order to pull cash from the linked dedicated cash account (in the event the floor is breached) or push cash to the linked dedicated cash account (in the event the ceiling is breached).

It is up to CLM participants to create and maintain the linked dedicated cash accounts in CRDM.

3.2.12 Floor/ceiling (completed)

For each main cash account, a CLM participant can define in CRDM a minimum (“floor”) and maximum (“ceiling”) amount that shall remain on the respective account. The CLM participant can choose between the following behaviours that the system shall apply in the event the floor or ceiling on an account is breached (after the settlement of a central bank operation):

1. CLM generates a notification that is sent to the CLM participant informing about the floor/ceiling breach (upon which the CLM participant can take action); or
2. CLM generates automatically an inter-service liquidity transfer order to pull cash from the defined d in RTGS (in the event the floor is breached) or push cash to the defined dedicated cash account in RTGS (in the event the ceiling is breached).

It is up to CLM participants to create and maintain the floor/ceiling information in CRDM.

3.2.13 Current reservation (completed)

Liquidity can be reserved and modified intra-day by CLM participants for the execution of central bank operations.

This information is defined at the level of the main cash account and it is up to CLM participants to set up and manage the current reservations in CLM.

3.2.14 Standing liquidity transfer order (completed)

A standing liquidity transfer order is an instruction of a CLM participant to transfer regularly a fixed amount of liquidity, upon a certain event, from a main cash account to another account over a period with or without a predefined end date.

This information is defined at the level of the main cash account and it is up to the CLM participant to create and manage its standing liquidity transfer orders information in CRDM.

3.2.15 Standing order for reservation (completed)

A standing order for reservation is an instruction of a CLM participant to set up an urgent reservation of a fixed amount for a business day on a main cash account without a predefined end date.

This information is defined at the level of the main cash account and it is up to the CLM participant to create and manage its standing order for reservation information in CRDM.

3.2.16 Notification message subscription (completed)

Message subscription shall allow a CLM participant to elect another party to receive some pre-defined messages either instead or in addition.

This information is defined at the level of the main cash account and it is up to the CLM participant to create and manage the notification message subscription in CRDM.

3.2.17 Report configuration (completed)

The CLM participant can configure standard reports that CLM shall create at certain times during a business day or at certain business day events. CLM participants can specify in their report configuration, whether such report shall be sent to the recipient immediately in A2A mode or be stored for later querying in A2A mode or downloading via GUI. Such standard reports are available for later querying and downloading until the next report based on the same configuration is created.

Report configuration shall also allow a CLM participant to elect another party to receive the report either instead or in addition.

This information is defined at the level of the main cash account and it is up to the CLM participant to create and manage the report configuration in CRDM.

3.2.18 Reference data for accounts in CLM (completed)

This chapter provides an overview of the attributes of the reference data objects previously described and does not give any indication on the structure of CRDM reference data tables.

The following table shows an exhaustive list of account reference data attributes in CLM.

Table 6 - Account reference data attributes

Attribute	Description
Account number	It specifies the number of the account (unique across all services).
Account type	Type of account. The exhaustive list of account types in CLM is as follows: - main cash account - dedicated transit account - central bank account - overnight deposit account - marginal lending account - NCB's ECB account

Attribute	Description
	- ECB mirror account
Currency	It specifies the currency of the account.
Account owner	It specifies the BIC11 of the party owning the account (unique within CLM).
Status	Blocking status for the account. Exhaustive list of possible values: - blocked for credit - blocked for debit - blocked for credit and debit - unblocked
Floor	It specifies a lower threshold which may trigger the sending of a notification message and/or a liquidity transfer order if it is breached from above (absolute numbers).
Ceiling	It specifies an upper threshold which may trigger the sending of a notification message and/or a liquidity transfer order if it is breached from below (absolute numbers).
Target amount after breaching floor	It specifies the target amount to be reached if the floor is breached.
Target amount after breaching ceiling	It specifies the target amount to be reached if the ceiling is breached.
Maximum amount for direct debit per day	It specifies the maximum amount of direct debits which can be debited each day on the main cash account.
Party to be billed	It specifies the party to whom the invoice is addressed.
Party to be charged	It specifies the party to whom the billable item is assigned, due to a contractual agreement.
Main cash account to be debited	It specifies the main cash account to be debited within the billing process.
Linked dedicated cash account	It specifies the linked dedicated cash account.
Minimum reserve party	It specifies the party for which this account is included for minimum reserve calculation.
Management of minimum reserve	It specifies the method by which the minimum reserve is managed. Possible values are: - direct - pool - no
Default flag	It indicates whether the account is the default choice of the party.

Attribute	Description
Account monitoring group identifier	It specifies the unique technical identifier of an account monitoring group.
Opening date	Opening date of the account.
Closing date	Closing date of the account.

Each main cash account is linked to one and only one CLM participant (i.e. the account owner); similarly, each dedicated transit account is linked to one and only one central bank (the European Central Bank for the euro dedicated transit accounts, the relevant central bank for any other settlement currency).

Furthermore, each main cash account may be linked to one or many liquidity transfer groups and to one or many account monitoring groups.

The following table shows an exhaustive list of liquidity transfer group reference data attributes in CLM.

Table 7 - Liquidity transfer group reference data attributes

Attribute	Description
Liquidity transfer group identifier	It specifies the unique technical identifier of the liquidity transfer group.
Liquidity transfer group name	It specifies the name of the liquidity transfer group.
Account(s)	It specifies the account(s) belonging to the liquidity transfer group.

The following table shows an exhaustive list of direct debit reference data attributes in CLM.

Table 8 - Direct debit reference data attributes

Attribute	Description
Direct debit identifier	It specifies the unique technical identifier of the direct debit mandate.
Account number	It specifies the account on which the direct debits are authorised.
Payee party identifier	It specifies the party from whom payment requests have been authorised under this mandate and to whom the corresponding payments are made.
Payee reference	The reference provided by the payee party to be included in the payment details for recognition of the payment.
Maximum amount (counterpart)	It specifies the maximum amount the authorised issuer is able to direct debit during the single business day.

Attribute	Description
Maximum amount per payment	It specifies the maximum amount the authorised issuer is able to direct debit in a single direct debit.
Valid from date	It specifies the date from which the direct debit instruction is valid.
Valid to date	It specifies the date until which the direct debit instruction is valid.

The following table shows an exhaustive list of the standing liquidity transfer order reference data attributes in CLM.

Table 9 - Standing liquidity transfer order reference data attributes

Attribute	Description
Standing order identifier	It specifies the unique technical identifier of the standing order.
Transfer type	It specifies the type of the liquidity transfer. The exhaustive list of transfer type options in CLM is as follows: l Inter-service liquidity transfer from main cash account to dedicated cash account l Intra-service liquidity transfer to another main cash account
Reference of instruction	It specifies the reference given by the original instructor of the liquidity transfer.
Transfer amount	It specifies the amount to be debited with the liquidity transfer.
Currency	It specifies the currency of the amount to be debited with the liquidity transfer.
Main cash account to be debited	It specifies the main cash account to be debited in CLM.
Account to be credited	It specifies the account (dedicated cash account and/or main cash account) to be credited.
Trigger event	It specifies the event type that triggers the transfer of liquidity.
Valid from date	It specifies the date from which the standing order is valid.
Valid to date	It specifies the date until which the standing order is valid.

The following table shows an exhaustive list of the standing order for reservation reference data attributes in CLM.

Table 10 - Standing order for reservation reference data attributes

Attribute	Description
Standing order for reservation identifier	It specifies the unique technical identifier of the standing order for reservation.
Priority type	It specifies the type of priority. The exhaustive list of priority class options is as follows: - urgent
Reservation amount	It specifies the amount of the required reservation.
Account	It specifies the account number of the main cash account for which the reservations are made.
Valid from date	It specifies the date from which the standing order for reservation is valid.
Valid to date	It specifies the date until which the standing order for reservation is valid.

The following table shows an exhaustive list of the message subscription reference data attributes in CLM.

Table 11 - Message subscription reference data attributes

Attribute	Description
Message subscription identifier	It specifies the unique technical identifier of the message subscription.
Message identifier	It specifies the identifier of the message subscribed to by the CLM participant.
Account	It specifies the account number of the main cash account for which the message has been subscribed.
Recipient	It specifies the identifier of the party subscribing to the message for the account.
Alternative recipient identifier	It specifies the identifier of the party nominated to receive the message either instead of or in addition to the recipient.
Additional copy	This flag indicates that the recipient still receives the message in addition to the nominated alternative recipient.
Business case	It specifies the business case for which a message has to be sent.
Subscription valid from	It specifies the date from which the subscription is valid.
Subscription valid to	It specifies the date until which the subscription is valid.

The following table shows an exhaustive list of the report configuration reference data attributes in CLM.

Table 12 - Report configuration reference data attributes

Attribute	Description
Report configuration identifier	It specifies the unique technical identifier of the report configuration.
Report identifier	It specifies the configured report for the account.
Account	It specifies the account number of the main cash account for which the report has been configured.
Recipient	It specifies the identifier of the party configuring the report for the account.
Parameters for report	It specifies the whether the relevant report is received in full or delta mode, and whether in push or pull mode.
Scheduled time	It specifies the scheduled time when the report is provided. Either scheduled time or scheduled event must be specified, but not both.
Scheduled event	It specifies the event that shall trigger the report to be produced. Either scheduled time or scheduled event must be specified, but not both.
Configuration valid from	It specifies the date from which the subscription is valid.
Configuration valid to	It specifies the date until which the subscription is valid.

3.3 Shared reference data (completed)

3.3.1 CLM calendar

The CLM calendar specifies the calendar days when CLM is open and follows the defined business day schedule. Different calendars per currency will be set up to operate different closing days.

3.3.2 CLM scheduled events

The CLM scheduled events specifies the scheduled events that will automatically trigger a specified process within CLM.

The following table shows the attributes of the CLM scheduled events.

Table 13 - Attributes of the CLM scheduled events

Attribute	Description
Scheduled event identifier	It specifies the unique technical identifier of a scheduled event.
Process identifier	It specifies the unique technical identifier of a business process.
Scheduled event status	It indicates whether the scheduled event has occurred and the business process has been initiated.
Event triggered timestamp	It specifies the system date and time at which the scheduled event occurred and the business process was triggered.
Repeat flag	It indicates whether another instance of the scheduled event should be created when this instance has occurred.
Trigger date	It specifies either the trigger date and trigger time or the trigger event identifier must be populated.
Trigger event identifier	It specifies the unique technical identifier of another scheduled event that shall trigger this scheduled event when it occurs.

3.3.3 CLM currency (completed)

The CLM currency specifies the available settlement currencies in CLM.

The following table shows the attributes of the CLM currency in CLM.

Table 14 - Attributes of the CLM currency

Attribute	Description
Currency code	It specifies the three-character ISO currency identifying the currency.
Currency name	It specifies the name of the currency.
Number of decimals	It specifies the number of decimals for the currency.

3.4 Interaction with CRDM (completed)

CRDM provides features that allow duly authorised users to set up, update, delete and query all reference data that are shared by multiple services/components (e.g. CLM or RTGS) for their processing activities.

The access to CRDM is possible in U2A mode (for all functions) and in A2A mode (for a subset of functions) via ESMIG.

In order to ensure a timely and consistent propagation of common reference data to the relevant components, CRDM implements a publish-subscribe feature allowing each component to receive all the common reference data (and their changes) they require for their processing.

In a nutshell:

- CRDM publishes all changes (in push mode) of common reference data (e.g. creations of new objects, updates of already existing objects).
- Other subscriber services/components get these changes too and apply them to their LRDM component, according to their needs.

Further detailed information can be found in chapter CRDM features.

As far as CLM is concerned, all reference data setup and maintenance operations are performed in CRDM while changes on local data are performed in CLM directly. The reference data are then propagated from CRDM to CLM asynchronously on a daily basis. However, the immediate update of specific reference data (e.g. blocking of main cash account) is done directly in CLM and is not propagated from CRDM.

Every CRDM opening day (T), an ad-hoc event triggers the propagation of all CLM reference data from CRDM to CLM. The event takes place at the end of day phase of CRDM business day, to ensure a smooth and complete reference data propagation before CLM receives the notification that a new business day is starting. The propagated reference data is then loaded into CLM during the start of day phase.

The set of reference data that CLM receives on business day T+1 includes all the active data of the mentioned business date T. If an item, propagated on date T, contains a validity-date in the future (e.g. T+2), CLM acquires it during the daily propagation but the item is available in CLM only when the validity date is reached.

The following diagram shows a conceptual overview of the interactions between CRDM and CLM.

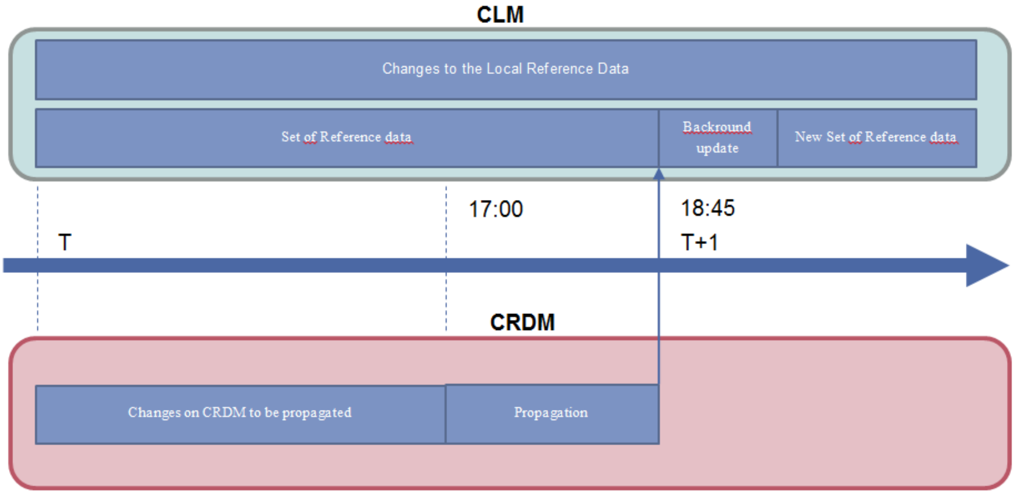

CLM
Changes to the Local Reference Data
Set of Reference data
Backround update
New Set of Reference data
T
17:00
18:45
T+1
CRDM
Changes on CRDM to be propagated
Propagation

Figure 2 - Interaction between CRDM and CLM

4 Business day (to be completed in Version 2.0)

5 Business and features description

5.1 Settlement of payments linked to central bank operations (completed)

5.1.1 Overview (completed)

In CLM the following central bank operations [1] are processed and settled on the main cash accounts of the CLM participant:

- update of credit line (cash side)
- marginal lending and overnight deposits (summarized as standing facilities)
- cash withdrawals
- monetary policy operations (e.g. open market operations like the main refinancing operation or the longer-term refinancing operations)
- debit of the invoiced amount
- interest payment orders linked to marginal lending, overnight deposits, minimum reserves and excess of reserves
- any other activity carried out by central banks in their capacity as central bank of issue

All central bank operations are settled with priority and are either fully executed or queued, i.e. payments linked to central bank operations are never settled partially.

Central bank operations can be initiated by the central bank in A2A or in U2A mode. The following payment types can be submitted:

- credit transfers or
- direct debits (e.g. used for the execution of cash withdrawals, repayment of monetary policy operations and collection of fees)
- connected payments
- warehoused payments

A payment order linked to a central bank operations leads to a debit (or credit) of the main cash account with the simultaneous credit (debit) of the central bank account/marginal lending account/overnight deposit account.

[1] Generally within this CLM UDFS the term central bank operation covers one of the here mentioned operations initiated by central banks.

With the exception of overnight deposits, which are initiated by a liquidity credit transfer (camt.050), a central bank can send the above mentioned central bank operations (depending on the underlying business case) as:

- a credit transfer (pacs.009) or
- a direct debit (pacs.010) to CLM (for further details please refer to chapter Flow of payments [▸ 53]).

Note: In case central bank sends a direct debit in general no direct debit mandate is needed. Only in case the central bank wants to have a direct debit booked on a main cash accounts of a participant not belonging to “its” own banking community a direct debit mandate is needed.

Beside that the central bank can send credit transfers and/or direct debits as **connected payments**. They are called “connected payments”, due to the link between payment (an immediate debit/credit of its main cash account) and a corresponding change of credit line. For further details please refer to chapter Connected payment [▸ 177].

Within the payment, central banks have the possibility to define the execution time (Definition of execution time [▸ 51]) It is possible to set

- an “earliest debit time indicator” (FROTIME) and
- a “latest debit time indicator” (REJTIME)

Furthermore, payments can be submitted as “**warehoused payments**” which means that the central bank operation is sent up to 10 calendar days in advance. In this case, the payment is warehoused until CLM opens for the settlement on the intended settlement day.

5.1.2 Definition of execution time (completed)

In general, the above mentioned central bank operations can be processed throughout the whole business day with the exception of the end of day processing and the maintenance window. Connected payments are processed up until the central bank general cut-off for the use of standing facilities (i.e. 18:40).

In addition, CLM participants have the possibility to determine the execution time of their payments. The following options are available:

- an “earliest debit time indicator”
- a “latest debit time indicator”

The following table describes payments with a set execution time.

Table 15 - Payments with set execution time indicators

	Earliest debit time indicator	Latest debit time indicator
Features	Payments to be executed from a certain time (message element: FROTIME)	Payments which should be executed up to a certain time and is rejected, if that is not the case (message element: REJTIME)
Effect	Payment is stored until the indicated time (with status earmarked). At the earliest debit time, the payment runs through the entry disposition.	REJTIME the payment is rejected, if it could not be executed until the latest debit time.
Processing	If the payment cannot be settled at the earliest debit time, it is queued until the cut-off time for payment type.The payment ordere can be revoked.	If the payment with the REJTIME indicator cannot be settled until the indicated debit time, the payment is rejected.

In case a payment with a “latest debit time indicator” is not executed 15 minutes prior to the defined time, an automatic notification in the GUI is triggered. The (optional) notification is directly displayed on top of all screens of the participant whose main cash account is debited.

It is possible to combine the “earliest debit time indicator” with the “latest debit time indicator”. The payment is meant to be executed during the indicated period.

The defined execution time of a payment can be changed if the payment is not executed yet. Effect of changing settlement time see chapter Amendment of payments [▷ 63] as well as chapter Comprehensive queue management [▷ 78].

Note: It is not possible to change the “earliest debit time indicator” of a payment which is queued due to the fact that the original “earliest debit time indicator” has been reached and it was already tried to settle this payment.

5.1.3 Warehouse functionality (completed)

Basics

It is possible to submit payments up to ten calendar days [2] in advance. In this case, the payment message is generated at the day of its submission and warehoused until CLM opens for the settlement of processing on the intended settlement day.

[2] The number of days are a parameter which is up to 10 calendar days but could be changed in future.

Note: In case a change in ISO20022 standards or formats is performed, warehoused payments with an execution time beyond this point in time cannot be stored. Such payments are rejected and a respective message (pacs.002) is sent to the sender of the payment.

Rules

The validation of warehoused payments is based on a three layer approach:

- ISO20022 format checks, meaning well-formedness check of ESMIG on the day of submission
- schema validations already on the day of submission
- business validations on the day of submission

On the intended settlement day, the warehoused payments undergo the same validations for a second time.

No checks are made in the time between submission day and value day.

Processing on the intended settlement day

On the value date with the start of the processing time of the respective order type (e.g. liquidity transfer, connected payment and direct debit) the warehoused payments are processed on top of the queue of incoming payments. They are immediately settled if enough liquidity is available (normal processing of payments in the entry disposition, see chapter Entry disposition [▸ 73]). Otherwise they are queued until the settlement attempt is successful (see chapter Dissolution of the payment queue [▸ 80]).

Exception: Warehoused payments with a set execution time indicator are treated according to the table Table 15 - Payments with set execution time indicators [▸ 52].

Information and control functions

Warehoused payments benefit from the same functionality via U2A or A2A:

- transparency about the status and other detailed information about the payment (order)
- revocation
- change of execution time (earliest and latest debit time indicator)

5.1.4 Flow of payments (partially completed)

5.1.4.1 Payments initiated by central bank - credit transfer (partially completed)

Only central banks can send a credit transfer linked to a central bank operation to a CLM participant that holds a main cash account.

The credit transfer is used in case of:

- provision of marginal lending
- payment of allocated open market operations or
- update of the credit line (cash side)
- payment of interests

Positive case of central bank operation credit transfer initiated by the central bank

In case the technical and business validation is passed successfully, the central bank operation is settled.

Message flow

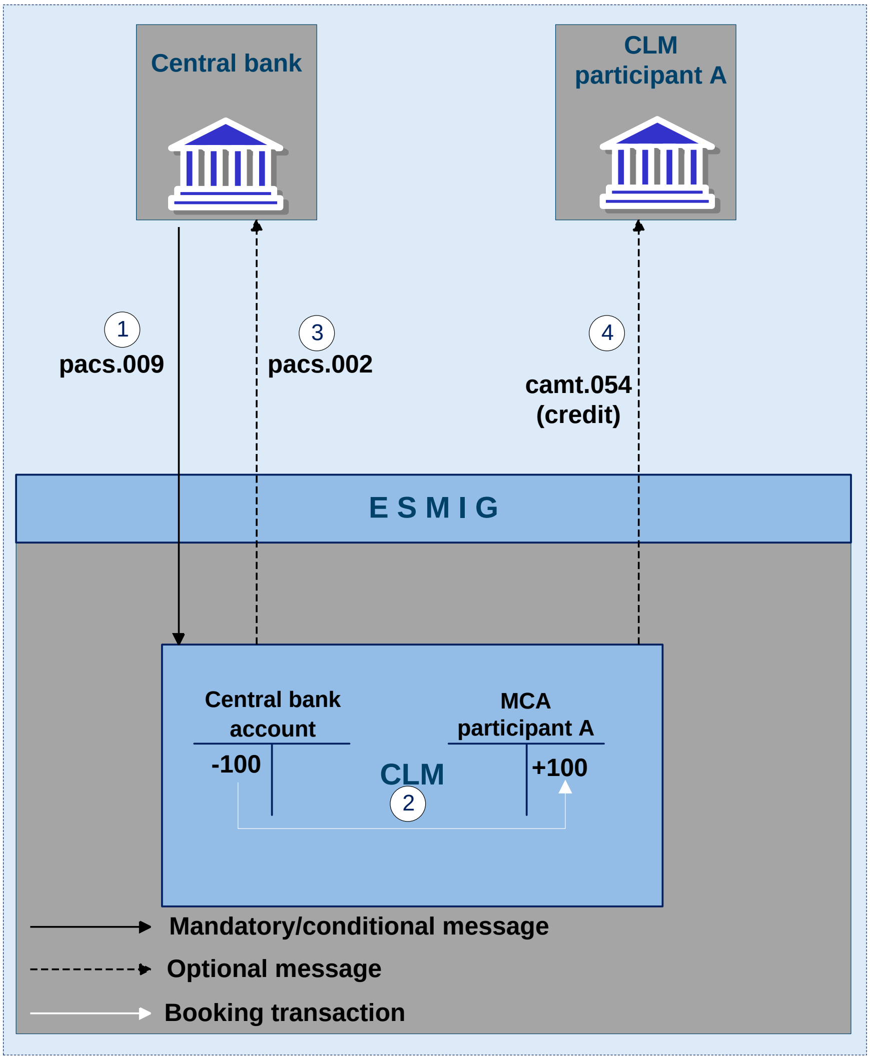


Figure 3 - pacs.009 centra bank operations

Process description

Table 16 - Central bank credit transfer (technical and business validations passed)

Step	Processing in/between	Description
1	Central bank via ESMIG to CLM	The central bank sends a pacs.009 via ESMIG to CLM
2	CLM	Booking takes place in CLM if CLM message check and validations are positive
3	CLM via ESMIG to central bank	CLM creates and forwards pacs.002 via ESMIG to central bank (optional)
4	CLM via ESMIG to CLM participant	CLM creates and forwards a camt.054 (credit) via ESMIG to CLM participant A (optional)

Used messages

- pacs.009 FI Credit Transfer (GEN and COV)
- BankToCustomerDebitCreditNotification (camt.054) [▸ 261] Bank to Customer Debit Credit Notification
- pacs.002 Payment Status Report

Technical validation failure

CLM performs the technical validations. For further details please refer to Rejection of payments [▸ 62].

CLM continues the technical validation even if a first error has been detected.

If the technical validation fails CLM rejects the central bank operation and provides all negative results in form of error codes in a single message (please refer to message flow described below).

In case the central bank instructed the central bank operation via U2A, the rejection notification is displayed directly on the screen. For further details please refer to the CLM user handbook.

Message flow

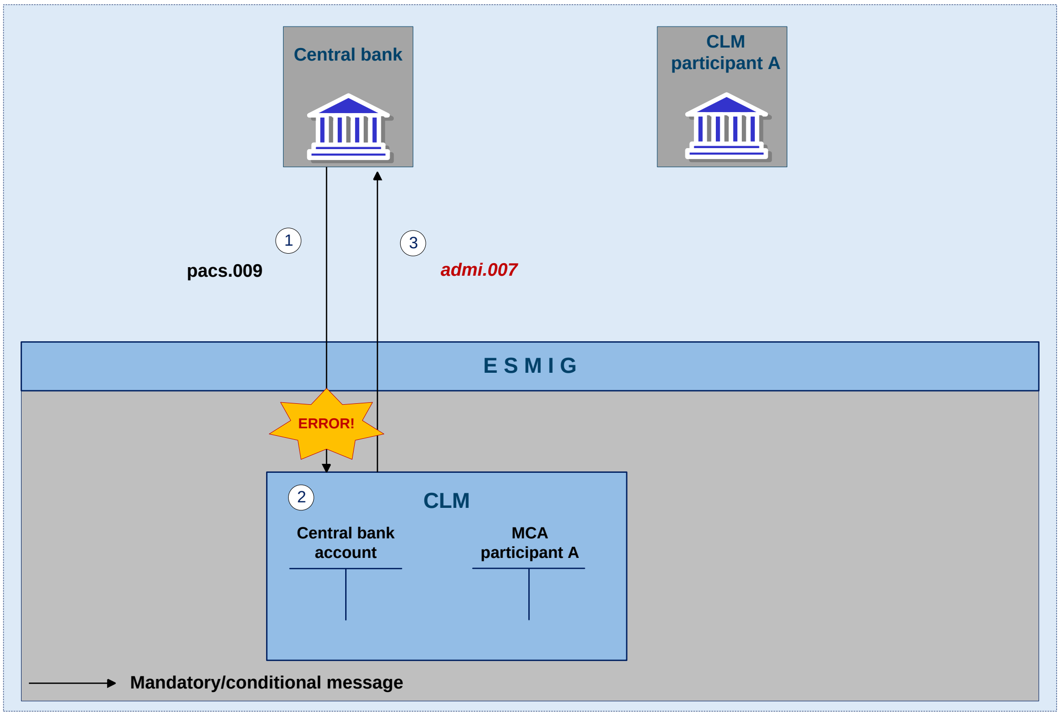


Figure 4 - pacs.009 central bank operations technical validation failed

Process description

Table 17 - Central bank credit transfer (technical validation failure)

Step	Processing in/between	Description
1	Central bank via ESMIG to CLM	The central bank sends a pacs.009 via ESMIG to CLM
2	CLM	CLM technical validation failed
3	CLM via ESMIG to central bank	CLM creates and forwards an admi.007 via ESMIG to the central bank

Used messages

- FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▸ 274]
- ReceiptAcknowledgement (admi.007) [▸ 224]

Business validation failure

CLM performs the business validations. For further details please refer to Rejection of payments [▸ 62].

If the business validation fails CLM rejects the central bank operation and provides the rejection notification to the central bank which submitted the central bank operation (please refer to message flow described below).

In case the central bank instructed the central bank operation via U2A, the rejection notification is displayed directly on the screen. For further details please refer to the CLM user handbook.

Message flow

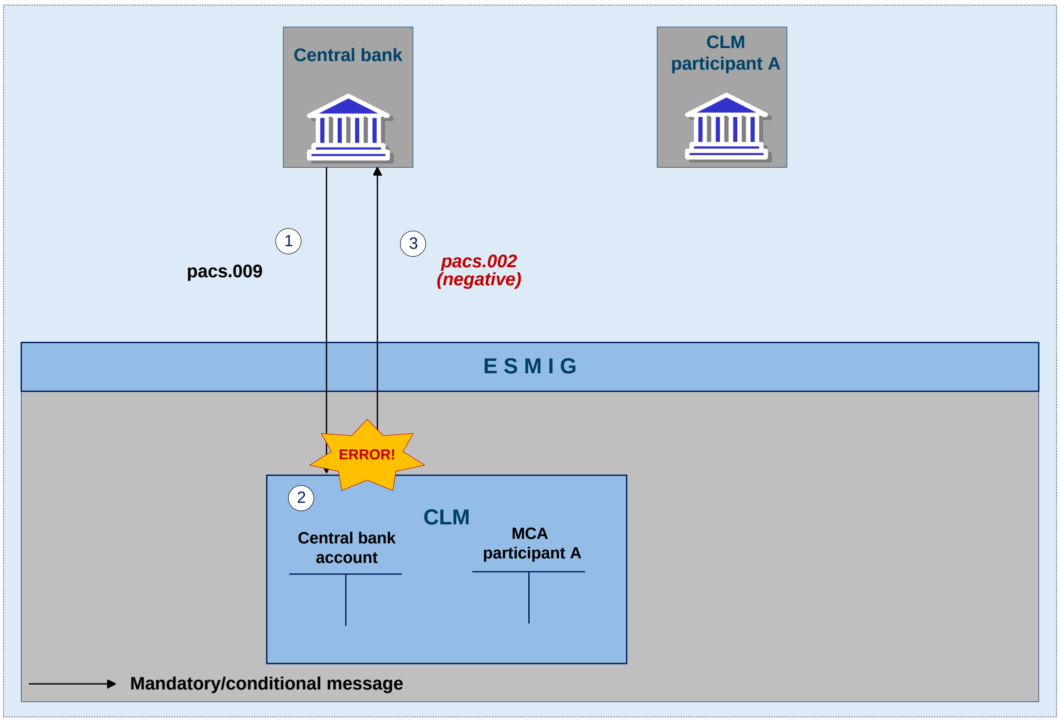


Figure 5 - pacs.009 central bank operations business validation failed

Process description

Table 18 - Central bank credit transfer (business validation failure)

Step	Processing in/between	Description
1	Central bank via ESMIG to CLM	The central bank sends a pacs.009 via ESMIG to CLM
2	CLM	CLM business validation failed
3	CLM via ESMIG to central bank	CLM creates and forwards a negative pacs.002 via ESMIG to central bank

Used messages

- pacs.009)
- pacs.002

5.1.4.2 Payments initiated by central bank - direct debit (partially completed)

Only a central bank can send a direct debit linked to a central bank operation to a CLM participant. Central banks are allowed to send direct debits within its market by default. No direct debit mandate is required in CRDM for central bank operations.

The direct debit is used in case of:

- marginal lending reimbursement
- reimbursement of open market operations
- update of credit line (cash side)
- debit of invoiced amounts
- cash withdrawals
- debit of interest

Positive case of central bank operation direct debit initiated by the central bank

In case the technical and business validation is passed successfully, the central bank operation is settled.

Message flow

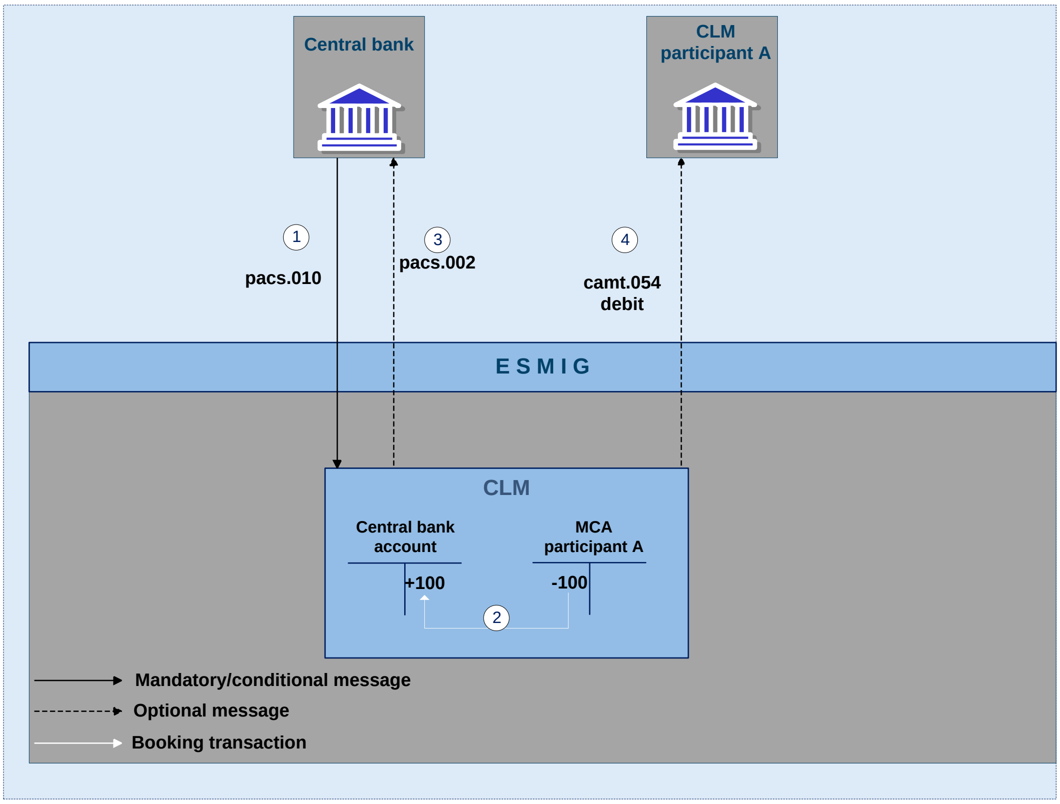


Figure 6 - pacs.010 central bank operations

Process description

Table 19 - Central bank direct debit (technical and business validations passed)

Step	Processing in/between	Description
1	Central bank via ESMIG to CLM	The central bank sends a pacs.010 via ESMIG to CLM
2	CLM	Booking takes place in CLM if CLM message check and validations are positive
3	CLM via ESMIG to central bank	CLM creates and forwards a pacs.002 via ESMIG to central bank (optional)
4	CLM via ESMIG to CLM participant	CLM creates and forwards a camt.054 (debit) via ESMIG to CLM participant A (optional)

Used messages

- FinancialInstitutionDirectDebit (pacs.010) [▸ 276]
- BankToCustomerDebitCreditNotification (camt.054) [▸ 261]
- PaymentStatusReport (pacs.002) [▸ 272]

CLM performs the technical validations. For further details please refer to Rejection of payments [▸ 62].

CLM continues the technical validation even if a first error has been detected.

If the technical validation fails CLM rejects the central bank operations and provides all negative results in form of error codes in a single message.

In case the central bank instructed the central bank operations via U2A, the rejection notification is displayed directly on the screen. For further details please refer to the CLM user handbook.

Message flow

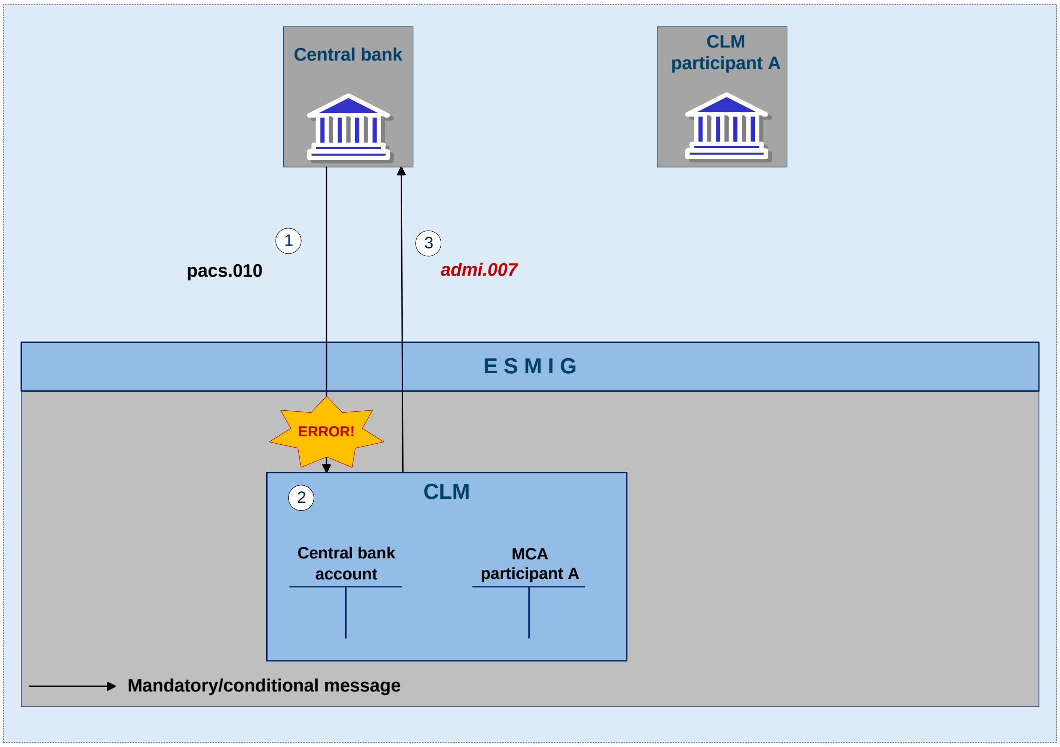


Figure 7 - pacs.010 central bank operations - technical validation failed

Process description

Table 20 - Central bank direct debit (technical validation failure)

Step	Processing in/between	Description
1	Central bank via ESMIG to CLM	The central bank sends a pacs.010 via ESMIG to CLM
2	CLM	CLM technical validation failed
3	CLM via ESMIG to central bank	CLM creates and forwards an admi.007 via ESMIG to central bank

Used messages

- FinancialInstitutionDirectDebit (pacs.010) [▸ 276]
- ReceiptAcknowledgement (admi.007) [▸ 224]

Business validation failure

CLM performs the business validations. For further details please refer to Rejection of payments [▸ 62].

If the business validation fails, CLM rejects the central bank operation and provides the rejection notification to the central bank which submitted the central bank operation.

In case the central bank instructed the central bank operation via U2A, the rejection notification is displayed directly on the screen. For further details please refer to the CLM user handbook.

Message flow

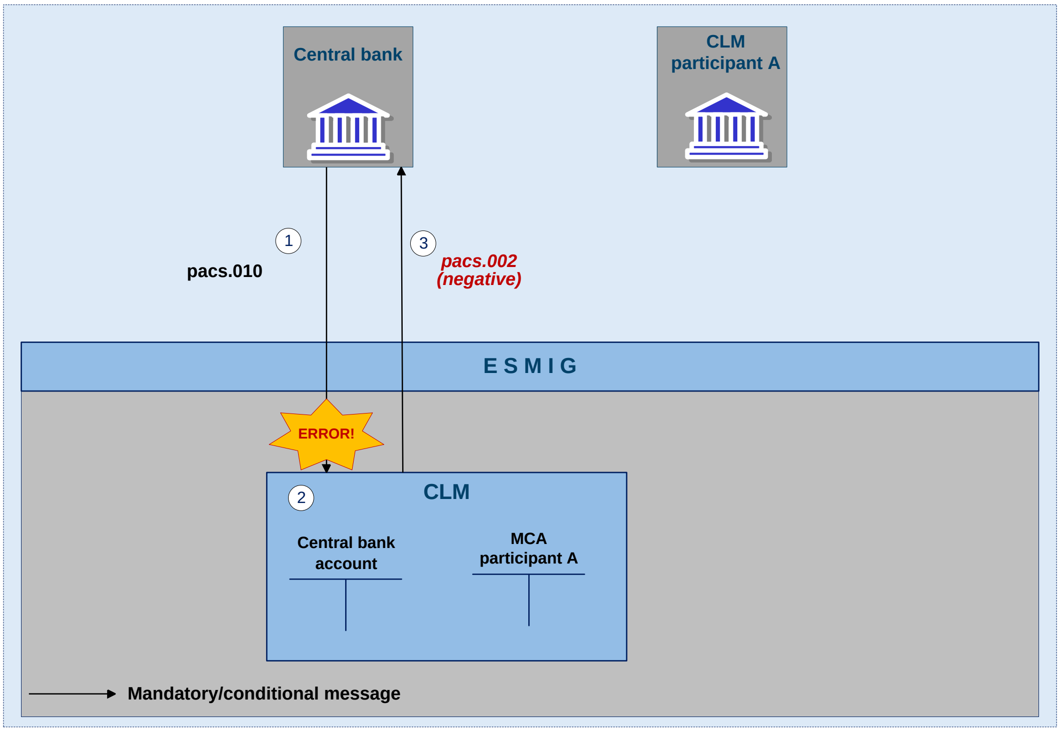


Figure 8 - pacs.010 central bank operations - validation failed

Process description

Table 21 - Central bank direct debit (business validation failure)

Step	Processing in/between	Description
1	Central bank via ESMIG to CLM	The central bank sends a pacs.010 via ESMIG to CLM
2	CLM	CLM business validation failed
3	CLM via ESMIG to central bank	CLM Creates and forwards a negative pacs.002 via ESMIG to central bank

Used messages

- FinancialInstitutionDirectDebit (pacs.010) [▷ 276]
- PaymentStatusReport (pacs.002) [▷ 272]

5.1.5 Rejection of payments (completed)

Payment orders sent to CLM have to pass several validations before the payment is effectively settled. Validations include technical checks and format checks (both technical validations) as well as checks for the correct content (business validations). A payment is rejected by CLM if either the technical validation or the business validation fails.

- Index of business rules and error codes [▸ 284]

Technical validation

The following technical validations are inter alia performed in CLM interface:

- Schema validation - syntax, format and structure of the message are compliant (e.g. check that all mandatory field in the message are populated)

In general CLM continues the technical validation even if a first error has been detected. In case the technical validation was not successful an admi.007 is sent to the instructing party (meaning the central bank) indicating which error occurred (all negative results in form of error codes are included).

In case the central bank instructed the central bank operation via U2A, the rejection notification is displayed directly on the screen. For further details please refer to the CLM user handbook.

Business validation

The validations described below are performed in one step in order to capture all the possible breaches; the checks therefore do not stop after the first breach occurring, as if there could be further breaches in the subsequent checks. If the validation failed, a rejection notification with appropriate reason codes for all breaches occurred is sent to the instructing party.

The following business validations are inter alia performed in CLM interface:

- check for duplicate submission for incoming central bank operations including the fields:
  - Sender of the Message
  - Message Type
  - Receiver
  - Transaction Reference Number
  - Related Reference
  - Value Date
  - Amount
- process specific access rights/authorisation checks:
  - Is the sender of the payment order the owner of the account to be debited or by another actor operating on its behalf?

  - In case of direct debit: is the sender of the payment order the owner of the account to be credited?
  - Is the central bank allowed to send central bank operations for the provided main cash acoounts?
  - In case a central bank acts on behalf of a credit institution: does the credit institution belong to the acting central bank?
- check on value date
  - If the value date is in the future (up to ten calendar days), it is treated as warehoused payment.
  - If the value date is the current business day, it is treated as like any other payment.
- payment type specific checks
- field and reference data checks:
  - Field value validation - codes are valid, domain values are within allowed range.

    (e.g. the main cash account and the central bank account mentioned in the central bank operation exist and are active for settlement in the relevant currency or the main cash account owner is not blocked at account or party level.)
  - Cross-field validation - all provided values are valid according to predefined values or cross-field validations.
  - database checks, e.g. existence of parties and accounts
- direct debit check
- check of back-up payments
- account checks

Error codes for possible rejections are listed in chapter Index of business rules and error codes [▷ 284].

If business validation fails CLM creates and forwards a pacs.002 (negative – payment status report) to the instructing party (meaning the central bank). The pacs.002 refers to the original transaction reference number and a set of elements from the original instruction. The pacs.002 message is a conditional message, i.e. it is mandatory in case of failed business validation.

5.1.6 Amendment of payments (completed)

As long as a central bank operation initiated in CLM is not settled (including warehoused payments), the central bank has the ability to change certain parameters of this payment.

The amendment of central bank operations is possible throughout the whole business day with the exception of the end-of-day processing and the maintenance window. Central banks can initiate an amendment in U2A mode only.

If the message content is valid (see chapter Rejection of payments [▷ 62]) CLM checks the status of the original central bank operation the amendment is referring to. The central bank operation to be amended has to be in an intermediate (i.e. not final) status to be eligible for amendment.

If the amendment operation succeeds, CLM modifies the original central bank operation according to the amendment request and send a success notification to the submitting central bank.

If the amendment operation fails, a reject notification with appropriate reason code is sent to the central bank.

Two different types of amendment are possible in CLM:

Table 22 - Possible amendment types in CLM

Parameter/action	Actor
Re-ordering within the respective queue (increase/decrease position)	Central bank
Change of set execution time (if defined before sending to CLM)	Central bank

These features enable a central bank to react on changed conditions and to react on behalf of the CLM participant in case of a changed liquidity situation during the day.

In principle, amendments can be provided to CLM in U2A. A description of the respective screen can be found in the CLM user handbook.

As a consequence of the amendment of central bank operations the dissolution of the payment queue process might be started. For further details please refer to chapter Dissolution of the payment queue [▸ 80].

Case: re-ordering the queued transactions

A central bank can change the queue position of central bank operations. The selected central bank operation can be placed:

- to the top of the queue
- to the end of the queue

The re-ordering can be done at any time during the business day. A detailed description of the process and the effect of the re-ordering can be found in chapter Comprehensive queue management [▸ 78].

Case: changing the execution time

Central bank operations can include a time that indicates when they should be settled, i.e. when the first settlement attempt is started (transactions with an “earliest debit time indicator”) and/or a time that indicates until when they should have been settled, i.e. after which no further settlement attempt takes place (transactions with a “latest debit time indicator”).

The execution time can be changed in CLM via U2A (i.e. the time may be advanced or postponed). The change has no impact on the payment processing, but on the queue management.

The change of the execution time can be done at any time during the business day. A detailed description of the process and the effect of the changed execution time can be found in chapter Comprehensive queue management [▸ 78].

Successful amendment

Message flow

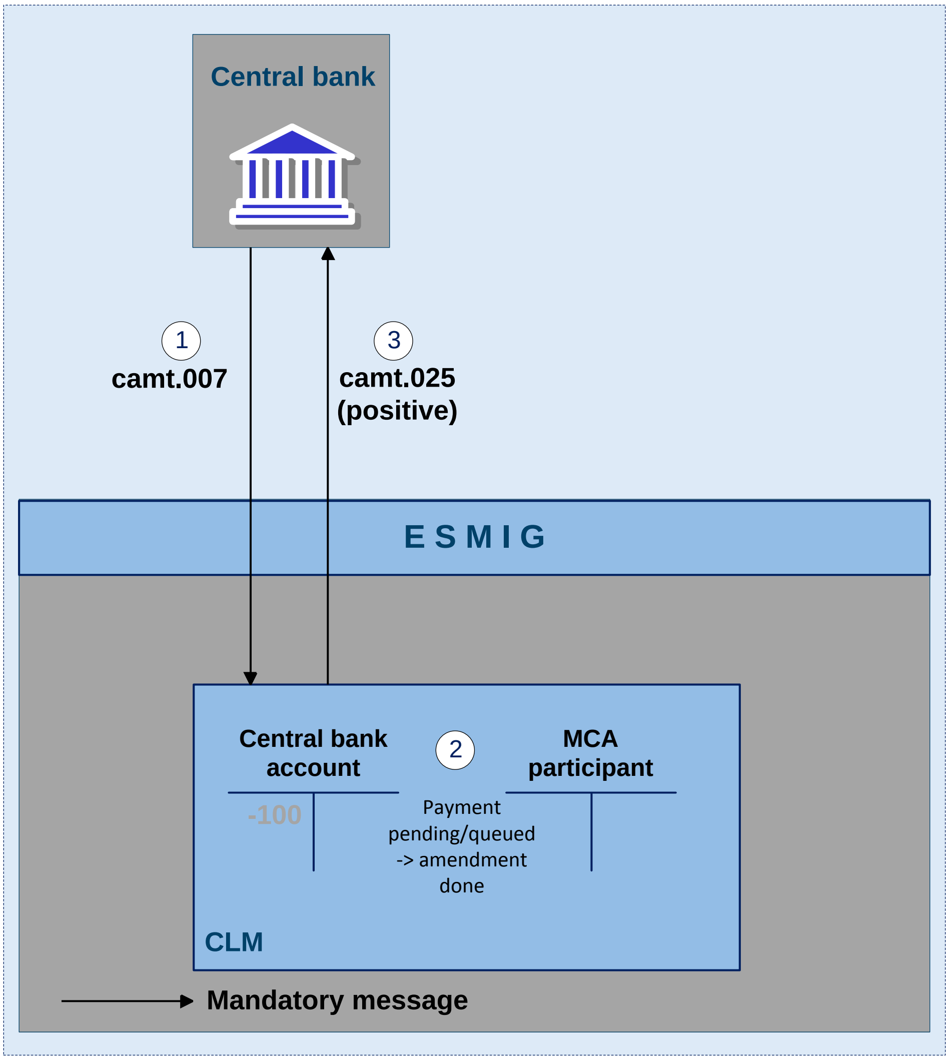


Figure 9 - Amend payment succeeded

Process description

Table 23 - Successful amendment of payment

Step	Processing in/between	Description
1	Central bank to CLM via ESMIG	The central bank sends a camt.007 via ESMIG to CLM.
2	CLM	CLM business validation passed. CLM processes the requested amendment.
3	CLM to central bank via ESMIG	CLM creates and forwards a positive camt.025 via ESMIG to the central bank.

Used messages

- ModifyTransaction (camt.007) [237]
- Receipt (camt.025) [243]

Failed amendment

Message flow

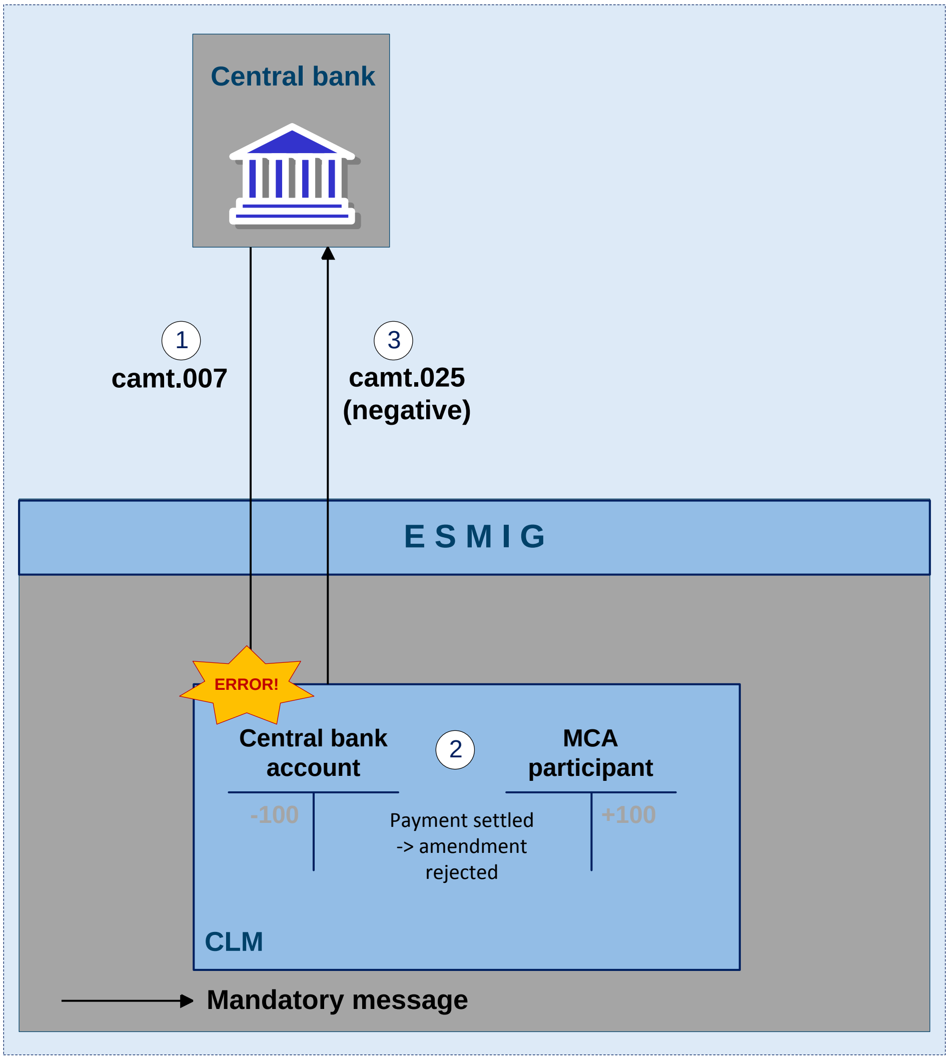


Figure 10 - Amend payment failed

Process description

Table 24 - Failed amendment of payment

Step	Processing in/between	Description
1	Central bank to CLM via ESMIG	The central bank sends a camt.007 via ESMIG to CLM.
2	CLM	CLM business validation failed.
3	CLM to central bank via ESMIG	CLM creates and forwards a negative camt.025 via ESMIG to the central bank.

Used messages

- ModifyTransaction (camt.007) [▸ 237]
- Receipt (camt.025) [▸ 243]

5.1.7 Revocation of payments (completed)

As long as a central bank operation is not settled (including warehoused payments), a central bank has the ability to revoke this payment.

To revoke a payment the following pre-conditions apply:

- a central bank operation has been initiated in CLM and
- the status of the payment is not final, i.e. the payment is in the CLM queue or it is warehoused

The revocation of central bank operations is possible throughout the whole business day with the exception of the end-of-day processing and the maintenance window. Standing facilities transactions (i.e. operations for marginal lending and overnight deposits) can additionally be revoked during the end-of-day processing, up until the cut-off time for standing facilities. Central banks can initiate a revocation in A2A as well as in U2A mode. A description of the individual U2A process can be found in the CLM user handbook.

A cancellation request can be sent to revoke central bank operations which were sent via pacs.009 or pacs.010. For each central bank operation submitted the central bank needs to send a dedicated cancellation request (FIToFIPaymentCancellationRequest (camt.056) [▸ 263]).

If the message content is valid (Rejection of payments [▸ 62]) CLM checks the status of the original central bank operation the revocation is referring to. The central bank operation to be revoked has to be in an intermediate (i.e. not final) status to be eligible for revocation. If the revocation operation succeeds, CLM cancels the original central bank operation and send a revoke success notification to the central bank as initiator. Where the revocation operation fails, a revocation reject notification with appropriate reason code is sent to the central bank (Index of business rules and error codes [▸ 284]).

Successful revocation

Message flow

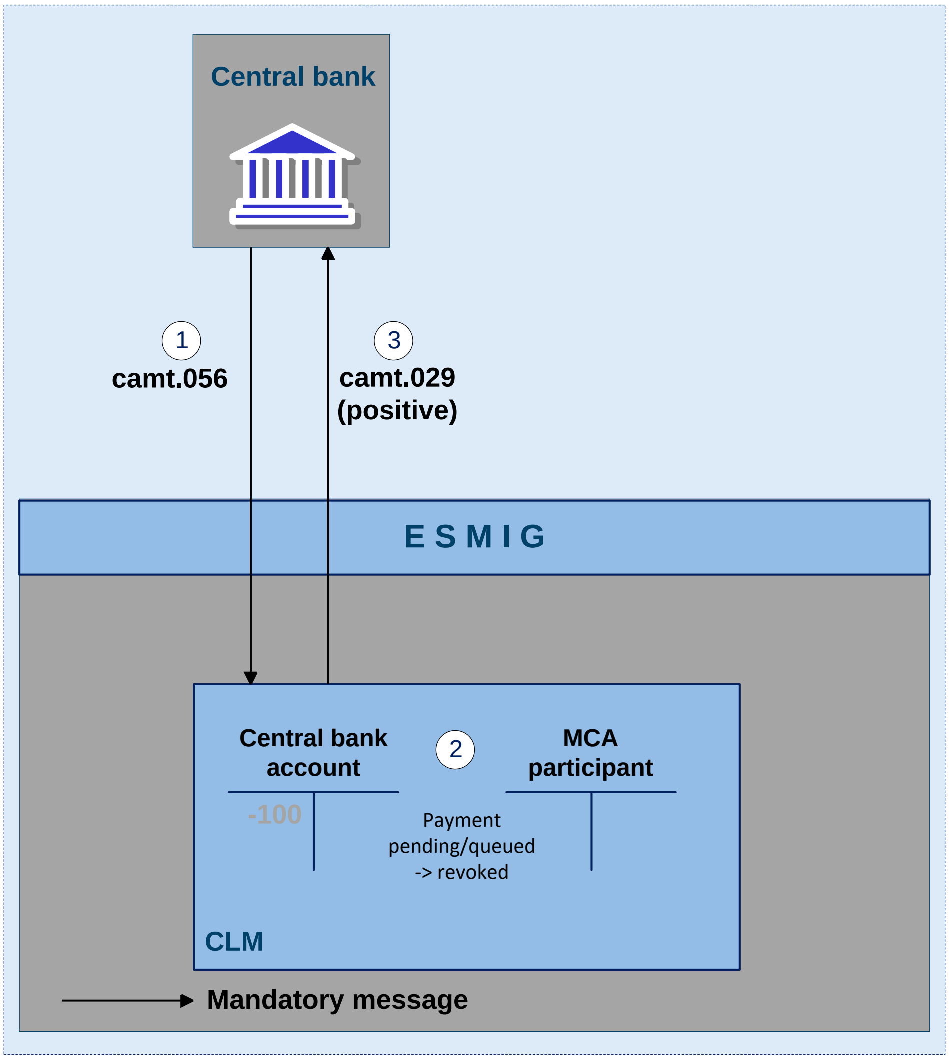


Figure 11 - Revoke payment via camt.056 – positive

Process description

Table 25 - Successful revocation of pending payment

Step	Processing in/between	Description
1	Central bank to CLM via ESMIG	Central bank sends a camt.056 (via ESMIG) to request revocation of payment to CLM.
2	CLM	CLM checks status of requested payment (not final); the payment is revoked and deleted from payment queue.
3	CLM to central bank via ESMIG	CLM send a positive camt.029 to confirm the revocation.

Used messages

- ResolutionOfInvestigation (camt.029) [▸ 245]
- FIToFIPaymentCancellationRequest (camt.056) [▸ 263]

Failed revocation

Message Flow

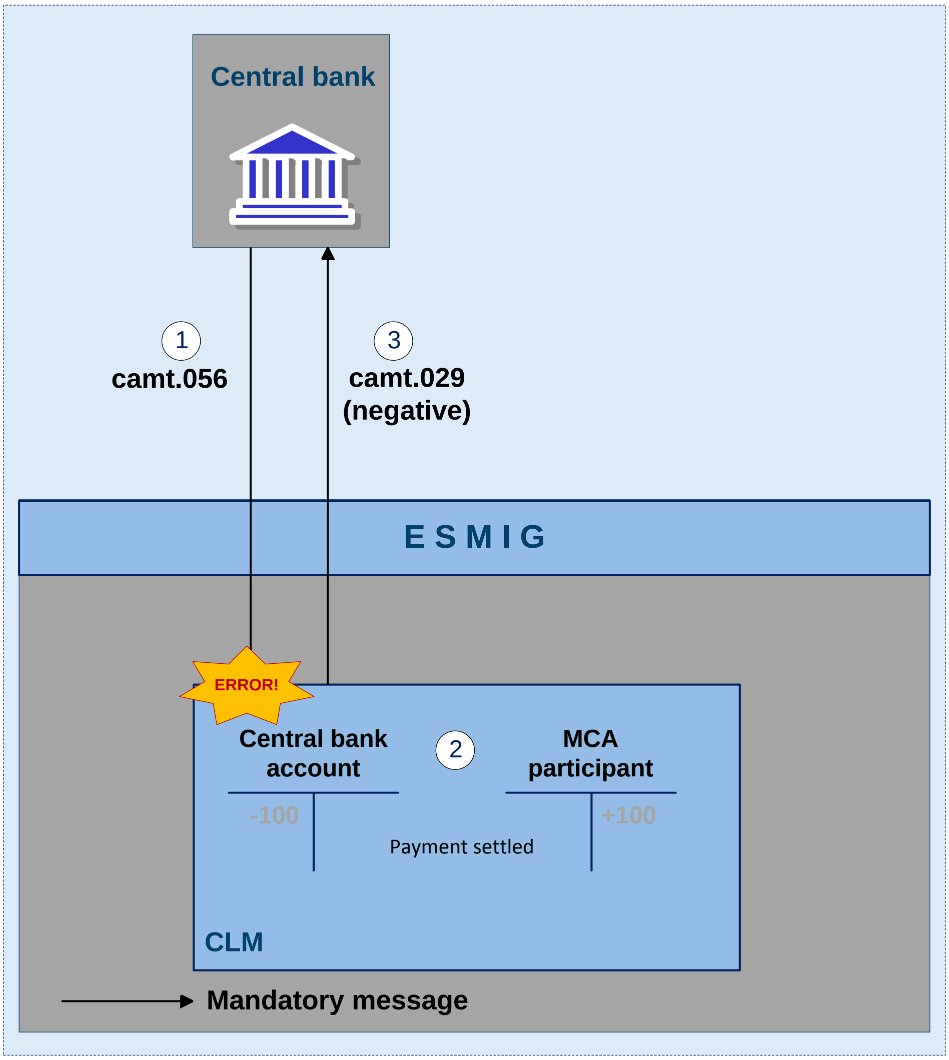


Figure 12 - Revoke payment via camt.056 - negative

Process description

Table 26 - Failed revocation of payment

Step	Processing in/between	Description
1	Central banks to CLM via ESMIG	Central bank sends a camt.056 (via ESMIG) to request revocation of payment to CLM.
2	CLM	CLM checks the status of requested payment (settled); revocation request not executed.
3	CLM to central banks via ESMIG	CLM send a negative camt.029 to notify a failed revocation.

Used messages

- ResolutionOfInvestigation (camt.029) [▸ 245]
- FIToFIPaymentCancellationRequest (camt.056) [▸ 263]

Technical validation

CLM performs the technical validations. For further details please refer to chapter Rejection of payments [▸ 62].

If the validation fails, a rejection notification with appropriate reason code is sent to the initiator of the revocation request (depending on the submission channel, an admi.007 in A2A mode or a broadcast on the screen in U2A mode).

Business validation

CLM performs the business validations. For further details please refer to chapter Rejection of payments [▸ 62].

5.1.8 Processing of payment orders (partially completed)

Basics

The efficient management of liquidity and the settlement of central bank operations in an optimised manner are of key importance. Therefore, offering a broad set of liquidity management features helps fulfilling the objectives of the CLM service.

These features may inter alia

- result in faster settlement, with a reduced amount of liquidity
- increase transparency for CLM participants
- contribute to achieve a higher degree of efficiency

- allow achieving a flexible and need-based control of payment flows.

The features are implemented in CLM on a flexible and optional basis to allow the CLM participant to meet its individual needs.

Objective for settlement

The aim of the processing in the CLM service is a fast settlement of central bank operations with the following characteristics:

- settlement in central bank money
- immediate, irrevocable booking of settled central bank operations

Moreover, it is the aim of the CLM processing to enable an efficient allocation of liquidity among the various services or components and its fast, immediate irrevocable settlement.

Influencing factors

The payment processing in the CLM service is influenced by the following factors

- balance on the main cash account
- credit line connected to the main cash account
- used reservations for central bank operations
- order of payment orders (including central bank operations and liquidity transfers) submitted
- set execution time

Basic principles

The following principles apply to the processing of central bank operations in CLM:

- All central bank operations have the same priority. They are not distinguished between urgent and normal payments.
- Attempt to settle immediately after their submission - with the exception of central bank operations with a set earliest debit time indicator “FROTIME”.

  In case a “FROTIME” is defined, these central bank operations are included in the settlement process only from that time indicated as earliest debit time.
- The central bank operation can include the latest debit indicator “REJTIME”.

  In case a “REJTIME” is defined, the central bank operations are excluded from the settlement process and are rejected at that time indicated as latest debit time.

- Warehoused payments can be initiated by default ten calendar days in advance [3]. The payment message passes the schema validation and the business validation of CLM and is warehoused until the start of day of CLM of that date.
- Offsetting mechanisms are not necessary in CLM and are not used.
- For central bank operations a defined amount of liquidity can be reserved in advance to separate it from the “non-reserved” part of the main cash account used inter alia for liquidity transfers.
- Central bank operations that are not yet executed can be revoked.
- Central bank operations that cannot settle immediately are queued. The orders within the queue is then processed following the FIFO-principle. CLM participants can intervene on queued central bank operations by
  – changing the set execution time

    Note: This is only possible in case an execution time has been set in the original payment order.

  – re-ordering of queued central bank operations
  – revoking the queued central bank operations
- CLM continuously attempts to settle the central bank operations in the queue.
- The entry disposition and the optimization procedures for queues can run at the same time.

5.1.8.1 Entry disposition (partially completed)

General remarks

In CLM the available liquidity of the main cash account can be divided into a non-reserved part and into a part reserved for central bank operations (see chapter Available liquidity [81]).

Central bank operations use the available liquidity in the dedicated reserved part of the main cash account first. Only in case this reserved part does not include any (or not enough) liquidity, the liquidity on the non-reserved part of the available liquidity on the main cash account is used in a second step. Moreover, the FIFO-principle applies among all central bank operations.

Liquidity transfers use only the liquidity in the non-reserved part of the main cash account. Liquidity transfers are only settled immediately. Therefore, no FIFO-principle is needed. Standing liquidity transfer orders are treated like immediate liquidity transfer as soon as the triggering event occurred. The only difference is that standing liquidity transfer orders could also settle partially in case of insufficient liquidity in the non-reserved part of the main cash account.

Offsetting mechanisms are not required in CLM. They are neither used for central bank operations nor for liquidity transfers.

[3] The number of days are defined as a parameter that indicates the number of days payments can be submitted to CLM in advance.

Unsuccessful entry disposition

If the submitted central bank operation cannot settle in the entry disposition, it is placed into the queue of central bank operations according to the FIFO-principle.

Note: In general, liquidity transfers are not placed into a queue and are rejected with appropriate error code in case of insufficient liquidity.

Settlement of payments in the entry disposition

Generally central bank operations have the highest priority. Therefore, CLM checks first which kind of payment order the CLM participant has submitted, whether it is a payment (meaning a central bank operation) or a liquidity transfer.

Central bank operations:

- First, the liquidity on the reserved part for central bank operations of the available liquidity on the main cash account is checked.
- In case of sufficient liquidity on the reserved part, the central bank operation is settled.
- In case of insufficient liquidity, the liquidity on the non-reserved part of the available liquidity on the main cash account is checked.
  - If there is overall sufficient liquidity, the central bank operation is settled.
  - If there is not sufficient liquidity, the central bank operation is queued. In case of queued central bank operations, CLM creates and sends an automated inter-service liquidity transfer to pull the missing liquidity from the linked RTGS dedicated cash account.
- When the payment is submitted to CLM and in case there are already other central bank operations queued due to insufficient available liquidity on the main cash account, the submitted payment is queued as well and it is put at the end of the queue following the FIFO-principle. When putting a new payment at the end of the queue CLM again creates and sends a new automated inter-service liquidity transfer to the RTGS component to pull liquidity from the linked RTGS dedicated cash account. The amount within this new automated inter-service liquidity transfer is the sum of all pending central bank operations that are currently in the queue minus the available liquidity (that is still not sufficient to settle the first central bank operation in the queue).

Note: As soon as a new automated inter-service liquidity transfer arrives in the RTGS, the RTGS component deletes the previous automated inter-service liquidity transfer and consider only the current one with the sum of all queued central bank operations.

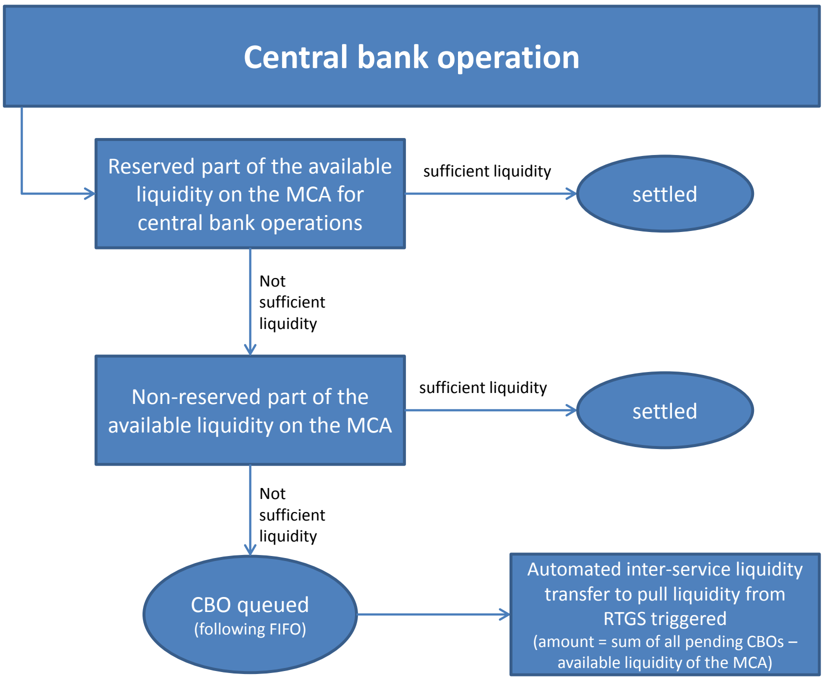

Central bank operation
Reserved part of the available liquidity on the MCA for central bank operations
sufficient liquidity
settled
Not sufficient liquidity
Non-reserved part of the available liquidity on the MCA
sufficient liquidity
settled
Not sufficient liquidity
CBO queued
(following FIFO)
Automated inter-service liquidity transfer to pull liquidity from RTGS triggered
(amount = sum of all pending CBOs – available liquidity of the MCA)

Figure 13 - Entry disposition of central bank operations

Liquidity transfers:

For liquidity transfers only the non-reserved part of the available liquidity on the main cash account can be used for the settlement. In case the liquidity is sufficient and there are no pending central bank operations queued, the liquidity transfer is immediately settled. In case the liquidity on the non-reserved part of the available liquidity on the main cash account is not sufficient the behaviour of CLM depends on the way of initiation of the liquidity transfer:

- *Immediate liquidity transfers:* In case the liquidity on the non-reserved part of the main cash account is not sufficient, the immediate liquidity transfer is rejected and a camt.025 receipt is sent to the CLM participant who submitted the original liquidity transfer.
- *Standing liquidity transfer order:* In case the liquidity on the non-reserved part of the main cash account is not sufficient and in case there are no pending central bank operations in the queue, the standing liquidity transfer order is partially settled up to the amount that is available. For the remaining amount that could not settle in the first settlement attempt no further attempt takes place.

Note: In case there is no liquidity at all available in the non-reserved part of the main cash account, the partial settlement takes place with the amount of zero. The CLM participant is informed accordingly via a camt.054 BankToCustomerDebitCreditNotification.

- *Event-based liquidity transfer orders e.g. stemming from floor/ceiling functionality:* The behaviour is analogue to the standing liquidity transfer orders, that means, in case the liquidity on the non-reserved part of the main cash account is not sufficient and there are no pending central bank operations in the queue, the event-based liquidity transfer orders, e.g. stemming from floor/ceiling functionality is partially settled up to the amount that is available. For the remaining amount that could not settle in the first settlement attempt no further attempt takes place.

Note: In case there is no liquidity at all available in the non-reserved part of the main cash account, the partial settlement takes place with the amount of zero. The CLM participant is informed accordingly via a camt.054 BankToCustomerDebitCreditNotification.

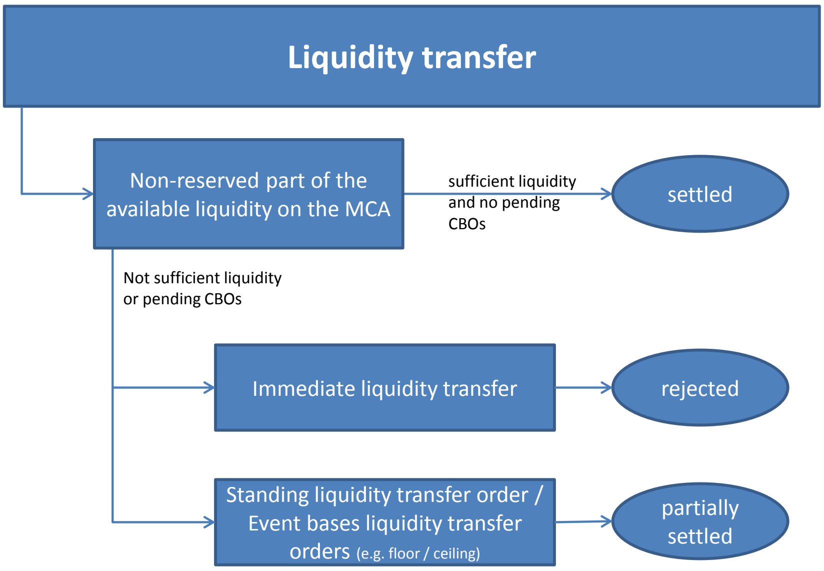

Liquidity transfer
Non-reserved part of the available liquidity on the MCA
sufficient liquidity and no pending CBOs
settled
Not sufficient liquidity or pending CBOs
Immediate liquidity transfer
rejected
Standing liquidity transfer order / Event bases liquidity transfer orders (e.g. floor / ceiling)
partially settled

Figure 14 - Entry disposition of liquidity transfers

Table 27 - Example - entry disposition of liquidity transfers

Action	Reserved part of the main cash account for central bank operations	Non-reserved part of the main cash account	Queued central bank operations	Automated inter-service liquidity transfer pending in RTGS	Remarks
Starting situation	100	50	0		
First central bank operation - amount: debiting 50	50 ⇩	50	0		
Second central bank operation – amount: debiting 500	50	50	500 ⇧	400 ⇧	

Action	Reserved part of the main cash account for central bank operations	Non-reserved part of the main cash account	Queued central bank operations	Automated inter-service liquidity transfer pending in RTGS	Remarks
Inter-service liquidity transfer from T2S – amount: crediting 10	50	60 ⇧	500	400	
Third central bank operation – amount: debiting 150	50	60	650 ⇧	540 ⇧	
Intra-service liquidity transfer – amount: debiting 30	50	60	650	540	Rejected due to queued central bank operations
Automated inter-service liquidity transfer from RTGS – amount: crediting 300	50	360 ⇧	650	540, whereas only 240 remain pending	
Automated inter-service liquidity transfer from RTGS – amount: crediting 240	0 ⇩	0 ⇩	0 ⇩	0 ⇩	

Rejection during end-of-day processing

If queued central bank operations cannot be settled until the end-of-day and are still queued due to lack of liquidity, these payments are rejected during end-of-day processing.

5.1.8.2 Comprehensive queue management (partially completed)

If a submitted central bank operation cannot be settled in the entry disposition, it is placed in the queue.

As long as central bank operations are not settled, the central bank of the CLM participant has the ability to change parameters of the payment.

Three different control options are offered:

- changing the set of execution time (if already defined in the central bank operation before sending it to CLM)
- re-ordering the queued payments

- revocation of a queued payment

These control options enable the central bank to react on changed conditions and to react on behalf of the CLM participant in case of a changed liquidity situation during the day. It is possible to modify a single central bank operation or several central bank operations at the same time. In case it is not possible to execute a modification request, the central bank is informed accordingly. Further details on the interventions done in U2A can be found in the CLM user handbook.

In case of successful interventions, the process to resolve the queue in CLM is started.

Changing the set of execution time

In principle, central bank operations can be submitted with a defined execution time. It is possible to include an earliest debit time indicator and/or a latest debit time indicator (see chapter Definition of execution time [▸ 51]).

In case a submitted central bank operation includes an earliest debit time indicator and/or a latest debit time indicator it is possible to change the earliest debit time indicator and/or the latest debit time indicator via U2A as long as the time is not reached. Such a change has no impact on the processing of the central bank operation, but on the queue management as the time indication only supports the queue management.

Table 28 - Effect of changing the execution time

Action	Effect
Deleting the earliest debit time indicator of a central bank operation (FROTIME)	This central bank operation is not in the queue yet, as the earliest debit time indicator is not reached so far. With the deletion, the entry disposition is done by CLM and a first settlement attempt takes place. As a result the central bank operation is either settled or put at the end of the queue.
Changing the earliest debit time indicator of a central bank operation (FROTIME)	The central bank operation is included into the settlement process from the new indicated time.

Note: Since the deletion or modification of the latest debit time indicator has no direct effect on the queue management it has not been considered in the table.

Re-ordering the queued payments

The central bank can change the queue position for a single or a sequence of central bank operations via U2A. The central bank operation selected can be placed:

- to the top of the queue of central bank operations
- to the end of the queue of central bank operations

Table 29 - Effect of changing the order of queued central bank operations

Action	Effect
Moving a central bank operation to the top of the queue	Immediate check whether the new central bank operation on top (and possibly any further in the queue) can be executed
Moving a central bank operation from the top to the end of the queue	
Moving a central bank operation that is not on top to the end of the queue	The action is taken into account during the next settlement process – no immediate attempt to settle.

In case of such a change, the central bank operation:

- keeps its original submission time
- is placed in the queue according to the change

Revocation of a queued payment

A central bank can revoke central bank operations that are queued and not yet successfully settled. The revocation can be done via U2A and A2A at any time during the day. The queue of central bank operations is reduced by the revoked payment.

For further details please refer to chapter Revocation of payments [▸ 68].

5.1.8.3 Dissolution of the payment queue (partially completed)

The queue is resolved in an event-oriented way starting with the central bank operation on top.

Table 30 - Origin of possible events

Events	By …
Liquidity increase	- incoming settled central bank operations (i.e. credits) - incoming settled intra-service liquidity transfers - incoming inter-service liquidity transfers from other services/components
Intervention on queue level	- If the central bank operation on top of the queue is changed – change of order – revocation – rejection of the central bank operation due to the fact that the latest debit time is reached

As soon as one of the above mentioned events occur, further settlement attempts take place to settle the central bank operations starting with the one on top of the queue. There are no additional algorithms as it is the case for the RTGS component.

The resolving queue process and the entry disposition are handled in the same way. If a single central bank operation cannot be settled, it remains in the queue (at maximum until the end of the business day).

5.2 Liquidity management (partially completed)

This chapter describes the tools and processes for an efficient management and usage of liquidity across the TARGET services in a harmonised and generic way. It covers the different kinds of liquidity transfers, liquidity reservations, floor/ceiling management as well as the standing facilities.

5.2.1 Available liquidity (partially completed)

The main cash account is used for the settlement of:

- liquidity transfers - to ensure an efficient liquidity provisioning for the settlement in T2S, RTGS and TIPS
- central bank operations

The main cash account may either have a zero or a positive balance.

In principle the available liquidity of a main cash account consists of:

- the positive balance on the main cash account
- the credit line linked to the main cash account

  Note: In case a CLM participant holds more than one main cash account, the credit line can only be linked to one main cash account. Only this main cash account (with the linked credit line) can be used for the central bank operations of that CLM participant.

It is up to the CLM participant to decide whether the available liquidity should be divided into:

- the reserved part for central bank operations and
- the non-reserved part

This would be done by using the reservation function.

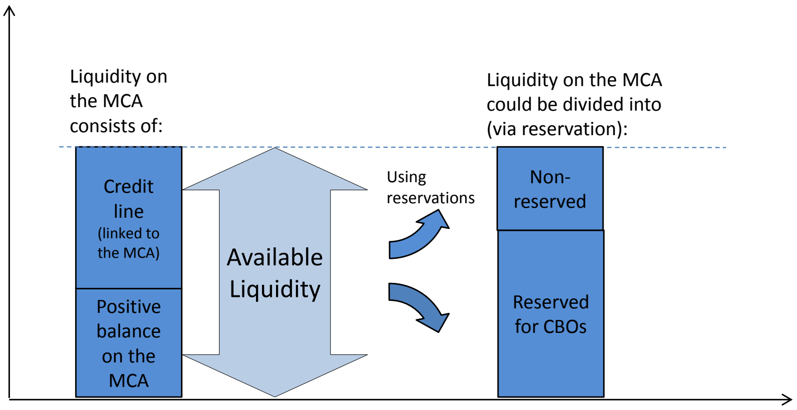

Liquidity on the MCA consists of:
Credit line (linked to the MCA)
Positive balance on the MCA
Available Liquidity
Using reservations
Liquidity on the MCA could be divided into (via reservation):
Non-reserved
Reserved for CBOs

Figure 15 - Illustration of the available liquidity

Without using the reservation function (please also refer to chapter Liquidity reservation [▸ 94] the main cash account just consists out of the available liquidity that is used for central bank operations and liquidity transfers in the same way.

With using the reservation function the liquidity in the reserved part for central bank operations could not be used for liquidity transfers. The reserved liquidity is only available for the settlement of central bank operations.

5.2.2 Liquidity transfer (completed)

5.2.2.1 Overview (completed)

The main cash account is the central source of liquidity for the different settlement services the CLM participant joined in. Therefore CLM has to ensure the efficient liquidity provision by liquidity transfers within CLM, to dedicated cash accounts of other services or components. Furthermore CLM optimises the efficient usage of liquidity for the different services/components and transfers liquidity between them.

Liquidity transfers are not classified as payments (i.e. pacs); they are cash management instructions using camt messages. The liquidity transfer order message (camt.050) is exchanged between users and the system in order to instruct the transfer of cash from one cash account to another cash account.

Liquidity can be transferred:

- between different main cash accounts within the CLM (under certain preconditions – for further details see chapter Liquidity transfer between two main cash accounts [▸ 89].
- between the main cash accounts and the dedicated cash accounts of the different settlement services/components
- between dedicated cash accounts within the same settlement service/component (out of scope of this UDFS)
- between dedicated cash accounts of different settlement services/components (via CLM transit account)

The following types of liquidity transfers exist:

- immediate liquidity transfer order
- automatically triggered inter service liquidity transfer order - if there are not sufficient funds for central bank operations
- liquidity transfer orders triggered by floor or ceiling amount
- standing liquidity transfer order

In general, liquidity transfers are never queued. They are either immediately settled (full or partially) or rejected. Only under following conditions automatically generated liquidity transfers can become pending.

- The main cash account has insufficient liquidity for a central bank operation and there is not sufficient liquidity on the RTGS dedicated cash account for an automatically triggered liquidity transfer to the main cash account.
- Any incoming liquidity (up to the required amount) on the RTGS dedicated cash account is then transferred stepwise (partially) to the main cash account until the pending central bank operation can be settled.
- The pending automated inter-service liquidity transfer from CLM is set on the top of the payment queue in RTGS.

For the transfer of liquidity the following rules apply:

Within CLM, liquidity can be transferred between main cash accounts belonging to the same party or liquidity transfer group. Liquidity transfer groups are configured by the respective central bank. For further details please refer to chapter Account monitoring groups [▸ 38].

The rules for liquidity transfer groups do not apply for central banks. That means a liquidity transfer within CLM is always possible as soon as a central bank account is involved.

5.2.2.2 Initiation of liquidity transfers (completed)

A liquidity transfer can be submitted via U2A or A2A (camt.050) to the CLM by

- a participant in CLM
- another actor on behalf of the participant
- central bank

A liquidity transfer can be initiated as

- immediate liquidity transfer order - the amount is transferred after initiation immediately
- automatically triggered inter-service liquidity transfer order - if there is not sufficient available liquidity for the settlement central bank operations
- liquidity transfer orders triggered by floor or ceiling amount
- standing liquidity transfer orders - the amount is transferred regularly at predefined event

A partial execution of a liquidity transfer takes place for standing orders and liquidity transfer orders triggered by a floor amount breach. For several standing orders, where the sum of all standing orders of the participant to be settled at the same event is larger than the available liquidity, CLM reduces all respective standing orders in a pro-rata mode.

The characteristics of different kind of liquidity transfers are summarised in the following table.

Table 31 - Underlying liquidity transfer characteristics

	Creation “on behalf”	Execution	Partial execution	Frequency of execution
Immediate liquidity transfer	No	Immediate	No	Once
	Yes	Immediate	No	Once

	Creation “on behalf”	Execution	Partial execution	Frequency of execution
Automatically triggered liquidity transfer order regarding missing available liquidity for central bank operations	Automatically triggered by CLM	Automatically triggered by event, then immediate	Not in CLM (in RTGS where the liquidity has to be pulled, partial execution might occur)	When the event takes place
Triggered by floor/ceiling amount	Triggered by CLM, if a target amount is created by the participant or another party on its behalf	Automatically triggered by event, then immediate	No, for liquidity transfers triggered by ceiling amount. Partially execution in RTGS for liquidity transfers triggered by floor amount.	When the event takes place
Standing order	Yes	Triggered by event, then immediate	Yes (after it is generated as an immediate liquidity transfer)	On a regular basis

The liquidity provisioning for the settlement of all payment types in the main cash account shall be processed following the FIFO principle. For further details refer to chapter Processing of payment orders [▸ 71].

Detailed information regarding the initiation of liquidity transfers in U2A mode can be found in the CLM user handbook.

5.2.2.3 Liquidity transfer process (completed)

In the following process descriptions successful transfers are described. The unsuccessful processes are described in chapter 5.2.2.3.5

5.2.2.3.1 Liquidity transfer from main cash account to dedicated cash account (completed)

A CLM participant can transfer liquidity from his main cash account to any dedicated cash account within another settlement service/component (T2S, RTGS or TIPS).

Message flow

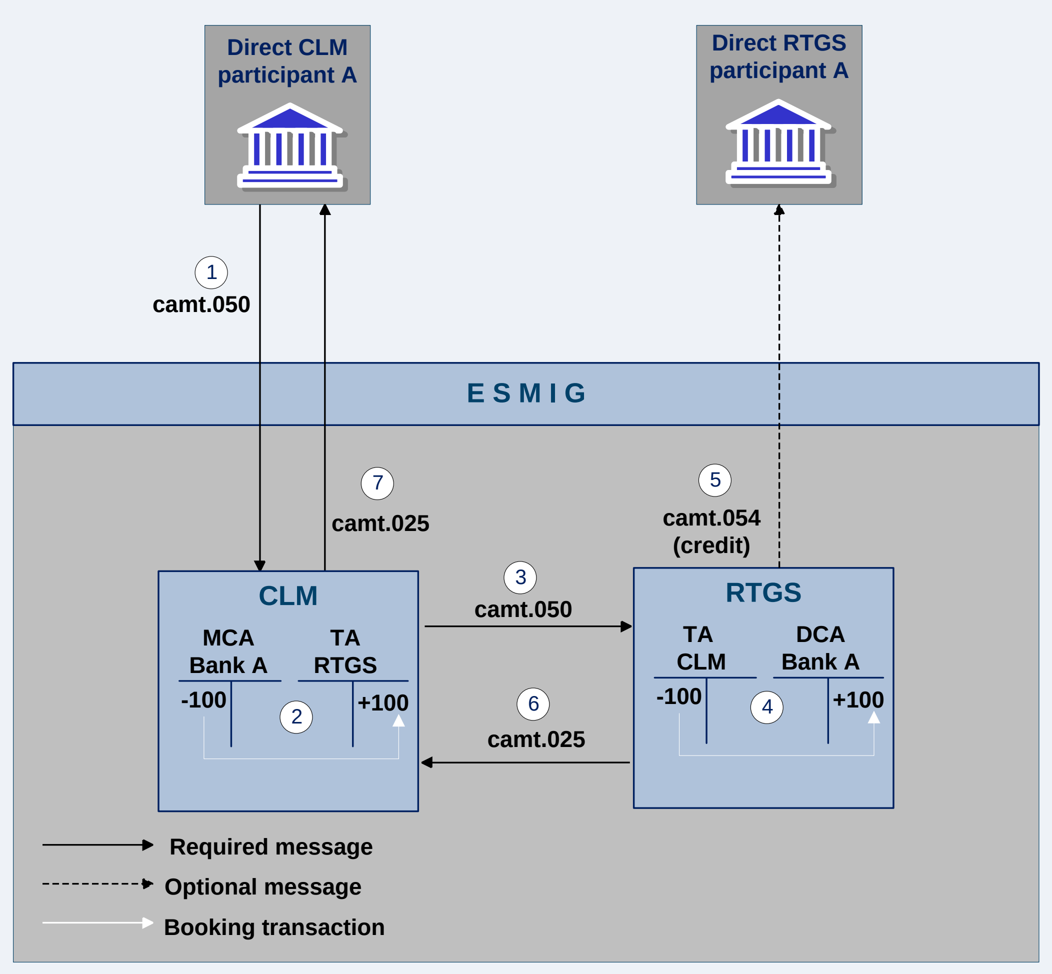


Figure 16 - camt.050 liquidity transfer from main cash account to RTGS dedicated cash account

Process description

Table 32 - Liquidity transfer from main cash account to RTGS dedicated cash account

Step	Processing in/between	Description
1	CLM participant via ESMIG to CLM	A camt.050 is sent from a direct CLM participant via ESMIG to CLM.
2	CLM	Booking on main cash accounts (main cash account to transit account RTGS)
3	CLM to RTGS	A camt.050 is forwarded to RTGS.
4	RTGS	Booking on RTGS dedicated cash accounts (transit account-CLM to RTGS dedicated cash account)

Step	Processing in/between	Description
5	RTGS via ESMIG to direct RTGS participant	A camt.054 (credit) is sent by RTGS via ESMIG to the direct RTGS participant (optional).
6	RTGS to CLM	For execution a camt.025 generated in RTGS is sent to CLM.
7	CLM via ESMIG to CLM participant	A camt.025 is sent by CLM via ESMIG to the CLM participant.

Used messages

- LiquidityCreditTransfer (camt.050) [▸ 255]
- BankToCustomerDebitCreditNotification (camt.054) [▸ 261]
- Receipt (camt.025) [▸ 243]

5.2.2.3.2 Liquidity transfer from dedicated cash account to main cash account (completed)

A settlement service/component participant can transfer liquidity from his dedicated cash account within a settlement service/component (T2S, RTGS or TIPS) to any main cash account.

Message flow

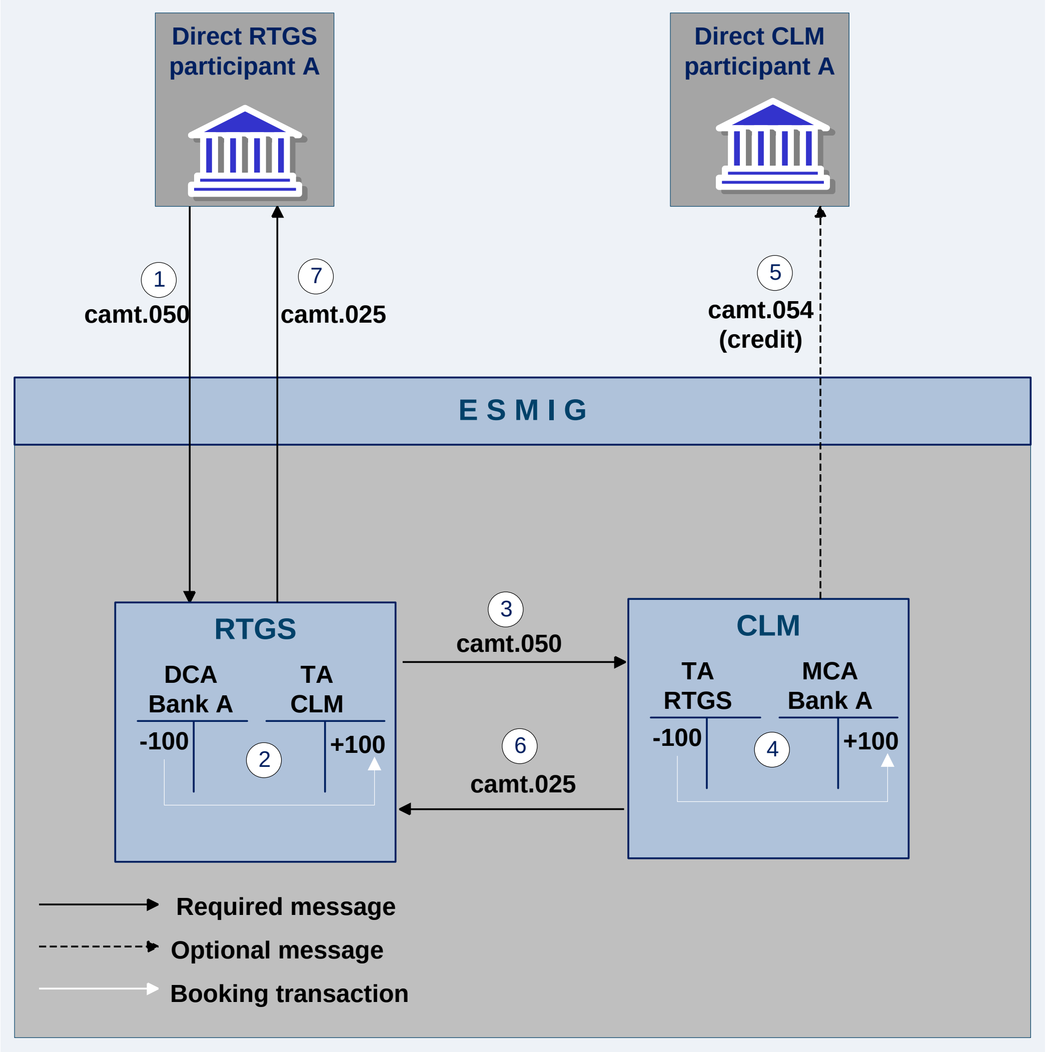


Figure 17 - camt.050 liquidity transfer from RTGS dedicated cash account to main cash account

Process description

Table 33 - Liquidity transfer from RTGS dedicated cash account to main cash account

Step	Processing in/between	Description
1	RTGS participant via ESMIG to RTGS	A camt.050 is sent from a direct RTGS participant via ESMIG to RTGS.
2	RTGS	Booking on RTGS dedicated cash accounts (RTGS dedicated cash account to transit account-CLM)
3	RTGS to CLM	A camt.050 is forwarded to CLM.
4	CLM	Booking on main cash account (transit account-RTGS to main cash account)

Step	Processing in/between	Description
5	CLM via ESMIG to CLM participant	A camt.054 (credit) is sent by CLM component via ESMIG to the CLM participant (optional).
6	CLM to RTGS	For execution a camt.025 generated in CLM is sent to RTGS component.
7	RTGS via ESMIG to direct RTGS participant	For execution a camt.025 is sent by RTGS via ESMIG to the direct RTGS participant.

Used messages

- LiquidityCreditTransfer (camt.050) [▷ 255]
- BankToCustomerDebitCreditNotification (camt.054) [▷ 261]
- Receipt (camt.025) [▷ 243]

5.2.2.3.3 Liquidity transfer between two main cash accounts (completed)

A CLM participant can transfer liquidity from one main cash account to another main cash account. The owners of the main cash accounts have to be in the same party or the accounts have to be in the same liquidity transfer group to work with the main cash account to be credited.

Message flow

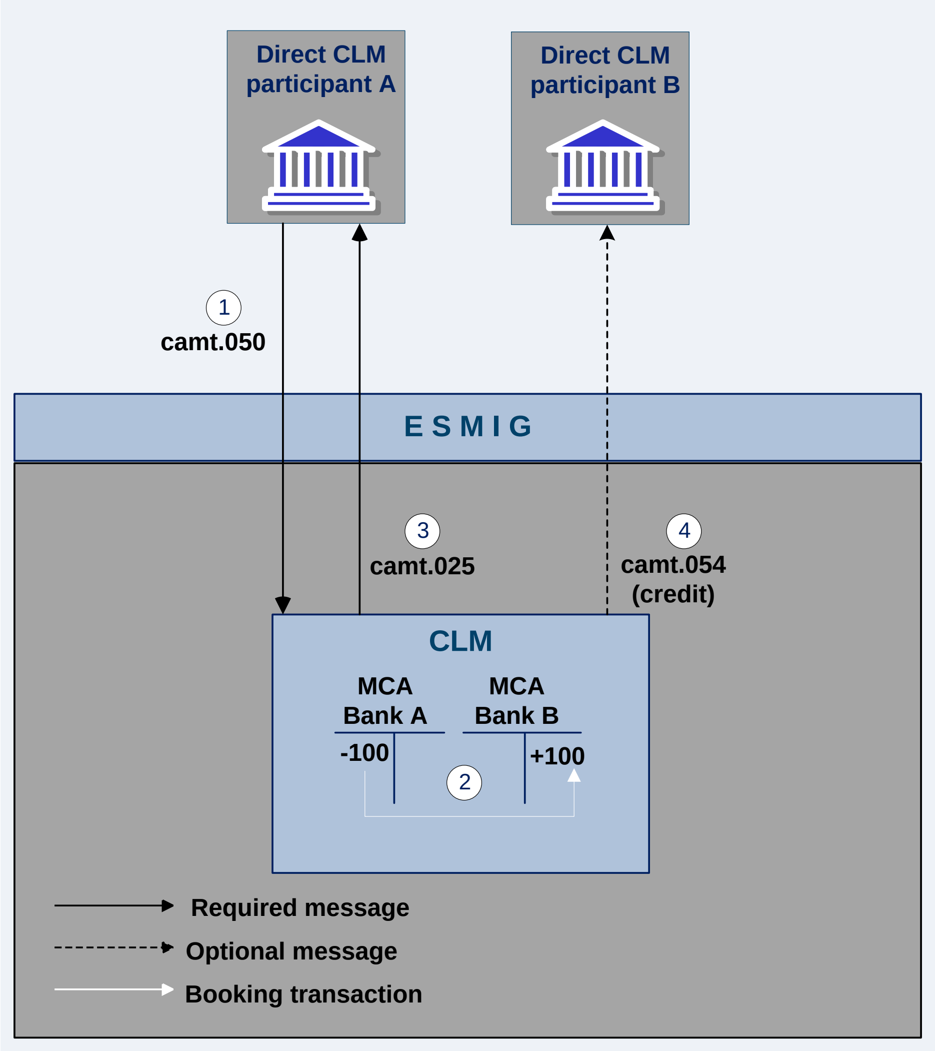


Figure 18 - camt.050 liquidity transfer intra-CLM

Process description

Table 34 - Liquidity transfer intra-CLM

Step	Processing in/between	Description
1	CLM participant A via ESMIG to CLM	A camt.050 is sent from a CLM participant A via ESMIG to CLM.
2	CLM	Booking on main cash accounts
3	CLM via ESMIG to CLM participant A	A camt.025 is sent by CLM via ESMIG to CLM participant A.
4	CLM via ESMIG to CLM participant B	A camt.054 is sent by CLM via ESMIG to CLM participant B (optional).

Used messages

- LiquidityCreditTransfer (camt.050) [▷ 255]
- BankToCustomerDebitCreditNotification (camt.054) [▷ 261]
- Receipt (camt.025) [▷ 243]

5.2.2.3.4 Liquidity transfer between two dedicated cash accounts in different settlement services/components (completed)

A settlement service/component participant can transfer liquidity from a dedicated cash account in one settlement service/component to a dedicated cash account within another settlement service/component.

Message flow

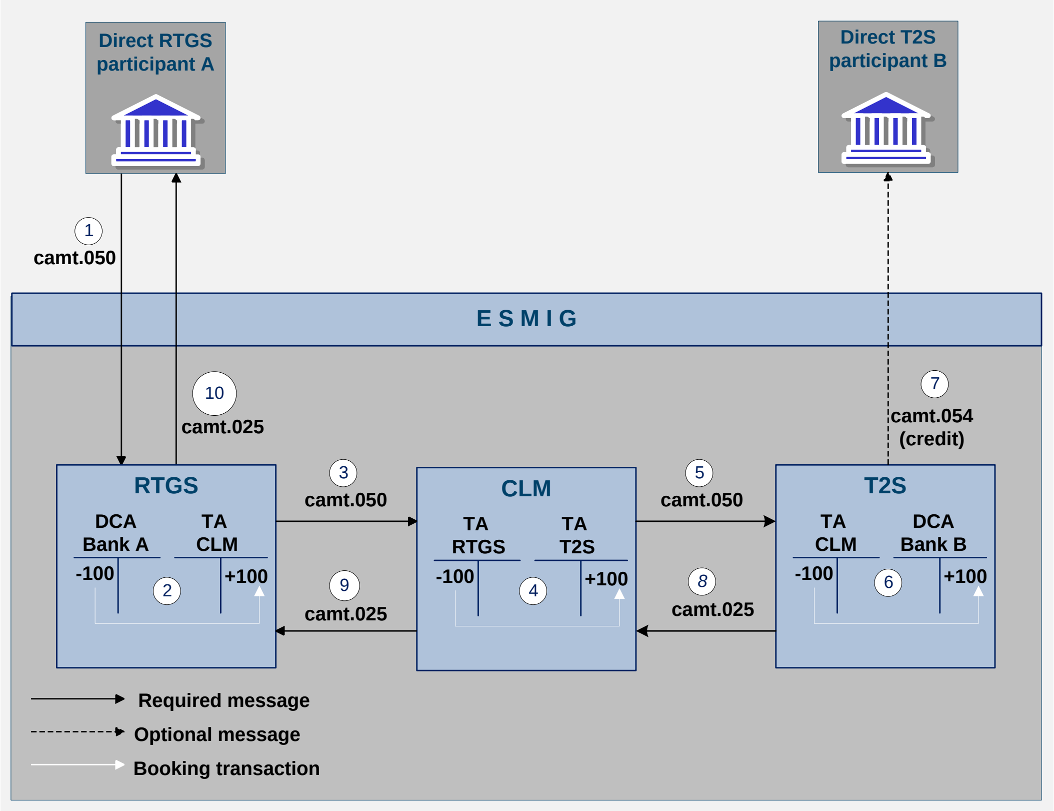


Figure 19 - camt.050 liquidity transfer inter-service/component

Process description

Table 35 - Liquidity transfer inter-service/component

Step	Processing in/between	Description
1	RTGS participant via ESMIG to RTGS	A camt.050 is sent from a direct RTGS participant via ESMIG to RTGS.
2	RTGS	Booking on RTGS dedicated cash accounts (RTGS dedicated cash account to transit account-CLM)
3	RTGS to CLM	A camt.050 is forwarded to CLM.
4	CLM	Booking on technical accounts in CLM (transit account -RTGS to transit account-T2S)
5	CLM to T2S	A camt.050 is forwarded to T2S.
6	T2S	Booking on T2S accounts (transit account –CLM to T2S dedicated cash account)
7	T2S via ESMIG to T2S participant	A camt.054 (credit) is sent by T2S via ESMIG to the T2S participant (optional).
8	T2S to CLM	A camt.025 generated in T2S is sent to CLM.
9	CLM to RTGS	CLM forwards the camt.025 to the RTGS.
10	RTGS via ESMIG to direct RTGS participant	A camt.025 is sent by RTGS via ESMIG to the direct RTGS participant.

Used messages

- LiquidityCreditTransfer (camt.050) [▷ 255]
- BankToCustomerDebitCreditNotification (camt.054) [▷ 261]
- Receipt (camt.025) [▷ 243]

5.2.2.3.5 Rejection of liquidity transfer orders (completed)

Liquidity transfer orders sent to the CLM have to pass several validations before the liquidity is effectively transferred. Validations performed include technical checks, format checks as well as checks for the correct content.

For different reasons a liquidity transfer can be rejected and a notification with the reason code for rejection is returned to the sending actor (see chapter Index of business rules and error codes [▷ 284]). The validations are distinguished in two types:

Technical validations

The system reactions on errors during technical validation differentiate between “well-formedness check in ESMIG” and “the schema validation in CLM”.

- In case the well-formedness check in ESMIG is not successful a ReceiptAcknowledgement (admi.007) is sent to the sending actor indicating which error occurred.
- In case the well-formedness check in ESMIG is successful, the service performs the schema validation.
- In case the schema validation in CLM is not successful, the service sends a ReceiptAcknowledgement (admi.007) to the sending actor indicating which error occurred.

Syntax/schema checks

CLM shall parse the message and perform a field level validation, e.g. on correct data type and size. CLM shall check whether all mandatory fields are populated. If the validation fails, a rejection notification with appropriate reason code is sent to the sender of the message (depending on the submission channel, a message in A2A mode or an error message on the screen in U2A mode).

Business validations

Rejections after the business validations result in a receipt message (camt.025) is sent to the sending actor including the respective error code(s) according to chapter Index of business rules and error codes [▸ 284].

Check for duplicate liquidity transfer

CLM carries out a duplicate submission control for incoming liquidity transfers. This control shall include the following fields: Sender of the message, Message Type, Receiver, Transaction Reference Number, Related Reference, Value Date and Amount.

Process specific authorisation checks

CLM performs specific authorisation checks. The liquidity transfer order can also be triggered by the scheduler in the case of standing orders.

Liquidity transfer group

For intra-service liquidity transfers, CLM checks whether both accounts belong to the same party or to participants within the same liquidity transfer group or not. If not, the order is rejected. This check is not performed for central bank accounts.

Field and reference data checks

The service performs the following field and reference data checks:

- field value validation - codes are valid, domain values are within allowed range
- cross-field validation – e.g. currency of the accounts involved equals the amount currency

- common reference data checks – e.g. existence of active parties and accounts

The validations described above is performed in one step in order to capture all the possible breaches; the checks therefore have not to stop after the first breach occurring, if there can be further breaches in the subsequent checks. If the validation fails overall, a rejection notification with appropriate reason codes for all breaches which occurred has to be sent to the sender. This principle applies to technical and business validations separately.

Subsequent processes and checks

- check available liquidity vs. amount to be transferred - CLM checks whether enough liquidity is available. Where there is a lack of liquidity the rules for partial execution apply.

  Note: In case reservations are used, only the non-reserved part of the available liquidity can be used for liquidity transfers.
- partial execution - if the liquidity transfer is initiated by an automatic trigger, CLM settles the liquidity transfer partially. For several standing orders, where the sum of all standing orders of the participant to be settled at the same event is larger than the available liquidity, CLM reduces all respective standing orders in a pro-rata mode.
- update cash balances - CLM books the liquidity transfer finally and irrevocably on both accounts and updates the defined value. CLM sends a (partly) success notification to the sending party and to the owner of the debited account.

5.2.3 Liquidity management features (completed)

5.2.3.1 Liquidity reservation (completed)

5.2.3.1.1 Overview (completed)

CLM offers the possibility to reserve cash of the main cash account, so that the main cash account has two types of source of liquidity:

- reserved for central banks
- non-reserved

The available liquidity in the reserved part of the main cash account is used for central banks (e.g. reimbursement of marginal lending) or for credit line decreases [4].

[4] The latter one uses the reserved part of the main cash account only in case there is not enough liquidity on the non-reserved part of the main cash account.

Reservations can be created, modified and deleted by the owner of the main cash account (or another actor acting on behalf of the main cash account owner) using U2A or A2A. Further details on the U2A functionality can be found in the CLM user handbook.

In general, the owner of the main cash account (or another actor acting on behalf of the main cash account owner) has the following possibilities.

- create and/or modify reservations with immediate effect during the current business day as a one-time reservation in CLM. This includes:
  - establishing a specific amount during the current day with immediate effect as a one-time reservation (e.g. setting a new reservation of 300)
  - "resetting" to zero the liquidity reserved for the current business day only with immediate effect.
  - changing the amount on demand during the day with immediate effect (e.g. from 300 to 200 or from 300 to 400).
- create, modify or delete a standing order reservation in CRDM valid as of the following day (i.e. valid as of the next business day until next change or the deletion of the standing order)

The liquidity reservation (with immediate effect as well as standing order reservation) is possible throughout the whole business day with the exception of the end of day processing and the maintenance window.

Standing order reservation

Standing order reservations are created and managed in CRDM. The amount defined in the standing order for reservation is valid at the start-of-day, even if the amount of the reservation is changed during the preceding business day with immediate effect (such a change is only valid for the respective business day).

At the start-of-day, reservations are set according to the standing orders and up to the available liquidity on the main cash account.

In case that the amount of non-reserved available liquidity is not sufficient to fulfil the liquidity reservation set up via standing order, the reservation is partially executed. CLM continues attempting to reserve the remaining amount until the reservation amount is reached whenever there is an increase of non-reserved liquidity on the main cash account.

Note: central banks always have a higher priority compared to pending reservations. CLM always settles the queued central banks first, before the reservation is fully executed.

One-time reservation with immediate effect

One-time reservations are created and managed directly in CLM. As outlined above it is possible to create a reservation for the current business day only. Moreover, it is possible to modify an existing reservation and to "reset to zero" the amount of the reservation with immediate effect for the current business day only. Owing to the asynchronous processing in CLM incoming liquidity might be blocked and used by a parallel booking process before the attempt to increase the reservation is performed.

Upon receipt of end of day notification and a new reservation order, CLM stops processing the original (queued) reservation order.

In case that the amount of non-reserved available liquidity is not sufficient to fulfil the liquidity reservation order, the reservation is partially executed. CLM attempts to reserve the remaining amount until the reservation amount is reached whenever there is an increase of non-reserved liquidity on the main cash account.

Note: central banks always have a higher priority compared to pending reservations. CLM always settles the queued central banks first, before the reservation is fully executed.

5.2.3.1.2 Liquidity reservation process (completed)

Reservation process – one-time reservation with immediate effect

The following message flows illustrate the reservation creation, the amendment (ModifyReservation (camt.048) [▸ 250]) and the “reset to zero”(DeleteReservation (camt.049) [▸ 253] in CLM.

Note: The creation and the management of standing order reservations are done in CRDM.

Message flow

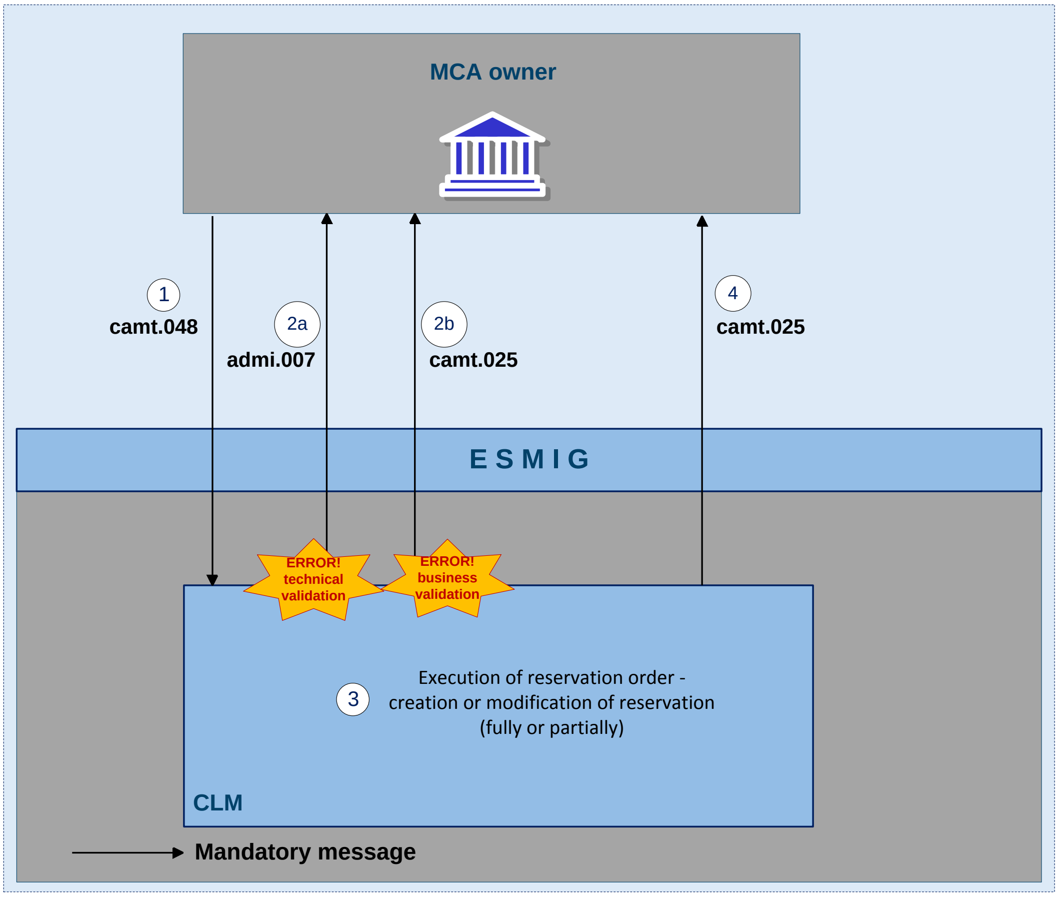

MCA owner
1
camt.048
2a
admi.007
2b
camt.025
4
camt.025
E S M I G
ERROR! technical validation
ERROR! business validation
3
Execution of reservation order - creation or modification of reservation (fully or partially)
CLM
Mandatory message

Figure 20 - One time reservation with immediate effect

Process description

Table 36 - Creation of a one-time liquidity reservation with immediate effect

Step	Processing in/between	Description
1	Owner of the main cash account via ESMIG to CLM	The owner of the main cash account sends a camt.048 via ESMIG to CLM.
2a	CLM	CLM perform a message check and technical validation. In case of a negative technical validation, an error message (admi.007) is sent. In case of a successful technical validation, CLM performs the business validation checks.

Step	Processing in/between	Description
2b	CLM	CLM business validation. In case of a negative business validation, an error message (camt.025) is sent. In case of successful business validation, CLM starts executing the reservation
3	CLM	CLM executes one-time reservation with immediate effect: In case of successful business validation, CLM starts executing the reservation.
4	CLM via ESMIG to the owner of the main cash account	CLM sends a camt.025 via ESMIG to owner of the main cash account **Note:** Only in case the total amount could be reserved, a notification (camt.025) is sent to the owner of the main cash account (or another actor acting on behalf).

Used messages

- ModifyReservation (camt.048) [▸ 250]
- Receipt (camt.025) [▸ 243]
- ModifyTransaction (camt.007) [▸ 237]

Modification of a reservation with immediate effect

Message flow

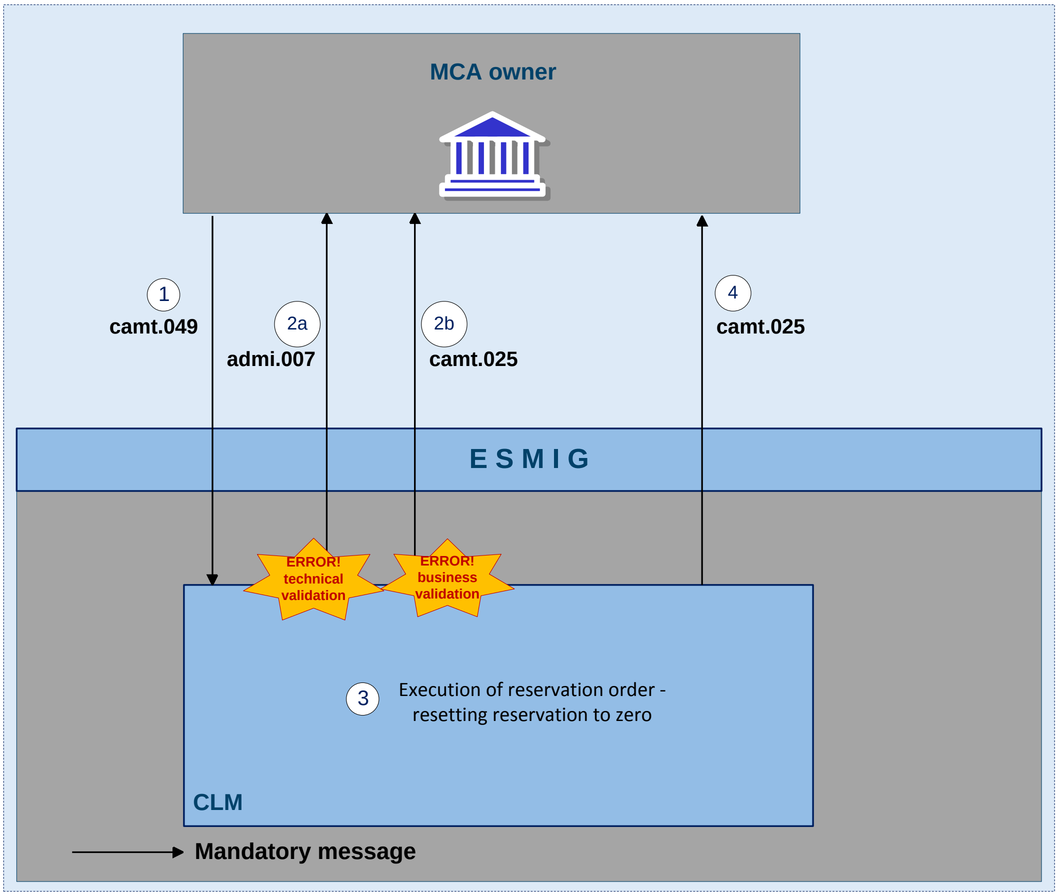

MCA owner
1
camt.049
2a
admi.007
2b
camt.025
4
camt.025
E S M I G
ERROR! technical validation
ERROR! business validation
3
Execution of reservation order - resetting reservation to zero
CLM
Mandatory message

Figure 21 - One time reservation with immediate effect

Since the same messages are used for creating a reservation as well as modifying a reservation, the message flow for creating a one-time reservation applies here, too.

Process description

Table 37 - Modification of a one-time liquidity reservation with immediate effect

Step	Processing in/between	Description
1	Owner of the main cash account via ESMIG to CLM	The owner of the main cash account sends a camt.048 via ESMIG to CLM in order to modify the reservation with immediate effect.
2a	CLM	CLM perform a message check and technical validation. In case of a negative technical validation, an error message (admi.007) is sent. In case of a successful technical validation, CLM performs the business validation checks.
2b	CLM	CLM business validation. In case of a negative business validation, an error message (camt.025) is sent. In case of successful business validation, CLM starts executing the reservation
3	CLM	CLM executes the modification of reservation: in case of successful business validation, CLM starts executing the reservation modification.
4	CLM via ESMIG to the owner of the main cash account	CLM sends a camt.025 via ESMIG to owner of the main cash account **Note:** Only in case the total amount could be reserved, a notification (camt.025) is sent to the owner of the main cash account (or another actor acting on behalf).

Used messages

- ModifyReservation (camt.048) [▸ 250]
- Receipt (camt.025) [▸ 243]
- ModifyTransaction (camt.007) [▸ 237]

“Resetting to zero” reservation

Message flow

Note: Owing to the fact that the messages used are the same for one-time reservation with immediate effect and standing order reservation, the message flow applies for both cases.

Process description

Table 38 - "Resetting to zero" reservation

Step	Processing in/between	Description
1	Owner of the main cash account via ESMIG to CLM	Owner of the main cash account (or another actor acting on behalf of the main cash account owner) sends a camt.049 via ESMIG to CLM in order to delete the reservation.
2a	CLM	CLM perform a message check and technical validation. In case of a negative technical validation, an error message (admi.007) is sent. In case of a successful technical validation, CLM performs the business validation checks.
2b	CLM	CLM business validation. In case of a negative business validation, an error message (camt.025) is sent. In case of successful business validation, CLM starts executing the reservation
3	CLM	CLM executes “resetting to zero”: in case of successful business validation, CLM starts executing the reservation by setting the reservation for the current business day to zero.
4	CLM via ESMIG to the owner of the main cash account	CLM sends a camt.025 via ESMIG to owner of the main cash account. **Note:** Only in case of a successful “reset to zero”, a notification (camt.025) is sent to the owner of the main cash account (or another actor acting on behalf).

Used messages

- ModifyReservation (camt.048) [250]
- Receipt (camt.025) [243]
- ModifyTransaction (camt.007) [237]

5.2.3.1.3 Effect of liquidity reservation (completed)

Activity	Balance on main cash account of Bank A	Liquidity reserved for central bank operations	Non-reserved liquidity
Start	1,000	300	700
Settlement liquidity transfer = 50 (debit)	950 ⇩	300 ⬄	650 ⇩
Reimbursed marginal lending to central bank = 200	750 ⇩	100 ⇩	650 ⬄
Receiving liquidity transfer from Bank C = 20 (credit)	770 ⇧	100 ⬄	670 ⇩
Set-up of overnight deposit = 100	670 ⇩	0 ⇩	670 ⬄
Incoming liquidity dedicated cash accountnsfer from RTGS dedicated cash account = 80	750 ⇧	0 ⬄	750 ⇧
Creation of one-time liquidity reservation = 200	750 ⬄	200 ⇧	550 ⇩
Set-up of overnight deposit = 150	600 ⇩	50 ⇩	550 ⬄
Submitting a "resetting to zero" reservation = 50	600 ⬄	0 ⇩	600 ⇧

5.2.3.2 Floor/ceiling (completed)

5.2.3.2.1 Definition of floor/ceiling threshold (completed)

The floor/ceiling threshold manages the behaviour of CLM after the successful settlement of a payment (central bank operation) whenever the amount of the account undercuts the floor amount or exceeds the ceiling amount. Since this functionality is optionally, it is up to the user to define the floor/ceiling threshold in CRDM.

The owner of the main cash account (or another actor acting on behalf of the main cash account owner) can define a minimum (“floor”) or maximum (“ceiling”) liquidity amount for its main cash account(s). The CLM participant has the option to choose the behaviour of CLM once the balance is below the defined floor or above the defined ceiling amount. Two options are available:

1. CLM generates a notification to be sent to the owner of the main cash account (or to another actor on behalf of the main cash account owner) informing about the floor/ceiling breach (upon which the CLM participant can take action); or
2. CLM automatically generates an inter-service liquidity transfer to pull cash from the CLM participant's RTGS dedicated cash account used for payments (where the floor is breached) or push cash to the CLM participant's RTGS dedicated cash account used for payments (where the ceiling is breached).

The floor/ceiling functionality itself is only triggered after the settlement of a central bank operation. So it is not relevant for liquidity transfers.

5.2.3.2.2 Breach of floor/ceiling threshold - notification (completed)

If the CLM participant choses the first option, CLM generates and sends out a notification with the information that the account balance is below the floor or that the account balance is above the ceiling respectively:

- in U2A (see user handbook) or
- in A2A mode (via ReturnAccount (camt.004) [▸ 229]; Floor and ceiling processing [▸ 191]).

The notification is sent every time the threshold is undercut (floor) or exceeded (ceiling). However, CLM does not sent the notification if -after passing the threshold- the main cash account balance remains consistently below the floor or above the ceiling.

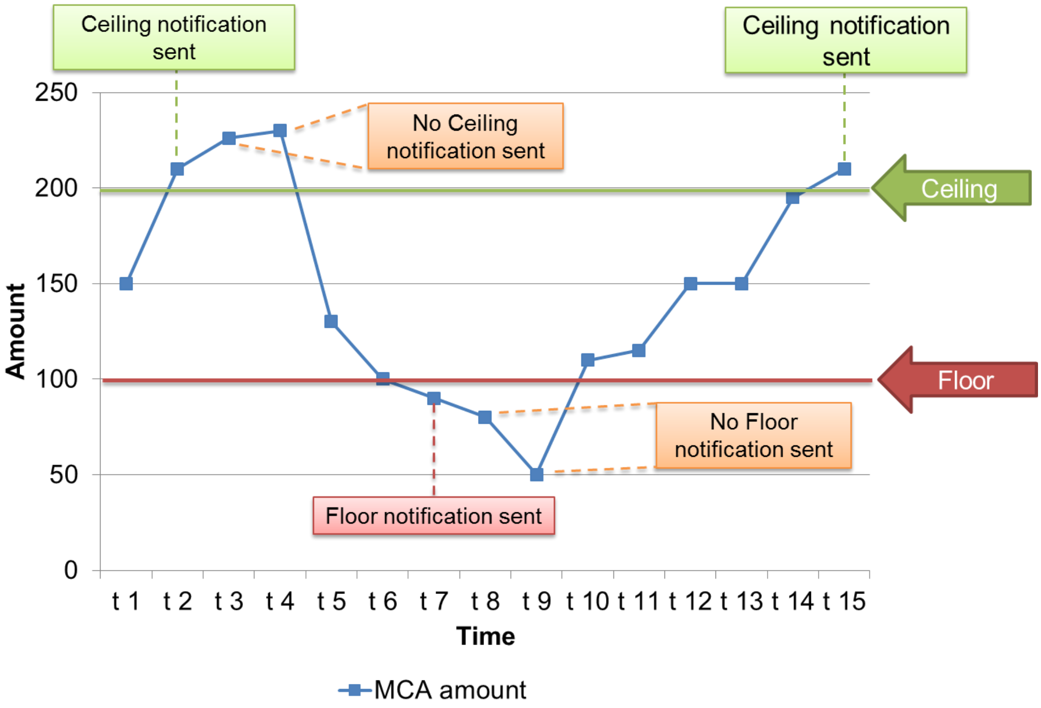


Figure 22 - Breach of floor/ceiling threshold – notification

5.2.3.2.3 Breach of floor/ceiling threshold - automatic liquidity transfer (completed)

If the CLM participant choses the second option, CLM creates and releases an inter-service liquidity transfer.

- In case of a breach of the floor threshold a certain amount is pulled from the RTGS dedicated cash account and credited the main cash account.
  - The used RTGS dedicated cash account is linked to the main cash account as defined in CRDM.
  - The amount to be transferred is the difference between the current main cash account balance and the predefined target amount, whereas the target amount can be different but equal or above the floor amount. If the available liquidity on the RTGS account is not sufficient, the liquidity transfer is partially settled.
- In case of a breach of the ceiling threshold a certain amount is pushed to the RTGS dedicated cash account and debited the main cash account.
  - The used RTGS dedicated cash account is the same as for the floor threshold, meaning it is linked to the main cash account as defined in CRDM.
  - The amount to be transferred to the RTGS dedicated cash account is the difference between the current main cash account balance and the predefined target amount.

The target amount could be different but is below the ceiling amount.

– The target amount for ceiling is a different one as the target amount of the floor threshold.

After the successful execution of the inter-service liquidity transfer the amount on the main cash account is within the boundaries of the floor or ceiling amount again.

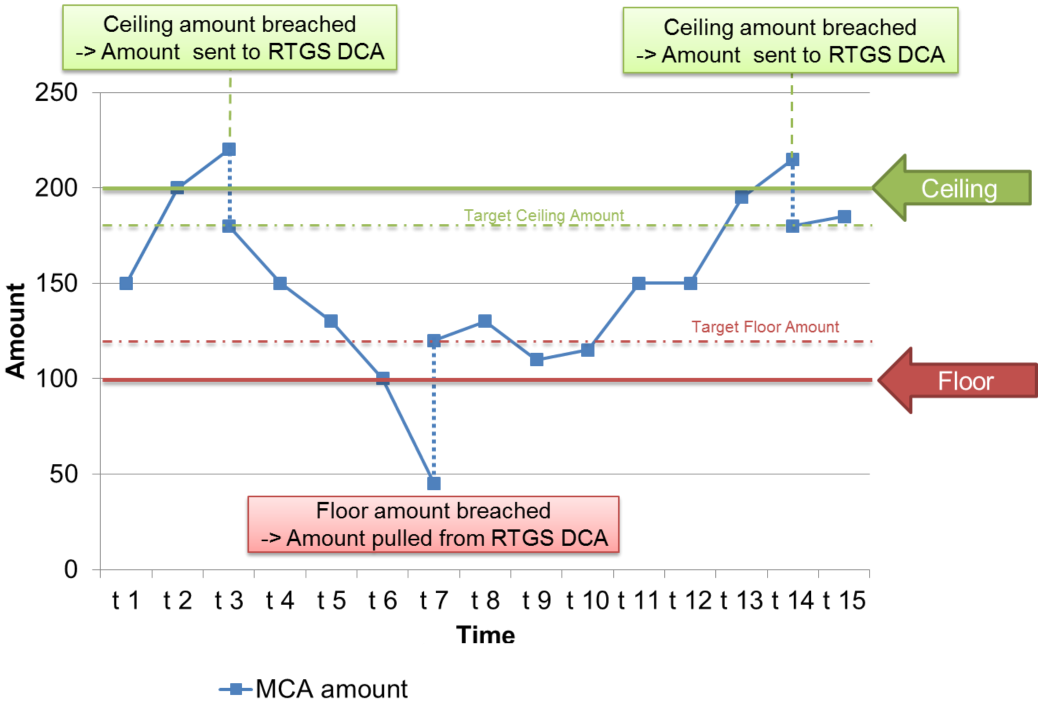


Figure 23 - Breach of floor/ceiling threshold - automatic liquidity transfer

5.3 Reserve management (to be completed in iteration 4)

5.3.1 Overview (to be completed in iteration 4)

5.3.2 Reserve management process (to be completed in iteration 4)

5.4 Standing facilities management

5.4.1 Overnight deposit (completed)

5.4.1.1 Overview (completed)

The overnight deposit process is an element of the central liquidity management standing facilities and breaks down into three parts:

- setup of an overnight deposit
- overnight deposit reverse transaction
- overnight deposit reimbursement and interest calculation

CLM participants can use the deposit facility to make overnight deposits with their national central banks.

To setup an overnight deposit, CLM participants are able to transfer liquidity from their main cash account to the relevant overnight deposit account.

Note: The owner of overnight deposit account to be set up is the central bank. A central bank has to open a separate overnight deposit account per CLM participant using the overnight deposit functionality.

It is also possible to activate a reverse transaction in order to reduce the amount deposited in the overnight deposit account. This has to be initiated before the deadline for the usage of standing facilities. CLM calculates the interest to be paid on the overnight deposit and, at the start of the next business day, returns automatically the capital amount and credits the interest on the CLM participant’s main cash account. In case of a negative interest rate, CLM calculates the interest to be paid by the CLM participants on the overnight deposit and, at the start of the next business day, returns automatically the capital amount to CLM and debits the interest to be charged from the CLM participant’s main cash account.

Note: For central banks outside the Eurosystem interests are always accumulated and cleared on a monthly basis. CLM calculates the accumulated interest at the end of a calendar month and clears ten days after the first business day of the following month (warehoused payment). The respective connected central bank has the possibility to check the calculated interest and to cancel the warehoused payment if the calculation is not correct.

Preconditions

A participant wishing to initiate an overnight deposit needs to:

- be a CLM participant
- be eligible to the overnight deposit facility
- have a main cash account
- have dedicated overnight deposit account(s) set up in CLM.
- for reverse transactions only: an overnight deposit for that business day has been set up previously

Furthermore, a control mechanism is placed in order to verify that the total amount envisaged for non-Eurosystem central banks are not exceeded.

Triggers

The setup and reversal of an overnight deposit can be initiated by:

- an overnight deposit or reverse transaction request sent by the CLM participant in A2A or

- manual input via U2A screen by the CLM participant (or central bank operator acting on behalf of the CLM participant)

The reimbursement of deposited capital and calculation of interest is triggered by the start of the next business day. CLM triggers automatically the liquidity transfer for the repayment of the capital amount and the interest payment. Interest for non-Eurosystem central banks is processed differently.

Definition of execution times

It is possible for CLM participants to set up and/or to reverse an overnight deposit from the opening time of CLM (i.e. 19:00 and after overnight deposit, marginal lending reimbursement and interest calculation) until the general cut-off for the use of standing facilities (i.e. 18:15 with additional fifteen minutes on the last day of the reserve maintenance period) with the exception of the maintenance window.

Settlement principles

The following principles apply to the processing of liquidity transfer orders linked to overnight deposits.

- Attempt to settle liquidity transfer immediately after its submission.
- Liquidity transfer orders: are either settled completely or cancelled (no partial settlement).
- Liquidity transfer orders are not be queued.
- Liquidity from RTGS-dedicated cash account(s) are be used to supplement insufficient liquidity on the main cash account.

5.4.1.2 Overnight deposit process (completed)

5.4.1.2.1 Setup overnight deposit (completed)

Message flow

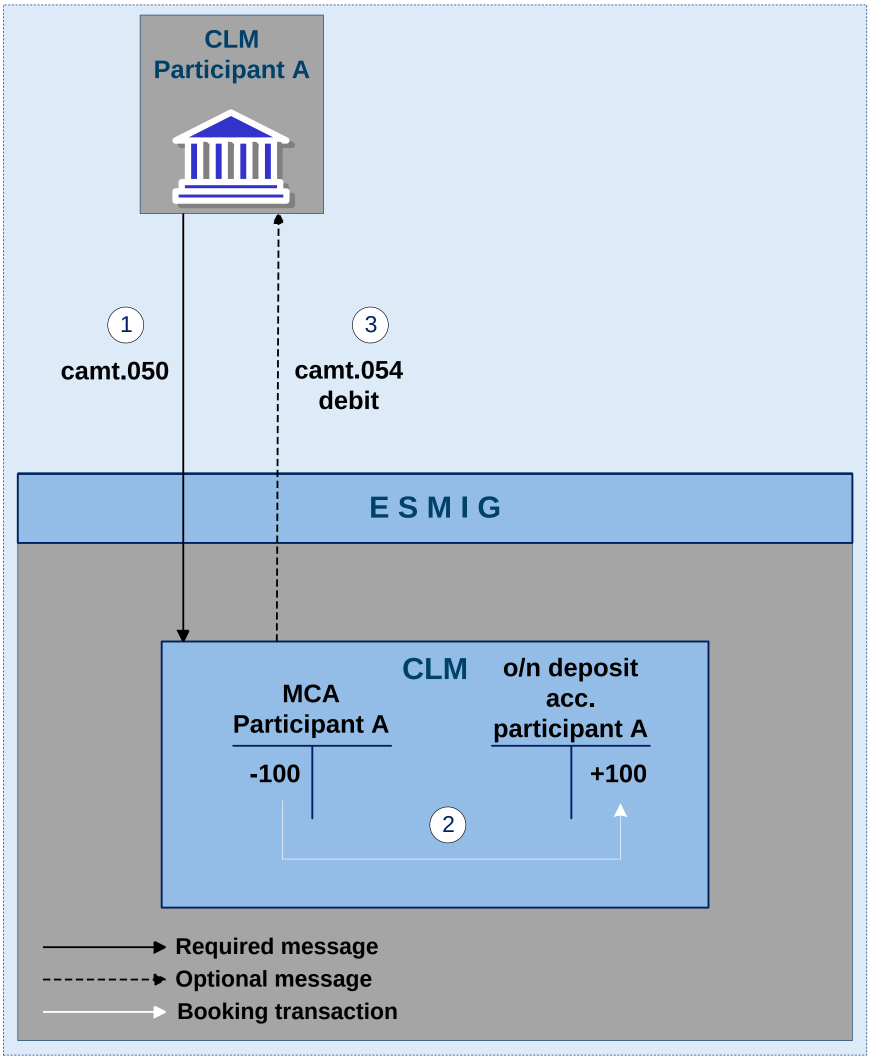


Figure 24 - camt.050 - setup overnight deposit

Process description

The process of setting up an overnight deposit in CLM consists of the following steps.

Table 39 - Setup overnight deposit

Step	Processing/between	Description
1	CLM participant via ESMIG to CLM	The CLM participant sends a camt.050 to CLM.
2	CLM	CLM credits the overnight deposit account of the central bank and debit the main cash account of the participant, if validations are positive.
3	CLM via ESMIG to CLM participant	CLM send a receipt (camt.025) to the CLM participant.
4	CLM via ESMIG to CLM participant	CLM sends an optional notification (camt.054 debit) to the CLM participant.

Used messages

- Receipt (camt.025) [▸ 243]
- LiquidityCreditTransfer (camt.050) [▸ 255]
- BankToCustomerDebitCreditNotification (camt.054) [▸ 261]

Expected results

The set-up of an overnight deposit leads to a transfer of liquidity from the participant's main cash account to the overnight deposit account of the central bank. Participants are allowed to send multiple camt.050 to set-up overnight deposit. Each new instruction increases the deposited amount.

Technical validations

At the reception of an overnight deposit request, the CLM interface performs technical validations. For further details please refer to Rejection of payments [▸ 62].

After encountering the first negative validation result, the service interface continues to validate as far as possible and reports all negative results combined in a single reply message. The CLM interface rejects the order not until performing all possible technical validations. In case of a negative result of the technical validation the request is rejected and a negative notification (admi.007) is sent to the instructing CLM participant.

If all technical validations are passed without any error, the request is sent to CLM for further processing, i.e. business validations.

Business validations

Once the technical validations are positively completed, the overnight deposit request proceeds the business validations. For further details please refer to Rejection of payments [▸ 62].

If any of the business validations fails, the overnight deposit request is rejected and a negative receipt (camt.025) is sent to the instructing CLM participant.

5.4.1.2.2 Overnight deposit reverse transaction (completed)

Once CLM participants have sent a set-up overnight deposit order, it is possible for the CLM participant (before the deadline for the usage of standing facilities) to activate a reverse transaction in order to reduce the amount deposited in the overnight deposit account.

Message flow

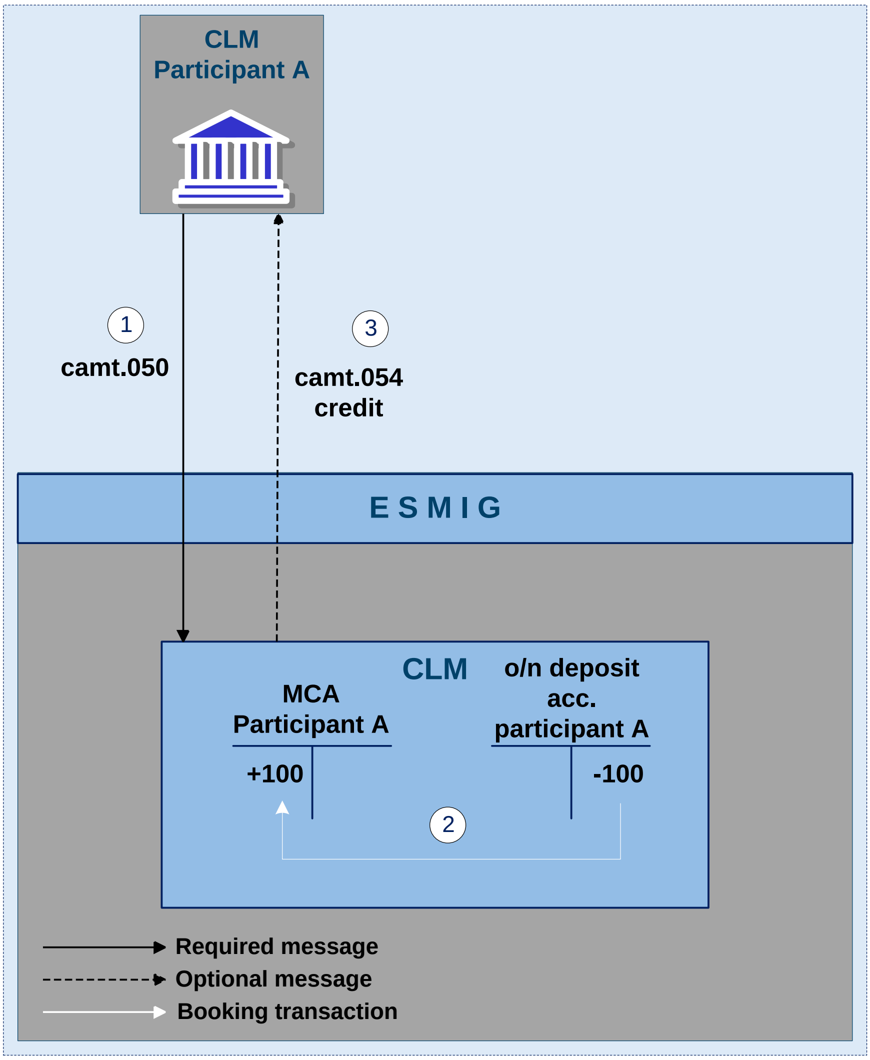


Figure 25 - camt.050 - reverse overnight deposit

Process description

The process of reversing an overnight deposit in CLM consists of the following steps.

Table 40 - Reverse overnight deposit

Step	Processing in/between	Description
1	CLM participant via ESMIG to CLM	The CLM participant sends a camt.050 to CLM.
2	CLM	CLM debits the overnight deposit account of the central bank and credits the main cash account of the participant if business validations are positive.
3	CLM via ESMIG to CLM participant	CLM send a receipt (camt.025) to the CLM participant.
4	CLM via ESMIG to CLM participant	CLM sends an optional notification (camt.054 credit) to the CLM participant.

Used messages

- Receipt (camt.025) [▸ 243]
- LiquidityCreditTransfer (camt.050) [▸ 255]
- BankToCustomerDebitCreditNotification (camt.054) [▸ 261]

Expected results

The reverse transaction leads to a transfer of liquidity from the central banks overnight deposit account to the CLM participant's main cash account.

Technical and business validations

Technical and business validations check that a corresponding overnight deposit is set-up previously. Apart from this additional check the same technical and business validations apply as described in chapter Rejection of payments.

5.4.1.2.3 Overnight deposit reimbursement and interest calculation (completed)

At start of the next business day CLM calculates the interest to be booked on the overnight deposit and automatically books the capital amount and the interest amount to the participants main cash accountOvernight deposit reimbursement and interest calculation [▸ 114].

Message flow

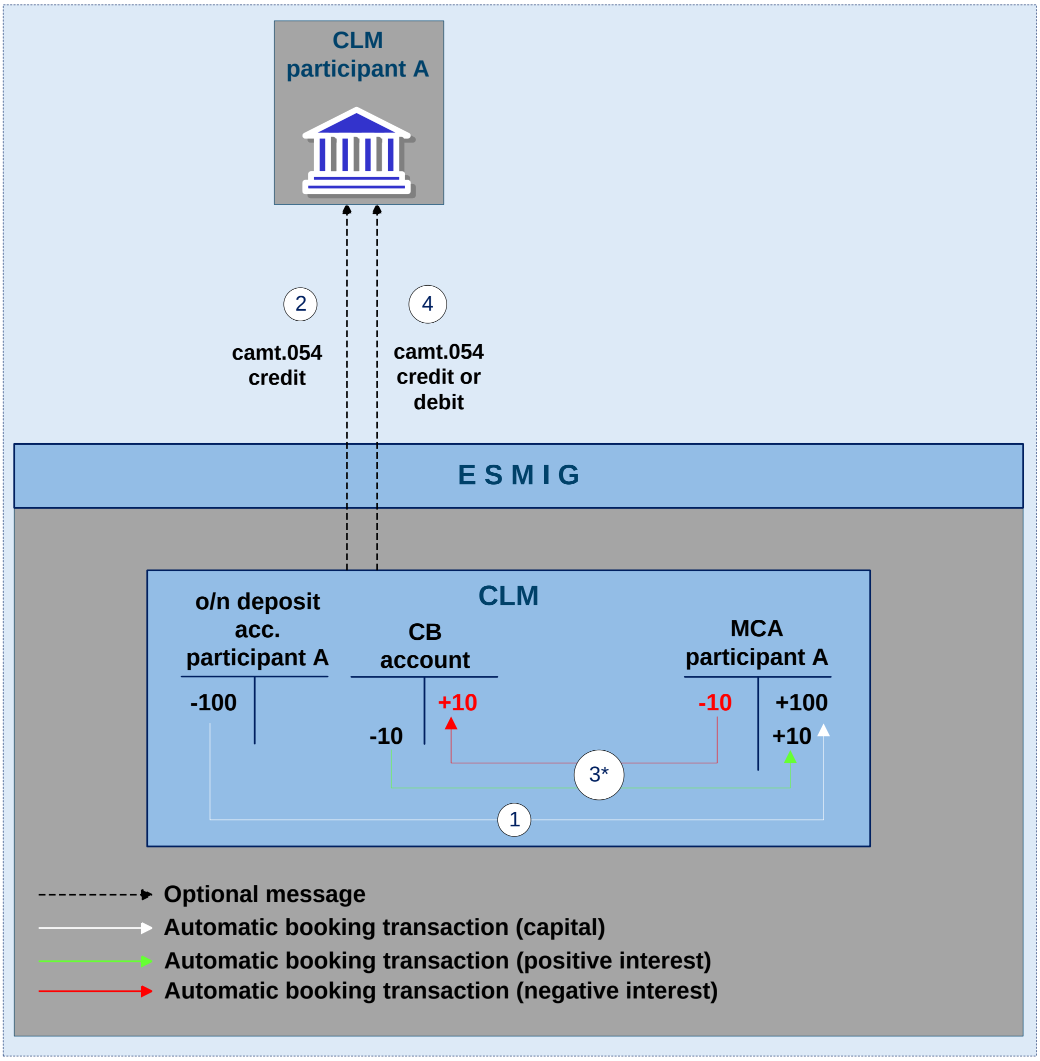


Figure 26 - Reimburse overnight deposit

Process description

The process of overnight deposit reimbursement and interest calculation in CLM consists of the following steps.

Table 41 - Reimburse overnight deposit

Step	Processing in/between	Description
1	CLM	CLM automatically debits the overnight deposit account of the central bank with the deposited amount and credits the main cash account of the CLM participant.
2	CLM via ESMIG to direct participant	CLM sends an optional notification (camt.054 credit) to the CLM participant.
3	CLM	CLM automatically debits the central bank account and credit the main cash account of the CLM participant, if the overnight deposit rate is positive [5]. CLM automatically credits the main cash account of the central bank and debit the main cash account of the CLM participant, if the overnight deposit rate is negative [6].
4	CLM via ESMIG to direct participant	CLM sends an optional notification (camt.054 credit or debit) to the CLM participant.

Used messages

- BankToCustomerDebitCreditNotification (camt.054) [▸ 261]

Expected results

The reimbursement of overnight deposit leads to the automatic transfer of liquidity (deposited capital) from the central overnight deposit account to the CLM participant's main cash account in CLM.

In addition, CLM automatically debits (or credits) the central bank account and credits (or debits) the participant’s main cash account with the calculated interest (depending on whether the overnight deposit rate is positive or negative).

Interest for non-Eurosystem central banks is processed differently.

Technical and business validations

The same validation processes as for setup of overnight deposits apply (see chapter “Technical validation” in chapter Setup overnight deposit [▸ 108]).

[5] CLM generates an interest payment even if the overnight deposit rate is zero.

[6] CLM generates an interest payment even if the overnight deposit rate is zero.

Interest calculation and payment for non-Eurosystem central banks is done at the end of the calendar month.

5.4.2 Marginal lending on request (to be completed in iteration 4)

5.4.2.1 Overview (to be completed in iteration 4)

5.4.2.2 Marginal lending on request process (to be completed in iteration 4)

5.4.2.2.1 Setup marginal lending on request (to be completed in iteration 4)

5.4.2.2.2 Marginal lending reimbursement and interest calculation (to be completed in iteration 4)

5.4.3 Automatic marginal lending (to be completed in iteration 4)

5.4.3.1 Overview(to be completed in iteration 4)

5.4.3.2 Automatic marginal lending process (to be completed in iteration 4)

5.4.3.2.1 Process automatic marginal lending (to be completed in iteration 4)

5.4.3.2.2 Marginal lending reimbursement and interest calculation (to be completed in iteration 4)

5.5 Information management for CLM

5.5.1 CLM status management (completed)

5.5.1.1 Concept (completed)

CLM informs its CLM actors of the processing results. This information is provided to the CLM actors via a status reporting which is managed by the status management. The communication of status to CLM actors is complemented by the communication of reason codes in case of negative result of a CLM process (e.g. validation failure notifications).

5.5.1.2 Overview (completed)

The status management process manages the status updates of the different instructions existing in CLM in order to communicate these status updates through status advice messages to the CLM actors throughout the lifecycle of the instruction. Status information on push basis is only available in A2A mode. Respective status advice messages are pushed via store-n-forward network service.

The status management handling also provides the reason codes to be sent to CLM actors in case of negative result of a CLM process (e.g. to determine the reason why an instruction is unsuccessfully validated or settled).

The status of an instruction is indicated through a value, which is subject to change through the lifecycle of the instruction. This value provides CLM actors with information about the situation of this instruction with respect to a given CLM process at a certain point in time.

Since each instruction in CLM can be submitted to several processes, each instruction in CLM may have several status. However, each of these status has one single value at a certain moment in time that indicates the instruction’s situation at the considered moment. Depending on its instruction type, an instruction is submitted to different processes in CLM. Consequently, the status featuring each instruction depend on the considered instruction type.

The following sections provide:

- the generic principles for the communication of status and reason codes to CLM actors
- the list of status featuring each instruction type as well as the possible values for each of these status

Reason codes are not exhaustively detailed below but are provided in chapter Index of status value and codes [▸ 286].

5.5.1.3 Status management process (completed)

Communication of status and reason codes to CLM actors

CLM actors can query the status values and reason codes of their instructions (e.g. payment orders, liquidity transfers, tasks, reference data updates) during the day.

The status can be classified in the following two types, common to all types of instructions.

- “Intermediate status” - in general an instruction has more than one status in its lifetime. If the status of an instruction is not a final status type, then the instruction is still process in CLM.

With each step in the process of the instruction the status changes until a final status is reached. Further status updates are communicated to the CLM actors if reached.

- "Final status" - this is the last status of an instruction (i.e. the status of an instruction when processing ends). At a point in time, any instruction in CLM reaches a final status and all respective processes are completed.

For some status updates mandatory information is provided. For other status updates, the status management process informs the CLM actor of the status change by means of the sending of status advice messages (according to their message subscription configuration).

Statuses and status values in CLM

The detailed status concept is provided in iteration 4.

5.5.2 CLM report generation (completed)

5.5.2.1 Concept (completed)

CLM provides the possibility to create the predefined report "statement of account" periodically. The CLM component triggers the generation of the "statement of account" report based on the reference data configuration. It is only foreseen at the business event "end of day". The report is not created intraday. Depending on the CLM participant's preferences the report is either sent out directly after creation or stored for later retrieval.

Report name	ISO message	ISO code
Statement of accounts	BankToCustomerStatement	BankToCustomerStatement (camt.053) [▸ 258]

The respective business process is described in chapter Receive report [▸ 198].

5.5.2.2 Overview (completed)

The report "statement of account" includes information on one single main cash account of a CLM participant. It is not possible to receive one combined "statement of account" for more than one main cash account. Furthermore it does not include information from other components, i.e. there is no report including combined information of CLM and RTGS.

The report provides information about all items that are booked on the main cash account and balance information of the current business day.

It is provided as a complete report, i.e. no delta version is offered.

The configuration of a report is independent from the message subscription for notifications, i.e. no message subscription reference data is needed in case the report should be sent (push mode).

5.5.2.3 Report generation process (completed)

Preconditions for report creation

In order to avoid unnecessary processing and storage CLM does not create reports automatically. To initiate the creation of a report, the report receiver has to configure the report in advance. The configuration is done via the graphical user interface for the reference data, which is described in the CLM UHB.

This configuration is stored as reference data and is valid until the report receiver decides that the report has not to be created anymore or until the “valid to” date stored within the report configuration is reached.

Moment of data extraction

The creation of a “statement of account” report is always triggered at the end of day of the CLM component after finalisation of booking processes [business event “EOD”]. A new report configuration can be set up for the next business day at the earliest. The possible validity limitations have to be specified when the report is configured for the first time. The respective component only creates those reports, for which the underlying report configurations is valid at the current business day.

Availability of the report in CLM

A generated report is available for download until it is replaced by a new version of it, i.e. a report that is created at the end of day of the current business day replaces the report that was created at the end of day of the previous business day. The replaced report is no longer available for download in CLM. In A2A mode CLM pushes the specific report, provided that the push preference for the report is stored for the respective recipient in reference data (i.e. report configuration). The message is sent out based on the routing information stored for the CLM participant. Otherwise the report is just stored after generation and can be downloaded in pull mode via U2A. Additionally a resend request allows the actor to initiate a re-delivery of the last report, which was pushed before.

CRDM parameter synthesis

The following parameters are created and updated by the CRDM actor (see Table 50 - Report configuration [▸ 134]) for the set-up of a report.

Table 42 - Parameters for the set-up of a report

Parameter	Mandatory/ optional	Possible values	Hint
Report type	Mandatory	Statement of accounts	
Concerned account	Mandatory	Main Cash Account	
Possible recipient of a report	Mandatory	CLM participant	
Communication channel	Mandatory	Push mode, pull mode	
Valid from	Mandatory	ISO-date	
Valid to	Optional	ISO-date	The field „Valid To" is the only field that can be amended after the report configuration has been stored.

Concerned account

Each report provides information on a certain scope of data. The data scope is indicated by the main cash account for which it is configured. The concerned account has to be specified, when the report is configured for the first time. It is necessary to store one configuration per main cash account for which the report should be created.

Possible recipients of a report

All reports can be received by the technical address of

- concerned account owner
- another authorised party (e.g. co-manager)

A created report can be received by one or several receivers. Each CLM participant can decide, if it wishes to receive a report directly after its creation or if it wants to query it ad-hoc via U2A.

If a recipient wishes to receive a report directly after its creation, this has to be stored in the reference data configuration of the report in CRDM.

If a recipient does not wish to receive a report directly after its creation but to request it afterwards, this CLM behaviour has to be stored in the reference data configuration of the report as well. Furthermore this recipient is stored as recipient of a report.

As a general principle the recipient(s) of a report can be different from the concerned account owner, but have to be configured in the same system entity. For information about the setup of report configuration for specific concerned account owners and recipients of a report, please see CLM UHB chapters related to report configuration setup.

It is allowed to request a resending of the currently available statement of account. Hereafter, the respective cases are described.

Case: resend request with positive validation and re-delivery

A resend request allows delivering the report message once more to the same technical address as used for the initial report delivery.

Message flow

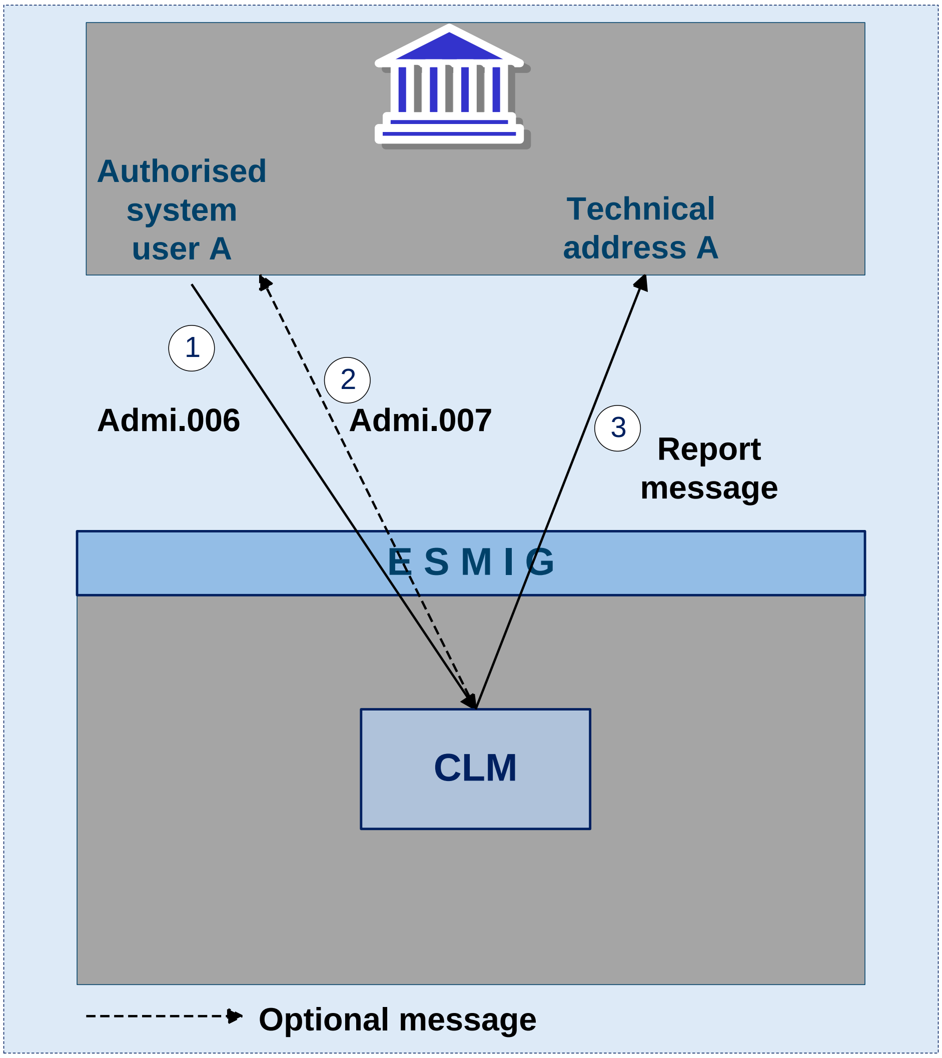


Figure 27 - camt.007 - amendment positive

Process description

Table 43 - Resend request with positive validation and re-delivery

Step	Processing in/between	Description
1	CLM participant via ESMIG to CLM	An authorised system user of a CLM participant A sends a admi.006 via ESMIG to CLM.
2	CLM	CLM message check and validation positive
3	CLM via ESMIG to CLM participant	Admi.007 including positive validation result via ESMIG to CLM participant A generated by CLM (optional)
4	CLM via ESMIG to CLM participant	Re-delivery of report message camt.053 to the original technical address (mandatory)

Used messages

- BankToCustomerStatement (camt.053) [▸ 258]
- ResendRequest (admi.006) [▸ 222]
- ReceiptAcknowledgement (admi.007) [▸ 224]

Case: Resend request with negative validation

Message flow

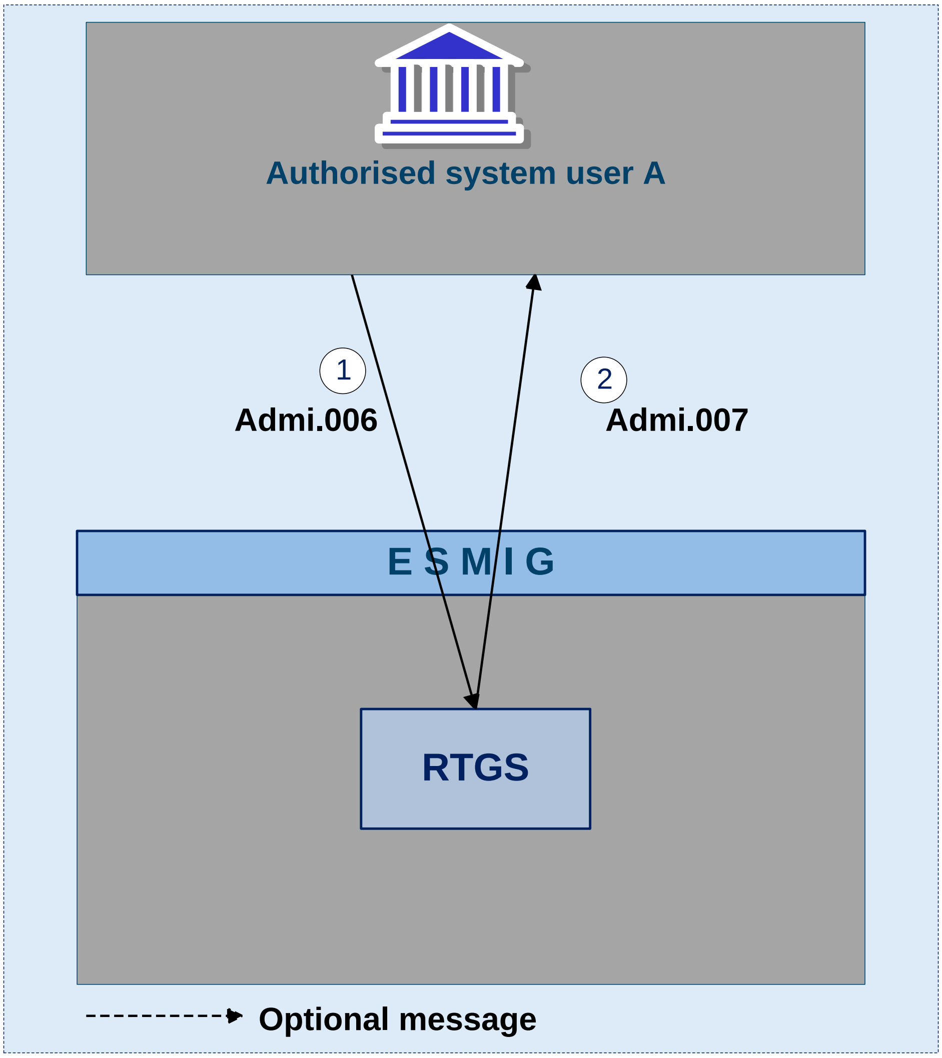


Figure 28 - camt.007 - amendment negative

Process description

Table 44 - Resend request with negative validation

Step	Processing in/between	Description
1	CLM participant via ESMIG to RTGS	An authorised system user of a CLM participant A sends a admi.006 via ESMIG to CLM.
2	CLM	CLM message check and validation negative
3	CLM via ESMIG to CLM participant	Admi.007 including negative validation result via ESMIG to CLM participant A generated by CLM (mandatory)

Used messages

- ResendRequest (admi.006) [▸ 222]
- ReceiptAcknowledgement (admi.007) [▸ 224]

5.5.3 Query management for CLM, CRDM, scheduler and billing (completed)

5.5.3.1 Concept for CLM, CRDM, scheduler and billing (completed)

Queries are provided by CLM, CRDM, scheduler and billing to the submitting actor as a means of satisfying his information needs on demand. The submitting actor can obtain information on different business items by submitting query requests to the mentioned components. These are answered on the basis of the latest data available.

For requests on CLM only queries using the specified (optional and mandatory) search and return criteria are available. Thus actors are not able to define these criteria by themselves.

The respective business process is described in chapter Execute query [▸ 195].

5.5.3.2 Overview for CLM, CRDM, scheduler and billing (completed)

CLM, CRDM, Scheduler and Billing provide a range of predefined query types, which the submitting actor can use to request information on business items. The offered queries are available for all authorised submitting actors of the respective service/component.

They can send query requests to components in A2A mode or in U2A mode. Generally, all these query requests are processed in real time. Exceptions occur during the maintenance window. During the maintenance window query management does not service any requests. In case ESMIG is available and the network interface is not closed, an A2A query request during maintenance window is handled by using timeout management. In case the network interface is closed the network service provider informs the authorised submitting actor about the closure of the real-time channel.

5.5.3.3 Query management process for CLM, CRDM, scheduler and billing (completed)

Initiating queries for CLM, CRDM, scheduler and billing

In order to obtain the desired information the submitting actor needs to submit a query request to a component. For the communication with components in A2A mode all query and response messages are set up as XML messages compliant with the ISO20022 standard. For the communication with components in U2A mode a graphical user interface based on a standard browser application is provided.

In general an authorised submitting actor can send each query request in A2A mode as well as in U2A mode. However, there are some queries which are only accessible via U2A mode. Query availability in the respective communication mode is shown in the table below. Query request and return criteria are described in detail in CLM UHB for U2A mode and in chapter 11 with link to MyStandards for A2A mode.

Table 45 - Initiating queries

Related component	Query type	Initiation via GUI (U2A mode)	Initiation via XML message (A2A mode)
CLM	Account Statement Query [7]	X	-
CLM	Audit Trail for CLM Query	X	X
CLM	Available Liquidity CLM Query	X	X
CLM (- overall)	Available Liquidity Overall Query	X	-
CLM	Broadcast Query	X	-
CLM	CLM Payment Order Query	X	X
CLM	Message Query	X	-
CLM	Current Reservations Query	X	X
CLM	Minimum Reserve Query	X	X
CRDM	Audit Trail for CRDM Query	X	X
CRDM	Calendar Query	X	X
CRDM	Central Bank Query	X	-
CRDM	Direct Debit Mandate Query	X	X
CRDM	Directory Query	X	X
CRDM	Error Code Query	X	X
CRDM	Event Query	X	X
CRDM	MCA Reference Data Query	X	X
CRDM	Message Subscription Query	X	-
CRDM	Liquidity Transfer Group Query	X	-
CRDM	Participant Reference Data Query	X	X

[7] A request for delivery of account statement in A2A is implemented via the resend request, i.e. admi.006.”

Related component	Query type	Initiation via GUI (U2A mode)	Initiation via XML message (A2A mode)
CRDM	Party Reference Data Query	X	X
CRDM	Role Query	X	-
CRDM	Standing Order Liquidity Transfer Query	X	X
CRDM	Standing Order Reservations Query	X	X
CRDM	User Query	X	-
Scheduler	System Time Query	X	X
Billing	VAT Query	X	-
Billing	Invoice Query	X	-

The different types of queries in components are static regarding the set of selection parameters, which can be mandatory, optional or conditional.

Preconditions for successful processing of queries

The relevant component validates the plausibility of search criteria that were specified by the submitting actor. In addition, the relevant component ensures that the submitting actor of the query request is allowed to initiate the query and to retrieve the requested data by checking, whether the submitting actor possesses all necessary privileges granted in advance (taking into account the validity dates) and ensuring the data scope.

Providing data for queries

If all checks performed by respective component were successful, it extracts the requested business information from the production data. The submitting actor receives the latest available data. If one or more of plausibility or authorisation checks performed by respective component fail, the submitting actor receives a response indicating the error that has occurred which is specified using the respective error code.

Retrieving the query response

In case the extraction of the query data is successful, the respective component sends a query response containing the requested business information back to the requesting actor. In case the extraction of the query data returns a zero result, the submitting actor receives appropriate information. If a retrieval of the query result fails, then an error response is provided to the submitting actor.

If the submitting actor sends the query via U2A mode, the response is given to the submitting actor in U2A mode. The U2A dialogue is described more in detail in the CLM UHB.

If the submitting actor sends the query via A2A mode, the response is given to the same submitting actor in A2A mode. The respective component does not allow the routing of the query response to a dedicated technical address.

Parameter synthesis

No specific query configuration from the submitting actor is needed.

6 Overview of used common components in CLM component

6.1 CRDM features (completed)

6.1.1 Concept (completed)

The CRDM common component allows duly authorised users to create and maintain reference data objects. CRDM objects specify reference data for the configuration of parties, cash accounts and rules and parameters.

6.1.2 Overview (completed)

The CRDM common component is in charge of executing reference data maintenance instructions for the creation or the maintenance of reference data objects.

Duly authorised users belonging to central banks, payment banks and to the operator can trigger CRDM according to their own specific access rights, i.e. using the functions and maintaining the common reference data objects they have been granted.

Duly authorised users of the operator are responsible for system configuration tasks and for the management of common reference data for central banks. These users can also act on behalf of other CRDM actors in order to perform some specific actions or within some pre-defined contingency scenarios.

CRDM common component executes immediately all reference data maintenance instructions. The related reference data changes become effective in the relevant TARGET service(s), common component(s) or back-office applications in a deferred way, by means of a daily reference data propagation process. The process takes place every business day and is scheduled in order to ensure a smooth and complete reference data propagation depending on the operational schedule of the relevant service(s).

All common reference data objects can be created and maintained in U2A mode, whereas only a sub-set of them can be maintained also through the DMT (see chapter Reference data maintenance types [▸ 158]). All reference data changes performed in U2A mode can be executed either in two-eyes or in four-eyes mode. Duly authorised actors can specify the applicable mode for the functions and the common reference data objects they manage (see chapter Access rights [▸ 127]).

Versioning facilities and validity periods allow the implementation of data revision and data history features, in order to keep track of all past data changes, to enter changes meant to become effective as of a future date and to define common reference data objects with limited or unlimited validity.

6.1.3 Access rights (completed)

This section provides information on access rights management in the CRDM. More into detail, chapter Access rights concepts [▸ 127] presents some basic concepts (e.g. user, privilege, role and data scope) related to access rights management. On this basis, chapter Access rights configuration [▸ 143] illustrates all the available options for the configuration of access rights. Finally, chapter Access rights configuration process [▸ 151] describes the access rights configuration process that each type of CRDM actor has to put in place in order to set up the appropriate assignment of roles and privileges for all its users.

6.1.3.1 Access rights concepts (completed)

This chapter presents the main concepts related to access rights management in the CRDM.

6.1.3.1.1 User function (completed)

Data migration tool files, XML messages and GUI functions are the atomic elements users can trigger through the data migration tool and in A2A and U2A mode respectively to interact with CRDM as well as other services, common components or back-office applications. Based on these set of files, XML messages and GUI functions, it is possible to define the set of all user functions, i.e. of all the possible actions that a user can trigger in CRDM or other services, common components or back-office application services, either in the DMT or in A2A or U2A mode.

6.1.3.1.2 Privilege (completed)

A privilege identifies the capability of triggering one or several user functions and it is the basic element to assign access rights to users. This means that a user U_X owns the access right to trigger a given user function F_Y if and only if U_X was previously granted with the privilege P_Y identifying the capability to trigger F_Y.

The following tables provide the exhaustive list of privileges covering all the user functions available:

- table access rights management
- table party data management
- table cash account data management
- table message subscription configuration
- table report configuration
- table reference data queries
- table TIPS functions
- table other

Table 46 - Access rights management

Privilege	User function	Data scope
Administer party [8]	n/a	n/a
Create certificate distinguish name	Certificate DN – new	Any certificate DN
Create DN-BIC routing	DN-BIC routing - new	DN-BIC routing data within own system entity (for central banks) or for DNs linked to own users and BICs authorised to own cash accounts (for payment banks).
Create role	Role – new	Roles within own system entity (for central banks).
Create user	User – new	Users within own system entity (for central banks) or own party (for payment banks).
Create user certificate distinguish name link	User certificate DN link – new	Links within own system entity (for central banks) or for own users (for payment banks).
certificate distinguish name	Certificate DN – delete/restore	Any certificate DN
Delete DN-BIC routing	DN-BIC routing - delete/restore	DN-BIC routing data within own system entity (for central banks) or for DNs linked to own users and BICs authorised to own cash accounts (for payment banks).
Delete role	Role – delete/restore	Roles within own system entity (for central banks).
Delete user	User – delete/restore	Users within own system entity (for central banks) or own party (for payment banks).
Delete user certificate distinguish name link	User certificate DN link – delete/restore	Links within own system entity (for central banks) or for own users (for payment banks).

[8] This privilege enables a user to act as party administrator for their own party.

Privilege	User function	Data scope
Grant privilege	Grant privilege	Privileges granted to parties, roles and users within own system entity (for central banks) or to own users (for payment banks)
Grant/revoke role	Grant/revoke role	Roles granted to parties and users within own system entity (for central banks) or to own users (for payment banks)
Revoke privilege	Revoke privilege	Privileges granted to parties, roles and users within own system entity (for centrals) or to own users (for payment banks)
Update DN-BIC routing	DN-BIC routing - edit	DN-BIC routing data within own system entity (for central banks) or for DNs linked to own users and BICs authorised to own cash accounts (for payment banks).
Update role	Role – edit	Roles within own system entity (for central banks)
Update User	User – edit	Users within own system entity (for central banks) or own party (for payment banks).

Table 47 - Party data management

Privilege	User function	Data scope
Create banking group	Banking group – new	Banking groups within own system entity (for central banks)
Create monetary financial institution	Monetary financial institution – new	Monetary financial institutions within own system entity (for central banks)
Create party	Party – new	Parties within own system entity (for central banks)
Create party-service link	Party-service link - new	Links within own system entity (for central banks)

Privilege	User function	Data scope
Create technical address network service link	Technical address network service link - new	Links within own system entity (for central banks)
Delete banking group	Banking group – delete/restore	Banking groups within own system entity (for central banks)
Delete monetary financial institution	Monetary financial institution – delete/restore	Monetary financial institutions within own system entity (for central banks)
Delete party	Party – delete/restore	Parties within own system entity (for central banks) excluding own party
Delete party-service link	Party-service link - delete/restore	Links within own system entity (for central banks)
Delete technical address networks service link	Technical address network service link - delete/restore	Links within own system entity (for central banks)
Update banking group	Banking group – edit	Banking groups within own system entity (for central banks)
Update monetary financial institution	Monetary financial institution – edit	Monetary financial institutions within own system entity (for central banks)
Update party	Party – edit	Parties within own system entity (for central banks)
Update party-service link	Party-service link - edit	Links within own system entity (for central banks)

Table 48 - Cash account data management

Privilege	User function	Data scope
Create account monitoring group	Account monitoring group – new	Account monitoring groups within own system entity (for central bank)
Create authorised account user	Authorised account user - new	Links within own system entity (for central bank) or for own cash accounts (for payment bank).
Create cash account	Cash account – new	Cash accounts within own system entity (for central bank) or CMBs linked to cash accounts owned by own party (for payment bank)

Privilege	User function	Data scope
Create direct debit mandate	Direct debit mandate - new	Direct debit mandates on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Create limit	Limit – new	Limits on CMBs defined on cash accounts within own system entity (for central bank) or linked to cash accounts owned by own party (for payment bank)
Create liquidity transfer order	Liquidity transfer order – new	Liquidity transfer orders on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Create liquidity transfer order group	Liquidity transfer order group – new	Liquidity transfer order groups containing liquidity transfer orders on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Create standing order for limit	Standing order for limit – new	Standing orders for limit on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Create standing order for reservation	Standing order for reservation – new	Standing orders for reservation on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Delete account monitoring group	Account monitoring group – delete/restore	Account monitoring groups within own system entity (for central bank)
Delete authorised account user	Authorised account user - delete/restore	Links within own system entity (for central bank) or for own cash accounts (for payment bank).
Delete cash Account	Cash account – delete/restore	Cash accounts within own system entity (for central bank) or CMBs linked to cash accounts owned by own party (for payment bank)

Privilege	User function	Data scope
Delete direct debit mandate	Direct debit mandate – delete/restore	Direct debit mandates on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Delete limit	Limit – delete/restore	Limits on CMBs defined on cash accounts within own system entity (for central bank) or linked to cash accounts owned by own party (for payment bank)
Delete liquidity transfer order	Liquidity transfer order – delete/restore	Liquidity transfer orders on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Delete liquidity transfer order group	Liquidity transfer order group – delete/restore	Liquidity transfer order groups containing liquidity transfer orders on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Delete standing order for limit	Standing order for limit – delete/restore	Standing orders for limit on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Delete standing order for reservation	Standing order for reservation – delete/restore	Standing orders for reservation on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Update account monitoring group	Account monitoring group – edit	Account monitoring groups within own system entity (for central bank)
Update authorised account user	Authorised account user - edit	Links within own system entity (for central bank) or for own cash accounts (for payment bank).
Update cash account	Cash account – edit	Cash accounts within own system entity (for central bans) or CMBs linked to cash accounts owned by own party (for payment bank)

Privilege	User function	Data scope
Update direct debit mandate	Direct debit mandate – edit	Direct debit mandates on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Update limit	Limit – edit	Limits on CMBs defined on cash accounts within own system entity (for central bank) or linked to cash accounts owned by own party (for payment bank)
Update liquidity transfer order	Liquidity transfer order – edit	Liquidity transfer orders on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Update liquidity transfer order group	Liquidity transfer order group – edit	Liquidity transfer order groups containing liquidity transfer orders on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Update standing order for limit	Standing order for limit – edit	Standing orders for limits on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Update standing order for reservation	Standing order for reservation – edit	Standing orders for reservation on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)

Table 49 - Message subscription configuration

Privilege	User function	Data scope
Create message subscription rule	Message subscription rule – new	Message subscription rules within own system entity (for central banks) or for own party (for payment banks)
Create message subscription rule set	Message subscription rule set – new	Message subscription rule sets within own system entity (for central banks) or for own party (for payment banks)

Privilege	User function	Data scope
Delete message subscription rule	Message subscription rule – delete/restore	Message subscription rules within own system entity (for central banks) or for own party (for payment banks)
Delete message subscription rule set	Message subscription rule set – delete/restore	Message subscription rule Sets within own system entity (for central banks) or for own party (for payment banks)
Update message subscription rule	Message subscription rule – edit	Message subscription rules within own system entity (for central banks) or for own party (for payment banks)
Update message subscription rule set	Message subscription rule set – edit	Message subscription rule sets within own system entity (for central banks) or for own party (for payment banks)

Table 50 - Report configuration

Privilege	User function	Data scope
Create report configuration	Report configuration – new	Report configurations within own system entity (for central banks) or for own party (for payment banks)
Delete report configuration	Report configuration – delete/restore	Report configurations within own system entity (for central banks) or for own party (for payment banks)
Update report configuration	Report configuration – edit	Report Configurations within own system entity (for central banks) or for own party (for payment banks)

Table 51 - Reference data queries

Privilege	User function	Data scope
Account monitoring group query	Account monitoring group – list	Account monitoring group
Authorised account user query	Authorised account user – list	Links within own system entity (for central banks) or for own cash accounts (for payment banks).
Banking group query	Banking group – list	Any banking group
BIC query	BIC query	Any BIC

Privilege	User function	Data scope
Cash account audit trail query	Revisions - selection criteria + list	Data within own system entity (for central bank) or linked to own party (for payment bank)
Cash account list query	Cash account list query	Cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Cash account reference data query	Cash account reference data query	Cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Certificate query	Certificate query	Any certificate DN
Country query	Countries – select + list	Any country
Currency query	Currencies – select + list	Any currency
Data changes of a business object details query	Data changes of a business object details query	Data within own system entity (for central banks) or linked to own party (for payment banks)
Data changes of a business object list query	n/a	Data within own system entity (for central banks) or linked to own party (for payment banks)
Direct debit mandate details query	Direct debit mandate – details	Direct debit mandates on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Direct debit Mandate List query	Direct debit mandate – list	Direct debit mandates on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Granted roles list query	Granted roles – search	Roles granted to parties and users within own system entity (for central banks) or to own users (for payment banks)
Granted roles list query	Grant/revoke role – details	Roles granted to parties and users within own system entity (for central banks) or to own users (for payment banks)

Privilege	User function	Data scope
Granted system privileges list query	Grant/revoke system privileges list query	Privileges granted to parties, roles and users within own system entity (for central banks) or to own users (for payment banks)
Limit query	Limit query	Limits on CMB defined on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Liquidity transfer order details query	Liquidity transfer order – details	Liquidity transfer orders on sash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Liquidity transfer order list query	Liquidity transfer order – list	Liquidity transfer orders on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Liquidity transfer order group query	Liquidity transfer order group – list	Liquidity transfer order groups within own system entity (for central bank) or containing cash accounts owned by own party (for payment bank)
Market-specific restriction list query	Market-specific restriction list query	Restrictions defined by the operator
Market-specific restriction type rule detail query	Market-specific restriction type rule – detail query	Restrictions defined by the operator
Market-specific restriction type rule parameter details query	Market-specific restriction type rule parameter details query	Restrictions defined by the operator
Market-specific restriction type rule set list query	Market-specific restriction type Rule set list query	Restrictions defined by the operator
Message subscription rule list query	Message subscription rule list query	Message subscriptions within own system entity (for central banks) or for own party (for payment banks)
Message subscription rule set details query	Message subscription rule sets details query	Message subscriptions within own system entity (for central banks) or for own party (for payment banks)

Privilege	User function	Data scope
Message subscription rule set list query	Message subscription rule set list query	Message subscriptions within own system entity (for central banks) or for own party (for payment banks)
Monetary financial institution query	Monetary financial institution – list	Any monetary financial institution
Network service list query	Network service list query	Any network service
Party audit trail query	Static data audit trail query	Data within own system entity (for central bank) or linked to own party (for payment bank)
Party list query	Party list query	Parties within own system entity (for central bank) or own party (for payment bank)
Party reference data query	Party reference data query	Parties within own system entity (for central bank) or own party (for payment bank)
Party-service link list query	Party-service link list query	Links within own system entity (for central banks) or linked to own party (for payment banks)
Party-service link query	Party-service link query	Links within own system entity (for central banks) or linked to own party (for payment banks)
Privilege query	Privilege – selection criteria + list	Any privilege
Queued data changes query	Queued data changes – select + list	Data within own system entity (for central banks) or linked to own party (for payment banks)
Report configuration details query	Report configuration details query	Report configurations within own system entity (for central banks) or for own carty (for payment banks)
Report configuration list query	Report configuration list query	Report configurations within own system entity (for central banks) or for own party (for payment banks)
Residual static data audit trail query	Static data audit trail query	Data within own system entity (for central banks) or linked to own party (for payment banks)

Privilege	User function	Data scope
Role list query	Role list query	Roles created or granted to parties and users within own system entity (for central banks) or to own users (for payment banks)
Service list query	Service list query	Any service
Standing order for limit details query	Standing order for limit – details	Standing orders for limit on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Standing order for limit list query	Standing order for limit – list	Standing orders for limit on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Standing order for reservation details query	Standing order for reservation – details	Standing orders for reservation on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
Standing order for reservation list query	Standing order for reservation – list	Standing orders for reservation on cash accounts within own system entity (for central bank) or owned by own party (for payment bank)
System entity query	System entities – select + list	Own system entity (for central banks)
System user link query	System user link query	Links within own system entity (for central banks) or linked to own users (for payment banks)
Technical address network service link details query	Technical address network service link details query	Links within own system entity (for central banks) or linked to own party (for payment banks)

Table 52 - TIPS functions

Privilege	User function	Data scope
Adjust CMB limit	Adjust CMB limit	Data within own system entity (for central bank) or linked to own party (for payment bank)
Instruct instant payment	Initiate instant payment Confirm/reject instant payment Request instant payment recall Confirm instant payment recall Reject instant payment recall Instant payment status investigation	Data related to accounts within own system entity (for central bank) or for which own party is set as authorised user (for payment bank)
Instruct liquidity transfer	Initiate outbound liquidity transfer	Accounts within own system entity (for central bank) or owned by own party (for payment bank)
Modify all blocking status	Block/unblock participant Block/unblock account Block/unblock CMB	Data within own system entity (for central bank) or linked to own party (for payment bank)
Modify CMB blocking status	Block/unblock CMB	Data within own system entity (for central bank) or linked to own party (for payment bank)
Query all	Query account balance and status Query CMB limit and status Query instant payment transaction	Data related to accounts within own system entity (for central bank) or owned by own party (for payment bank)
Query as reachable party	Query CMB limit and status Query instant payment transaction	Data related to accounts within own system entity (for central bank) or for which own party is set as authorised user (for payment bank)

Table 53 - Other

Privilege	User function	Data scope
Data migration tool access	n/a	n/a

See chapter Configuration of privileges [▸ 143] for information on the configuration of privileges.

6.1.3.1.3 Role (completed)

A role is a set of privileges. See chapter Configuration of roles [▸ 150] for information on the configuration of roles.

6.1.3.1.4 User (completed)

A user is an individual or application that interacts with CRDM triggering the available CRDM user functions. See chapter Configuration of users [▸ 143] for information on the configuration of users.

6.1.3.1.5 Common reference data objects and the hierarchical party model (completed)

All parties in the CRDM are linked to each other according to a hierarchical model. As shown in the following diagram and on the basis of this hierarchical party model, the operator is the only party at level 1, all the central banks are level 2 parties, all payment banks are level 3 parties [9]. All the other reference data objects are linked to a party. For example:

- a cash account is linked to its central bank or payment bank.

[9] Participation types may be further detailed with information specific to each individual service, if the service foresees this possibility.

- a restriction type is linked to the operator.

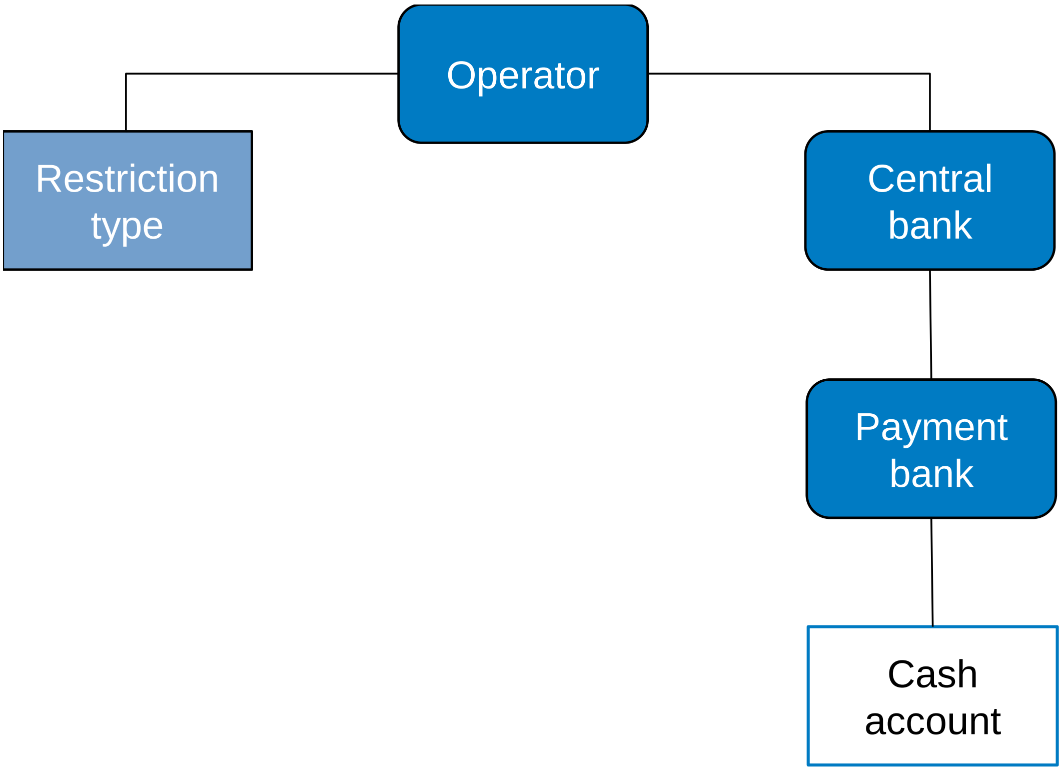


Figure 29 - Common reference data objects and the hierarchical party model

6.1.3.1.6 Data scope (completed)

For each privilege, the hierarchical party model determines the data scope of the grantee, i.e. the set of reference data objects on which the grantee can trigger the relevant user function. More precisely:

- users of the operator have visibility on all reference data objects and can act on objects belonging to participants only in exceptional circumstances, following a specific agreement;
- users of the central banks have visibility on all reference data objects belonging to the same system entity [10];
- users of the payment banks have visibility on reference data objects that are (directly or indirectly) linked to the same party.

The following example describes the concept of data scope [11].

[10] A system entity in CRDM corresponds to a partition of data equating to the scope of a central bank or of the operator. For example, the system entity of a central bank includes all the data related to its payment banks.

[11] The following example presents only the configuration data that are relevant for the example. All the possible configuration options are defined in the following sections.

Example – data scope

Three users, X, Y and Z, belonging to a payment bank, to a central bank and to the operator respectively, are granted with the same privilege to query cash accounts:

Table 54 - User privileges (data scope)

User	Privilege
X	Cash account reference data query
Y	Cash account reference data query
Z	Cash account reference data query

The following diagram shows the data scopes stemming from this access rights configuration for the three users.

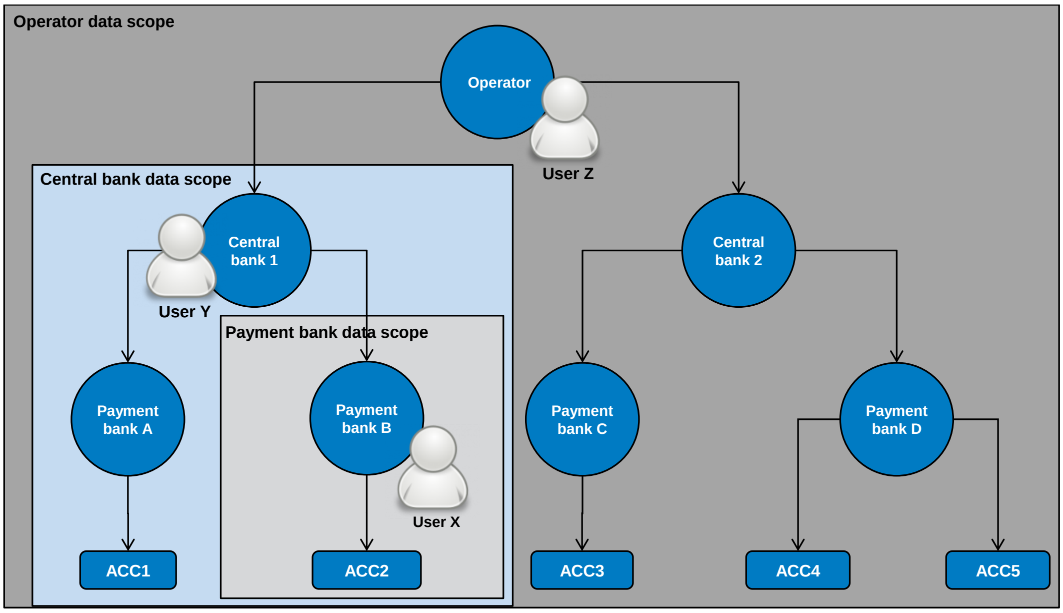


Figure 30 - Data scopes

The diagram shows that users X, Y and Z are given different data scopes, owing to the fact that they belong to different parties located at different levels of the hierarchical party model. More precisely:

- User X of payment bank B gets a data scope including the cash account ACC2 only, as ACC2 is the only account of payment bank B. User X cannot query any other cash account in CRDM.
- User Y of central bank 1 gets a data scope including cash accounts ACC1 and ACC2, as these accounts belong to payment banks of central bank 1. User Y cannot query any other cash account in CRDM, i.e. any cash account falling under the data scope of any other central bank.
- User Z of the operator gets a data scope including all cash accounts in CRDM, as the operator is at the top level of the hierarchical party model.

6.1.3.2 Access rights configuration (completed)

This section presents how roles and privileges can be configured in CRDM in order to grant each user with the appropriate set of access rights.

6.1.3.2.1 Configuration of users (completed)

Links between users and parties

Each new user is linked to the same party which the creator user belongs to. An exception takes place when creating the first user of a party, i.e.

- when a CRDM operator system administrator creates a new system administrator for a central bank
- when a central bank system administrator creates a new system administrator for one of its payment banks

In all these cases the created user is linked to the party this user is going to administer.

Through the link with the relevant party, each user inherits a data scope (see chapter Data scope [▸ 141]).The link between a user and a party cannot be changed, i.e. a user is always linked to the same party.

Party administrators

Each party must have at least one party administrator, i.e. a user being granted specific system privileges that allow its grantee to grant any roles and privileges previously granted to the grantee's party.

6.1.3.2.2 Configuration of privileges (completed)

Availability of privileges

Each privilege, just after its creation, is available to the party administrator(s) of the operator only. This means that party administrators of all the other parties cannot grant this privilege to their users.

A privilege becomes available to a party administrator of a party different from the operator only after this privilege has been granted to this party. From this moment on, the party administrator can grant this privilege, according to the rules defined in the following sections.

This implies that a two-step process is required in order to grant a specific privilege to a user belonging to a party different from the operator. In the first step, the privilege is granted to the relevant party (so that it becomes available to the party administrator(s) of this party). With the second step, one of the party administrators grants the privilege to the relevant user.

The following diagram illustrates the access rights configuration steps needed to grant a user Z of a party B a given privilege P that is already available to the party administrator X of another party A. [12]

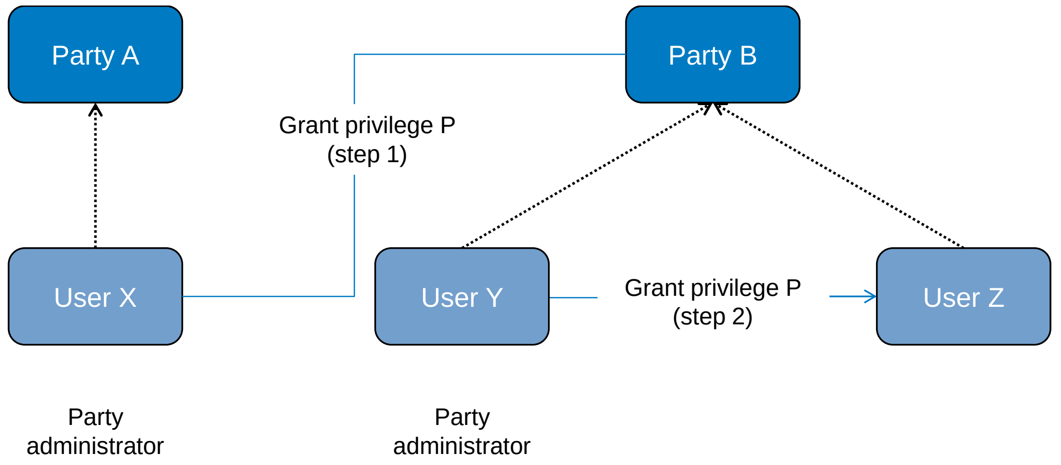


Figure 31 - Access rights configuration steps

The two configuration steps are as follows:

- User X, as a party administrator of party A, grants privilege P to party B. From this moment on, privilege P becomes available to the party administrator Y of party B.
- User Y, as a party administrator of party B, grants privilege P to user Z. From this moment on, user Z can trigger the user functions linked to privilege P.

At Party level, access rights are propagated following the hierarchical party model, i.e. the operator propagates access rights to central banks which in turn propagate them to their payment banks. If necessary, the operator can act on behalf of a central bank following a specific request to propagate access rights directly to its payment banks.

While the features described above apply to all privileges related to CRDM functions, it should be noted that TIPS privileges cannot be granted directly to parties or users, but can only be granted to roles, which can in turn be granted to parties and users. This implies that the above described configuration steps remain valid for TIPS as well, but in this case privileges have to be granted to roles in the first place and then roles can be granted to parties and users. For details on the configuration of roles see chapter Configuration of roles [▸ 150].

Granting privileges

Most privileges can be granted to roles, users and parties, with the exception of TIPS privileges that can be granted to roles only. When granting a privilege, the grantor specifies appropriate values for the three following assignment options: deny option, administration option and four-eyes option.

[12] Party A may be the operator or any other party which was previously granted privilege P.

Table 55 - Privilege assignment options

Option	Description
Deny	This option specifies whether the associated user function is allowed (deny is false) or explicitly denied (deny is true).
Administration	If the grantee of the privilege is a user or a role, this option specifies whether the grantee is allowed to grant the same privilege to another user or role of the same party (administrator is true) or not (administrator is false). If the grantee of the privilege is a party, this option specifies whether the party administrators of the grantee party is allowed to grant the same privilege only to users and roles of the same party (administrator is false) or also to other parties (administrator is true).
Four-eyes	This option specifies whether the grantee of the privilege is allowed to use the function associated to the privilege according to the two-eyes (four-eyes is false) or four-eyes (four-eyes is true) principles. This option is relevant only when the deny option is set to false and it is always not relevant for privileges related to queries.

Example - assignment of privileges to roles

The following table shows some examples of assignment of privileges to roles:

Table 56 - Assignment of privileges to roles

Row	Role	Privilege	Deny	Admin	Four-eyes
1	Cash account management	Cash account reference data query	False	False	Not relevant
2	Cash account administration	Cash account reference data query	True	True	Not relevant
3	Party management	Create party	False	False	True
4	Party management	Update party	False	False	True
5	Party management	Delete party	False	False	True
6	Party management	Party reference data query	False	True	Not relevant

For each assignment of a privilege to a role, three additional attributes define the features of such assignment.

For example, according to row 1, the privilege to query cash account data is assigned to the cash account management role:

- without deny, i.e. users linked to the cash account management role can query cash account data [13];
- without admin, i.e. users linked to the cash account management role cannot grant the privilege to query cash account data to other roles and users.

According to row 2, the privilege to query cash account data is assigned to the cash account administration role.

- with deny, i.e. users linked to the cash account administration role cannot query cash account data;
- with admin, i.e. users linked to the cash account administration role can grant the privilege to query cash account data to other roles and users of the same party.

As a whole, rows 1 and 2 result in a segregation of duties between business users and access rights administrators. In fact, users linked to the cash account management role can query accounts, but they cannot configure the same access rights for any other user. On the contrary, users linked to the cash account administration role cannot query accounts, but they can configure these access rights for other users.

According to row 3, the privilege to create parties is assigned to the party management role:

- without deny and with four-eyes set to true, i.e. users linked to the party management role can create parties according to the four-eyes principle only;
- without admin, i.e. users linked to the party management role cannot grant the privilege to create parties to other roles and users.

As per rows 4 and 5, the privileges to maintain and delete parties are assigned to the party management role with the same assignment options.

Finally, according to row 6, the privilege to query parties is assigned to the party management role:

- without deny, i.e. users linked to the party management role can query parties;
- with admin, i.e. users linked to the party management role can grant the privilege to query parties to other roles and users of the same party.

As a whole, rows from 3 to 6 only result in a partial segregation of duties between business users and access rights administrators. In fact:

- business users linked to the party management role can create, maintain, delete and query parties, they can only configure the same access rights for any other user limited to the query privilege;

[13] In this case the setting for the four eyes assignment option is not applicable, as the privilege refers to a query.

- on the contrary, access rights administrators linked to the party management role, and whose party is also linked to the same role, can create, maintain, delete and query parties and they can also grant the same privilege to other users of the same party; in addition, they can also grant the query privilege to other parties.

Example - assignment of privileges to users

The following table shows two examples of assignment of privileges to users:

Table 57 - Assignment of privileges to users

Row	Privilege	User	Deny	Admin	Four-eyes
1	Create cash account	U_X	False	False	False
2	Create cash account	U_Y	True	True	False

For each assignment of a privilege to a user, three additional attributes define the features of such assignment.

According to row 1, the privilege to create cash accounts is assigned to user U_X:

- without deny, i.e. user UX can create cash accounts according to the two-eyes principle (as the privilege is assigned without four-eyes);
- with admin, i.e. user UY can grant the privilege to create cash accounts to other roles and users of the same party, according to the two-eyes principle or to the four-eyes principle (as the privilege is assigned without four-eyes).

Similarly, row 2 stipulates that the privilege to create cash accounts is assigned to user U_Y:

- with deny, i.e. user UY cannot create cash accounts;
- with admin, i.e. user UY can grant the privilege to create cash accounts to other roles and users of the same party, according to the two-eyes principle or to the four-eyes principle (as the privilege is assigned without four-eyes).

As a whole, this configuration results in a full segregation of duties between business users and access rights administrators. In fact, user UX can create cash accounts, but without having the possibility to grant the same privilege to any other user. Vice versa, user UY can configure this privilege for other users, but without having the possibility to use it.

Example - assignment of privileges to parties

The following table shows one example of assignment of a privilege to a party:

Table 58 - Assignment of privileges to parties

Privilege	Party	Deny	Admin	Four-eyes
Cash account reference data query	Payment bank A	False	True	False

For each assignment of a privilege to a party, three additional attributes define the features of such assignment. In this example, the privilege to query cash accounts is assigned to the payment bank A:

- without deny, i.e. party administrators of the payment bank A can grant the privilege to query cash accounts to other roles and users of the same party;
- with admin, i.e. party administrators of the payment bank A can grant the privilege to query cash accounts to other parties.

The four-eyes attribute is set to false but it is not relevant for this example, as the privilege refers to a query.

Revoking privileges

Privileges can be revoked from roles, users and parties. When revoking a privilege from the user, this just results in the removal of the privilege from the list of privileges linked to the user. When revoking a privilege from a role, this results in the removal of the privilege from the list of privileges linked to the role. Consequently, all the users and parties linked to the role are not linked anymore to the privilege, with immediate effect. When revoking a privilege from a party, CRDM applies a cascade effect. This results in the removal of the privilege:

- from the list of privileges linked to the party and
- from the list of privileges linked to all the roles and users of the party

The following table shows all the possible scenarios for revoking privileges that are allowed in CRDM, their link with the cascade process and how party administrators of central banks can ensure that all the privileges revoked from one of their parties are revoked also from all the users of the same party:

Table 59 - Cascade process when revoking privileges

Function	From	Cascade	Propagation to user
Revoke privilege	User	n/a	As the grantee is already a user, there is no need to trigger any cascade process.
Revoke privilege	Role	n/a	If the party administrator of the payment bank granted a privilege included in the role directly to other users of the payment bank, then the removal of this privilege from the role would not revoke the same privilege from these users. In fact, when revoking a privilege from a role, CRDM does not trigger the cascade process as this may result in unintended removal of privileges from the users of the payment bank. For example, even a simple movement of a privilege between two roles assigned to the same payment bank (i.e. revoking the privilege from the first role and granting it to the latter) would imply the removal of the same privilege from all the users of this payment bank and this would oblige the party administrator of the payment bank to grant again this privileges to all the impacted users. In order to ensure that the relevant privilege is revoked also from the users of the payment bank (if this is the intended goal), the party administrator of the central bank should grant directly this privilege to the payment bank and then revoke it, as this triggers the cascade process related to the revoke privilege function from party (see next row of this table).
Revoke privilege	Party	Yes	CRDM triggers automatically the cascade process, which ensures that privileges revoked from a party are also revoked from all the users and roles of the same party.

The cascade process is automatically triggered in a deferred mode one time per business day. However, in case the party administrator needs the cascade process to take place immediately, this can be achieved by contacting the operator, as the operator can trigger this process on demand also intraday.

6.1.3.2.3 Configuration of roles (completed)

Links between roles

CRDM supports a role-based access control (RBAC) model. This results in the possibility to inherit privileges from one or more roles.

Granting roles

Roles can be granted to users and parties. When granting a role to a user, the grantee user immediately inherits all the privileges of the granted role, i.e. all the privileges linked to the granted role. When granting a role to a party, the grantee party immediately inherits all the privileges of the granted role, i.e. all the privileges linked to the granted role.

Revoking roles

Roles can be revoked from users and parties. When revoking a role from a user, this user immediately loses all the privileges of the revoked role, i.e. all the privileges linked to the revoked role. When revoking a role from a party, this party immediately loses all the privileges of the revoked role, i.e. all the privileges linked to the revoked role. Both when revoking roles from users and from parties, CRDM does not apply a cascade effect. The following table shows all the possible scenarios for revoking roles that are allowed in CRDM, their link with the cascade process and how party administrators of central banks can ensure that all the roles revoked from one of their parties (and all the privileges included in these roles) are revoked also from all the users of the same party:

Table 60 - Cascade process when revoking roles

Function	From	Cascade	Propagation to user
Revoke role	User	n/a	As the grantee is already a user, there is no need to trigger any cascade process.
Revoke role	Party	n/a	If the party administrator of the payment bank granted the role (or a privilege included in the role) directly to other users of the payment bank, then the removal of this role from the party would not revoke the same role (or the privilege included in the role) from these users. In fact, when revoking a role from a party, CRDM does not trigger the cascade process as this may result in unintended removal of roles (or privileges) from the users of the payment bank. In order to ensure that the relevant role is revoked also from the users of the payment bank, the party administrator of the central bank should revoke all the privileges included in the role from the role itself and then delete the role. It should be noted that this approach can be applied without unintended side effects on other payment banks only if the role was specifically created for (and assigned to) the relevant payment bank only, otherwise the procedure just described would also have an effect on all payment banks (and on all their users) being granted with the same role. Furthermore, in order to ensure that any privilege belonging to the role and that was granted directly to users of the payment bank is also revoked from these users, the party administrator of the central bank should grant directly this privilege to the payment bank and then revoke it, as this triggers the cascade process related to the revoke privilege function from party (see Table 11 – cascade process when revoking privileges).

6.1.3.3 Access rights configuration process (completed)

As described in chapter Configuration of privileges [▸ 143], before the party administrator of a given party can grant a privilege to a user of the same party, the same privilege has to be granted to the same party, so that it becomes available to the party administrator(s) of the party.

On this basis, the following diagram illustrates the steps needed for granting a given privilege P to the users of a central bank (identified as party A in the diagram).

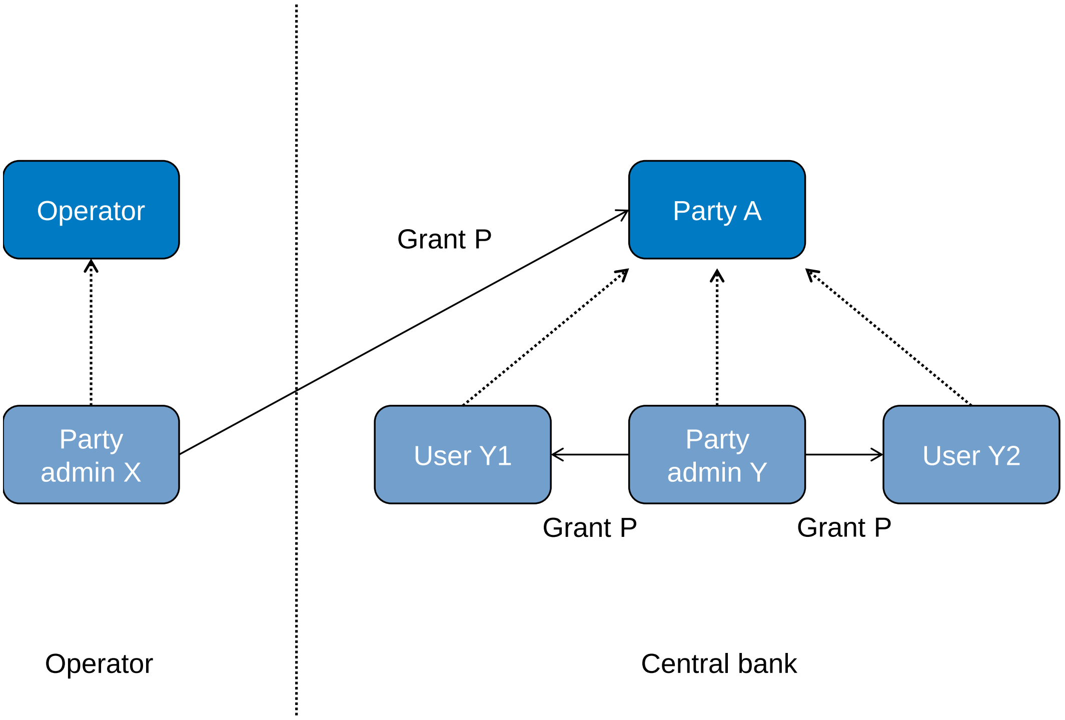


Figure 32 - Access rights configuration process (A)

The diagram shows that the two required steps are as follows:

- user X, as a party administrator of the operator, grants the privilege P to the party A;
- user Y, as a party administrator of the party A, grants the privilege P to all the relevant users (in this case, users Y1 and Y2).

The same process applies when a central bank needs to configure access rights for their payment banks. The following diagram illustrates all the steps needed for granting a given privilege P to the users of a payment bank (party B in the diagram), via the relevant central bank (party A in the diagram).

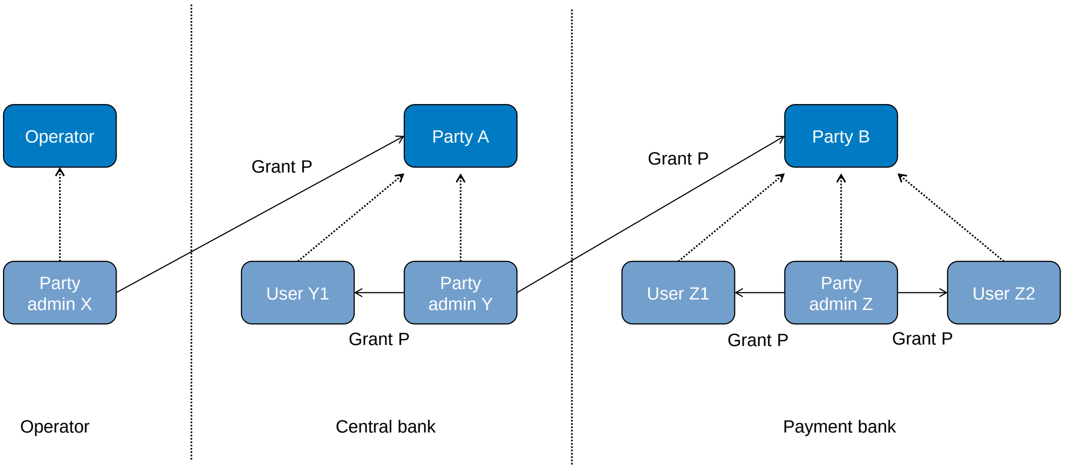


Figure 33 - Access rights configuration process (B)

The diagram shows that the three required steps are as follows:

- user X, as a party administrator of the operator, grants the privilege P to the party A (i.e. to a central bank);
- user Y, as a party administrator of the party A, grants the privilege P to the party B (i.e. to a payment bank);
- user Z, as a party administrator of the party B, grants the privilege P to the relevant users (in this case users Z1 and Z2).

In addition, the diagram shows that user Y, as a party administrator of the party A, can also grant the privilege P to the user Y1, as this user belongs to the same party.

These two examples illustrates that the access rights configuration process in the CRDM consists in two main tasks:

- configuration of access rights at party level;
- configuration of access rights at user level.

As stated in chapter Configuration of privileges [143] , the above process is not directly applicable for TIPS privileges; in this case privileges have to be granted to roles in the first place and then roles can be granted to parties and users. For details on the configuration of roles see chapter Configuration of roles [150].

6.1.3.3.1 Configuration of access rights at party level (completed)

This task consists in the assignment of the relevant set of roles and privileges to a given party in CRDM. A party administrator of the operator performs this task for the configuration of access rights of central banks.

The following diagram shows an example in which the party administrator of the operator grants to all the central banks the same set of roles and privileges. This set includes all the privileges needed by the central banks and all the privileges needed by the payment banks.

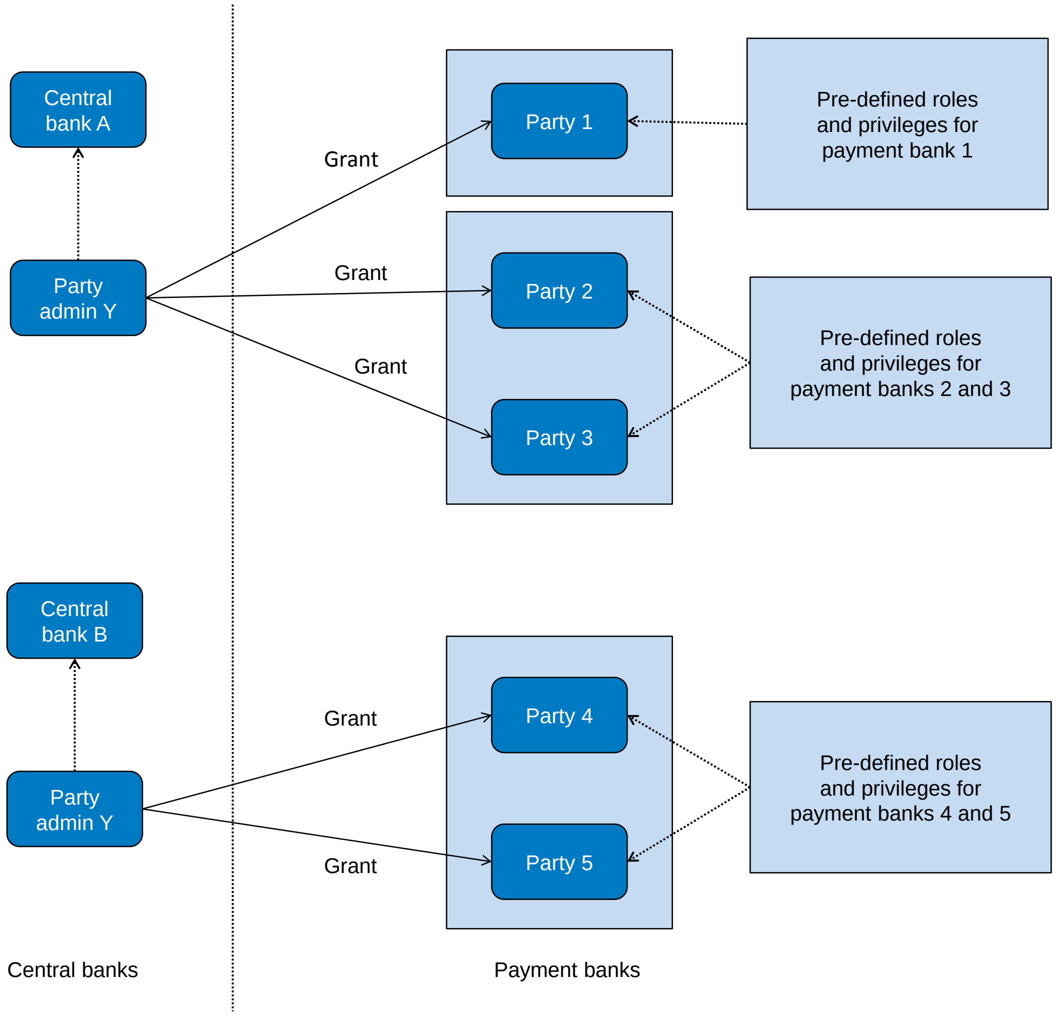

Central bank A
Party admin Y
Grant
Grant
Grant
Party 1
Party 2
Party 3
Pre-defined roles and privileges for payment bank 1
Pre-defined roles and privileges for payment banks 2 and 3
Central bank B
Party admin Y
Grant
Grant
Party 4
Party 5
Pre-defined roles and privileges for payment banks 4 and 5
Central banks
Payment banks

Figure 34 - Example - configuration of access rights at party level by the operator

A party administrator of each central bank assigns the relevant set of roles [14] and privileges to all its payment banks. In this example the party administrator of a central bank A configures the relevant access rights for three payment banks party 1, party 2 and party 3. This results in two different set of roles and privileges, the first one being granted to the payment bank party 1 only, the latter being assigned to both payment banks party 2 and party 3. Similarly, the party administrator of a central bank B assigns the relevant access rights to two payment banks party 4 and party 5, this task resulting in the configuration of the same set of access rights for both payment banks party 4 and party 5.

[14] New roles can only be created and maintained by the operator and central bank parties. Payment banks can only grant/revoke roles that have previously been granted to them by their central banks.

6.1.3.3.2 Configuration of access rights at user level (completed)

After the configuration of access rights at party level has been set up for a given party, its party administrator(s) can perform the configuration of access rights at user level, in order to assign the appropriate roles and privileges to all the users of the given party.

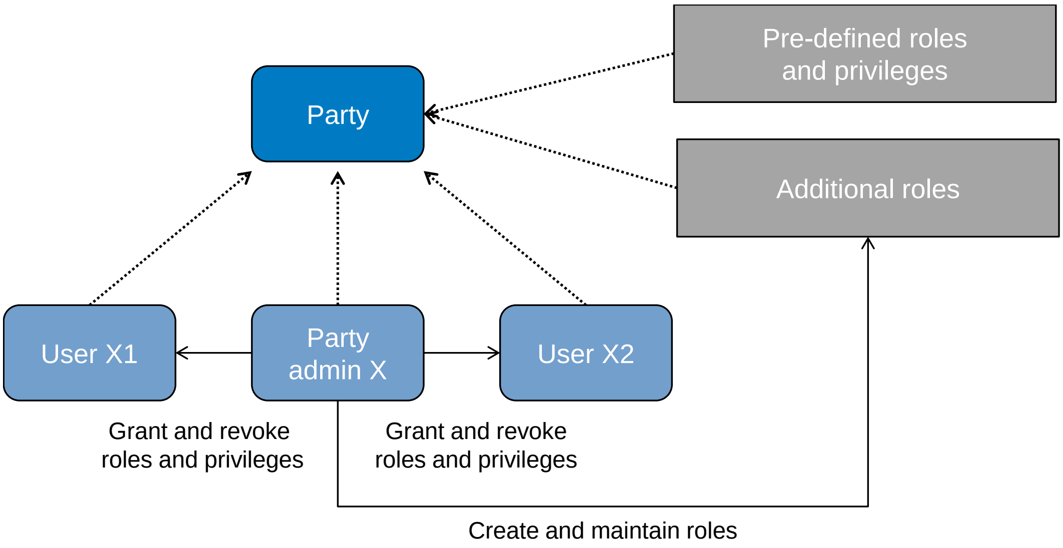

Pre-defined roles and privileges
Party
Additional roles
User X1
Party admin X
User X2
Grant and revoke roles and privileges
Grant and revoke roles and privileges
Create and maintain roles

Figure 35 - Configuration of access rights at user level

The above diagram shows that the party administrator(s) can set up the appropriate access rights configuration for the users of the same party:

- by possibly creating and maintaining [15] additional roles, besides the ones previously granted at party level [16]
- by granting (and revoking) the (default and additional) roles and the (default) privileges to the users of the same party

6.1.4 Message subscription

To be provided in a future version.

[15] New roles can only be created and maintained by the operator and central bank parties. Payment Banks can only grant/revoke roles that have previously been granted to them by their central banks.

[16] These additional roles can only be granted with available privileges, i.e. privileges previously granted at party level.

6.1.5 Instructing scenarios

6.1.6 Reference data maintenance process

6.1.6.1 Reference data objects (completed)

Duly authorised actors manage common reference data by creating and maintaining common reference data objects. A common reference data object is a set of logically related, self-consistent information. Parties and cash accounts are examples of common reference data objects. The following table provides the exhaustive list of common reference data objects defined in CRDM and the CRDM actors that are responsible for their management, i.e. for creating and maintaining them:

Table 61 - Common reference data objects

Area	Object	Responsible CRDM actors [17] [18]
Party	Party	Operator, central bank
	Party service link	Operator, central bank
	Banking group	Central bank
	Monetary financial institution	Central bank
Cash account	Cash account	Central bank
	Limit	Payment bank
	Authorised account user	Payment bank
	Account monitoring Group	Central bank
	Standing liquidity transfer order	Payment bank
	Liquidity transfer group	Payment bank
	Direct debit mandate	Payment bank
	Standing order for reservation	Payment bank
	Floor/ceiling	Payment bank

[17] "All" indicates that all types of CRDM actors (operator, central banks, payment banks) have the ability to manage the object type.

[18] The actor types listed for each function refer to the default responsible actor in normal operating conditions. However it is possible for the operator to act on behalf of central banks (and of payment banks, upon request of the relevant central bank) and for the central banks to act on-behalf of their payment banks, under well-defined contingency scenarios.

Area	Object	Responsible CRDM actors [17] [18]
Access rights management	User Role Privilege Certificate DN User-certificate DN link Role user [19] Role party [20] Grantee privilege [21]	All Operator, central bank Operator All All All Operator, central bank Operator, central bank, payment bank
Message subscription configuration	Message subscription rule Message subscription rule set	Central bank, payment bank Central bank, payment bank
Network configuration	DN BIC routing Network service Technical address network service link	Payment bank Operator Operator, central bank
Report configuration	Report configuration	Payment Bank
Restriction type management	Restriction type	Operator
Billing configuration	Service Item	Operator
Configuration parameters	Country Currency Currency service link System entity BIC directory Service	Operator Operator Operator Operator Operator Operator

[19] This object is related to the granting/revoking of roles to/from users.

[20] This object is related to the granting/revoking of roles to/from parties.

[21] This object is related to the granting/revoking of privileges to/from roles, parties and users.

A common reference data object consists of one or more classes of information. For example, a party is a common reference data object, consisting of the following classes of information:

- party
- party code
- party address
- party technical address

Each class of information includes a defined set of attributes. For example, the class of information party name of the common reference data object party includes the following attributes:

- the long name of the party
- the short name of the party
- the starting validity date of the party name

CRDM common component provides functions to maintain all common reference data objects (see chapter Reference data maintenance types [▸ 158]). Each maintenance operation on a common reference data object results in a new version of the same object. Each version of a common reference data object is called a revision of the object. Consequently, at any point in time, CRDM stores one or many revisions of each common reference data object, more precisely only one revision for newly created objects that were never maintained after their creation and N revisions for objects that were maintained N-1 times after they were created. The first revision of each common reference data object includes all the attribute values provided at creation time. After that, each maintenance request successfully processed creates a new revision for the object. This means that each revision may entail changes of many attributes of the same common reference data object at the same time. A new revision is also created when deleting and restoring a common reference data object.

Some classes of information are subject to data history, i.e. classes of information having multiple occurrences with continuous and non-overlapping validity periods. For example, the classes of information party name and party code of the common reference data object party can be subject to data history. In fact, they include a valid from attribute which determines the valid value of these classes of information at any given point in time.

6.1.6.2 Reference data maintenance types (completed)

CRDM allows a duly authorised actor to perform the following types of reference data maintenance operations on common reference data objects:

- create: creates a new common reference data object.

- update: updates an already existing common reference data object. It is possible, with a single update, to create, update or delete one or many classes of information of a common reference data object at the same time.
- delete: it deletes an already existing common reference data object. Deletion is always logical and not physical. Physical deletion is performed automatically by CRDM when performing the purge process following the archiving process (see chapter Reference data archiving and purging [▸ 164]).
- restore [22]: it reactivates a previously deleted common reference data object, i.e. it updates the approval status of this object from deleted to active.

Besides these operations, CRDM provides some specific types of reference data maintenance operations for the configuration of access rights (See section Access rights [▸ 127] for a detailed description of these operations).

CRDM allows all reference data maintenance types on all reference data objects in U2A mode, whereas it allows them only on a subset of reference data objects through the DMT and A2A mode respectively. The following tables show the exhaustive list of all the available reference data maintenance types that are possible in the DMT and in A2A mode:

Table 62 - Management of reference data objects in DMT

Area	Object	DMT function
Party data management	Party	Create
	Technical address network service link	Create
Cash account data management	Cash account	Create
	Authorised account user	Create
	Limit	Create

[22] This function is available in U2A mode only and it is granted, for each object, with the system privilege that allows deleting the same object as well.

Area	Object	DMT function
Access rights management	User Role Privilege Certificate DN User-certificate DN link	Create Create, grant Grant Create Create
Message subscription configuration	Message subscription rule set Message subscription rule	Create Create
Report configuration	Report configuration	Create

Table 63 - Management of reference data objects in A2A mode

Area	Object	DMT function
Party data management	Party	Create, update, delete
Cash account data management	Cash account Liquidity transfer order Limit	Create, update, delete Update, delete Update, delete

6.1.6.3 Validity of reference data objects (completed)

Some common reference data objects include attributes limiting the validity period of these objects. For example, each party service link, which defines the participation of a given payment bank in a specific service, common component or back-office application, includes two attributes specifying the date from which and the date to which the link is valid, i.e. the period in which said payment bank can operate in that service, common component or back-office application. Between the creation date and the deletion date of the link, but outside the validity period just defined, the payment bank is not allowed to operate in the Service, even though it is active in CRDM repository and it can be queried and maintained by a duly authorised user.

CRDM common component makes a distinction between the following two categories of common reference data objects:

- common reference data objects with unlimited validity period
- common reference data objects with limited validity period

The following table shows the exhaustive list of all the common reference data objects with unlimited validity period:

Table 64 - Common reference data objects with unlimited validity period

Area	Object
Party	Banking group Monetary financial institution
Cash account	Account monitoring group Liquidity transfer group
Access rights management	User Role Privilege Certificate DN User-Certificate DN link Role user link Role party link Privilege role link
Network configuration	Network service Technical address network service link
Configuration parameters	Country Currency Currency service link System entity Service Currency service link

This type of common reference data object starts being valid in CRDM immediately after it has been created. Similarly, a common reference data object with unlimited validity period may be immediately updated or deleted by a duly authorised user. However, in both cases the reference data change, i.e. the creation of a new object or the update or deletion of an already existing object is made effective in the relevant Eurosystem market infrastructure service(s) only by means of the daily reference data propagation process.

Regardless of the way common reference data object with limited validity period are propagated to the relevant Eurosystem market infrastructure service(s), between the creation date and the deletion date of this object, it is active in the CRDM common component and it can be queried and maintained by a duly authorised actor.

Common reference data objects with limited validity period can be updated either intraday, i.e. while they are in their validity period or as of a future date, i.e. before they become valid.

The following table shows the exhaustive list of all the common reference data objects with limited validity period, with the columns on the right specifying the possible maintenance operations depending on the validity period:

Table 65 - Common reference data objects with limited validity period [23]

Area	Object	Creation	Update	Deletion
Party	Party	Validity date may take the value of the current date.	May take effect on the current date [24].	May be performed only on objects that are not valid on the current date.
	Party service link	Validity date may take the value of the current date.	May take effect on the current date.	May be performed only on objects that are not valid on the current date.
Cash account	Cash account	Validity date may take the value of the current date.	May take effect on the current date.	May be performed only on objects that are not valid on the current date.
	Standing liquidity transfer order	Validity date may take the value of the current date.	May take effect on the current date.	May be performed only on objects that are not valid on the current date.
	Standing order for reservation	Validity date may take the value of the current date.	May take effect on the current date.	May be performed only on objects that are not valid on the current date.

[23] In the following table, the columns 'Creation/Update/Deletion' clarify whether it is possible to perform a given maintenance operation on each object with immediate effect in CRDM. For example, if a user updates an object on which updates "may take effect on the current date", they are able, should they wish to do so, to perform changes that become immediately valid in CRDM. On the contrary, if the update "may take effect only as of a future date" then it is not possible to perform intraday changes on the object. The possibilities described in the table represent the level of flexibility offered to the user. Within these limitations, the user decides exactly when a specific modification should take effect.

[24] This is not applicable to the party code, which cannot be updated if it is currently active.

Area	Object	Creation	Update	Deletion
	Direct debit mandate	Validity date may take the value of the current date.	May take effect on the current date.	May be performed only on objects that are not valid on the current date.
	Authorised account user	Validity date may take the value of the current date.	May take effect on the current date.	May be performed only on objects that are not valid on the current date.
	Floor/ceiling	Validity date may take the value of the current date.	May take effect on the current date.	May be performed only on objects that are not valid on the current date.
Message subscription	Message subscription rule set	Validity date may take value of the next business day at the earliest.	May take effect only as of a future date.	May be performed only on objects that are not valid on the current date.
	Message subscription rule	Validity date may take value of the next business day at the earliest.	May take effect only as of a future date.	May be performed only on objects that are not valid on the current date.
Report configuration	Report configuration	Validity date may take value of the next business day at the earliest.	May take effect only as of a future date.	May be performed only on objects that are not valid on the current date.
Restriction type management}	Restriction type	Validity date may take value of the next business day at the earliest.	May take effect only as of a future date.	May be performed only on objects that are not valid on the current date.
Network configuration	DN-BIC routing	Validity date may take the value of the current date.	May take effect on the current date.	May be performed only on objects that are not valid on the current date.
Configuration parameters	BIC directory	Validity date may take the value of the current date.	May take effect on the current date.	May be performed only on objects that are not valid on the current date.

For parties and cash accounts the validity period is defined by an opening date and a closing date attribute. Between these two dates the common reference data object, i.e. the party or the cash account, is valid, meaning that Eurosystem market infrastructure services can use it for processing (e.g. for settlement purposes). Outside this period, the common reference data object can only be queried or maintained in the CRDM common component by a duly authorised user.

6.1.6.4 Reference data archiving and purging (completed)

CRDM archives new reference data and their changes three calendar months after they were created or changed. CRDM purges, i.e. physically deletes reference data from the production data base three calendar months after they were deleted. For example, a party has to be deleted before CRDM can purge it. This implies that a party is never purged, unless a duly authorised user makes the decision to delete it.

The following example illustrates how CRDM archives and purges the different revisions of a generic common reference data object.

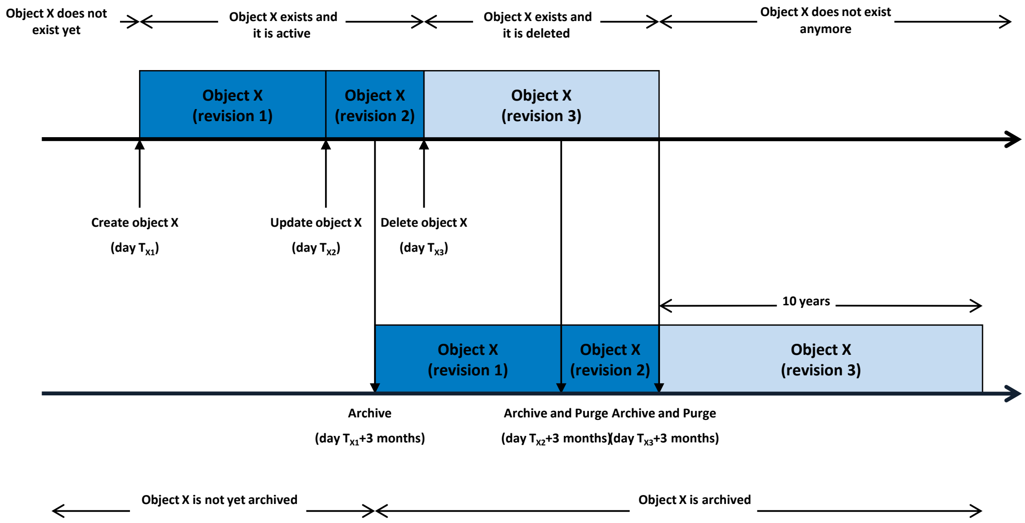

Object X does not exist yet
Object X exists and it is active
Object X exists and it is deleted
Object X does not exist anymore
Object X (revision 1)
Object X (revision 2)
Object X (revision 3)
Create object X (day T_{X1})
Update object X (day T_{X2})
Delete object X (day T_{X3})
10 years
Object X (revision 1)
Object X (revision 2)
Object X (revision 3)
Archive (day T_{X1}+3 months)
Archive and Purge (day T_{X2}+3 months)
Archive and Purge (day T_{X3}+3 months)
Object X is not yet archived
Object X is archived

Figure 36 - Example - archiving and purging after deletion of a common reference data object

In this example, a duly authorised user creates intra-day, on business day T_{X1}, a common reference data object X. This results in the creation of the first revision of the object X. During business day T_{X2} (with $T_{X2}<T_{X1}$ + three calendar months) a duly authorised user updates the common reference data object X changing one (or many) of its attribute(s). This results in the creation of a new revision (2) for X.

On business day T_{X1+} three calendar months, the archiving process copies the first revision of the common reference data object X into the archiving data base. It is worth mentioning that:

- CRDM does not purge the archived revision, as it still refers to a period of time that expired on T_{X2}, i.e. since less than three calendar months.
- CRDM does not archive the second revision of the common reference data object X, as it was created on T_{X2}, i.e. since less than the duration of the retention period.

During business day T_{X3} (with T_{X3}<$_{TX2}$ + three calendar months), a duly authorised user deletes the common reference data object X. This results in the creation of a new revision (3) for the same object. On business day T_{X2+} three calendar months, the archiving process copies the second revision of the common reference data object X into the archiving data base. In this case:

- CRDM does not purge this second revision, as it still refers to a period of time that expired on T_{X3}, i.e. since less than three calendar months.
- CRDM does not archive the third revision of the common reference data object X, as it was created on T_{X3}, i.e. since less than three calendar months.
- CRDM purges the first revision of the common reference data object X, as it refers to a period of time that expired exactly since three calendar months.

Finally, on business day T_{X3+} three calendar months, the archiving process copies the third and final revision of the common reference data object X into the archiving data base. On the same day, just after the archiving process is successfully performed, CRDM purges the common reference data object X, by physically deleting the last two revisions of the object X that are still present in the production data base.

From this moment on, all revisions of the common reference data object X are available only in the archiving data base, where the archiving common component keeps them for a period of ten years.

6.1.6.5 Lifecycle of reference data objects (completed)

This section puts together all the concepts described so far and provides a general description of the lifecycle of common reference data objects.

Lifecycle of common reference data objects with unlimited validity period

The following diagram illustrates the lifecycle of a common reference data object with unlimited validity period both in the production data base and in the archiving data base:

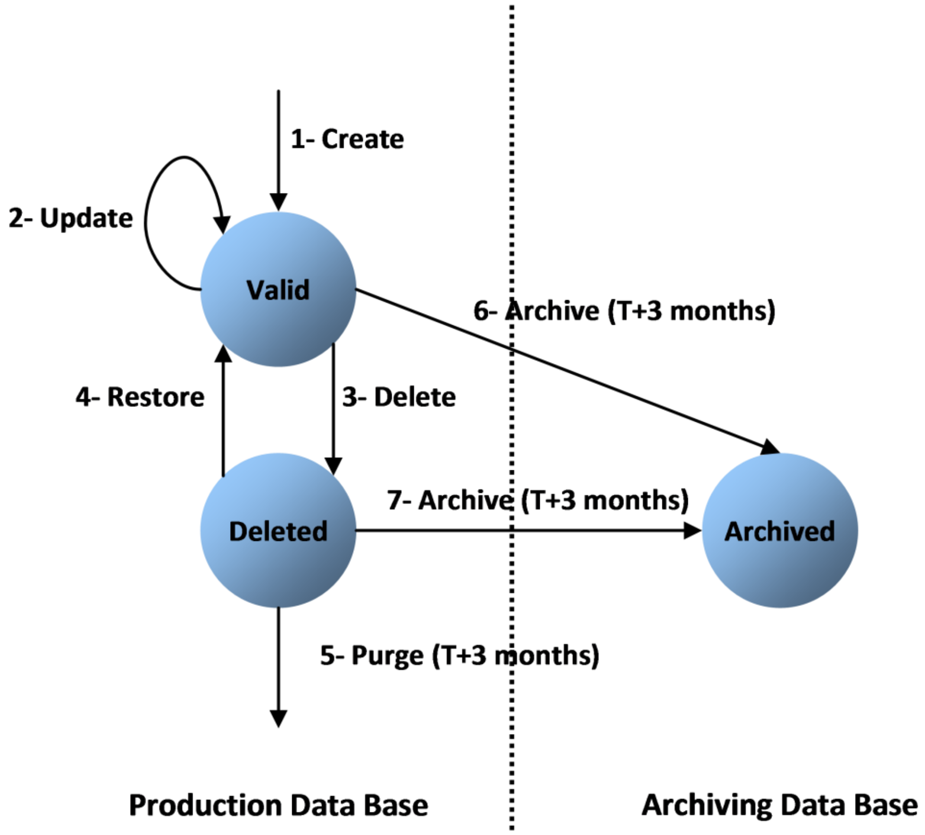


Figure 37 - Lifecycle of common reference data objects with unlimited validity period

When a duly authorised actor submits a reference data maintenance instruction to CRDM to create a common reference data object with unlimited validity period, CRDM processes it and, in case of successful processing, it creates the relevant object. This object is valid and it exists in the production data base only (transition 1).

From this moment on, a duly authorised user may submit to CRDM one or many reference data maintenance instructions to update the common reference data object. Regardless of the result of CRDM processing, i.e. whether the reference data maintenance instruction is successfully or unsuccessfully processed, the common reference data object remains valid (transition 2).

When a duly authorised user submits to the CRDM reference data maintenance instruction to delete a common reference data object, the CRDM processes it and, in case of successful processing, it deletes the relevant object. This object is logically deleted (transition 3), even if it is still physically present in the production data base.

From this moment on and within a period of three calendar months, if a duly authorised user submits to CRDM a reference data maintenance instruction to restore a previously deleted common reference data object, CRDM processes it and, in case of successful processing, it restores the relevant object. As a result, the object becomes valid again (transition 4).

Three calendar months after a common reference data object is deleted, CRDM physically deletes it from the production data base. This results in the object being purged by the production data base (transition 5), i.e. it exists only in the archiving data base.

Three calendar months after a common reference data object is created, updated or deleted, CRDM copies the revision of the common reference data object resulting from this reference data maintenance instruction from the production data base to the archiving data base. As a result the common reference data object is both in the production data base and archived in the archiving data base, in case it was created or updated, or only in the archiving data base, in case it was deleted (transitions 6 and 7).

Lifecycle of common reference data objects with limited validity period

The following diagram illustrates the lifecycle of a common reference data object with limited validity period both in the production data base and in the archiving data base

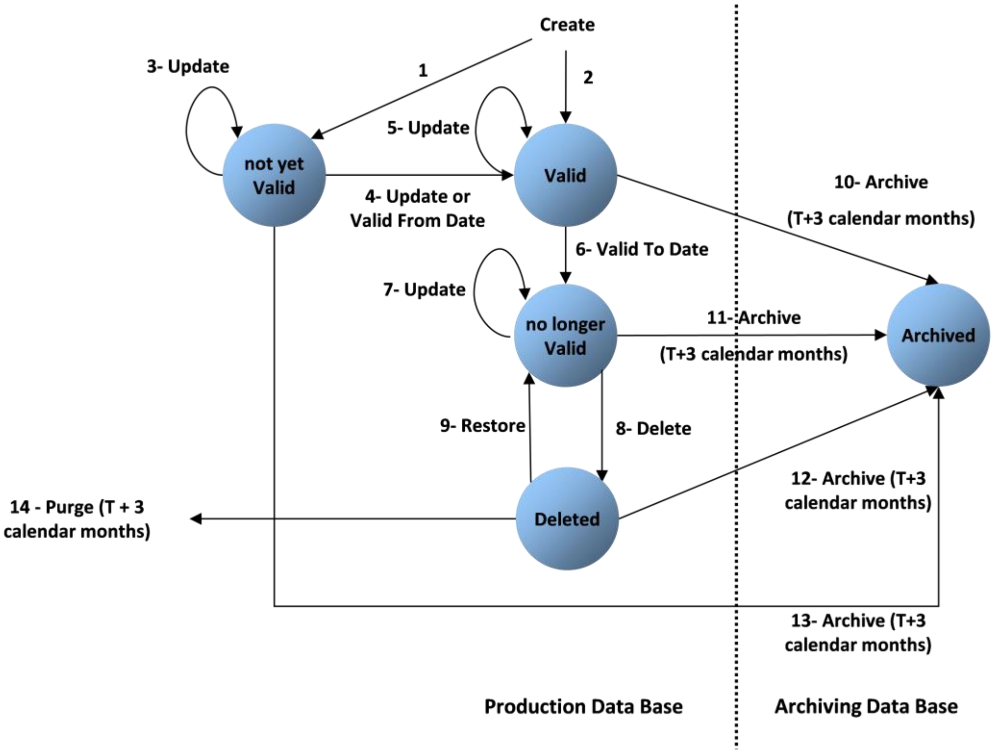


Figure 38 - Lifecycle of common reference data objects with limited validity period

When a duly authorised user submits to CRDM a reference data maintenance instruction to create a common reference data object with limited validity period, CRDM processes it and, in case of successful processing, it creates the relevant object. This object is either valid or not yet valid, depending on the starting date of its validity period, and it exists in the production data base only (transitions 1 and 2).

From this moment on, a duly authorised user may submit to the CRDM one or many reference data maintenance instructions to update the common reference data object. If the object is valid, then it remains valid, regardless of the result of CRDM processing, i.e. whether the reference data maintenance instruction is successfully or unsuccessfully processed (transition 5). If the object is not yet valid, two sub-cases are possible:

- If the reference data maintenance instruction also updates the starting date of the validity period to the current business date and it is successfully processed, then the common reference data object becomes valid (transition 4).
- In all other cases, whether the reference data maintenance instruction is successfully or unsuccessfully processed, the common reference data object remains not yet valid (transition 3).

A common reference data object becomes valid from the starting business date of the validity period (transition 4).

A common reference data object is valid until the end of day of the final date of the validity period (transition 6).

When a duly authorised user submits to CRDM a reference data maintenance instruction to delete a common reference data object, CRDM processes it and, in case of successful processing, it deletes the relevant object. This object is logically deleted (transition 8), even if it is still physically present in the production data base.

From this moment on and within a period of three calendar months, if a duly authorised user submits to the Common Reference Data Management service a reference data maintenance instruction to restore a previously deleted common reference data object, CRDM processes it and, in case of successful processing, it restores the relevant object. As a result, the object becomes no longer valid again (transition 9).

Three calendar months after a common reference data object has been deleted, CRDM physically deletes it from the production data base. This results in the object being purged by the production data base (transition 14), i.e. it exists only in the archiving data base.

Three calendar months after a common reference data object is created, updated or deleted, CRDM copies the revision of the common reference data object resulting from this reference data maintenance instruction from the production data base to the archiving data base. As a result the object is both in the production data base (as a not yet valid, valid, no longer valid or deleted object) and in the archiving data base archived, in case it was created or updated, or only in the archiving data base, in case it was deleted (transitions 10, 11, 12 and 13).

6.1.6.6 Reference data propagation (completed)

CRDM allows users to configure reference data to be used in the local reference data management of other TARGET services (e.g. TIPS, CLM and RTGS).

Data set up in CRDM is propagated to other services, common components or back-office applications on a regular basis, typically once a day, at a preset time before the change of business date. If needed, participants can request an ad-hoc propagation to be run at different times of day for a specific service, common component or back-office application. There is no technical limit on the number of times a data propagation can run during a given business date.

No data propagation flow exists from TIPS, CLM and RTGS to CRDM. Since CRDM contains data belonging to different services, common component or back-office application, specific segregation principles are put in place to make sure that relevant data is made available in each service, common component or back-office application depending on the individual needs. In this respect certain objects (e.g. country, currency) are fully shared – they are made available to every service, common component or back-office application without distinction. Other objects are service-specific, and are made available in full to a single service (example includes banking group for CLM). Finally, certain objects are shared among multiple services, but the data is segregated and made available in a given service based on the values of specific attributes that link each instance to a specific service, either directly or indirectly. Examples of this type of objects include party and cash account.

The following table lists the possible CRDM reference data objects and their relevance for each service, as well as the data segregation principles defining which instances are propagated to which service.

Table 66 - CRDM data segregation per service/component

Area	Object	Service(s)/component	Segregation principles
Party	Party	CLM, RTGS, T2S, TIPS	All data is available in T2S. Parties with a party service link to CLM, RTGS or TIPS are available in that service/component.
	Party service link	None	Only relevant for CRDM; defines the availability of party data for a given service.
	Banking group	CLM	All data is available in CLM.
	Monetary financial institution	CLM	All data is available in CLM.
Cash account	Cash account	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the cash account type attribute; each possible value of this attribute identifies a type of cash account used by a single service.

Area	Object	Service(s)/component	Segregation principles
	Authorised account user	TIPS	All data is available in TIPS.
	Account monitoring group	CLM	All data is available in CLM.
	Standing liquidity transfer order	CLM, RTGS, T2S	Data is available in different services depending on the cash account type attribute of the cash account it refers to.
	Liquidity transfer group	CLM, RTGS	Data is available in different services depending on the cash account type attribute of the cash accounts it refers to.
	Limit	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the cash account type attribute of the cash account it refers to.
	Direct debit mandate	CLM, RTGS	Data is available in different services depending on the cash account type attribute of the cash account it refers to.
	Standing order for limit	RTGS	All data is available in RTGS.
	Standing order for reservation	CLM, RTGS	Data is available in different services depending on the cash account type attribute of the cash accounts it refers to.
	Floor/ceiling	CLM, RTGS	Data is available in different services depending on the cash account type attribute of the cash account it refers to.

Area	Object	Service(s)/component	Segregation principles
Access rights management	User	CLM, RTGS, T2S	All data is available in T2S. Data related to parties with a party service link to CLM or RTGS is available in that service.
	Role	CLM, RTGS, T2S, TIPS	All data is available in T2S. Data containing privileges related to CLM, RTGS or TIPS is available in that service.
	Privilege	T2S	All data is available in T2S. It is not available in other services, but it is used by CRDM to determine the availability of other access rights data in those Services. Each privilege includes a link to a single service which defines the service that contains the user function activated by the privilege.
	Certificate DN	CLM, RTGS, T2S, TIPS	All data is available in T2S. Data linked to users flagged as main users for TIPS is available in TIPS. Data linked to users under parties with a party service link to CLM or RTGS is available in that service.
	User-certificate DN link	CLM, RTGS, T2S, TIPS	All data is available in T2S. Data linked to users flagged as main users for TIPS is available in TIPS. Data linked to users under parties with a party service link to CLM or RTGS is available in that service.

Area	Object	Service(s)/component	Segregation principles
	Role user	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the service the privileges contained in the role refer to.
	Role party	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the service the privileges contained in the role refer to.
	Grantee privilege	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the service the privilege refers to.
Message subscription configuration	Message subscription rule set	CLM, RTGS, T2S, TIPS	All data is available in T2S. Data containing message subscription rules that reference data from CLM, RTGS or TIPS is available in those services.
	Message subscription rule	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the underlying reference data objects the rule refers to.
Network configuration	Network service	CLM, RTGS, T2S, TIPS	Data is available in different Services based on an attribute that defines a direct reference to a single Service.
	Technical address network service link	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the service the related network service refers to.
	DN BIC routing	TIPS	All data is available in TIPS.
Report configuration	Report configuration	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the specific type of report being subscribed.

Area	Object	Service(s)/component	Segregation principles
Restriction type management	Restriction type	RTGS, T2S, TIPS	Data is available in different services based on an attribute that defines a direct reference to a single service.
Billing configuration	Service item	None	Only relevant for CRDM and Billing.
Configuration parameters	Country	CLM, RTGS, T2S, TIPS	All data is available in all services.
	Currency	CLM, RTGS, T2S, TIPS	All data is available in all services.
	Currency service link	CLM, RTGS, T2S, TIPS	Data is available in different services depending on the service the link refers to.
	System entity	CLM, RTGS, T2S, TIPS	All data is available in all services.
	BIC directory	CLM, RTGS, T2S, TIPS	All data is available in all services.
	Service	None	Only relevant for CRDM.

6.2 Data warehouse (to be completed in version 2.0)

6.2.1 Introduction

6.2.2 Scope of the data warehouse

6.2.3 Access

6.2.3.1 Connectivity

6.2.3.2 Authentication and authorisation

6.2.4 User roles and access rights

6.2.4.1 Overview

6.2.4.2 User rights

6.2.4.3 User profiles

6.2.5 Data warehouse queries and reports

6.2.5.1 Overview

6.2.5.2 Types of queries and reports

6.2.5.3 Predefined queries and reports

6.3 Billing (to be completed in version 2.0)

6.4 Legal archiving (to be completed in version 2.0)

7 Contingency services (to be completed in version 2.0)

8 Operations and support (to be completed in version 2.0)

8.1 Business application configuration

8.2 Calendar management

8.3 Business day management

8.4 Business and operations monitoring

8.5 Archiving management

8.6 Trouble management

9 Additional information for central banks

9.1 Role of central banks in CLM (to be completed in iteration 4)

9.2 Reference data for central banks (to be completed in iteration 4)

9.2.1 Specific data for central banks

9.2.2 Setup of CLM related reference data

9.3 Settlement of payments - specific functions for central banks (to be completed in iteration 4)

9.3.1 Payments linked to monetary policy operations

9.3.2 Cash withdrawals

9.4 Credit line management

9.4.1 Credit line update (to be completed in iteration 4)

9.4.1.1 Overview

9.4.1.2 Credit line update process

9.4.2 Connected payment (completed)

9.4.2.1 Overview (completed)

A connected payment is a payment initiated by a central bank system or central bank operator that triggers a change in the credit line of the CLM participant and debit/credit of its account simultaneously to compensate the change in its credit line. Therefore the CLM participant needs a main cash account.

The processing of connected payments is not possible between the central bank general cut-off for the use of standing facilities (i.e. 18.40 on normal business day) and the start of the provisioning of liquidity for the new business day (i.e. 19.00 on normal business day), as well as during the maintenance window.

A connected payment leads to the increase or decrease of the CLM participants credit line and at the same time to a corresponding debit or credit booking on its main cash account. (**Note:** The connected payment is processed on all or nothing basis). Connected payments are not queued and can therefore not be revoked.

To decrease a credit line and credit the main cash account a FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▸ 274] message is used.

To increase a credit line and debit the main cash account a FinancialInstitutionDirectDebit (pacs.010) [▸ 276] message is used.

9.4.2.2 Connected payment process (completed)

The following payment flow illustrates a connected payment with positive validation and settlement on the basis of a FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▸ 274] .

Message flow

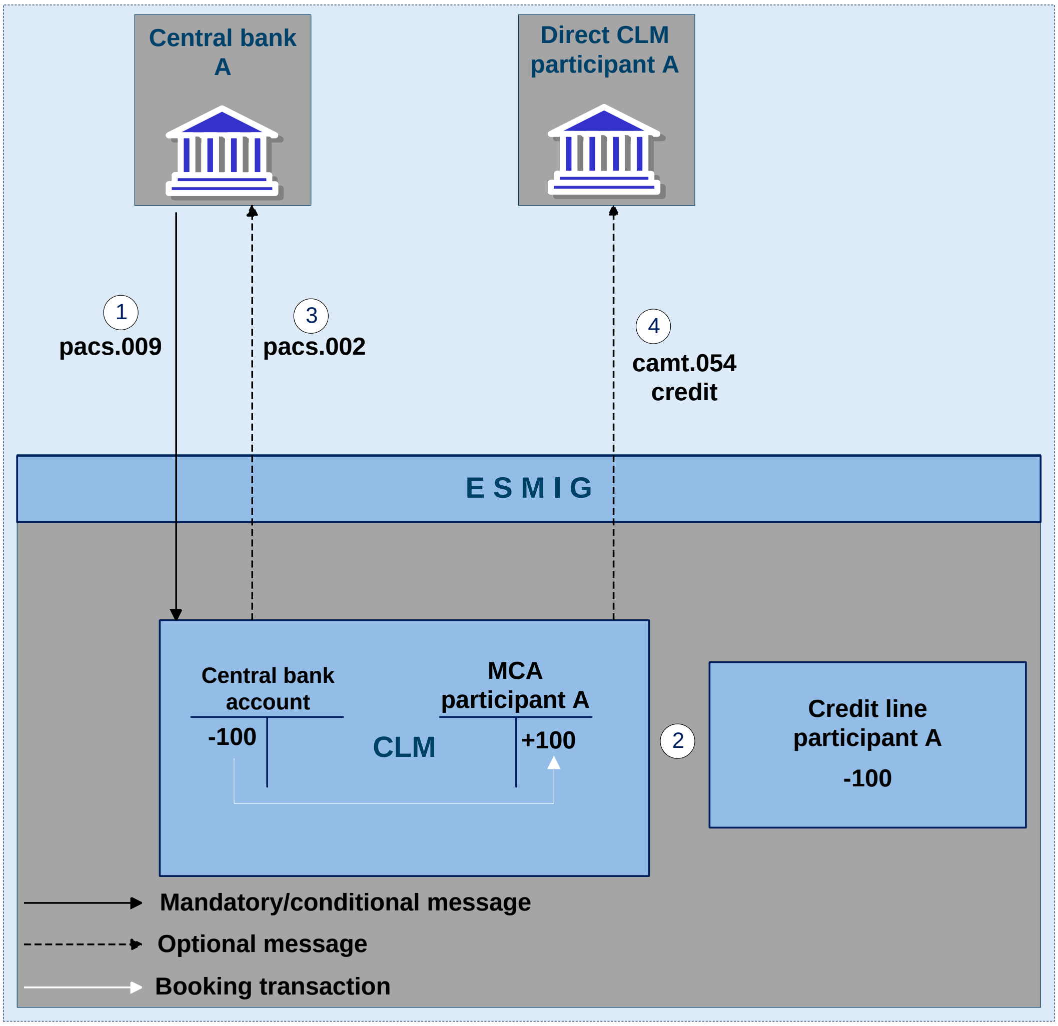


Figure 39 - pacs.009 connected payment

Process description

Table 67 - Connected payment (pacs.009)

Step	Processing in/between	Description
1	Central bank via ESMIG to CLM	The central bank sends a pacs.009 including message element CONPAY via ESMIG to the CLM
2	CLM	CLM check and validation positive Debit central bank account and credit main cash account participant A simultaneously decrease credit line for participant A (settlement amount is not necessarily equal to credit line change) if business validation positive
3	CLM via ESMIG to central bank	Creation and forwarding of pacs.002 by the CLM (optional) via ESMIG to central bank
4	CLM via ESMIG to CLM participant	Creation and forwarding of camt.054 (credit) by the CLM via ESMIG to CLM participant A (optional)

Used messages

- FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▹ 274]
- PaymentStatusReport (pacs.002) [▹ 272]
- BankToCustomerDebitCreditNotification (camt.054) [▹ 261]

The following payment flow illustrates a connected payment with positive validation and settlement on the basis of a FinancialInstitutionDirectDebit (pacs.010) [▹ 276] .

Message flow

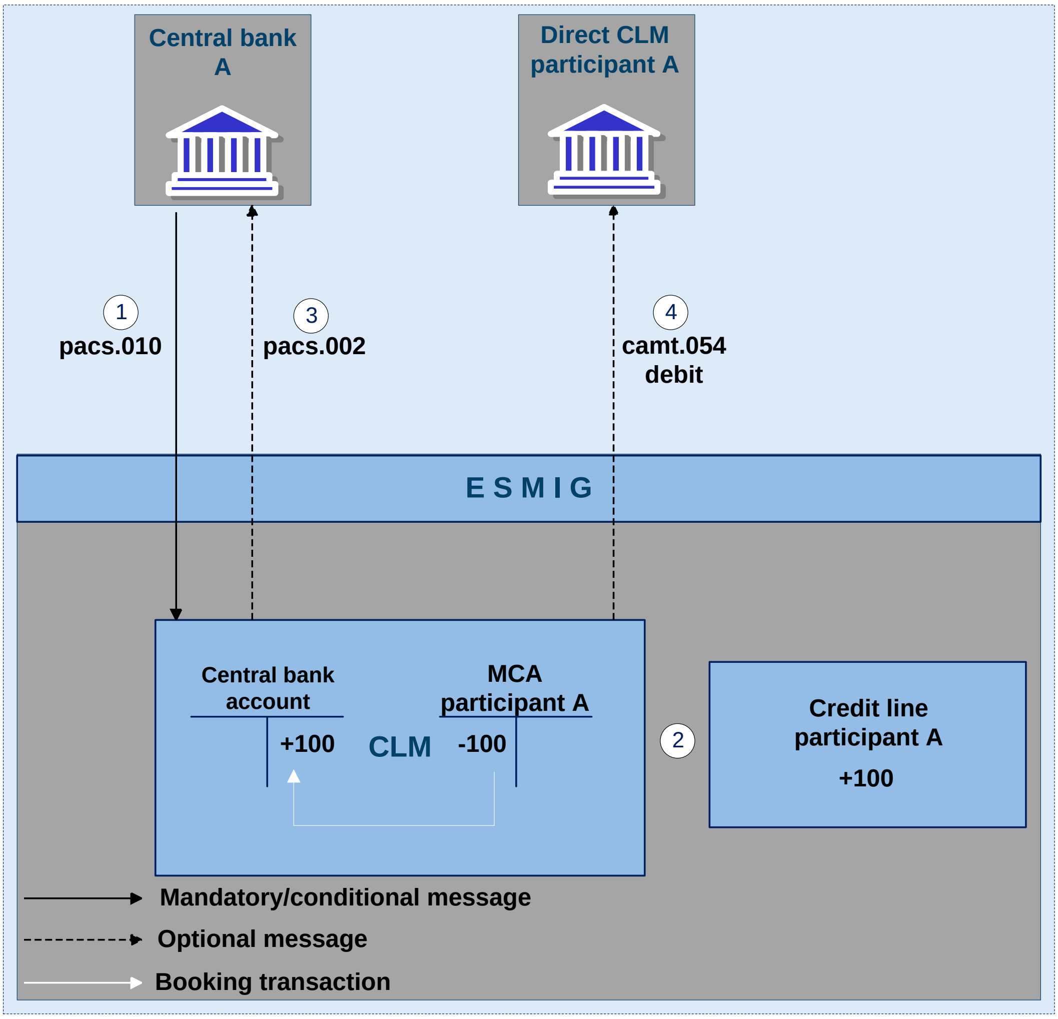


Figure 40 - pacs.010 connected payment

Process description

Table 68 - Connected payment (pacs.010)

Step	Processing in/between	Description
1	Central bank via ESMIG to CLM	The central bank sends a pacs.010 with message element CONPAY and the credit line change via ESMIG to the CLM
2	CLM	CLM check and validation positive Credit central bank account and debit main cash account participant A simultaneously increase credit line for participant A (settlement amount is not necessarily equal to credit line change) if business validation positive
3	CLM via ESMIG to central bank	Creation and forwarding of pacs.002 by the CLM (optional) via ESMIG to central bank
4	CLM via ESMIG to CLM participant	Creation and forwarding of camt.054 (debit) by the CLM via ESMIG to CLM participant A (optional)

Used messages

- FinancialInstitutionDirectDebit (pacs.010) [▸ 276]
- PaymentStatusReport (pacs.002) [▸ 272]
- BankToCustomerDebitCreditNotification (camt.054) [▸ 261]

9.5 End-of-day procedures (to be completed in iteration 4)

9.6 Query management - central bank specific queries (to be completed in iteration 4)

9.7 Business/liquidity monitoring for central banks (to be completed in iteration 4)

9.8 Reserve management - specific functions for central banks (to be completed in iteration 4)

9.9 Standing facilities - specific functions for central banks (to be completed in iteration 4)

9.10 Data warehouse - specific functions for central banks (to be completed in version 2)

9.11 Billing - specific functions for central banks (to be completed in version 2)

9.12 Contingency services - specific functions for central banks (to be completed in version 2)

9.13 Specific requirements for central banks of "out" countries (to be completed in version 2)

II Dialogue with the CLM participant

10 Processes with CLM components

10.1 Interface processing - send file (to be completed in iteration 4)

10.2 Reference data management - maintain local reference data object (to be completed in iteration 4)

10.3 Payment order processing

10.3.1 Send payment order (completed)

This process starts

- when the submitting actor sends one of the following messages via ESMIG to the CLM component:

Table 69 - Messages sent by the submitting actor to CLM component

Message	Message name
FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▸ 274]	FinancialInstitutionCreditTransfer
FinancialInstitutionDirectDebit (pacs.010) [▸ 276]	FinancialInstitutionDirectDebit
LiquidityCreditTransfer (camt.050) [▸ 255]	LiquidityCreditTransfer

- when the CLM component receives a message from the file splitting process (refer to interface process "Interface processing - send file [▸ 185]").

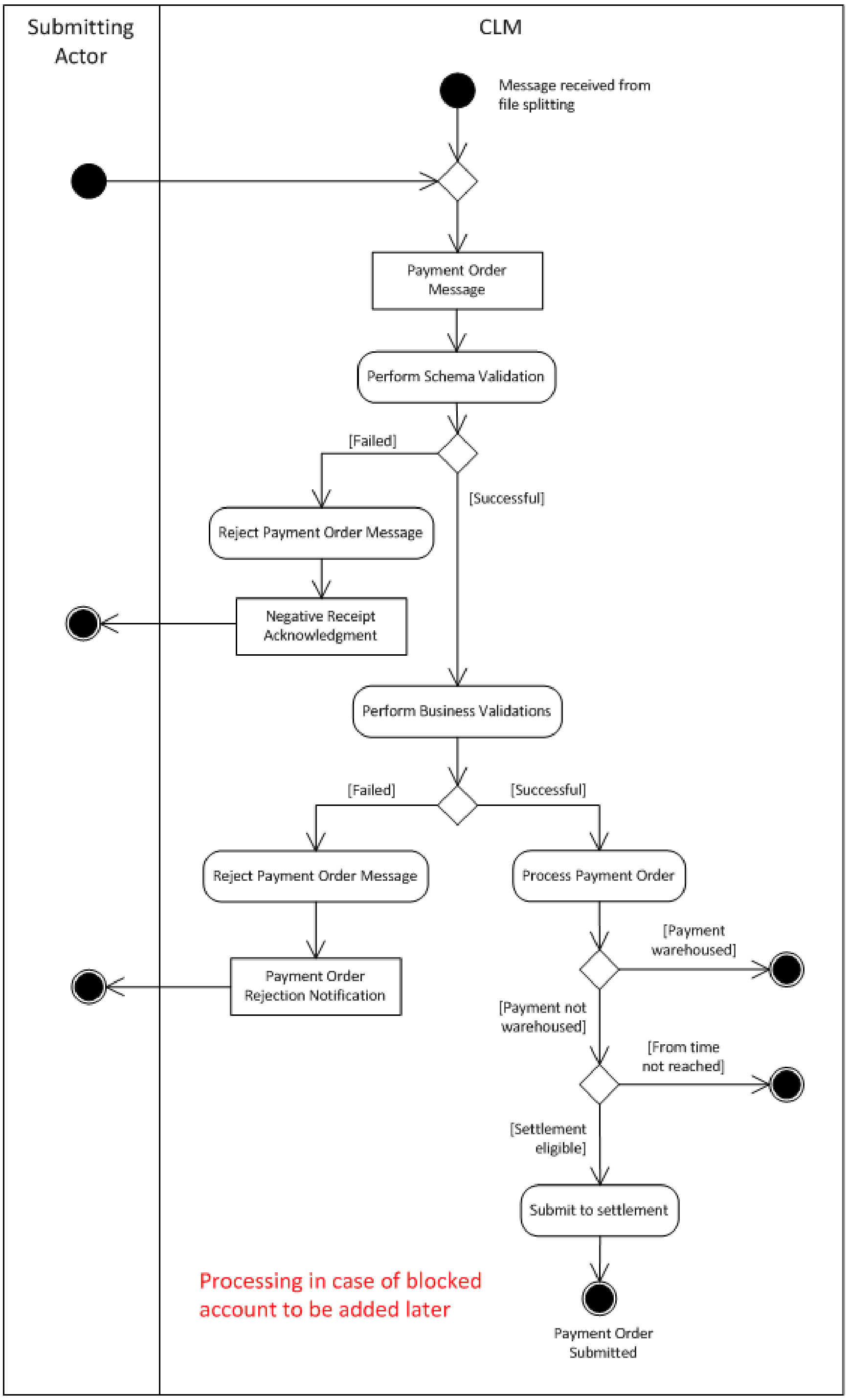

Submitting Actor
CLM
Message received from file splitting
Payment Order Message
Perform Schema Validation
[Failed]
[Successful]
Reject Payment Order Message
Negative Receipt Acknowledgment
Perform Business Validations
[Failed]
[Successful]
Reject Payment Order Message
Process Payment Order
Payment Order Rejection Notification
[Payment warehoused]
[Payment not warehoused]
[From time not reached]
[Settlement eligible]
Submit to settlement
Processing in case of blocked account to be added later
Payment Order Submitted

Figure 41 - Send CLM payment order

Schema validation:

In the first step, the CLM component performs the schema validation of the payment order message.

- **[Failed]** In case the schema validation fails, the CLM component rejects the payment order message and the submitting actor receives a "*Negative Receipt Acknowledgement*" ReceiptAcknowledgement (admi.007) [224].

Note: CLM identifies all possible schema validation errors and does not stop the schema validation after the first error is found.

- **[Successful]** In case of a successful schema validation, the CLM component continues with the business validation.

Business validation:

In the second step, CLM performs the business validation with possible outcomes being:

- **[Failed]** In case the business validation fails, the CLM component rejects the payment order message and the submitting actor receives a "*Payment Order Rejection Notification*" PaymentStatusReport (pacs.002) [▸ 272].

  Note: The CLM component continues with all possible business validations even after the business validation identifies one or more errors. It does not stop after identifying the first business validation error.

- **[Successful]** In case the business validation is successful, CLM continues with the processing of the payment order.

  As part of this processing step, the CLM component determines

  – whether the payment order is a warehoused payment;

  – whether the defined "From Time" when specified in the payment has not been reached;

  – whether the payment order is directly eligible for the settlement.

The processing submits the payment order directly to the Standard CLM settlement [▸ 188] process when it is directly eligible for settlement.

10.3.2 Revoke cancel payment order (to be completed in iteration 4)

10.3.3 Amend payment order (to be completed in iteration 4)

10.3.4 Send reserve management instruction (to be completed in iteration 4)

10.3.5 Execute standing order (to be completed in iteration 4)

10.3.6 Reservation management (to be completed in iteration 4)

10.3.7 Reject pending payment orders at end of day (to be completed in iteration 4)

10.3.8 Settle CLM payment orders

10.3.8.1 Standard CLM settlement (completed)

The process “*attempt payment order settlement*” starts

- after receiving a successfully validated payment order [Payment Order Submitted],
- in case of an inter-service liquidity transfer initiated in the CLM component could not be successfully booked in the other service or components and the amount needs to be credited back to the main cash account [A] or
- for a successfully validated payment order that specifies “From Time” and the “From Time” has been reached [Payment Order From Time Reached]

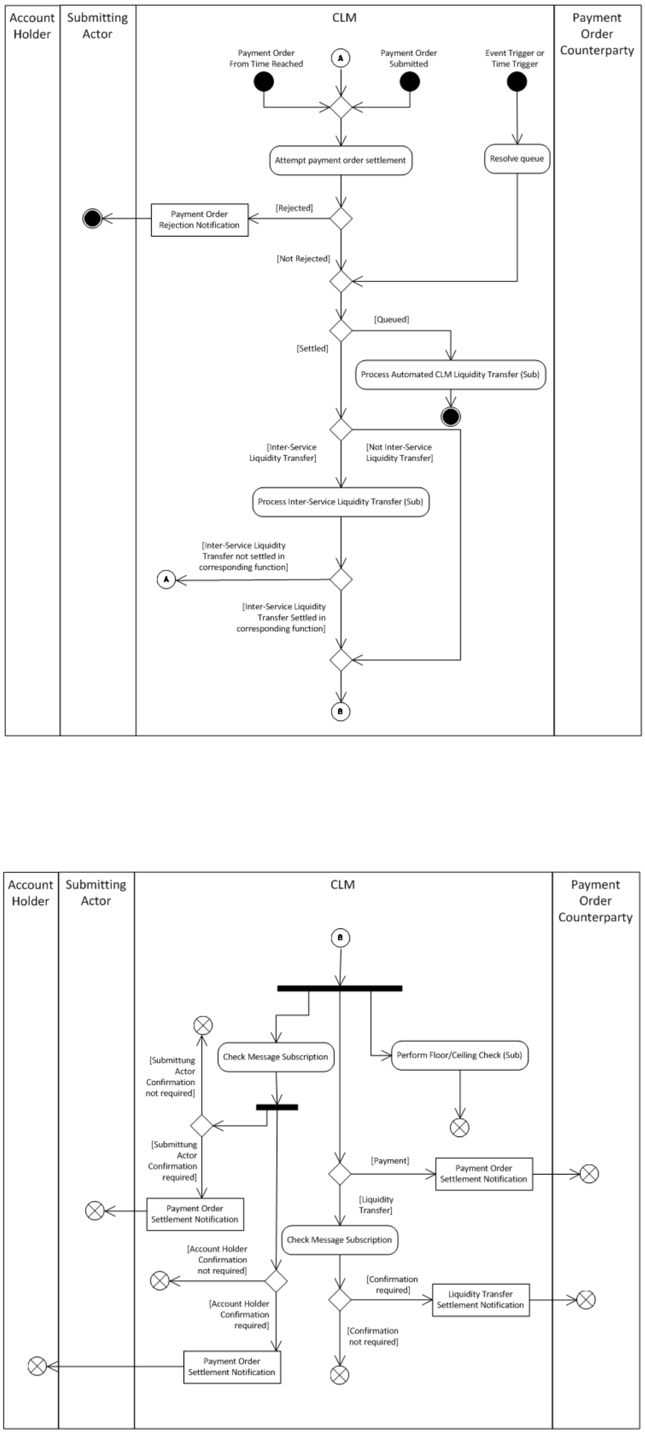

Account Holder
Submitting Actor
CLM
Payment Order Counterparty
Payment Order From Time Reached
A
Payment Order Submitted
Event Trigger or Time Trigger
Attempt payment order settlement
Resolve queue
Payment Order Rejection Notification
[Rejected]
[Not Rejected]
[Queued]
[Settled]
Process Automated CLM Liquidity Transfer (Sub)
[Inter-Service Liquidity Transfer]
[Not Inter-Service Liquidity Transfer]
Process Inter-Service Liquidity Transfer (Sub)
[Inter-Service Liquidity Transfer not settled in corresponding function]
A
[Inter-Service Liquidity Transfer Settled in corresponding function]
B
Account Holder
Submitting Actor
CLM
Payment Order Counterparty
B
Check Message Subscription
Perform Floor/Ceiling Check (Sub)
[Submittung Actor Confirmation not required]
[Submittung Actor Confirmation required]
[Payment]
Payment Order Settlement Notification
Payment Order Settlement Notification
[Liquidity Transfer]
Check Message Subscription
[Account Holder Confirmation not required]
[Account Holder Confirmation required]
[Confirmation required]
Liquidity Transfer Settlement Notification
[Confirmation not required]
Payment Order Settlement Notification

Figure 42 - Standard CLM settlement

In the first step, the process *“attempt payment order settlement”* tries to settle the submitted payment order, resulting in one of the following outcomes:

- **[Rejected]** In case settlement of the liquidity transfer is not possible due to insufficient liquidity, the process rejects the liquidity transfer and sends a “Payment Order Rejection Notification” camt.025 Receipt [243] to the submitter of the original incoming camt.050.
- **[Not Rejected]**

  - the payments (central bank operations) settle or queue.
  - the liquidity transfers settle.

In the second step

- for all accepted (not rejected) payment orders
- as well as for all queued payments forwarded to the process "*resolve queue*" in case of an event trigger

the result of the process can be

- **[Queued]** the queueing of central bank operations which cannot settle, triggering the sub-process *automated CLM liquidity transfer*.

  Note: Queueing of liquidity transfers never takes place in CLM. Contrary to RTGS, queueing of liquidity transfers also not occurs in the case of an automatically triggered inter-service liquidity transfer from RTGS.

- **[Settled]** After successful settlement the *payment order counterparty* receives the following messages in case of
  - central bank operations:

    a "*Payment Order Settlement Notification*" camt.054 BankToCustomerDebitCreditNotification [▸ 261] provided that a respective message subscription configuration has been set up in advance.
  - liquidity transfers:

    a "*Liquidity Transfer Settlement Notification*" camt.054 BankToCustomerDebitCreditNotification [▸ 261] provided that a respective message subscription has been set up in advance.

    Note: CLM treats inter-service liquidity transfers that another component initiates as any other intra-service liquidity transfer.

- The *submitting actor* receives the following messages in case of
  - liquidity transfers initiated via camt.050 :

    a "*Payment Order Settlement Notification*" camt.025 Receipt [▸ 243] provided that a respective message subscription configuration has been set up in advance.
  - inter-service liquidity transfers initiated via camt.050 in CLM:

    a "*Payment Order Settlement Notification*" camt.025 Receipt [▸ 243] only after successful settlement in the other service or component provided that a respective message subscription configuration has been set up in advance.

    Note: In case the other service or component could not successfully book the settlement, CLM sends a negative "*Payment Order Settlement Notification*" camt.025 Receipt [▸ 243] including the first error code reported by the other service or component.
  - central bank operations:

    a "*Payment Order Settlement Notification*" pacs.002 PaymentStatusReport [▸ 272] provided that a respective message subscription configuration has been set up in advance.

- The *account holder* receives the following messages provided that the submitting actor and the account holder differ in case of
  - intra-service liquidity transfers initiated via camt.050 :

    a “*Payment Order Settlement Notification*” camt.054 BankToCustomerDebitCreditNotification [▸ 261] provided that a respective message subscription configuration has been set up in advance.
  - inter-service liquidity transfers initiated via camt.050 in CLM:

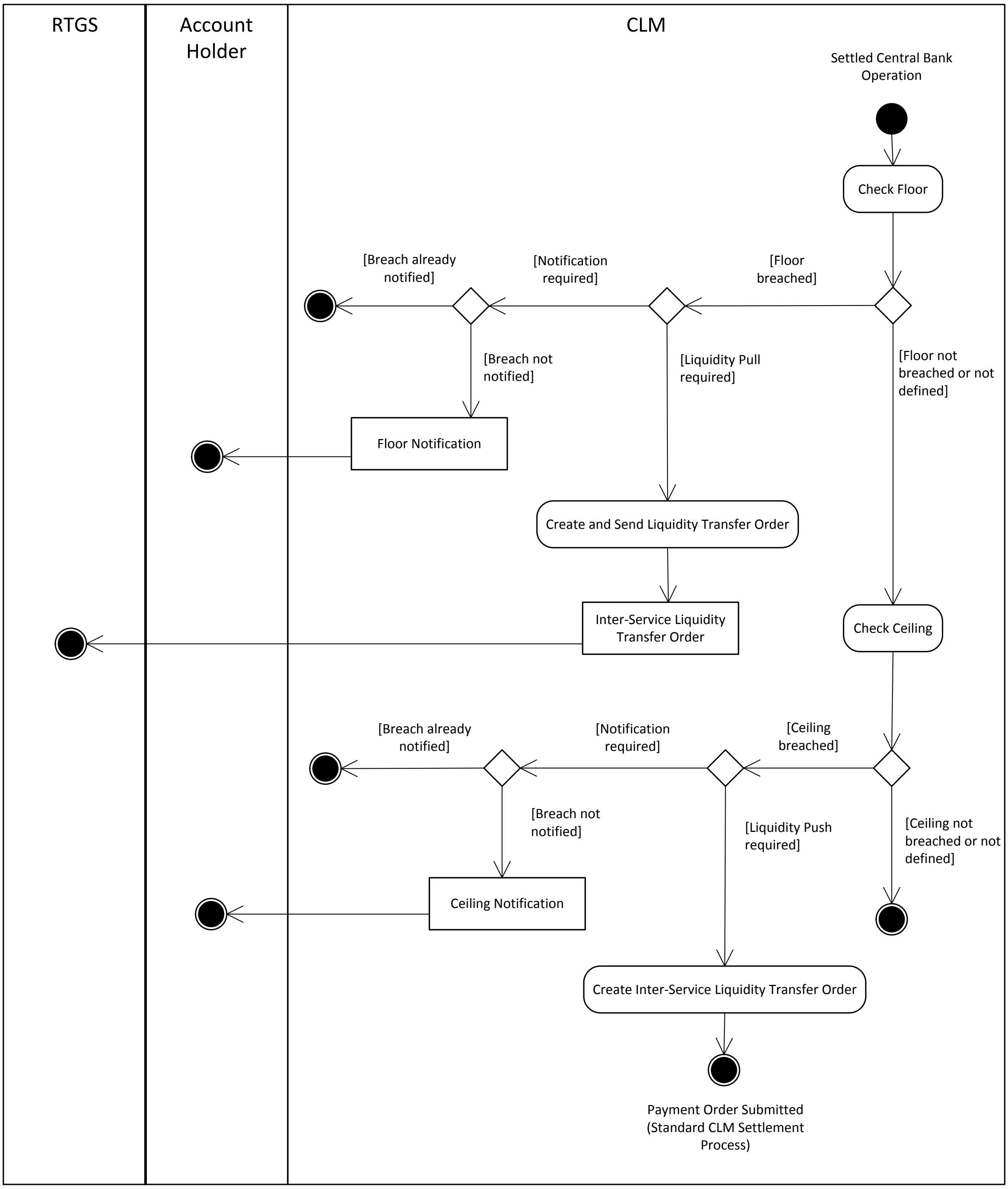
    a “*Payment Order Settlement Notification*” camt.054 BankToCustomerDebitCreditNotification [▸ 261] only after successful settlement in the other service provided that a respective message subscription configuration has been set up in advance.
  - central bank operations:

    a “*Payment Order Settlement Notification*” camt.054 BankToCustomerDebitCreditNotification [▸ 261] provided that a respective message subscription configuration has been set up in advance.

10.3.8.1.1 Floor and ceiling processing (completed)

This process starts after settlement of a central bank operation on the main cash account.

Note: The settlement of liquidity transfers on main cash accounts triggers no floor/ceiling processing.


RTGS
Account Holder
CLM
Settled Central Bank Operation
Check Floor
[Breach already notified]
[Notification required]
[Floor breached]
[Breach not notified]
[Liquidity Pull required]
[Floor not breached or not defined]
Floor Notification
Create and Send Liquidity Transfer Order
Inter-Service Liquidity Transfer Order
Check Ceiling
[Breach already notified]
[Notification required]
[Ceiling breached]
[Breach not notified]
[Liquidity Push required]
[Ceiling not breached or not defined]
Ceiling Notification
Create Inter-Service Liquidity Transfer Order
Payment Order Submitted (Standard CLM Settlement Process)

Floor Processing:

- In case
  - – of a breach of a previously defined floor,
  - – the configuration to receive a floor notification has been set up in advance and
  - – no prior notification of the breach to the account holder,

the main cash account holder receives a “*Floor Notification*” camt.004 (ReturnAccount) [229].

- In case
  - Of a breach of a previously defined floor and
  - the configuration to trigger an inter-service liquidity transfer to pull liquidity from the linked RTGS dedicated cash account has been set up in advance

CLM sends to the RTGS an inter-service liquidity transfer order as camt.050 (LiquidityCreditTransfer) [▸ 255] in order to pull liquidity up to the targeted floor amount.

Ceiling Processing:

- In case
  - of a breach of a previously defined ceiling,
  - the configuration to receive a ceiling notification has been set up in advance and
  - no prior notification of the breach to the account holder before,

the main cash account holder receives a “*Ceiling Notification*” camt.004 (ReturnAccount) [▸ 229].

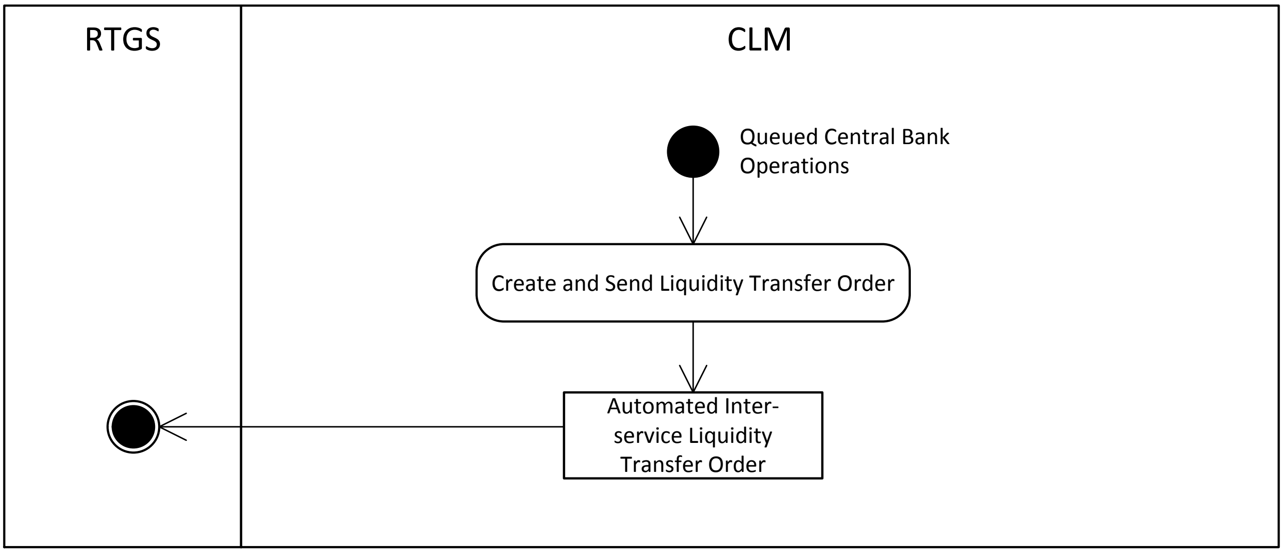
- In case
  - of a breach of a previously defined ceiling has been breached and
  - the configuration to trigger an inter-service liquidity transfer to push liquidity to the linked RTGS dedicated cash account has been set up in advance

CLM sends to the RTGS an inter-service liquidity transfer order as camt.050 (LiquidityCreditTransfer) [▸ 255] in order to push liquidity to reach the predefined target ceiling amount.

10.3.8.1.2 Automated liquidity transfer (completed)

This process starts when a central bank operation does not settle and, therefore is queued.

Note: This process does not apply to liquidity transfers in CLM.


RTGS
CLM
Queued Central Bank Operations
Create and Send Liquidity Transfer Order
Automated Inter-service Liquidity Transfer Order

CLM automatically creates a new automated inter-service liquidity transfer order and sends a camt.050 (LiquidityCreditTransfer) [▸ 255] to RTGS to pull the missing liquidity that the settlement of a central bank operation requires.

10.3.8.2 Connected payment settlement

10.3.8.3 Credit line modification

10.3.8.4 Till/reject time check

10.3.9 CLM end-of-day processing (to be completed in iteration 4)

10.3.9.1 Acquire general ledger data (to be completed in iteration 4)

10.3.9.2 Cross central bank turnover calculation (to be completed in iteration 4)

10.3.9.3 End-of-day cash position calculation (to be completed in iteration 4)

10.3.9.4 Automated marginal lending (to be completed in iteration 4)

10.3.10 CLM start-of-day processing (to be completed in iteration 4)

10.3.10.1 Marginal lending reversal (to be completed in iteration 4)

10.3.10.2 Overnight deposit processing (to be completed in iteration 4)

10.3.10.3 Minimum reserve processing (to be completed in iteration 4)

10.3.11 Revalidate warehoused payments at start of day (to be completed in iteration 4)

10.4 Information services

10.4.1 Execute query (partially completed)

This is a general process description for executing a query, which is similar in all components. In order to retrieve information from a component the submitting actor sends a query request message via ESMIG to the relevant component. Chapter Query management for CLM, CRDM, scheduler and billing [122] describes the respective business scope.

The following activity diagram provides respective processes in the context of the CLM component.

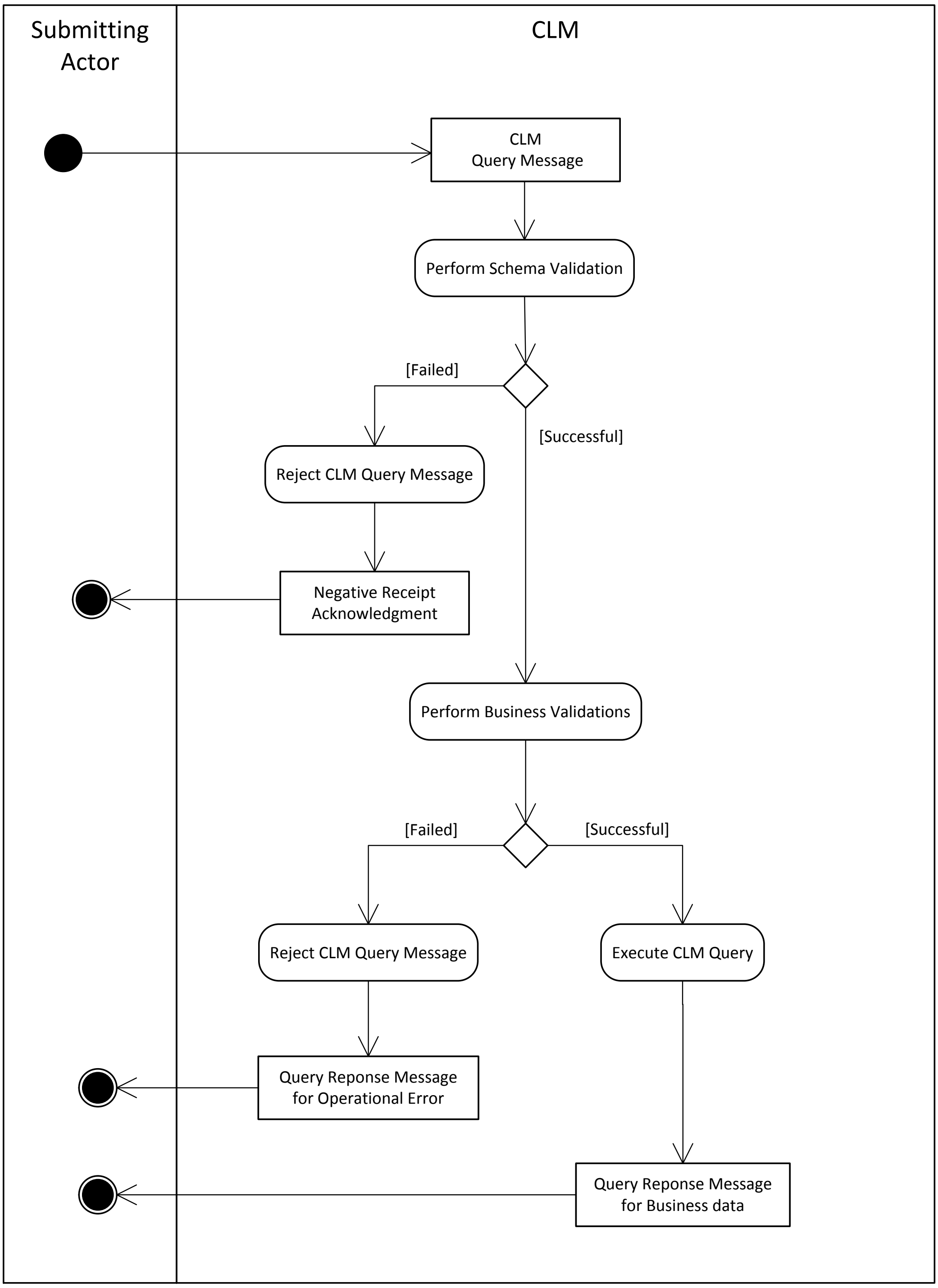

Submitting Actor
CLM
CLM Query Message
Perform Schema Validation
[Failed]
[Successful]
Reject CLM Query Message
Negative Receipt Acknowledgment
Perform Business Validations
[Failed]
[Successful]
Reject CLM Query Message
Execute CLM Query
Query Reponse Message for Operational Error
Query Reponse Message for Business data

Figure 43 - CLM send query

As a first step within the respective component, the process "Perform schema validation" performs the schema validation of the respective [Query Request Message] schema.

- **[failed]** The process sends an admi.007 message [Negative Receipt Acknowledgment] to the submitting actor including all information regarding the reasons for failed validation.
- **[successful]** The process triggers the business validation.

  After successful schema validation, the component performs the business validations (all business rules which are relevant for the respective query including access rights). The validation procedure continues with business validations to the extent possible even after the business validation identifies one or more errors. It reports all identified validation errors.
- **[failed]** The process “Reject query message”message sends a rejection that includes the reasons for failing [Query Response Message for Operational Error] to the submitting actor.
- **[successful]** The process “Execute CLM query” starts. The required business data are extracted; the [Query Response Message for Business Data] and sends the response via ESMIG to the ESMIG to the submitting actor.

The following tables provides a detailed list of A2A messages for query processing.

Table 70 - A2A messages for query processing

Related component	Query type	Query request message	Query response message for operational error	Query response message for business data
CLM	Audit Trail for CLM Query	GetAudit	ReturnAudit	ReturnAudit
CLM	Available Liquidity CLM Query	GetAccount (camt.003) [▸ 228]	ReturnAccount (camt.004) [▸ 229]	ReturnAccount (camt.004) [▸ 229]
CLM (- overall)	Available Liquidity Overall Query	GetAccount (camt.003) [▸ 228]	ReturnAccount (camt.004) [▸ 229]	ReturnAccount (camt.004) [▸ 229]
CLM	CLM Transactions Query	GetTransaction (camt.005) [▸ 232]	ReturnTransaction (camt.006) [▸ 234]	ReturnTransaction (camt.006) [▸ 234]
CLM	Current Reservations Query	GetReservation (camt.046) [▸ 247]	ReturnReservation (camt.047) [▸ 249]	ReturnReservation (camt.047) [▸ 249]
CLM	Minimum Reserve Query	GetAccount (camt.003) [▸ 228]	ReturnAccount (camt.004) [▸ 229]	ReturnAccount (camt.004) [▸ 229]
CRDM	Audit Trail for CRDM Query	GetAudit	ReturnAudit	ReturnAudit
CRDM	Calendar Query	GetCalendar	ReturnCalendar	ReturnCalendar
CRDM	Direct Debit Mandate Query	GetDirectDebit	ReturnDirectDebit	ReturnDirectDebit

Related component	Query type	Query request message	Query response message for operational error	Query response message for business data
CRDM	Directory Query	GetDirectory	ReturnDirectory	ReturnDirectory
CRDM	Error Code Query	GetErrorCode	ReturnErrorCode	ReturnErrorCode
CRDM	Event Query	GetBusinessDayInformation (camt.018) [▸ 238]	ReturnBusinessDayInformation (camt.019) [▸ 240]	ReturnBusinessDayInformation (camt.019) [▸ 240]
CRDM	Main Cash Account Reference Data Query	AccountQuery (acmt.025) [▸ 220]	AccountReport (acmt.026) [▸ 221]	AccountReport (acmt.026) [▸ 221]
CRDM	Participant Reference Data Query	PartyQuery (reda.015) [▸ 278]	PartyReport (reda.017) [▸ 279]	PartyReport (reda.017) [▸ 279]
CRDM	Party Reference Data Query	PartyQuery (reda.015) [▸ 278]	PartyReport (reda.017) [▸ 279]	PartyReport (reda.017) [▸ 279]
CRDM	Standing order Liquidity Transfer Query	GetStandingOrder (camt.069) [▸ 264]	ReturnStandingOrder (camt.070) [▸ 265]	ReturnStandingOrder (camt.070) [▸ 265]
CRDM	Standing order Reservations Query	GetReservation (camt.046) [▸ 247]	ReturnReservation (camt.047) [▸ 249]	ReturnReservation (camt.047) [▸ 249]
Scheduler	System Time Query	GetBusinessDayInformation (camt.018) [▸ 238]	ReturnBusinessDayInformation (camt.019) [▸ 240]	ReturnBusinessDayInformation (camt.019) [▸ 240]

10.4.2 Receive report (partially completed)

This is a general description of the CLM process “Receive report”. CLM uses reports to periodically provided CLM actors with a defined set of data according to their data scope and access rights.

The chapter CLM report generation [▸ 116] describes the respective business scope.

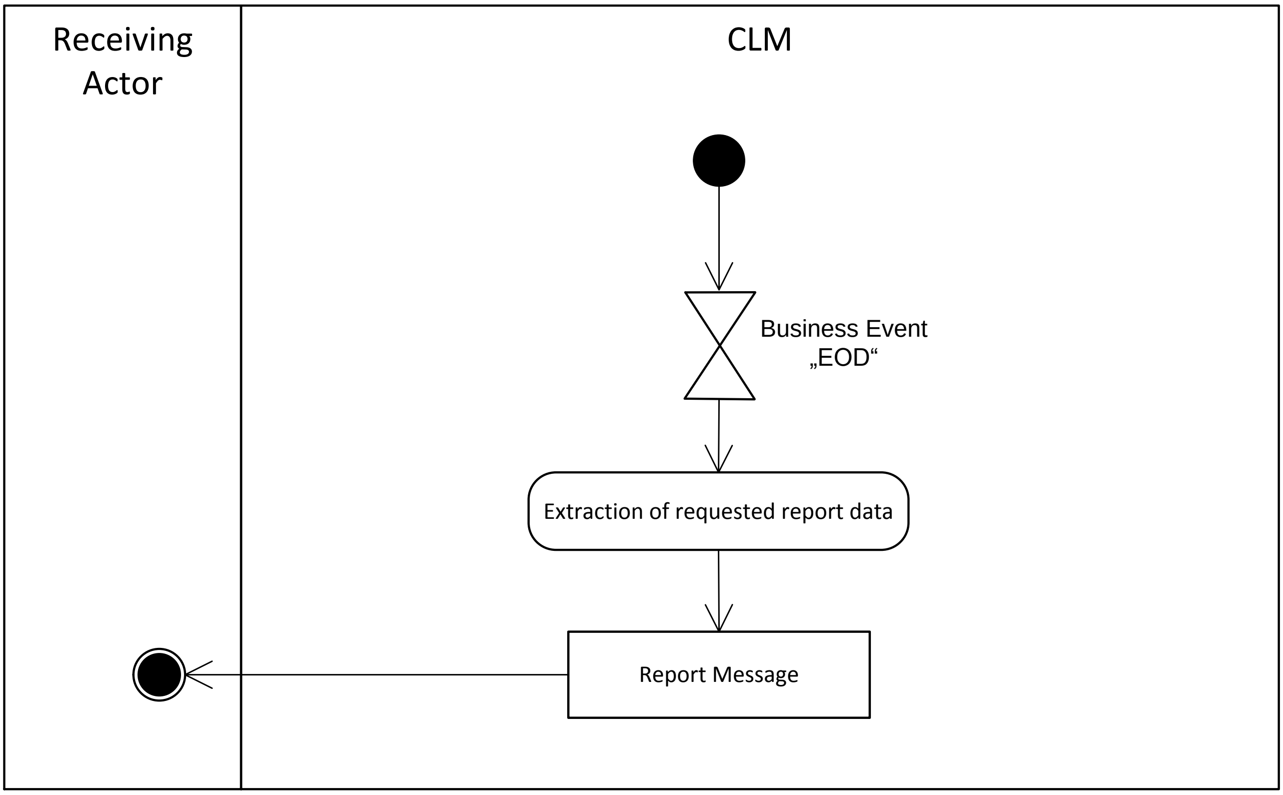


Figure 44 - CLM receive report

The defined business event end of day [EOD] triggers the process “Extraction of requested report data”. The CLM component creates the report, including making the necessary calculations on raw data for aggregated values and storing them for further processing. CLM sends the [Report message] via ESMIG to the receiving actor when a report configuration for the report is set up.

Table 71 - Receive report

Report name	ISO message	ISO code
Statement of accounts	BankToCustomerStatement	BankToCustomerStatement (camt.053) [258]

11 Dialogues and processes

11.1 Dialogues and processes between CRDM and CRDM actor

11.1.1 A2A Common reference data maintenance and query process (completed)

11.1.1.1 Reference data maintenance process (completed)

The common reference data maintenance process can be described as a common message flow that applies to every business scenario.

Upon the sending of a request instructed with an input message, a related response message or a technical validation error message is returned.

11.1.1.1.1 Reference data objects (completed)

The shared generic message flow is as follows:

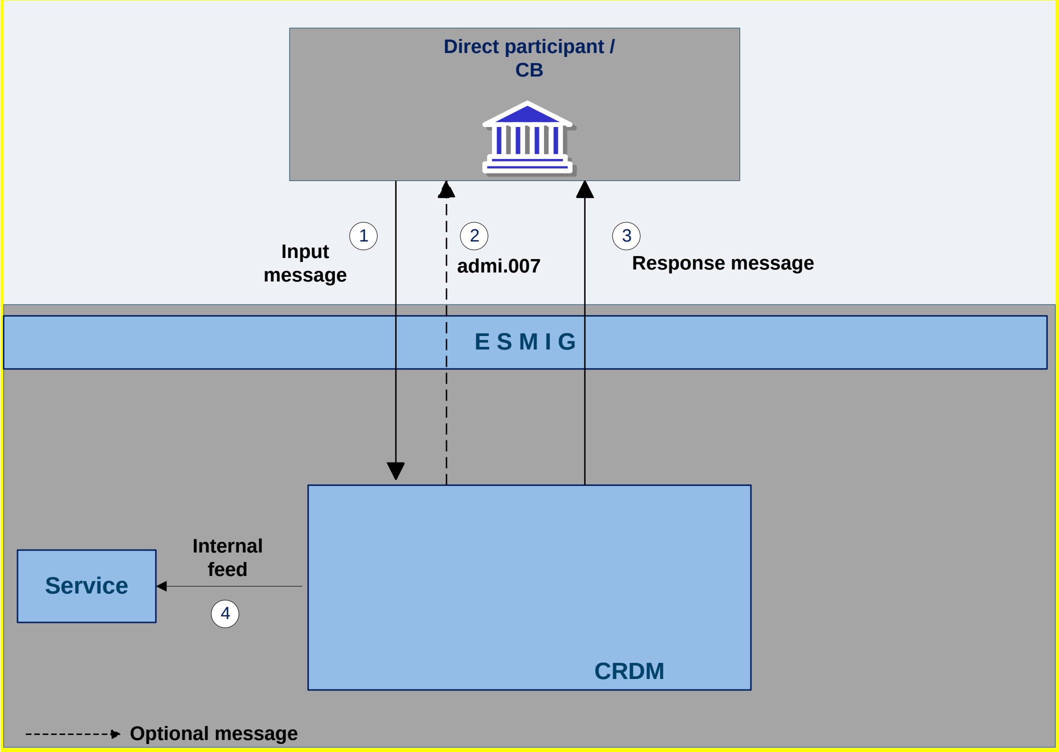


Figure 45 - Common reference data maintenance process

Table 72 - Common reference data maintenance process

Step	Activity
1	The authorised actor (participant, responsible central bank or another actor operating on behalf of the account owner under a contractual agreement) sends the input message to CRDM to create, modify or delete a common reference data entity.
2	In case of rejection upon technical validation, an admi.007, receipt acknowledgement is sent by CRDM to the sender of the originating request.
3	CRDM performs the business validation and sends to the authorised actor a response message to report processing result.
4	CRDM propagates the updated information to the subscribing services for their internal processing.

The messages used in the interaction change depending on the business scenario to be covered.

In the following table, for every concerned common reference data entity and related business scenario, the input and response messages are defined.

Table 73 - Common reference data maintenance messages

Business scenario	Input message	Response message
Create standing order	camt.024	camt.025
Modify standing order	camt.024	camt.025
Delete standing order	camt.071	camt.025
Modify limit	camt.011	camt.025
Delete limit	camt.012	camt.025
Modify standing order for reservation	camt.048	camt.025
Delete standing order for reservation	camt.049	camt.025
Create cash account	acmt.007	acmt.010
Create cash account	acmt.007	acmt.011
Delete cash account	acmt.019	acmt.010
Delete cash account	acmt.019	acmt.011
Modify cash account	acmt.015	acmt.010
Modify cash account	acmt.015	acmt.011
Create party	reda.014	reda.016
Modify party	reda.022	reda.016
Delete party	reda.031	reda.016

11.1.1.2 Common reference data query (completed)

The common reference data query can be described as a common message flow that applies to every business scenario.

Upon the sending of a query instructed with an input message, a related query response message or a technical validation error message is returned.

11.1.1.2.1 Reference data query message coverage (completed)

The shared generic message flow is as follows:

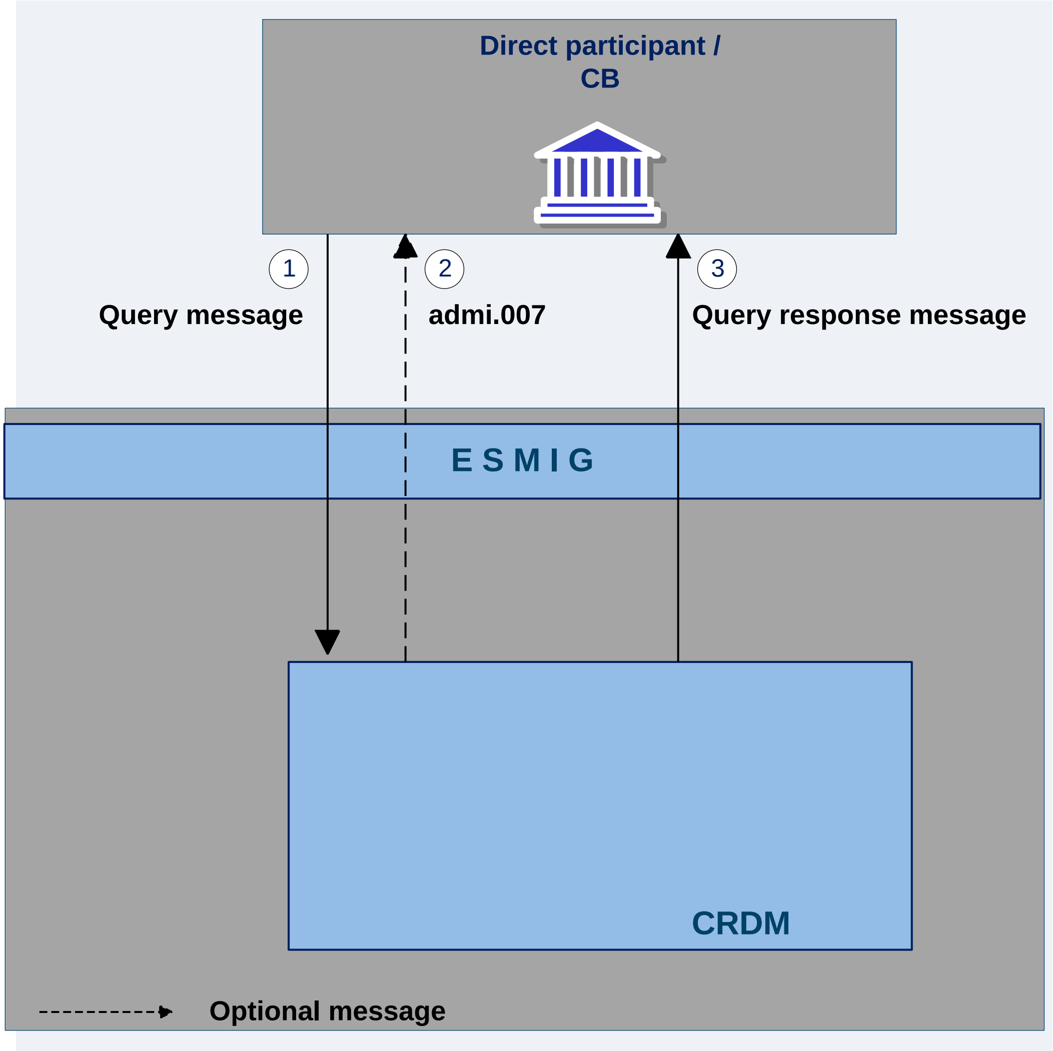


Figure 46 - Common reference data query process

Table 74 - Common reference data query process

Step	Activity
1	The authorised actor (participant or another actor operating on behalf of the owner under a contractual agreement) sends the query message to CRDM to retrieve a set of common reference data entity.
2	In case of rejection upon technical validation, an admi.007, receipt acknowledgement is sent by CRDM to the sender of the originating query.
3	CRDM performs the business validation and sends to the authorised actor a query response message to report processing result, that is retrieved records or business error found during the validation.

The messages used in the interaction change depending on the query to be performed.

In the following table, for every concerned common reference data entity, the query and query response messages are defined.

Table 75 - Common reference data query messages

CRDM entity	Query messages	Query response message
Standing order	camt.069	camt.070
Account	acmt.025	acmt.026
Account audit trail	reda.039	reda.040
Party	reda.015	reda.017
Party audit trail	reda.042	reda.043

11.1.2 Data migration tool file upload (completed)

11.1.2.1 Introduction (completed)

This use case covers the standard situation of a central bank or payment bank CRDM actor loading reference data into common CRDM common component. The upload use case is available via U2A through a dedicated section.

The user uploading the file is propagated to the related back-end functions and must have the appropriate access right configuration.

11.1.2.2 Activity diagram (completed)

The following diagram details all the processing steps of the DMT file upload use case.

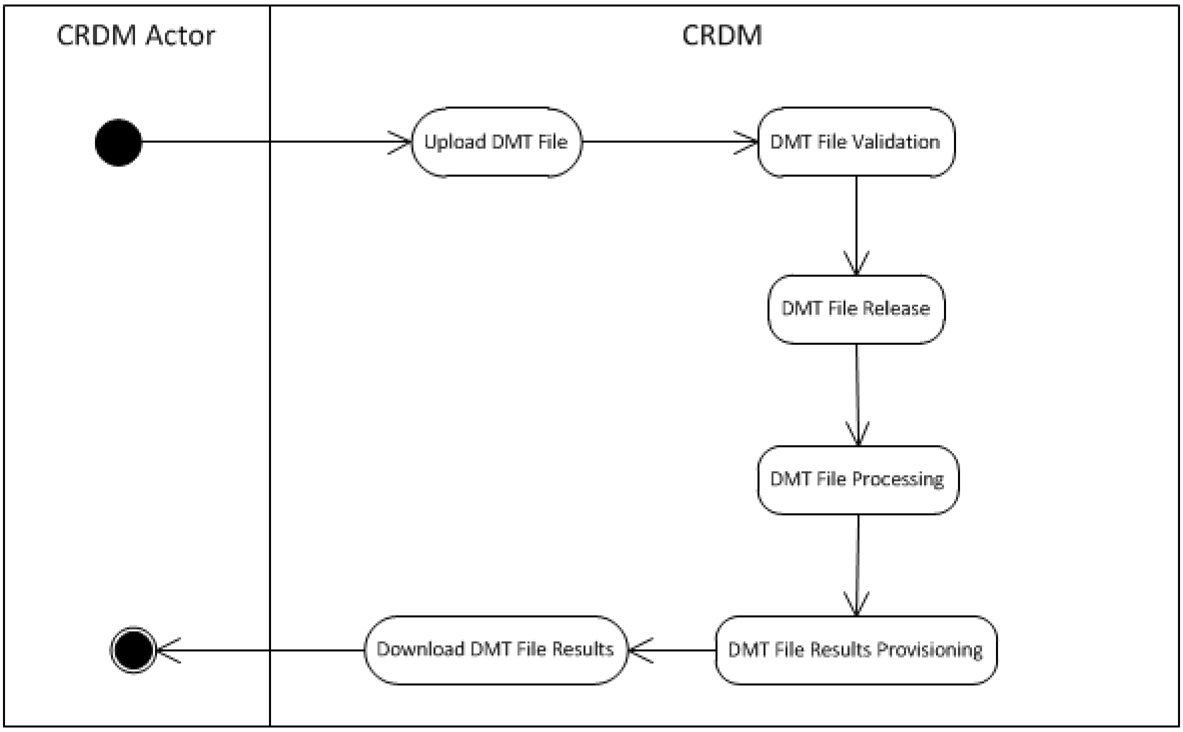


Figure 47 - DMT file upload process

11.1.2.2.1 Upload DMT file (completed)

The CRDM actor uploads the required DMT file containing the reference data to be created in CRDM.

The file can be generated in Excel or Comma Separated Value format and follows the specifications described in Catalogue of messages.

11.1.2.2.2 DMT file validation (completed)

CRDM performs a technical validation on the uploaded file to ensure that the technical constraints are respected.

11.1.2.2.3 DMT file release (completed)

The operator releases the file for the back end module processing as agreed with the actor.

This step triggers the back end module function required by the file as described in the record type label.

11.1.2.2.4 DMT file processing (completed)

The DMT triggers the related back end module function passing information record by record.

Every call to the back end module function generates a result processing.

11.1.2.2.5 DMT file results provisioning (completed)

Once all of the records in the uploaded file are sent and processed by the back end module which provides the related result, the DMT file result is consolidated.

For every record, the successful processing or the business errors receives from the back end module is included in the DMT file results.

The file is published for the CRDM actor to download.

11.1.2.2.6 Download DMT file results (completed)

CRDM actor downloads the result file reporting the number of migrated records and the detailed list of errors for rejected records.

The following table maps the reference data maintenance operations available in the DMT with the related reference data objects and the file specifications contained in chapter Catalogue of messages.

Table 76 - DMT files specifications

Reference data object	Operation	File specifications section
Authorised account user	Create	4.5.3.14
Cash account	Create	4.5.3.12
Certificate DN	Create	4.5.3.10
DN-BIC routing	Create	4.5.3.16
Limit	Create	4.5.3.13
Message subscription rule	Create	4.5.3.8
Message subscription rule set	Create	4.5.3.7
Party	Create	4.5.3.1
Party-service link	Create	4.5.3.15
Privilege	Grant	4.5.3.6
Report configuration	Create	4.5.3.9
Role	Create	4.5.3.4
Role	Grant	4.5.3.5

Reference data object	Operation	File specifications section
Technical address network service link	Create	4.5.3.2
User	Create	4.5.3.3
User certificate DN link	Create	4.5.3.11

11.2 Dialogues and processes between ESMIG and participant (to be completed in iteration 4)

11.3 Dialogues and processes with data warehouse (to be completed in iteration 4)

11.4 Dialogues and processes with billing (to be completed in iteration 4)

III Catalogue of messages

12 Messages - introduction (to be completed in iteration 4)

13 Messages - general information

13.1 Message validation

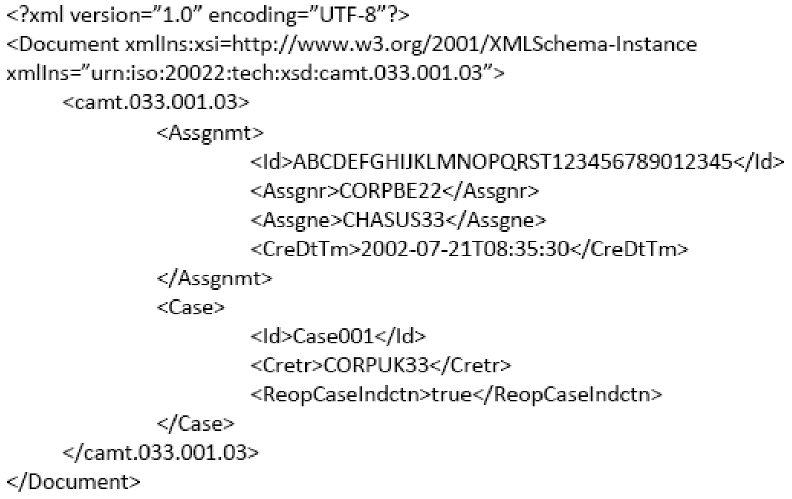
13.1.1 Structure of ISO 20022 messages (completed)

XML schema files conform to the compulsory overall structure foreseen for ISO 20022 messages.

Each schema file requires an XML declaration. This declaration provides information on the used XML version and the applicable character set within the message. XML declarations do not have an end tag as they are not part of the XML document itself and hence do not constitute an XML element.

Below the XML declaration, all schema files have a root element. This root element provides the name of the schema file, including information on the variant and the versionFußnote of the schema file. The actual content of the schema file is hence a sub-element of the root element. Similar to all other elements within the schema file, the root element also has an end tag at the end of the schema file.

The below example provides an indication of the overall structure of ISO 20022 messages.



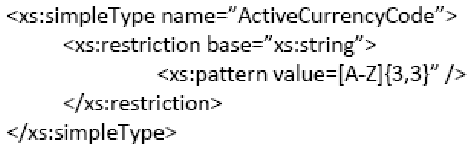
```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlIns:xsi=http://www.w3.org/2001/XMLSchema-Instance
xmlIns="urn:iso:20022:tech:xsd:camt.033.001.03">
        <camt.033.001.03>
                <Assgnmt>
                        <Id>ABCDEFGHIJKLMNOPQRST123456789012345</Id>
                        <Assgnr>CORPBE22</Assgnr>
                        <Assgne>CHASUS33</Assgne>
                        <CreDtTm>2002-07-21T08:35:30</CreDtTm>
                </Assgnmt>
                <Case>
                        <Id>Case001</Id>
                        <Cretr>CORPUK33</Cretr>
                        <ReopCaseIndctn>true</ReopCaseIndctn>
                </Case>
        </camt.033.001.03>
</Document>
```

When being sent as an ISO 20022 message, an XML document is referred to as message instance. The underlying schema file “explains” what makes up a valid message (i.e. it contains the necessary rules and definitions). The message instances themselves consist of message components, choice components and message elements.

Message components are items which are used for setting up a message. These message components contain a set of message elements. In ISO 20022 these message components are usually linked to a particular business component. A comprehensive overview of all standardized ISO 20022 message components is available in the data dictionary of ISO 20022.

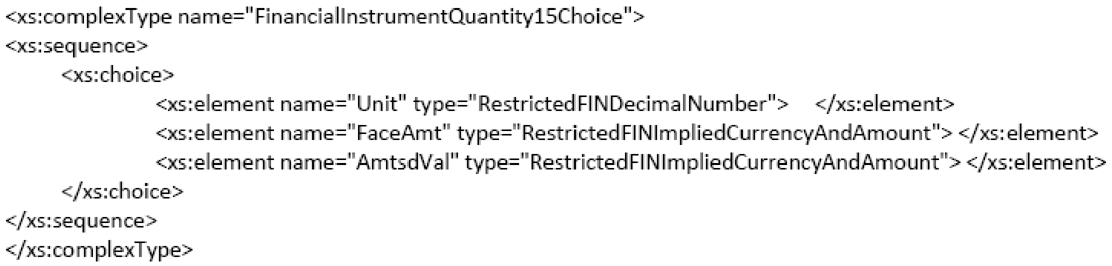
Message elements are the constituents of the message components and are uniquely identified in each component. In ISO 20022 these message elements are usually linked to a particular business element. Filled-in message elements occur as simple and complex data types. All message elements have such a particular type. These data types specify the format of the possible values of a message element.

Simple types serve as a prescription on how to fill the respective message element in the message instance. The simple type shown below prescribes the way in which the currency code must be entered.



```
<xs:simpleType name=”ActiveCurrencyCode”>
        <xs:restriction base=”xs:string”>
                <xs:pattern value=[A-Z]{3,3}” />
        </xs:restriction>
</xs:simpleType>
```

Complex types allow for choice and sequencing options within the message and do not (only) prescribe ways of filling message elements. They hence determine the structure of a message element. The complex type shown below allows for a choice on how to assure party identification in a message.



```
<xs:complexType name="FinancialInstrumentQuantity15Choice">
<xs:sequence>
        <xs:choice>
                <xs:element name="Unit" type="RestrictedFINDecimalNumber">    </xs:element>
                <xs:element name="FaceAmt" type="RestrictedFINImpliedCurrencyAndAmount"> </xs:element>
                <xs:element name="AmtsdVal" type="RestrictedFINImpliedCurrencyAndAmount"> </xs:element>
        </xs:choice>
</xs:sequence>
</xs:complexType>
```

ISO 20022 groups data types into standardized representation classes. These representation classes provide a set of possible data which can be inserted into the concerned message element.

For example, the message element “Bank Identifier” can be assigned to the representation class “BICIdentifier” or message element “Text” can be assigned to the representation class “Max35Text”.

Choice components allow the user of the message to choose between several possibilities. The message user may only choose one possible option in the instance.

Another term which specifies the partitioning within a message instance is the message item. Such a message item can be either a message building block or a message element. Message items which occur as XML tags within the message instance can appear at any level of nesting in the message.

A message building block is a message item which is specific to the concerned message (i.e. the user cannot find it in the ISO 20022 data dictionary). Within the corresponding schema file of the message the building block must be defined as an immediate child of the message. This is not to be confused with reusable groupings of one or more message elements, known as message components (i.e. that the user can find in the ISO 20022 data dictionary).

13.1.2 CLM-specific schema customisation (completed)

Based upon the enriched ISO schema files for its messages, once available (i.e. after the enrichment of newly-developed messages or after the publication of maintained messages in the context of a new standards release) these schema files were customised to adapt them to the specificities applicable in the context of CLM.

The customisation of the schema files used in CLM followed a particular approach which combines the needs of the CLM actors to have a coherent logic across the messages and the need within CLM to have a usable and efficient schema definition. CLM derived this approach from the following customisation principles:

- customised CLM schema files are compliant with the initial ISO 20022 schema files;
- when possible, CLM customisation drops all the message elements with no direct connection to the user requirements of CLM;
- when possible, CLM customisation restricts element types to the CLM-specific usage;
- CLM customisation defines the necessary content of mandatory fields which cannot be pruned (i.e. “removed”) from the ISO schema files;
- CLM customisation restricts the list of possible code values to the sole codes allowed in CLM;
- CLM customisation sets the length of the values to the length applicable in CLM;
- CLM customisation sets the occurrence of message elements to the occurrence applicable in CLM;
- CLM customisation makes optional message elements mandatory if their usage in CLM is always compulsory;
- CLM customisation restricts the allowed characters to those used in CLM with a pattern;
- CLM customisation restricts numeric fields applicable to CLM (e.g. for amounts).

Based on the chosen approach four scenarios apply to the customisation for CLM purposes.

1. A (part of a) message only contains elements which are supported by CLM and there is hence no need for any pruning;
2. CLM does not need a certain element but it cannot be pruned in the message because of a particular customer need;
3. neither CLM nor CLM actors need a certain element and therefore it is pruned;
4. neither CLM nor its users need a certain element but as mandatory element in the ISO schema file it cannot be pruned and may be filled with a dummy value in CLM.

For the scenarios 1, 3 and 4, CLM only allows message elements according to the customised schema file. CLM rejects any inbound message containing message elements which are not part of the CLM customised schema file. Message elements under the scope of scenario 4 are not subject to further processing in CLM. CLM actors can hence fill these fields either with dummy values or real data (inserting real data does not lead to any processing, either).

For scenario 2 an alternative procedure applies. If message elements are present in the message and in the CLM customised schema file although the message element is per se dispensable, CLM nevertheless processes the message. For these message elements only schema validations are applicable. CLM does not validate these elements against its business rules.

However, for all messages, CLM prunes elements which are not within the general scope of its functionalities.

CLM rejects messages during schema validation in cases where actors:

- use elements in the message which are not present in the CLM customised schema file;
- use values in allowed elements but do not respect the restrictions of these values foreseen in the CLM customised schema.

For CLM outbound messages the logic for filling message elements customised to be optional is derived from the concrete circumstances and purposes of the concerned messages.

- For query response messages the filled message elements for outbound messages are those necessary to convey the information requested by the corresponding query message;
- For report messages the same applies, in accordance to the concrete configuration for the subscribed reports;

For any other CLM outbound message the filling of optional fields also depends on either:

- the corresponding inbound message with its specific intention
- or the purpose of the CLM-generated outbound message in case no inbound message precedes.

The sections “The message in business context” may contain message usages and/or message samples in which the content of given fields for a specific purpose or as a reply to a specific inbound message are depicted.

13.1.3 XML character set (completed)

UTF-8 is a Unicode character encoding of variable length. It has the capacity to represent every character of the Unicode character set and is backwards compatible to ASCII (in contrast to UTF-16 or UTF-32). In the vast majority of character representations in UTF-8 it only takes one byte to code one characterFußnote.

UTF-8 is part of the ISO 10646 scheme which was published as a first draft in 1990. The idea is to assign a unique code point to every character (i.e. letters, numbers, symbols, ideograms, etc.) covered by this standard. Whereas the standard foresees a maximum amount of 1.1 million of such code points some 100.000 are attributed to abstract characters for the time being. The inclusiveness, however, is steadily augmenting as characters from previously unrepresented writing systems are added.

The ISO website offers a free-of-charge download of the complete definition of the ISO 10646 standard including all the later amendments (e.g. of additional languages).

Further restrictions to the character set will be defined.

13.1.3.1 Schema validation (completed)

All ISO 20022 messages which arrive at the CLM Interface for further processing are subject to validation rules related to the syntax and structure of the message itself. In this context one can distinguish between well-formedness and validity of the message sent to CLM.

An ISO 20022 message is well-formed if it satisfies the general syntactical rules foreseen for XML documents as outlined in the above chapter. The major aspects to be respected are the following.

The message only contains properly encoded Unicode characters.

- the specific syntax characters (e.g. “<” and “&”) are not used in the message except in their function as mark-up delineation;
- the element-delimiting tags (i.e. start, end and empty-element tags) are correctly nested and paired and none of them is missing or overlapping;
- the start and end tags match exactly and are case-sensitive.

The message has one root element which contains all other elements.

In contrast to other forms of representation the definition of XML documents is rather strict. XML processors cannot produce reasonable results if they encounter even slight violations against the principle of well-formedness. Any violation of this well-formedness automatically entails an interruption of the message processing and an error notification to the sender.

Every well-formed ISO 20022 message arriving at CLM undergoes a validity check according to the rules contained in the enriched CLM schema files. These CLM enriched schemas make the structure of the message visible to the user and provide all necessary explanations on the validations the message undergoes.

The CLM enriched schema files serve different purposes:

- they provide a definition of all the elements and attributes in the message;
- they provide a definition on what elements are child elements and on their specific order and number;
- they provide a definition of the data types applicable to a specific element or attribute;
- they provide a definition of the possible values applicable to a specific element or attribute.

CLM provides the CLM enriched schema file description in several formats: in xsd, Excel and pdf. This shall allow the user to accommodate himself with the format of his choice while having recourse to computer processable information to the largest extent.

A short extract from an xsd schema file for exemplary purposes:

[EXAMPLE xsd schema file of CLM will be added later on]}

Based on the relevant CLM enriched schema, the CLM interface performs the following validations for each incoming message instance:

- validation of the XML structure (starting from the root element);
- validation of the element sequencing (i.e. their prescribed order);
- validation of the correctness of parent-child and sibling relations between the various elements;
- validation of the cardinality of message elements (e.g. if all mandatory elements are present or if the overall number of occurrences is allowed);
- validation of the choice options between the message elements;
- validation of the correctness of the used character set;
- validation of the correctness of the code list values and their format.

13.1.3.1.1 Business validation (completed)

Besides validations which verify the correctness of the ISO 20022 message as XML document itself CLM also conducts validations which are based on the business context CLM operates in.

This business validation in CLM takes place on the basis of a set of pre-defined business rules which are available in the appendix to this document.

On a general level CLM verifies the validity of the transmitted message content against its reference data repository.

In case of violations against existing business rules, CLM transmits them to the relevant CLM actors directly via an outbound message. This message contains all the information the CLM actor needs to fully understand why e.g. an intended step of processing could not be completed by the system.

[EXAMPLE – extract of an outbound message sent in case of business rule validation will be added later on]

13.2 Communication infrastructure

13.2.1 Envelope messages

13.2.1.1 Business application header (completed)

Regardless of any (ongoing) standardisation discussions at ISO level a business application header (BAH) is defined in general for all messages which are used in CLM.

The BAH is not applicable when:

- referring to the acknowledgement of the receipt (admi.007) of a message within CLM;
- technical validation errors identified during the "A2A Business File Validation and Splitting process" are answered from CLM by a ReceiptAcknowledgement (admi.007)

Technically speaking, the application header is a separate XML document standing apart from the XML documents which represent the message instance itself.

The business application header facilitates the message processing as it stores the information necessary for the processing at one central place. Without business application header this information would be either inside the message instance or in the "RequestHeader" of the ISO 20022 message. A uniform appearance (structure) of relevant information in the business application header improves the routing of the message once it arrives at the addressee's interface.

The "Request Payload" stands for the whole communication data which is exchanged between and with CLM.

BAH and Business message (XML message instance) are part of this payload.

For example, the message element contained in the application header allows identifying immediately whether a sent message is a copy of a previously sent message.

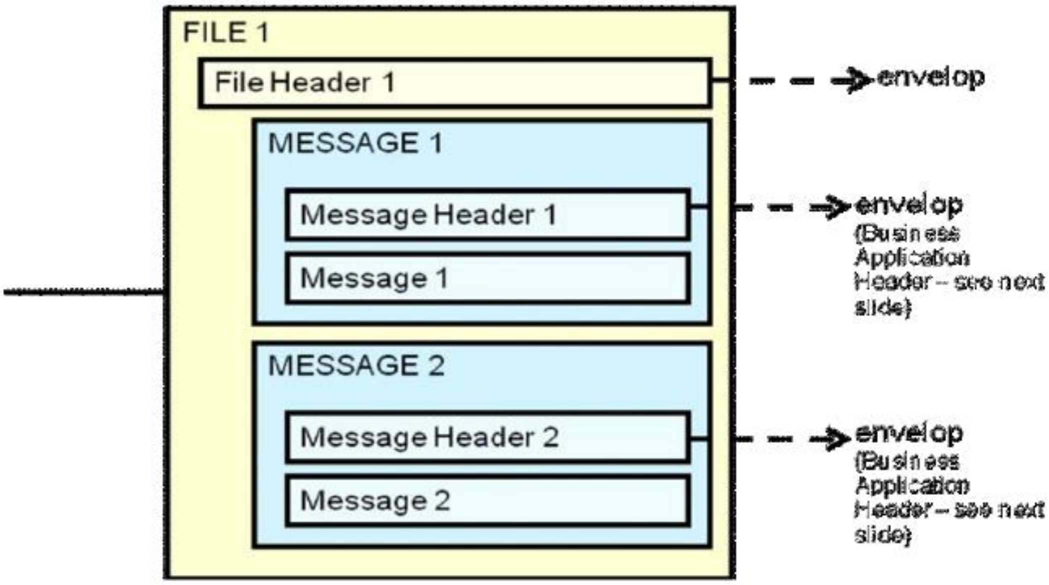
13.2.1.2 Business file header

Besides the sending of single messages CLM supports the exchange of message batches. Therefore, it is possible for the CLM actor to send and receive a file composed of several messages. CLM uses a file header to assure the appropriate processing of such message batch. The file structure within is compliant to the requirement of the "Giovannini Protocol: File Transfer Rulebook (May 2007)".

The file header contains information about the sender, the creation date of the file and the included number of messages. It therefore differs from the application header which is only used to contain additional information regarding one message (i.e. the following message).

Equivalent to all incoming single messages, A2A files arriving at CLM entail a receipt confirmation from CLM. After the successful authentication check CLM divides the file into single messages. Every message undergoes a separate validation (schema validation). CLM reports errors on message level either by the corresponding response message or by a status message.

To communicate a user or an application can send single messages at a different time or a file containing several messages. Both the message and the file are sent within an envelope which can be compared to a cover page as it contains information about the content.



[* An example of the usage of the business file header will be added later]

13.2.1.2.1 Digital Signature managed within the business layer (to be completed in iteration 4)

tbd

13.2.1.3 Time zones (completed)

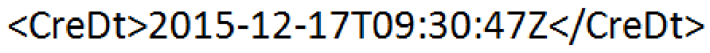
Messages exchanged between CLM and its users consist of the business application header and the message payload. Both parts of the message contain time indications.

The relevant reference for all inbound and outbound communication in CLM is Central European Time (CET) or Central European Summer Time (CEST). All indications contained in the payload of CLM messages (based on given timestamps e.g.) refer to CET/CEST. The attribution of timestamps in CLM solely occurs on CET/CEST basis. All possible information related to time within the payload of messages sent to CLM must refer to CET/CEST. The CLM calendar as the relevant framework for all operational issues of CLM contains CET/CEST only.

Due to the ISO definition of the application header the time indications within the application header refer to Zulu time. CLM users must take into account the difference between the two time formats when exchanging messages with CLM.

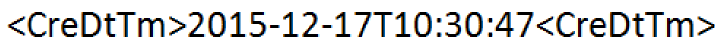
Example

A message sent to CLM on 17 December 2015 at 10:30:47 CET/CEST would need to contain the following field in the application header ("ZULU time"):



```
<CreDt>2015-12-17T09:30:47Z</CreDt>
```

In case the same message contains within the payload an additional reference to the creation date of the message, it would need to contain the following information within the payload ("CET/CEST time"):



```
<CreDtTm>2015-12-17T10:30:47<CreDtTm>
```

13.2.1.4 Outbound traffic exceeding given size limitations (to be completed in iteration 4)

tbd

13.2.1.5 Re-sending of messages (to be completed in iteration 4)

14 List of messages (partially completd)

Table 77 - List of messages

Chapter	Message code	Message name
Administration (admi)		
ResendRequest (admi.006) [▸ 222]	admi.006	ResendRequest
ReceiptAcknowledgement (admi.007) [▸ 224]	admi.007	ReceiptAcknowledgement
Cash Management (camt)		
GetAccount (camt.003) [▸ 228]	camt.003	GetAccount
ReturnAccount (camt.004) [▸ 229]	camt.004	ReturnAccount
GetTransaction (camt.005) [▸ 232]	camt.005	GetTransaction
ReturnTransaction (camt.006) [▸ 234]	camt.006	ReturnTransaction
ModifyTransaction (camt.007) [▸ 237]	camt.007	ModifyTransaction
GetBusinessDayInformation (camt.018) [▸ 238]	camt.018	GetBusinessDayInformation
ReturnBusinessDayInformation (camt.019) [▸ 240]	camt.019	ReturnBusinessDayInformation
Receipt (camt.025) [▸ 243]	camt.025	Receipt
ResolutionOfInvestigation (camt.029) [▸ 245]	camt.029	ResolutionOfInvestigation
GetReservation (camt.046) [▸ 247]	camt.046	GetReservation
ReturnReservation (camt.047) [▸ 249]	camt.047	ReturnReservation
ModifyReservation (camt.048) [▸ 250]	camt.048	ModifyReservation
DeleteReservation (camt.049) [▸ 253]	camt.049	DeleteReservation
LiquidityCreditTransfer (camt.050) [▸ 255]	camt.050	LiquidityCreditTransfer
BankToCustomerStatement (camt.053) [▸ 258]	camt.053	BankToCustomerStatement
BankToCustomerDebitCreditNotification (camt.054) [▸ 261]	camt.054	BankToCustomerDebitCreditNotification

Chapter	Message code	Message name
FIToFIPaymentCancellationRequest (camt.056) [▸ 263]	camt.056	FIToFIPaymentCancellationRequest
Headers (head)		
BusinessApplicationHeader (head.001) [▸ 268]	head.001	BusinessApplicationHeader
BusinessFileHeader (head.002) [▸ 270]	head.002	BusinessFileHeader
Payments Clearing and Settlement (pacs)		
PaymentStatusReport (pacs.002) [▸ 272]	pacs.002	PaymentStatusReport
FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▸ 274]	pacs.009	FinancialInstitutionCreditTransfer (GEN and COV)
FinancialInstitutionDirectDebit (pacs.010) [▸ 276]	pacs.010	FinancialInstitutionDirectDebit

14.1 Account management (acmt)

14.1.1 AccountQuery (acmt.025)

14.1.1.1 Overview and scope of the message

This chapter illustrates the AccountQuery message.

The AccountQuery is sent by an actor authorised to query cash account reference data.

In response to the AccountQuery, an acmt.026 containing the requested information is returned.

14.1.1.2 Schema

Outline of the schema

The AccountQuery message is composed of the following message building blocks:

References

This block is mandatory and contains an identification used to uniquely and unambiguously identify the message.

AccountServicerIdentification

This block is mandatory. It contains the identification of the party receiving the request.

Organisation

This block is mandatory. It contains the identification of the party sending the request.

Account Search Criteria

This block is mandatory and provides with all the search criteria that must be used to filter Account records in the CRDM coverage.

References/links

The schema and the related documentation in HTML/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

https://www.swift.com/mystandards/CSLD/acmt.025.001.002

14.1.2 AccountReport (acmt.026)

14.1.2.1 Overview and scope of the message

This chapter illustrates the AccountReport message.

The AccountReport is sent by CRDM to an authorised actor to provide with requested cash account information.

The AccountReport is sent in response to the acmt.025 message.

14.1.2.2 Schema

Outline of the schema

The AccountReport message is composed of the following message building blocks:

References

This block is mandatory and contains the identification assigned by the sending party to uniquely and unambiguously identify the message and the identification of the original message.

AccountServicerIdentification

This building block is mandatory. It contains the identification of the central bank responsible for the receiving party.

Organisation

This building block is mandatory. It contains the identification of the receiving party.

ReportOrError

This building block is mandatory. It provides either the information matching the search criteria or an error indication.

References/links

The schema and the related documentation in HTML/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

https://www.swift.com/mystandards/CSLD/acmt.026.001.002

14.2 Administration (admi)

14.2.1 ResendRequest (admi.006)

14.2.1.1 Overview and scope of the message

This chapter illustrates the ResendRequest message.

The *ResendRequest* message is sent by directly connected CLM participants to CLM. It is used to request the resending of a message or a file (a duplicate of the original message/file) supported by CLM.

The *ResendRequest* message supports resend requests for the following messages:

- BankToCustomerStatement (camt.053)

The *ResendRequest* message must provide party technical address of the CLM participant to receive the resent message.

In response to the ResendRequest message, CLM sends out either:

- ReceiptAcknowledgement (admi.007) [▸ 224] advising of an error

or, simultaneously:

- ReceiptAcknowledgement (admi.007) [▸ 224] advising of a successful validation
- the requested resend message

14.2.1.2 Schema

Outline of the schema.

The *ResendRequest* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It contains an identification assigned by the sending party to uniquely and unambiguously identify the request message.

ResendSearchCriteria

Defines the criteria required to unambiguously identify the information to be resent.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/admi.006.001.01_CLM

Business rules applicable to the schema

For business rules applicable to ResendRequest please refer to the business rules table below.

14.2.1.3 The message in business context

Usage case: Resend BankToCustomerStatement

Specific message requirements

All content must comply with the business rules for the message.

Table 78 - admi.006_ResendRequest_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: admi006.001.01_CLM_ResendRequest_Example.xml

14.2.2 ReceiptAcknowledgement (admi.007)

14.2.2.1 Overview and scope of the message

This chapter illustrates the *ReceiptAcknowledgement* message.

The *ReceiptAcknowledgement* message is sent by CLM to a directly connected CLM participant. It is used to reject the reception of a previously sent message, or to notify the success of a ResendRequest(admi.006).

Within CLM this message is generated after a negative authentication process. It can be also sent as an error reporting response to a report query or resend request and as a validation result notification to a resend request.

Within CLM, the *ReceiptAcknowledgement* message has the following usages:

- Missing Authentication (without BAH)
- Schema Validation Rejections
- Rejection Resend
- Validation Result Resend
- Oversize and Timeout

In general, the *ReceiptAcknowledgement* is sent by CLM without a BAH.

14.2.2.2 Schema

Outline of the schema.

The *ReceiptAcknowledgement* message is composed of the following message building blocks:

MessageIdentification

This building block is mandatory and provides a set of elements to uniquely identify the receipt acknowledgement message.

Report

This building block is mandatory and is composed of the individual RelatedReference and RequestHandling blocks.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/admi.007.001.07_CLM

Business rules applicable to the schema

No business rules are applicable to a *ReceiptAcknowledgement* message.

14.2.2.3 The message in business context

Usage case: Missing authentication

The Receipt-Acknowledgement message is used in this usage to report that CLM is not able to process an incoming message because of failed authentication of the sending party (sender authentication NOK or decryption NOK).

Specific message content

Table 79 - admi.007_MissingAuthentication_MessageContent

Message item	Data type/code	Utilisation
Reference Document/RctAck/Rpt/RltdRef/Ref	RestrictedFINXMax16Text	MsgID of the incoming message this ReceiptAcknowledgement is sent for
StatusCode Document/RctAck/Rpt/ReqHdlg/StsCd	Max4AlphaNumericText	Status Code indicating the error which occurred during the technical validation. Used in case of BR short names: list TBD
Description Document/RctAck/Rpt/ReqHdlg/Desc	RestrictedFINXMax140Text	Textual description of the technical validation error specified in the status code field. Used in case of BR short names: list TBD

Usage case example: admi.007.001.01_CLM_MissingAuthentication_Example.xml

Usage case: Inbound processing rejections

The ReceiptAcknowledgement is used in this usage by CLM to inform the sender that an incoming message has caused an error during its processing. It reports the error which occurred in an error code and, if available, in a textual description.

Specific message content

Table 80 - admi.007_InboundProcessingRejections_MessageContent

Message item	Data type/code	Utilisation
Reference Document/RctAck/Rpt/RltdRef/Ref	RestrictedFINXMax16Text	MsgID of the incoming message this ReceiptAcknowledgement is sent for
StatusCode Document/RctAck/Rpt/ReqHdlg/StsCd	Max4AlphaNumericText	Status Code indicating the error which occurred during the technical validation
Description Document/RctAck/Rpt/ReqHdlg/Desc	RestrictedFINXMax140Text	Textual description of the technical validation error specified in the status code field

Usage case example: admi.007.001.01_CLM_InboundProcessingRejections_Example.xml

Usage case: RejectionResend

The ReceiptAcknowledgement message is used in this usage to inform the sender about the rejection (check permission resend NOK) of an incoming message.

Specific message content

Table 81 - admi.007_RejectionResend_MessageContent

Message item	Data type/code	Utilisation
Reference Document/RctAck/Rpt/RltdRef/Ref	RestrictedFINXMax16Text	MsgID of the incoming message this ReceiptAcknowledgement is sent for
StatusCode Document/RctAck/Rpt/ReqHdlg/StsCd	Max4AlphaNumericText	Status Code specifiing the missing permission error
Description Document/RctAck/Rpt/ReqHdlg/Desc	RestrictedFINXMax140Text	Permission Denied

Usage case example: admi.007.001.01_CLM_Rejectionresend_Example.xml

Usage case: Validation Result-Resend

The ReceiptAcknowledgement Validation Result Resend message is used in this usage to inform the sender of a message that their request for resending a message could be successfully processed by CLM. It reports the positive status in a code.

Specific message content

Table 82 - admi.007_ValidationResultResend_MessageContent

Message item	Data type/code	Utilisation
Reference Document/RctAck/Rpt/RltdRef/Ref	RestrictedFINXMax16Text	MsgID of the incoming message this ReceiptAcknowledgement is sent for
StatusCode Document/RctAck/Rpt/ReqHdlg/StsCd	Max4AlphaNumericText	Status code “OK”

Usage case example: admi.007.001.01_CLM_ValidationResultResend_Example.xml

Usage case: Oversize and Timeout

The ReceiptAcknowledgement message is used in to inform the sender about an oversize and timeout scenario. The related reference indicates "NONREF". The correlation to the query has to be identified on network layer.

Specific message content

Table 83 - admi.007_OversizeAndTimeout_MessageContent

Message item	Data type/code	Utilisation
Reference Document/RctAck/Rpt/RltdRef/Ref	RestrictedFINXMax16Text	Always populated with NON-REF
StatusCode Document/RctAck/Rpt/ReqHdlg/StsCd	Max4AlphaNumericText	Status Code indicating the error which occurred during the technical validation. Used in case of BR short names: list TBD
Description Document/RctAck/Rpt/ReqHdlg/Desc	RestrictedFINXMax140Text	Textual description of the technical validation error specified in the status code field. Used in case of BR short names: list TBD

Usage case example: admi.007.001.01_CLM_OversizeAndTimeout_Example.xml

14.3 Cash management (camt)

14.3.1 GetAccount (camt.003)

14.3.1.1 Overview and scope of the message

This chapter illustrates the *GetAccount*message.

The *GetAccount* message is sent by a CLM participant (or on their behalf by an authorised party) to CLM. It is used to request balances including credit line of one CLM main cash account held at CLM.

The *GetAccount* message contains the criterion which is used to select the response information.

Within CLM, the *GetAccount* message has the following usages:

- query GetAccount (camt.003) [▹ 228]

This usage is described below, in the chapter "The message in business context".

In response to the *GetAccount* message, a ReturnAccount (camt.004) [▹ 229] message containing the requested information is returned.

14.3.1.2 Schema

Outline of the schema.

The *GetAccount* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

AccountQueryDefinition

This building block is mandatory. It contains detailed information related to the business query criteria about the account.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.003.001.06_CLM

Business rules applicable to the schema

For business rules applicable to *GetAccount* please refer to the business rules table below.

14.3.1.3 The message in business context

Usage case: Query available liquidity

Specific message requirements

All content must comply with the business rules for the message.

Table 84 - camt.003_GetAccount_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.003.001.06_CLM_GetAccount_Example.xml

14.3.2 ReturnAccount (camt.004)

14.3.2.1 Overview and scope of the message

This chapter illustrates the *ReturnAccount* message.

The *ReturnAccount* message is sent by CLM to a CLM participant (or a party authorised by them). It is used to provide information on the balances of one CLM main cash account held at CLM by the CLM participant.

Within CLM, the *ReturnAccount* message has the following usages:

- query Available Liquidity Response
- information to CLM particicpant – Floor notification
- information to CLM particicpant – Ceiling notification

These usages are described below, in the chapter “The message in business context”.

The *ReturnAccount* message is sent in response to a GetAccount (camt.003) [▸ 228] message, which requested the information.

14.3.2.2 Schema

Outline of the schema.

The *ReturnAccount* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query about account, or an error indication.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.004.001.07_CLM

Business rules applicable to the schema

No business rules are applicable to a *ReturnAccount* message.

14.3.2.3 The message in business context

Usage case: Query available liquidity response

Specific message content

Table 85 - camt.004_ReturnAccountQueryresponse_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example 1: camt.004.001.07_CLM_ReturnAccountQueryResponse_Example.xml

When CLM needs to report an error processing the request the following fields are used.

Table 86 - camt.004_ReturnAccountGetAccountQueryResponseErr_MessageContent

Message item	Data type/code	Utilisation
Error Code Document/RtrAcct/RptOrErr/OprlErr/Err/Prtry/	Max35Text	CLM code for the problem being informed.
Error Description Document/RtrAcct/RptOrErr/OprlErr/Err/Desc	Max140Text	Description of the problem being informed.

Usage case example 2: camt.004.001.07_CLM_ReturnAccountQueryResponseErr_Example.xml

Usage case: Information to CLM participant – Floor notification

Specific message content

Table 87 - camt.004_ReturnAccountQueryresponseFloorNotification_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.004.001.07_CLM_ReturnAccountQueryResponseFloorNotififcation_Example.xml

Usage case: Information to CLM participant – Ceiling notification

Specific message content

Table 88 - camt.004_ReturnAccountQueryresponseCeilingNotification_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.004.001.07_CLM_ReturnAccountQueryResponseCeilingNotififcation_Example.xml

14.3.3 GetTransaction (camt.005)

14.3.3.1 Overview and scope of the message

This chapter illustrates the *GetTransaction* message.

The *GetTransaction* message is sent by a CLM participant (or on their behalf by an authorised party) to CLM. It is used to request information about liquidity transfer orders, liquidity transfers, payment orders and payments held in CLM.

The *GetTransaction* message can be used to request payment information based upon multiple criteria.

Within CLM, the *GetTransaction* message has the following usages:

- query transactions of the banking community
- query transactions (CLM and overall)
- query SF transactions of the banking community

These usages are described below, in the chapter “The message in business context”.

In response to the *GetTransaction* message, a ReturnTransaction (camt.006) [▸ 234] message containing the requested information is returned.

14.3.3.2 Schema

Outline of the schema.

The *GetTransaction* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

TransactionQueryDefinition

This building block is mandatory. It contains detailed information related to the business query criteria about the transaction.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.005.001.07_CLM

Business rules applicable to the schema

For business rules applicable to *GetTransaction* please refer to the business rules table below.

14.3.3.3 The message in business context

Usage case: Query transactions of banking community

Specific message requirements

All content must comply with the business rules for the message.

Table 89 - camt.005_GetTransactionQryTxnsOfBankingCommunity_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.005.001.07_CLM_GetTransactionQryTxnsOfBankingCommunity_Example.xml

Usage case: Query transactions CLM and overall

Specific message requirements

All content must comply with the business rules for the message.

Table 90 - camt.005_GetTransactionQryTxnsOfCLMAndOverall_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.005.001.07_CLM_GetTransactionQryTxnsOfCLMAndOverall_Example.xml

Usage case: Query SF transactions of banking community

Specific message requirements

All content must comply with the business rules for the message.

Table 91 - camt.005_GetTransactionQrySFTxnsOfBankingCommunity_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.005.001.07_CLM_GetSFTransactionQryTxnsOfBankingCommunity_Example.xml

14.3.4 ReturnTransaction (camt.006)

14.3.4.1 Overview and scope of the message

This chapter illustrates the *ReturnTransaction*message.

The *ReturnTransaction* message is sent by CLM to a CLM participant (or a party authorised by them). It is used to provide information on the details of one or more liquidity transfer orders, liquidity transfers, payment orders and/or payments held in CLM.

The *ReturnTransaction* message contains such information based upon main cash accounts held at CLM by the CLM participant and upon the criteria provided in the request.

Within CLM, the *ReturnTransaction* message has the following usages:

- response to query transactions of the banking community (GetTransaction (camt.005) [▸ 232])
- response to query transactions (CLM and overall) (GetTransaction (camt.005) [▸ 232])
- response to query SF transactions of the banking community(GetTransaction (camt.005) [▸ 232])

This usage is described below, in the chapter “The message in business context”.

The *ReturnTransaction* message is sent in response to a GetTransaction (camt.005) [▸ 232] message, which requested the information.

14.3.4.2 Schema

Outline of the schema.

The *ReturnTransaction* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query about transaction, or an error indication.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.006.001.07_CLM

Business rules applicable to the schema

No business rules are applicable to a *ReturnTransaction* message.

14.3.4.3 The message in business context

Usage case: Response to query transactions of banking community

Specific message content

Table 92 - camt.006_ReturnTransactionQryTxnsOfBankingCommunityResp_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example 1: camt.006.001.07_CLM_ReturnTransactionQryTxnsOfBankingCommunityResp_Example.xml

When CLM needs to report an error processing the request the following fields are used.

Table 93 - camt.006_ReturnTransactionQryTxnsOfBankingCommunityRespErr_MessageContent

Message item	Data type/code	Utilisation
Error Code Document/RtrTx/RptOrErr/OprlErr/Err/Prtry/	ErrorHandling1Code	CLM code for the problem being informed.
Error Description Document/RtrTx/RptOrErr/OprlErr/Desc	Max140Text	Description of the problem being informed.

Usage case example 2: camt.006.001.07_CLM_ReturnTransactionQryTxnsOfBankingCommunityRespErr_Example.xml

Usage case: Response to query transactions of CLM and overall

Specific message content

Table 94 - camt.006_ReturnTransactionQryTxnsOfCLMAndOverallResp_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.006.001.07_CLM_ReturnTransactionQryTxnsOfCLMAndOverallResp_Example.xml

Usage case: Response to query SF transactions of banking community

Specific message content

Table 95 - camt.006_ReturnTransactionQrySFTxnsOfBankingCommunityResp_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.006.001.07_CLM_ReturnTransactionQrySFTxnsOfBankingCommunityResp_Example.xml

14.3.5 ModifyTransaction (camt.007)

14.3.5.1 Overview and scope of the message

This chapter illustrates the *ModifyTransactionV07* message.

The *ModifyTransaction* message is sent by a CLM participant (or on their behalf by an authorised party) to CLM. It is used to request modification to one liquidity transfer order or one payment order on the CLM participant’s main cash account.

The *ModifyTransaction message* may only be used for an order which is in a transient status (i.e. it has not reached a final status such as rejected, revoked or settled).

The *ModifyTransaction* message contains the new value that the CLM participant wants to be applied to the relevant feature of the order identified in the message.

Within CLM, the *ModifyTransaction* message has the following usages:

- amendment of a payment order

This usage is described below, in the chapter “The message in business context”.

In response to the *ModifyTransaction* message, a Receipt (camt.025) [▸ 243] is sent, indicating the success or rejection/failure of the modification.

To further verify the outcome of the request, the CLM participant may submit a GetTransaction (camt.005) [▸ 232] message with the appropriate search criteria.

14.3.5.2 Schema

Outline of the schema.

The *ModifyTransaction* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

Modification

This building block is mandatory and repetitive. It identifies the list of modifications to be executed.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.007.001.07_CLM

Business rules applicable to the schema

For business rules applicable to *ModifyTransaction* please refer to the business rules table below.

14.3.5.3 The message in business context

Usage case: Amendment of a payment order

Specific message requirements

All content must comply with the business rules for the message.

Table 96 - camt.007_ModifyTransaction_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.007.001.07_CLM_ModifyTransaction_Example.xml

14.3.6 GetBusinessDayInformation (camt.018)

14.3.6.1 Overview and scope of the message

This chapter illustrates the *GetBusinessDayInformation*message.

The GetBusinessDayInformation message is sent by a CLM participant (or on their behalf by an authorised party) to CLM. It is used to request information on different types of administrative data linked to the CLM system.

Within CLM, the *GetAccount* message has the following usages:

- query system time (GetSystemTime)
- query system time (GetBusinessDayInformation)

These usages are described below, in the chapter “The message in business context”.

In response to the *GetBusinessDayInformation* message, a ReturnBusinessDayInformation (camt.019) [▸ 240] message containing the requested information is returned.

14.3.6.2 Schema

Outline of the schema.

The *GetBusinessDayInformation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

BusinessDayInformationQueryDefinition

This building block is mandatory. It contains detailed information related to the business query criteria about the business day information.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.018.001.04_CLM

Business rules applicable to the schema

For business rules applicable to *GetBusinessDayInformation* please refer to the business rules table below.

14.3.6.3 The message in business context

Usage case: Get System Time

Specific message requirements

All content must comply with the business rules for the message.

Table 97 - camt.018_GetBusinessDayInformationGetSystemTime_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.018.001.04_CLM_GetBusinessDayInformationGetSystemTime_Example.xml

Usage case: Get Business Day Information

Specific message requirements

All content must comply with the business rules for the message.

Table 98 - camt.018_GetBusinessDayInformationGetBusinessDayInfo_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.018.001.04_CLM_GetBusinessDayInformationGetBusinessDayInfo_Example.xml

14.3.7 ReturnBusinessDayInformation (camt.019)

14.3.7.1 Overview and scope of the message

This chapter illustrates the *ReturnBusinessDayInformation* message.

The *ReturnBusinessDayInformation* message is sent by CLM to a CLM participant (or a party authorised by them). It is used to provide information on the details of on different types of administrative data linked to the CLM system.

The *ReturnBusinessDayInformation* message contains such administrative data information based upon the criteria provided in the request.

Within CLM, the *ReturnBusinessDayInformation* message has the following usages:

- query System time (GetSystemTime)
- query System time (GetBusinessDayInformation)

These usages are described below, in the chapter “The message in business context”.

The *ReturnBusinessDayInformation* message is sent in response to a GetBusinessDayInformation (camt.018) [▸ 238] message, which requested the information.

14.3.7.2 Schema

Outline of the schema.

The *ReturnBusinessDayInformation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query about business day information, or an error indication.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.019.001.06_CLM

Business rules applicable to the schema

No business rules are applicable to a *ReturnBusinessDayInformation* message.

14.3.7.3 The message in business context

Usage case: Get System Time

Specific message content

Table 99 - camt.019_ReturnBusinessDayInformationGetSystemTime_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example 1: camt.019.001.06_CLM_ReturnBusinessDayInformationGetSystemTime_Example.xml

When CLM needs to report an error processing the request the following fields are used.

Table 100 - camt.019_ReturnBusinessDayInformationGetSystemTimeErr_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example 2: camt.019.001.06_CLM_ReturnBusinessDayInformationGetSystemTimeErr_Example.xml

Usage case: Get Business Day

Specific message content

Table 101 - camt.019_ReturnBusinessDayInformationGetBusinessDayInfo_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example 1: camt.019.001.06_CLM_ReturnBusinessDayInformationGetBusinessDayinfo_Example.xml

When CLM needs to report an error processing the request the following fields are used.

Table 102 - camt.019_ReturnBusinessDayInformationGetBusinessDayInfoErr_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example 2: camt.019.001.06_CLM_ReturnBusinessDayInformationGetBusinessDayInfoErr_Example.xml

14.3.8 ModifyStandingOrder (camt.024)

14.3.8.1 Overview and scope of the message

This chapter illustrates the ModifyStandingOrder message.

The ModifyStandingOrder message is sent by an actor authorised to create or modify standing orders for liquidity transfers.

The ModifyStandingOrder message is replied by a camt.025 to return a positive technical response to the sender of the message or to provide detailed information in case of an error.

14.3.8.2 Schema

Outline of the schema

The ModifyStandingOrder message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides with the message identification provided by the requesting actor.

StandingOrderIdentification

This block is mandatory and provides with all the key information to identify an existing standing order to be amended or a new standing order to be created.

NewStandingOrderValueSet

This block is mandatory and provide with the pieces of information related to the standing order to be modified or created.

It includes the amount to be transferred, the required account references to perform the transfer, the intended validity period and the execution type in terms of event identification.

References/links

The schema and the related documentation in HTML/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

https://www.swift.com/mystandards/CSLD/camt.024.001.05

14.3.9 Receipt (camt.025)

14.3.9.1 Overview and scope of the message

This chapter illustrates the *ReceiptV04* message.

The *Receipt* message is sent by CLM to a CLM participant (or a party authorised by them). It is used to reply to a previously sent liquidity transfer order, payment order or order-related activity.

The *Receipt* message is used to inform the CLM participant regarding the following business activities:

- amendment of a payment order
- amendment of a standing liquidity order
- creation of immediate liquidity transfer order
- liquidity reservation (create, amend, delete)
- overnight deposit reverse transaction
- process liquidity transfer order from dedicated cash account to main cash account
- process liquidity transfer order from main cash account to dedicated cash account
- process liquidity transfer order
- revocation of a payment order (any type of payment order)
- set up overnight deposit

Within CLM, the *Receipt* message has the following usages:

- response to a previously sent message

This usage is described below, in the chapter “The message in business context”.

The *Receipt* message is sent in response to several situations, both as a response to an action, and as an unsolicited update related to a previous action. See above business actions for details.

14.3.9.2 Schema

Outline of the schema.

The *Receipt* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReceiptDetails

This building block is mandatory and non-repetitive. It contains information relating to the status of a previous instruction, with descriptive text if the status indicates a failure.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.025.001.04_CLM

Business rules applicable to the schema

For business rules applicable to *Receipt* please refer to the business rules table below.

14.3.9.3 The message in business context

Usage case: Response to a previously sent message

Specific message content

The actual status value used depends upon the nature of the original requested action, based upon the following table:

Action/Status table – TBD

Table 103 - camt.025_Receipt_MessageContent

Message item	Data type/code	Utilisation
Original message Document/Rct/RctDtls/OrgnlMsgId/MsgId	RestrictedFINXMax16Text	Unique message identification of the original instruction message.
Status Document/Rct/RctDtls/OrgnlMsgId/ReqHdlg/StsCd	Max4AlphaNumericText	Values TBD
Description Document/Rct/RctDtls/OrgnlMsgId/ReqHdlg/desc	RestrictedFINXMax140Text	Descriptive text explaining the reason for rejection of the action request.

Usage case example 1: camt.025.001.04_CLM_Receipt_Example.xml

Usage case example 2: camt.025.001.04_CLM_ReceiptErr_Example.xml

14.3.10 ResolutionOfInvestigation (camt.029)

14.3.10.1 Overview and scope of the message

This chapter illustrates the *ResolutionOfInvestigation* message.

The ResolutionOfInvestigation message is sent by CLM to a CLM participant (or a party authorised by them). It is used to inform of the status of a previously requested cancellation request.

The *ResolutionOfInvestigation* message only concerns the cancellation of one liquidity transfer order or one payment order.

Within CLM, the *ResolutionOfInvestigation* message has the following usages:

- successful cancellation of a liquidity transfer/payment order
- unsuccessful cancellation of a liquidity transfer/payment order

These usages are described below, in the chapter “The message in business context”.

The *ResolutionOfInvestigation* message is sent in response to a FIToFIPaymentCancellationRequest (camt.056) [▸ 263] message.

14.3.10.2 Schema

Outline of the schema.

The *ResolutionOfInvestigation* message is composed of the following message building blocks:

Assignment

Identifies the assignment of an investigation case from an assigner to an assignee. The assigner must be the sender of this message and the assignee must be the receiver.

Status

Indicates the status of the investigation/cancellation.

Cancellation Details

Specifies the details of the underlying transactions being cancelled.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.029.001.07_CLM

Business rules applicable to the schema

No business rules are applicable to a *ResolutionOfInvestigation* response message.

14.3.10.3 The message in business context

Message usage: Successful cancellation of a liquidity transfer/payment order

Specific message content

Table 104 - camt.029_ResolutionOfInvestigationSuccessfulPaymentCancel_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.029.001.08_CLM_ResolutionOfInvestigationSuccessfulPaymentCancel_Example.xml

Message usage: Unsuccessful cancellation of a liquidity transfer/payment

Specific message content

Table 105 - camt.029_ResolutionOfInvestigationUnsccuessfulPaymentCancel_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.029.001.08_CLM_ResolutionOfInvestigationSuccessfulPaymentCancel_Example.xml

14.3.11 GetReservation (camt.046)

14.3.11.1 Overview and scope of the message

This chapter illustrates the *GetReservation*message.

The *GetReservation* message is sent by a CLM participant (or on their behalf by an authorised party) to CLM. It is used to request details of one or more reservation facilities set by the CLM participant (or on their behalf by an authorised party).

The *GetReservation* message can be used to request reservation information based on several criteria.

Within CLM, the *GetReservation* message has the following usages:

- Query reservation

This usage is described below, in the chapter “The message in business context”.

In response to the *GetReservation* message, a ReturnReservation (camt.047) [▸ 249] message containing the requested information is returned.

14.3.11.2 Schema

Outline of the schema.

The *GetReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

ReservationQueryDefinition

Definition of the reservation query.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.046.001.06_CLM

Business rules applicable to the schema

For business rules applicable to *GetReservation* please refer to the business rules table below.

14.3.11.3 The message in business context

Usage case: Query reservation

Specific message requirements

All content must comply with the business rules for the message.

Table 106 - camt.046_GetReservation_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.046.001.04_CLM_GetReservation_Example.xml

14.3.12 ReturnReservation (camt.047)

14.3.12.1 Overview and scope of the message

This chapter illustrates the *ReturnReservationV05* message.

The *ReturnReservation* message is sent by CLM to a CLM participant (or a party authorised by them). It is used to respond to a reservation query.

The *ReturnReservation* message provides details of one or more reservation facilities set by the CLM participant (or on their behalf by and authorised party).

Within CLM, the *ReturnReservation* message has the following usages:

- response to Query reservation

This usage is described below, in the chapter "The message in business context".

The *ReturnReservation* message is sent in response to a GetReservation (camt.046) [▸ 247] message which requested the information.

14.3.12.2 Schema

Outline of the schema.

The *ReturnReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It contains an identification assigned by the sending party to uniquely and unambiguously identify the message and the original business query identification.

ReportOrError

This building block is mandatory and non-repetitive. It contains either the information matching the search criteria of the related business query about limit message, or an error indication. It includes sections such as limit type, the credit consumer identifier, the currency code, the limit amount, the date from which the credit limit is valid.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.047.001.07_CLM

Business rules applicable to the schema

No business rules are applicable to a *ReturnReservation* response message.

14.3.12.3 The message in business context

Usage case: Response to query reservation

Specific message content

Table 107 - camt.047_ReturnReservation_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example 1: camt.047.001.05_CLM_ReservationResponse_Example.xml

The returned business data in case of an error response.

Table 108 - camt.047_ReturnReservation_ErrorContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example 2: camt.047.001.05_CLM_ReservationResponseErr_Example.xml

14.3.13 ModifyReservation (camt.048)

14.3.13.1 Overview and scope of the message

This chapter illustrates the *ModifyReservation* message.

The *ModifyReservation* message is sent by a CLM participant (or on their behalf by an authorised party) to CLM. It is used to request modifications to the details of one particular reservation set by the CLM participant (or on their behalf by an authorised party).

The *ModifyReservation* message contains the new value that the CLM participant wants to be applied to the reservation facility identified in the message.

Within CLM, the *ModifyReservation* message has the following usages:

- Liquidity Reservation (Create)
- Liquidity Reservation (Amend)
- Standing order for Reservation (Create)
- Standing order for Reservation (Amend)

These usages are described below, in the chapter “The message in business context”.

In response to the *ModifyReservation* message, a Receipt (camt.025) [▹ 243] is sent, indicating the success or rejection/failure of the modification.

To further verify the outcome of the request, the CLM participant may submit a GetReservation (camt.046) [▹ 247] message with the appropriate search criteria.

14.3.13.2 Schema

Outline of the schema.

The *ModifyReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

ReservationIdentification

Identification of the reservation (current or default).

NewReservationValueSet

New reservation values.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.048.001.07_CLM

Business rules applicable to the schema

For business rules applicable to *ModifyReservation* please refer to the business rules table below.

14.3.13.3 The message in business context

Usage case: Liquidity reservation (Create)

Specific message requirements

All content must comply with the business rules for the message.

Table 109 - camt.048_ModifyReservationCreateLiquidityReservation_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.048.001.04_CLM_ModifyReservationCreateLiquidityReservation_Example.xml

Usage case: Liquidity reservation (Amend)

Specific message requirements

All content must comply with the business rules for the message.

Table 110 - camt.048_ModifyReservationAmendLiquidityReservation_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.048.001.04_CLM_ModifyReservationAmendLiquidityReservation_Example.xml

Usage case: Standing order for Reservation (Create)

Specific message requirements

All content must comply with the business rules for the message.

Table 111 - camt.048_ModifyReservationCreateReservationSO_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.048.001.04_CLM_ModifyReservationCreateReservationSO_Example.xml

Usage case: Standing order for Reservation (Amend)

Specific message requirements

All content must comply with the business rules for the message.

Table 112 - camt.048_ModifyReservationAmendReservationSO_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.048.001.04_CLM_ModifyReservationAmendReservationSO_Example.xml

14.3.14 DeleteReservation (camt.049)

14.3.14.1 Overview and scope of the message

This chapter illustrates the *DeleteReservationV04* message.

The *DeleteReservation* message is sent by a CLM participant (or on their behalf by an authorised party) to CLM to request the deletion of one particular reservation set by the CLM participant and managed by CLM.

The *DeleteReservation* message allows for the deletion of only one reservation facility.

Within CLM, the *DeleteReservation* message has the following usages:

- delete a standing order for Reservation

This usage is described below, in the chapter “The message in business context”.

In response to the *DeleteReservation* message, a receipt Receipt (camt.025) [▸ 243] is sent, indicating the success or rejection/failure of the deletion.

To further verify the outcome of the request, the CLM participant may submit a GetReservation (camt.046) [▸ 247] message with the appropriate search criteria.

14.3.14.2 Schema

Outline of the schema.

The *DeleteReservation* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

CurrentReservation

Identifies the current reservation to delete.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.049.001.07_CLM

Business rules applicable to the schema

For business rules applicable to *DeleteReservation* please refer to the business rules table below.

14.3.14.3 The message in business context

Usage case: Delete a standing order for Reservation

Specific message requirements

All content must comply with the business rules for the message.

Table 113 - camt.049_DeleteReservation_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.049.001.04_CLM_DeleteReservation_Example.xml

14.3.15 LiquidityCreditTransfer (camt.050)

14.3.15.1 Overview and scope of the message

The *LiquidityCreditTransfer* message is sent by a CLM participant (or on their behalf by an authorised party) to CLM.

The *LiquidityCreditTransfer* message is used to request a transfer of funds between

- two CLM main cash accounts belonging to the CLM participant, or
- two CLM main cash accounts within the same liquidity group of main cash accounts, defined within CLM

with each main cash account being identified using its BIC11.

Within CLM, the *LiquidityCreditTransfer* message has the following usages:

- creation of immediate liquidity transfer order
- overnight deposit reverse transaction
- process liquidity transfer order from dedicated cash account to main cash account
- process liquidity transfer order from main cash account to dedicated cash account
- process liquidity transfer order
- set up overnight deposit

These usages are described below, in the chapter “The message in business context”.

In response to the *LiquidityCreditTransfer* message, a Receipt (camt.025) [243] message containing the status is returned.

14.3.15.2 Schema

Outline of the schema.

The *LiquidityCreditTransfer* message is composed of the following message building blocks:

MessageHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message and type of query.

LiquidityCreditTransfer

This building block is mandatory. It contains detailed information related to the liquidity credit transfer being instructed

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.050.001.04_CLM

Business rules applicable to the schema

For business rules applicable to *LiquidityCreditTransfer* please refer to the business rules table below.

14.3.15.3 The message in business context

Usage case: Create immediate liquidity transfer

Specific message requirements

All content must comply with the business rules for the message.

Table 114 - camt.050_LiquidityCreditTransferImmediateLiquidityTransfer_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.050.001.04_CLM_LiquidityCreditTransferImmediateLiquidityTransfer_Example.xml

Usage case: Overnight deposit reverse

Specific message requirements

All content must comply with the business rules for the message.

Table 115 - camt.050_LiquidityCreditTransferOvernightDepositReverse_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.050.001.04_CLM_LiquidityCreditTransferOvernightDepositReverse_Example.xml

Usage case: Liquidity transfer order from dedicated cash account to main cash account

Specific message requirements

All content must comply with the business rules for the message.

Table 116 - camt.050_LiquidityCreditTransferInterServiceLTOrderDCAtoMCA_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.050.001.04_CLM_LiquidityCreditTransferInterServiceLTOrderDCAtoMCA_Example.xml

Usage case: Liquidity transfer order from main cash account to dedicated cash account

Specific message requirements

All content must comply with the business rules for the message.

Table 117 - camt.050_LiquidityCreditTransferInterServiceLTOrderMCAtoDCA_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.050.001.04_CLM_LiquidityCreditTransferInterServiceLTOrderMCAtoDCA_Example.xml

Usage case: Liquidity transfer order

Specific message requirements

All content must comply with the business rules for the message.

Table 118 - camt.050_LiquidityCreditTransferIntraServiceLTO_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.050.001.04_CLM_LiquidityCreditTransferIntraServiceLTO_Example.xml

Usage case: Set up overnight deposit

Specific message requirements

All content must comply with the business rules for the message.

Table 119 - camt.050_LiquidityCreditTransferOvernightDeposit_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.050.001.04_CLM_LiquidityCreditTransferOvernightDeposit_Example.xml

14.3.16 BankToCustomerStatement (camt.053)

14.3.16.1 Overview and scope of the message

This chapter illustrates the *BankToCustomerStatementV07* message.

The *BankToCustomerStatement* message is sent by CLM to a CLM participant (or a party authorised by them). It is used to inform of the entries booked to an account and account balance information at a given point in time.

The *BankToCustomerStatement* message provides information for cash management and/or reconciliation of information on booked/settled entries only. Optionally it can include details of underlying transactions that have been included in the entry.

Within CLM, the *BankToCustomerStatement* message has the following usages:

- provision of account statement
- query account statement
- query to request a copy of a Report on general ledger
- sending of settlement services' and CLM GL files to central banks

These usages are described below, in the chapter “The message in business context”.

The *BankToCustomerStatement* for account statement message is produced automatically at end of day.

14.3.16.2 Schema

Outline of the schema.

The *BankToCustomerStatement* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

Statement

This building block is mandatory and repetitive. It contains information on booked entries and balances for a CLM dedicated cash account.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.053.001.07_CLM

Business rules applicable to the schema

No business rules are applicable to a *BankToCustomerStatement* message.

14.3.16.3 The message in business context

Usage case: Provision of Account Statement

Specific message content

Table 120 - camt.053_BankToCustomerStatement_MessageContents

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.053.001.07_CLM_BankToCustomerStatement_Example.xml

Usage case: Query Account Statement

Specific message content

Table 121 - camt.053_BankToCustomerStatementQryAccountStatement_MessageContents

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.053.001.07_CLM_BankToCustomerStatementQryAccountStatement_Example.xml

Usage case: Query Copy of Report on General Ledger

Specific message content

Table 122 - camt.053_BankToCustomerStatementQryCopyReportonGL_MessageContents

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.053.001.07_CLM_BankToCustomerStatementQryCopyReportonGL_Example.xml

Usage case: Sending of Settlement Services

Specific message content

Table 123 - camt.053_BankToCustomerStatementSendingOfSettlementServices_MessageContents

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.053.001.07_CLM_BankToCustomerStatementSendingOfSettlementServices_Example.xml

14.3.17 BankToCustomerDebitCreditNotification (camt.054)

14.3.17.1 Overview and scope of the message

This chapter illustrates the *BankToCustomerDebitCreditNotificationV07* message.

The *BankToCustomerDebitCreditNotification* message is sent by CLM to CLM participants (or a party authorised by them). It is used to confirm the credit or the debit of a certain amount on one of their CLM main cash accounts.

The *BankToCustomerDebitCreditNotification* message is sent by CLM when the account-owner was not the instructor of the movement.

The *BankToCustomerDebitCreditNotification* message is only concerned with one single debit or credit movement on one single CLM main cash account.

Within CLM, the *BankToCustomerDebitCreditNotification* message has the following usages:

- credit/debit notification payments
- credit/debit notification connected payments
- credit/debit notification liquidity transfers

This usage is described below, in the chapter “The message in business context”.

The *BankToCustomerDebitCreditNotification* message is sent in response to a debit/credit movement activity within CLM.

14.3.17.2 Schema

<u>Outline of the schema.</u>

The *BankToCustomerDebitCreditNotification* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. It contains an identification assigned by the sending party to uniquely and unambiguously identify the message.

Notification

This building block is mandatory and repetitive. Each repetition notifies of a debit or credit entry for the CLM dedicated cash account.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.054.001.07_CLM

Business rules applicable to the schema

No business rules are applicable to a *BankToCustomerDebitCreditNotification* response message.

14.3.17.3 The message in business context

Message usage: Credit/debit Notification Payments

Specific message content

Table 124 - camt.054_BankToCustomerDebitCreditNotificationPayment-MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.054.001.07_CLM_BankToCustomerDebitCreditNotificationPayment_Example.xml

Message usage: Credit/debit Notification Connected Payments

Specific message content

Table 125 - camt.054_BankToCustomerDebitCreditNotificationConnectedPayment-MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.054.001.07_CLM_BankToCustomerDebitCreditNotificationConnectedPayment_Example.xml

Message usage: Credit/debit Notification Liquidity Transfers

Specific message content

Table 126 - camt.054_BankToCustomerDebitCreditNotificationLiquidityTransfer-MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.054.001.07_CLM_BankToCustomerDebitCreditNotificationLiquidityTransfer_Example.xml

14.3.18 FIToFIPaymentCancellationRequest (camt.056)

14.3.18.1 Overview and scope of the message

This chapter illustrates the *FIToFIPaymentCancellationRequestV07* message.

The *FIToFIPaymentCancellationRequest* message is sent by a CLM participant (or on their behalf by an authorised party) to CLM. It is used to request the cancellation of an original liquidity transfer order or payment order.

The *FIToFIPaymentCancellationRequest* message concerns only one original liquidity transfer order or one payment order.

Within CLM, the *FIToFIPaymentCancellationRequest* message has the following usages:

- revocation of a payment order (any type of payment order)

This usage is described below, in the chapter “The message in business context”.

In response to the *FIToFIPaymentCancellationRequest* message, a ResolutionOfInvestigation (camt.029) [▸ 245] is sent, indicating the success or rejection/failure of the cancellation.

14.3.18.2 Schema

Outline of the schema.

The *FIToFIPaymentCancellationRequest* message is composed of the following message building blocks:

Assignment

Identifies the assignment of an investigation case from an assigner to an assignee. The assigner must be the sender of this message and the assignee must be the receiver.

Underlying

Identifies the payment instruction to be cancelled.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/camt.056.001.07_CLM

Business rules applicable to the schema

For business rules applicable to *FIToFIPaymentCancellationRequest* please refer to the business rules table below.

14.3.18.3 The message in business context

Usage case: Revocation of a payment order

Specific message requirements

All content must comply with the business rules for the message.

Table 127 - camt.056_FiToFiPaymentCancellationRequest_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: camt.056.001.07_CLM_FiToFiPaymentCancellation_Example.xml

14.3.19 GetStandingOrder (camt.069)

14.3.19.1 Overview and scope of the message

This chapter illustrates the GetStandingOrder message.

The GetStandingOrder message is sent by an authorised actor to retrieve standing order information.

The GetStandingOrder message is replied by a camt.070 to return the retrieved standing order information or to provide detailed information in case of an error (e.g. no rows retrieved).

14.3.19.2 Schema

Outline of the schema

The GetStandingOrder message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides with the message Identification provided by the requesting actor.

RequestType

This block is optional and can be used to specify which kind of query must be performed.

StandingOrderQueryDefinition

This block is mandatory and provides with all the search criteria that must be used to filter standing order records in the CRDM coverage. Possible criteria are account and BIC.

References/links

The schema and the related documentation in HTML/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

https://www.swift.com/mystandards/CSLD/camt.069.001.02

14.3.20 ReturnStandingOrder (camt.070)

14.3.20.1 Overview and scope of the message

This chapter illustrates the ReturnStandingOrder message.

The ReturnStandingOrder message is sent by CRDM to an authorised actor to provide with requested standing order information.

The ReturnStandingOrder message is sent as a response to a previously sent camt.069.

14.3.20.2 Schema

Outline of the schema

The ReturnStandingOrder message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides with the message identification provided by the requesting actor as well as the original business query message identification and the request type specifying the kind of query that has been performed.

ReportOrError

This block is mandatory and includes either the retrieved records or the error occurred during the query processing (e.g. no records retrieved).

Report

This block is mandatory and provides with all the pieces of information related to the retrieved standing order.

References/links

The schema and the related documentation in HTML/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

https://www.swift.com/mystandards/CSLD/camt.070.001.03

14.3.21 DeleteStandingOrder (camt.071)

14.3.21.1 Overview and scope of the message

This chapter illustrates the DeleteStandingOrder message.

The DeleteStandingOrder message is sent by an actor authorised to delete standing orders for liquidity transfers.

The DeleteStandingOrder message is replied by a camt.025 to return a positive technical response to the sender of the message or to provide detailed information in case of an error.

14.3.21.2 Schema

Outline of the schema

The DeleteStandingOrder message is composed of the following message building blocks:

MessageHeader

This block is mandatory and provides with the message identification provided by the requesting actor.

StandingOrderDetails

This block is mandatory and provides with all the key information to identify an existing standing order to be deleted. Both identification and account identification must be provided.

References/links

The schema and the related documentation in HTML/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

https://www.swift.com/mystandards/CSLD/camt.071.001.02

14.3.22 GetCreditLine

14.3.22.1 Overview and scope of the message

14.3.22.2 Schema

14.3.22.3 The message in business context

14.3.23 ReturnCreditLine

14.3.23.1 Overview and scope of the message

14.3.23.2 Schema

14.3.23.3 The message in business context

14.4 Headers (head)

14.4.1 BusinessApplicationHeader (head.001)

14.4.1.1 Overview and scope of the message

This chapter illustrates the BusinessApplicationHeader (BAH) message.

For payment messages between bank A and bank B, FROM identifies bank A and TO identifies bank B. For service messages between bank A and the MI (e.g. pacs.009 connected payment, liquidity messages etc.), FROM identifies bank A and TO identifies the MI.

14.4.1.2 Schema

Outline of the schema

The BAH message is composed of the following message building blocks:

FROM

The sender that has created this message for the receiver that processes this message. FROM BIC must have exactly 11 characters.

TO

The receiver designated by the sender who ultimately processes this message. TO BIC must have exactly 11 characters.

BusinessMessageIdentifier

Identifies unambiguously the message. The BusinessMessageIdentifier has maximum 35 characters.

MessageDefinitionIdentifier

Contains the MessageIdentifier that defines the message. It must contain a MessageIdentifier published on the ISO 20022 website.

Business service (optional)

Specifies the business service agreed between the sender and the receiver under which rules this message is exchanged. To be used when there is a choice of processing services or processing service levels. Example: E&I.

CreationDate

Date and time when this message (header) was created.

CopyDuplicate (optional)

Indicates whether the message is a copy, a duplicate or a copy of a duplicate of a previously sent ISO 20022 message.

PossibleDuplicate (optional)

Is a flag indicating if the message exchanged between sender and receiver is possibly a duplicate.

Signature (optional)

Contains the digital signature of the business entity authorised to sign this message.

Related (optional)

Specifies the BAH of the message to which this message relates. It can be used when replying to a query; it can also be used when cancelling or amending.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/head.001.001.01_CLM

Business rules applicable to the schema

14.4.1.3 The message in business context

The BAH contains information to correctly process the message payload by means that every messages exchanged between CLM and the participants respectively CLM and the other services includes such an information. The relation between BAH and message payload is exactly one to one.

The BAH includes the following main information:

- document routing (e.g. sender, receiver, information about the message)
- document identification (e.g. MessageDefinitionIdentifier, creation date and time)
- document processing information (e.g. sender, service, COPY, possible duplicate)

14.4.2 BusinessFileHeader (head.002)

14.4.2.1 Overview and scope of the message

This chapter illustrates the *BusinessFileHeader*message.

The *BusinessFileHeader* is used by directly connected CLM participants to send several business messages within one file to CLM.

Under a single *BusinessFileHeader*, every message within the file has to be an ISO 20022 message together with its business application header (business message). A file can contain one or several business messages.

Within CLM, the *BusinessFileHeader* information is used for:

- consistency and completeness checks

This usage is described below, in the chapter “The message in business context”.

In response to an incoming file which fails validation, CLM sends a ReceiptAcknowledgement (admi.007 [▸ 224]) message containing information on negative validation.

Results from validation which is performed at file level, are sent by CLM without BAH information.

14.4.2.2 Schema

Outline of the schema.

The *BusinessFileHeader* is composed of the following building blocks:

PayloadDescription

The PayloadDescription is a mandatory block and contains the following information tags:

- PayloadDetails: with PayloadIdentifier; CreationDateAndTime and PossibleDuplicateFlag
- ApplicationSpecificInformation: which contains information about the total number of instances (messages) within the file
- PayloadTypeDetails: which declares the payload content (describes the type of business document being exchanged)
- ManifestDetails: with information to each document type and the number of instances (messages) for each declared type.

Payload

The Payload is a mandatory block and contains the set of business messages, each built of an ISO 20022 message together with its business application header.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/head.002.001.01_CLM

Business rules applicable to the schema

For business rules applicable to *BusinessFileHeader* please refer to the business rules table below.

14.4.2.3 The message in business context

Usage case: Resend BankToCustomerStatement

Specific message requirements

All content must comply with the business rules for the message.

Table 128 - head.002_BusinessFileHeader_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: head.002.001.01_CLM_IncomingMessageFileWithinCLM_Example.xml

14.5 Payments clearing and settlement (pacs)

14.5.1 PaymentStatusReport (pacs.002)

14.5.1.1 Overview and scope of the message

This chapter illustrates the *FIToFIPaymentStatusReport* message.

The *FIToFIPaymentStatusReport* message is sent by CLM to a CLM participant (or a party authorised by them). It is used to inform this party about the status of a previous payment order.

The *FIToFIPaymentStatusReport* message is treated as mandatory for all processing failure situations. To receive a *FIToFIPaymentStatusReport* message for normal successful processing situations, subscription is required.

The *FIToFIPaymentStatusReport* message is used as a response/update for the following CLM business activities:

- connected payment processing (FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009) [▸ 274])
- connected payment processing (FinancialInstitutionDirectDebit (pacs.010) [▸ 276])
- delete a standing order
- payments linked to central bank operations (credits)

Within CLM, the *FIToFIPaymentStatusReport* message has the following usages:

- success response to a previously sent message
- rejection response to a previously sent message

These usages are described below, in the chapter “The message in business context”.

The *FIToFIPaymentStatusReport* message is sent in response to several situations, both as a response to an action, and as an unsolicited update related to a previous action. See above business actions for details.

14.5.1.2 Schema

Outline of the schema.

The *FIToFIPaymentStatusReport* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

TransactionInformationAndStatus

Information concerning the original transactions, to which the status report message refers.

References/links

The CLM-specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/pacs.002.001.09_CLM

Business rules applicable to the schema

For business rules applicable to *FIToFIPaymentStatusReport* please refer to the business rules table below.

14.5.1.3 The message in business context

Usage case: Success response to a previously sent message

Specific message content

Table 129 - pacs.002_FIToFIPaymentStatusReportSuccessful_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example pacs.002.001.09_CLM_FIToFIPaymentStatusReportSuccessful_Example.xml

Usage case: Rejection response to a previously sent

Specific message content

Table 130 - pacs.002_FIToFIPaymentStatusReportRejection_MessageContent

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: pacs.002.001.09_CLM_FIToFIPaymentStatusReportRejection_Example.xml

14.5.2 FinancialInstitutionCreditTransfer (GEN and COV) (pacs.009)

14.5.2.1 Overview and scope of the message

This chapter illustrates the *FinancialInstitutionCreditTransfer* message.

This message type can be used for different CLM services:

- liquidity transfers
- central bank operations

High value payments can be sent by a

- direct CLM participant
- central banks as a direct participant or on behalf of a CLM participant (mandated payments)

Credited and debited CLM accounts must be denominated in the same currency.

Within CLM, the *FinancialInstitutionCreditTransfer* message has the following usages:

- connected payment processing
- creation of payment
- payments linked to central bank operations (credits)

These usages are described below, in the chapter “The message in business context”.

In response to the *FinancialInstitutionCreditTransfer* message, a PaymentStatusReport (pacs.002) [▸ 272] is returned.

14.5.2.2 Schema

Outline of the schema.

The *FinancialInstitutionCreditTransfer* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

CreditTransferTransactionInformation

Set of elements providing information specific to the individual credit transfer(s).

References/links

The CLM specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/pacs.009.001.07_CLM

Business rules applicable to the schema

For business rules applicable to *FinancialInstitutionCreditTransfer* please refer to the business rules table below.

14.5.2.3 The message in business context

Usage case: Connected payment processing

Specific message requirements

All content must comply with the business rules for the message.

Table 131 - pacs.009_FinancialInstutionCreditTransferConnectedPayment_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: pacs.009.001.07_CLM_FinancialInstitutionCreditTransferConnectedPayment_Example.xml

Usage case: Creation of payment

Specific message requirements

All content must comply with the business rules for the message.

Table 132 - pacs.009_FinancialInstutionCreditTransferCreatePayment_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: pacs.009.001.07_CLM_FinancialInstitutionCreditTransferCreatePayment_Example.xml

Usage case: Payments linked to central bank operations

Specific message requirements

All content must comply with the business rules for the message.

Table 133 - pacs.009_FinancialInstutionCreditTransferPaymentLinkedToCBO_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Usage case example: pacs.009.001.07_CLM_FinancialInstitutionCreditTransferPaymentLinkedToCBO_Example.xml

14.5.3 FinancialInstitutionDirectDebit (pacs.010)

14.5.3.1 Overview and scope of the message

This chapter illustrates the *FinancialInstitutionDirectDebitV* message.

The *FinancialInstitutionDirectDebit* message is sent by a CLM participant (or on their behalf by an authorised party) to CLM. It is used to move an amount from the CLM main cash account of another CLM participant, to a main cash account of the sending CLM participant.

The *FinancialInstitutionDirectDebit* message concerns only one direct debit movement.

Within CLM, the *FinancialInstitutionDirectDebit* message has the following usages:

- connected payment processing
- payments linked to central bank operations (debits)

These usages are described below, in the chapter “The message in business context”.

In response to the *FinancialInstitutionDirectDebit* message, a PaymentStatusReport (pacs.002) [▸ 272] message containing the status of the movement is returned to the sending CLM participant.

In addition, if the movement is successful, the *FinancialInstitutionDirectDebit* message is forwarded to the debited CLM participant (or a party authorised by them).

14.5.3.2 Schema

Outline of the schema.

The *FinancialInstitutionDirectDebit* message is composed of the following message building blocks:

GroupHeader

This building block is mandatory and non-repetitive. Set of characteristics shared by all individual transactions included in the status report message.

CreditInstruction

Characteristics that apply to the credit side of the payment transaction(s) included in the message.

References/links

The CLM specific schema and documentation in HTML/PDF format as well as the message examples are provided outside of this document under the following link:

http://www.swift.com/mystandards/CLM/pacs.010.001.02_CLM

Business rules applicable to the schema

For business rules applicable to *FinancialInstitutionDirectDebit* please refer to the business rules table below.

14.5.3.3 The message in business context

Usage case: Connected payment processing

Specific message requirements

All content must comply with the business rules for the message.

Table 134 - pacs.010_FIDirectDebitConnectedPayment_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Message example: pacs.010.001.02_CLM_FIDirectDebitConnectedPayment_Example.xml

Usage case: Payments linked to central bank operations

Specific message requirements

All content must comply with the business rules for the message.

Table 135 - pacs.010_FIDirectDebitPaymentLinkedToCBO_MessageRequirements

Message item	Data type/code	Utilisation
TBD	TBD	TBD

Message example: pacs.010.001.02_CLM_FIDirectDebitPaymentLinkedToCBO_Example.xml

14.6 Reference data (reda)

14.6.1 PartyQuery (reda.015)

14.6.1.1 Overview and scope of the message

This chapter illustrates the PartyQuery message.

The PartyQuery is sent by an actor authorised to query party reference data.

In response to the PartyQuery, a reda.017 containing the requested information is returned.

14.6.1.2 Schema

Outline of the schema

The PartyQuery message is composed of the following message building blocks:

MessageIdentification

This building block is mandatory. It must contain an identification assigned by the sending party to uniquely and unambiguously identify the message.

Search Criteria

This block is mandatory and it contains detailed information related to the business party query message.

References/links

The schema and the related documentation in HTML/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

https://www.swift.com/mystandards/CSLD/reda.015.001.001

14.6.2 PartyReport (reda.017)

14.6.2.1 Overview and scope of the message

This chapter illustrates the PartyReport message.

The PartyReport is sent by CRDM to an authorised actor to provide with requested party information.

The PartyReport is sent in response to the reda.015 message.

14.6.2.2 Schema

Outline of the schema

The PartyReport message is composed of the following message building blocks:

MessageHeader

It contains an identification assigned to uniquely and unambiguously identify the message and the identification of the original business query generating the report.

ReportOrError

This building block is mandatory. It contains either the information matching the search criteria of the related query or an error indication.

References/links

The schema and the related documentation in HTML/PDF format as well as the message examples are provided within the MyStandards repository under the following link:

https://www.swift.com/mystandards/CSLD/reda.017.001.001

15 Specific messages for central banks (to be completed in iteration 4)

15.1 camt.007 - specific for central banks

15.1.1 Overview and scope of the message

15.1.2 Schema

15.1.3 The message in business context

15.2 camt.025 - specific for central banks

15.2.1 Overview and scope of the message

15.2.2 Schema

15.2.3 The message in business context

15.3 camt.029 - specific for central banks

15.3.1 Overview and scope of the message

15.3.2 Schema

15.3.3 The message in business context

15.4 camt.056 - specific for central banks

15.4.1 Overview and scope of the message

15.4.2 Schema

15.4.3 The message in business context

IV Appendixes

16 Index and digital signature

16.1 Index of business rules and error codes (partially completed)

Table 136 - Business rules and error codes

BR Name	Description	Inbound messages	Reply messages	Reason code	Error text
	A message structure is valid according to the schema defined for a message.	any message	admi.007		The message is not valid. //Dynamic error including element name.//
	A message type has to be supported	head.001	admi.007		The received single message type is not known in CLM
	The system user sending the inbound A2A communication has to be known.	head.002	admi.007		The System User is not known
	The system user sending the inbound A2A communication must not be locked.	head.002	admi.007		The System User is blocked due to lockout.
	The file header tags which are necessary for authentication processing must be valid according to the XML schema.	head.002	admi.007		At least one BFH tag for authentication is not valid. //Dynamic error including element name.//

BR Name	Description	Inbound messages	Reply messages	Reason code	Error text
	The system user sending the inbound A2A communication has to be known.	head.001	admi.007		The System User sending the inbound A2A communication has to be known.
	The digital signature has to be valid for the business sending user.	head.002	admi.007		Digital signature is not valid for the Business Sending User.
	The business sending user has to be known.	head.002	admi.007		The Business Sending User is not known.
	The business application header tags which are necessary for authentication processing must be valid according to the XML schema.	head.001	admi.007		At least one BAH tag for authentication is not valid. //Dynamic error including element name.//
	The digital signature has to be valid for the business sending user.	head.001	admi.007		Digital signature is not valid for the Business Sending User.
	Business sending user is allowed to send for the system user reference.	head.002	admi.007		Business Sending User is not allowed to send for the system user reference.
	The business sending user has to be known.	head.001	admi.007		The Business Sending User is not known.

BR Name	Description	Inbound messages	Reply messages	Reason code	Error text
	The file must be valid according to the XML schema.	head.002	admi.007		The file is not valid. //Dynamic error including element name.//
	The file must contain at least one individual message.	head.002	admi.007		he file could not be processed, because it does not contain any individual message.
	The file must not have been already processed. The file was sent twice or the reference number of the file was used before by the same business sending party.	head.002	admi.007		The file was sent twice or the reference number of the file was used before. It could only be processed once.

16.2 Index of status value and codes (to be completed in iteration 4)

16.3 Index of instruction references (to be completed in iteration 4)

16.4 Digital signature on business layer (to be completed in iteration 4)

17 Glossary (partially completed)

Term	Definition	Acronym	Source [25]
4CB	The Deutsche Bundesbank (BBk), the Banco de España (BdE), the Banque de France (BdF) and the Banca d'Italia (BdI), collectively, in their capacity as the national central banks responsible for building, maintaining and running T2 services and common components, in accordance with the relevant contractual arrangements and with decisions of the ECB's Governing Council.		CLM/RTGS
4CB network	The 4CB network is the common internal technical network used by the providers of the market infrastructure services.		CLM/RTGS
A2A	See application-to-application.		CLM/RTGS
Account holder	Individual or entity which is authorised to perform transactions on behalf of an account.		CLM/RTGS
Account monitoring group	An optional clustering of accounts for liquidity purposes, e.g. consolidated monitoring, liquidity management.		CLM/RTGS
Act on behalf	Corresponds to the situation when a participant has been granted the authority to perform actions on behalf of one or more other account holders. Central banks are allowed to act on behalf of their participants.		CLM/RTGS
Actor	User defined a dedicated distinguished name which is allowed to interact with one or more T2 service.		CLM/RTGS

[25] In general definitions are taken from the sources provided in this column. Where small variations to the original text have been made, the source is marked with "*".

Term	Definition	Acronym	Source [25]
Algorithm	An algorithm is a mathematical method to provide a smooth, fast and liquidity saving resolution of the payment queue, for example by taking offsetting payment flows into account.	ALG	RTGS
Ancillary system	A system in which payments or securities are exchanged and/or cleared, while the ensuing monetary obligations are settled in another system, typically an RTGS system. Ancillary systems are e.g.: –retail payment systems (RS) –large value payment systems (LVPS) –foreign exchange (FX) systems –money market systems –clearing houses –securities settlement systems (SSS).	AS	RTGS
Application-to-application	A connectivity mode that enables the exchange of information between the application of the service provider and the software application(s) of the actors.	A2A	CLM/RTGS
AS	See ancillary system.		RTGS
Authentication	The methods used to verify the origin of a message or to verify the identity of a participant connected to a system and to confirm that a message has not been modified or replaced in transit.		CLM/RTGS
Availability	The ability of a configuration item or service/component to perform its agreed function when required.		CLM/RTGS
Available liquidity	Credit balance on the account plus collateralised credit line for overdraft (if available).		CLM/RTGS

Term	Definition	Acronym	Source [25]
Backup payments	In the event of a technical system outage a direct participant (affected participant) may lose its ability to send payments to and receive payments from RTGS. In order to give the affected participant the possibility to reduce the business impact of the technical failure, functionality is offered to generate payments via U2A, the so-called backup payments functionality.		RTGS
BAH	See business application header.		CLM/RTGS
Beneficiary	A recipient of funds (payee) or securities. Depending on the context, a beneficiary can be a direct participant in CLM or RTGS and/or a final recipient.		CLM/RTGS
BIC	See business identifier code.		CLM/RTGS
BIC11	In addition to the first eight characters of the BIC, an optional branch code of three characters is used to identify any branch or reference of an institution.		CLM/RTGS
BIC directory	Directory published by SWIFT, part. It contains the business identifier codes (BIC) that SWIFT has registered according to the ISO 9362 standard, and the names and addresses of the corresponding entities.		RTGS
Bilateral/multilateral limit	Instruction of a direct participant to define a bilateral/multilateral limit of a fixed amount within RTGS on a regular basis (time or event triggered).		RTGS
Broadcast	Information message simultaneously available to all or a selected group of participants in CLM and RTGS.		CLM/RTGS

Term	Definition	Acronym	Source[25]
Business application header	The message envelope for business application data that determines which business application the data are routed to and identifies the type of content.	BAH	CLM/RTGS
Business day	The business day comprises and defines the opening times and specific phases per T2 service.		CLM/RTGS
Business identifier code	Identification of financial or non-financial institutions within the financial services industry according to the International Organization for Standardization (ISO) Standard 9362.	BIC	CLM/RTGS
Bypass FIFO	See FIFO by-passing		
CB	See central bank		CLM/RTGS
CBO	See central bank operations		CLM/RTGS
CBS	See central bank services		CLM/RTGS
CCP	See central counterparty		RTGS
Ceiling	An upper threshold of an account balance defined by the participant for initiating a service-specific action.		CLM/RTGS
Central bank	A central bank is the institution responsible for monetary policy and the proper functioning of the monetary system in a country or area.	CB	CLM/RTGS
Central bank operations	Operations initiated by central banks in their capacity as central bank of issue, e.g. monetary policy operations, changes of the credit line.	CBO	CLM/RTGS
Central bank services	Business service managing central bank operations and meeting monetary policy requirements.	CBS	CLM/RTGS

Term	Definition	Acronym	Source [25]
Central counterparty	An entity that interposes itself between the counterparties to the contracts traded in one or more financial markets, becoming buyer to every seller and the seller to every buyer.	CCP	RTGS
Central European Time	Standard time which is one hour ahead of Coordinated Universal Time (UTC).	CET	CLM/RTGS
Central liquidity management	Business component of the T2 services managing and showing funds and credit lines for direct participants and central bank operations. In addition, central component for funding the RTGS component and T2S and TIPS.	CLM	CLM/RTGS
CET	See Central European Time.		CLM/RTGS
Clearing	The process of transmitting, reconciling and, in some cases, confirming payment or securities transfer orders prior to settlement, possibly including the netting of orders and the establishment of final positions for settlement.		CLM/RTGS
Clearing house	A central entity (or central processing mechanism) through which financial institutions agree to exchange transfer instructions for funds or securities. In some cases, the clearing house may act as central counterparty for the participants and therefore assume significant financial risks.		CLM/RTGS
CLM	see central liquidity management.		CLM/RTGS
CLS	See continuous linked settlement.		CLM/RTGS
Collateral	An asset or third-party commitment that is used by the collateral provider to secure an obligation vis-à-vis the collateral taker.		CLM/RTGS

Term	Definition	Acronym	Source [25]
Common reference data management	Business component managing centrally the reference data for all TARGET services and common components.	CRDM	CLM/RTGS
Connected payment	Payments by a central bank or an ancillary system to a participant that trigger a change in the credit line of this participant and an immediate debit/credit of its account to compensate the change in this credit line.		CLM/RTGS
Contingency services	Common component for the management of the emergency situations.		CLM/RTGS
Continuous linked settlement	Payment-versus-payment (PvP) mechanism offered by CLS bank, meaning that a foreign exchange operation is settled only if both counterparties simultaneously have an adequate position in the currency they are selling.	CLS	RTGS
COT	See Cut-off time		CLM/RTGS
CRDM	See common reference data management.		CLM/RTGS
Credit line	A commitment to grant intra-day credit on demand based on collateral provided to a central bank.		CLM/RTGS
Credit transfer	A payment order or, sometimes, a sequence of payment orders made for the purpose of placing funds at the disposal of the beneficiary. Both the payment instructions and the funds described therein move from the bank of the payer/originator to the bank of the beneficiary, possibly via several other banks as intermediaries and/or more than one credit transfer system.		CLM/RTGS

Term	Definition	Acronym	Source [25]
Customer	Entity which is not a participant (direct or indirect) and which uses the service of a participant to exchange transactions in the system.		CLM/RTGS
Cut-off time	The deadline defined by a system (or an agent bank) to accept transfer orders.	COT	CLM/RTGS
Data warehouse	Centralised collection of data from operational business applications in which data are aggregated and optimised for reporting and analysis.	DWH	CLM/RTGS
DCA	See Dedicated Cash Account.		CLM/RTGS
Dedicated cash account	An account dedicated for a single service/component e.g. TIPS, T2S, RTGS.	DCA	CLM/RTGS
Deposit facility	A standing facility of the Eurosystem which counterparties may use to make overnight deposits at a national central bank, which are remunerated at a pre-specified interest rate.		CLM
Direct participant	A participant in T2 services that directly carries out transactions with other participants in the system. He can perform all activities allowed in the T2 services without intermediary.		CLM/RTGS
Distinguished name	A name given to a person, company or element within a computer system or network that uniquely identifies it from everything else	DN	CLM/RTGS
DN	See distinguished name.		CLM/RTGS
DWH	See data warehouse.		CLM/RTGS
EBA	Euro Banking Association.		RTGS
ECB	European Central Bank.		CLM/RTGS
End of Day	End of the defined business day.	EOD	CLM/RTGS

Term	Definition	Acronym	Source [25]
Entry Disposition	A broad set of liquidity management features achieving a flexible and need-based control of the payment flows, thereby limiting possible liquidity risks.		CLM/RTGS
EOD	See end of day.		CLM/RTGS
ESMIG	See Eurosystem single market infrastructure gateway.		CLM/RTGS
Eurosystem single market infrastructure gateway	The common entry point for all interaction with the T2 services, T2S and TIPS. Based on common technical specifications, ESMIG is network agnostic. It allows participants to connect through one or multiple service providers for both A2A and U2A interfaces.	ESMIG	CLM/RTGS
Extensible Mark-up Language	An open standard developed and maintained by World Wide Web Consortium (W3C), for describing and structuring data for the transmission and exchange of information between computer applications and organisations/humans.	XML	CLM/RTGS
FIFO	First in first out.		CLM/RTGS
FIFO by-passing	The system tries to process the first transfer in the queue, but if that cannot be executed owing to lack of funds it then tries to settle the next transfer instead; also called Bypass fifo.		RTGS
File	A file is identified via the file header. It may include zero, one or many single individual messages.		CLM/RTGS
Final (finality)	Irrevocable, unconditional, or not annullable.		CLM/RTGS
Final settlement	Settlement which is irrevocable, unconditional, or not annullable.		CLM/RTGS

Term	Definition	Acronym	Source [25]
Floor	A lower threshold of an account balance defined by the participant for initiating a component-specific action.		CLM/RTGS
Graphical user interface	The interface that allows a user to interact with a software application through the use of graphical elements (e.g. windows, menus, buttons and icons) on a computer screen, using the keyboard and mouse.	GUI	CLM/RTGS
Gridlock	A situation that can arise in a funds or securities transfer system in which the failure of some transfer orders to be executed (because the necessary funds or securities are unavailable) prevents a substantial number of other orders from other participants from being executed.		RTGS
Gross settlement system	A transfer system in which the settlement of funds or securities occurs individually (on an instruction-by-instruction basis).		CLM/RTGS
Guarantee fund mechanism	Mechanism to provide the complementary liquidity needed according to predefined rules in case an ancillary system cannot settle using the settlement banks liquidity only.		RTGS
Guarantee funds account	Account used in case the optional guarantee mechanism has to be activated by an ancillary system or a central bank on its behalf.		RTGS
Guarantor	Owner of the guarantee funds account.		RTGS
GUI	See graphical user interface.		CLM/RTGS
Incident	An event which is not part of the standard operation of the service and which causes, or may cause, an interruption or a reduction of the quality of the T2 services.		CLM/RTGS

Term	Definition	Acronym	Source [25]
Indirect participant	A participant in a funds or securities transfer system with tiering arrangement using a direct participant as intermediary to perform some of the activities allowed in the T2 services.		CLM/RTGS
Instructions	Orders for a service/component e.g. payment order, liquidity transfer order, tasks.		CLM/RTGS
Intraday liquidity	Funds which can be accessed during the business day, usually to enable financial institutions to make payments on an intraday basis.		CLM/RTGS
ISO	International Organization for Standardization		CLM/RTGS
ISO 20022	The international standard for financial services messaging, maintained by the International Organization for Standardization (ISO).		CLM/RTGS
Limit	Amount for normal payments a direct participant is willing to pay to another participant/account (bilateral limit) or to the other participants/accounts (multilateral - limit towards whom no bilateral limit is defined), without having received payments (that are credits) first. For a direct participant it is possible to establish standing orders or current bilateral (respectively multilateral) limits.		RTGS
Liquidity transfer	Liquidity transfer is a payment, the main purpose of which is to transfer liquidity between different accounts of the same participant.	LT	CLM/RTGS
Liquidity transfer group	Liquidity transfer group refers to an optional grouping of cash accounts defined by a central bank for the purpose of arranging liquidity transfers.		CLM/RTGS

Term	Definition	Acronym	Source [25]
Liquidity transfer order	Liquidity transfer order is a payment order, the main purpose of which is to transfer liquidity between different accounts of the same participant. A liquidity transfer order is still not settled.	LTO	CLM/RTGS
Main cash account	Account kept in CLM for provision of credit lines, central bank operations and liquidity management incl. sourcing of dedicated cash accounts.	MCA	CLM
Mandated payment	Payment initiated by an entity that is not party to the transaction (typically by a central bank or an ancillary system in connection with ancillary system settlement) on behalf of another entity. A central bank sends a credit transfer (with specific message structure) on behalf of the failed direct participant (only in case of contingency situations).	MP	CLM/RTGS
Market infrastructure services	Services offered – in this case - by the Eurosystem in the area of payments and security settlements.	MIS	CLM/RTGS
MCA	See main cash account.		CLM/RTGS
Messages	Messages part of the interactive communication between user and service/component		CLM/RTGS
MIS	See market infrastructure services.		CLM/RTGS
Network service provider	A business entity, licensed – in this case - by the Eurosystem, that provides the technical infrastructure, including hardware and software, to establish a secure and encrypted network connection permitting the exchange of information between actors.	NSP	CLM/RTGS
Night-time settlement	Procedure during night time phase.	NTS	RTGS
NSP	See network service provider.		CLM/RTGS

Term	Definition	Acronym	Source [25]
NTS	See night-time settlement.		/RTGS
Offsetting	Offsetting in the RTGS aims at increasing the capacity of the system to settle payments, thereby reducing queues, speeding up the settlement process and reducing the need of intraday liquidity. A bilateral or multilateral offsetting mechanism considers payments in the queues of participants and tries to settle them simultaneously on a gross basis within one legal and logical second.		RTGS
Opening day	See TARGET opening day.		CLM/RTGS
Overnight credit	See marginal lending facility.		CLM
Overnight deposit	See deposit facility.		CLM
Partial settlement	The settlement of only part of a settlement instruction's original amount, when full settlement is not possible owing to lack of cash or securities.		CLM/RTGS
Participant	An entity which is identified/recognized by the system, is bound by rules of the system and is allowed to send and capable to receive transfer orders, either directly (as a direct participant) or indirectly (as an indirect participant).		CLM/RTGS
Party	Any entity defined in the system. This includes: central banks, payment banks, participants and ancillary systems.		CLM/RTGS
Payee	See beneficiary.		CLM/RTGS
Payer	The party to a payment transaction which issues the payment order or agrees to the transfer of funds to a payee.		CLM/RTGS
Payment	A payment is a transfer of funds which discharges an obligation on the part of a payer vis-à-vis a payee.		CLM/RTGS

Term	Definition	Acronym	Source [25]
Payment order	An order or message to initiate a payment .The order may relate either to a credit transfer or to a direct debit.		CLM/RTGS
Payment system	A payment system consists of a set of instruments, banking procedures and, typically, interbank funds transfer systems which facilitate the circulation of money.		CLM/RTGS
Payment versus payment	A mechanism in a foreign exchange settlement system which ensures that a final transfer of one currency occurs if, and only if, a final transfer of the other currency or currencies takes place (e.g. CLS).	PvP	RTGS
Priority	In general, payments are settled immediately, if sufficient liquidity is available on the cash account of the participant. Considering their urgency, they can be submitted and managed by the sender using different priorities.		CLM/RTGS
Privilege	A right, either granted or denied, to execute certain functions within an application or to access and/or update certain data.		CLM/RTGS
Problem	An abnormal state or condition at the component, equipment, or sub-system level, which may lead to a failure that produces incorrect or unexpected results, showing a discrepancy between the relevant specifications and the actual results.		CLM/RTGS
Pull mode	A communication model using the request/response (and query/response) message exchange pattern. A service consumer requests specific information from a service provider and then waits to receive the response.		CLM/RTGS

Term	Definition	Acronym	Source [25]
Push mode	A communication model in which the service provider actively passes event-driven or time-triggered messages to a service consumer based on a subscription by the consumer to the information.		CLM/RTGS
PvP	See payment versus payment.		RTGS
Query	A function to retrieve information from a database using selection criteria to fulfil ad hoc information demands.		CLM/RTGS
Real-time	At the same time as events actually happens.		CLM/RTGS
Real-time gross settlement	The continuous (real-time) settlement of funds or securities transfers individually on an order-by-order basis with intraday finality.	RTGS	CLM/RTGS
Real-time gross settlement system	A settlement system in which processing and settlement take place on a transaction-by-transaction basis in real-time.		CLM/RTGS
Receiver	A participant who obtains the respective message.		CLM/RTGS
Report	An event-driven or time-triggered publishing of information in a defined standard format to specific recipients.		CLM/RTGS
RTGS	See real-time gross settlement.		RTGS
RTGS component	Comprises the processing of high-value payments and ancillary system settlement.		RTGS
Securities settlement system	A transfer system for settling securities transactions. It comprises all of the institutional arrangements required for the clearing and settlement of securities trades and the provision of custody services for securities.	SSS	RTGS

Term	Definition	Acronym	Source [25]
Sender	A participant who initiates the process by sending the respective message to the T2 services.		CLM/RTGS
Service	A set of business functions and provisions.		CLM/RTGS
Service level	The measured and reported achievement against one or more service level targets.		CLM/RTGS
Service level management	The framework of the Eurosystem for specifying services, and monitoring the agreed service levels.	SLM	CLM/RTGS
Service level target	A commitment that is documented in the service level agreement. Service level targets are based on the service levels required to meet business objectives.		CLM/RTGS
Settlement bank	Direct participant who pertains to one or more ancillary systems. The participant may manage the ancillary system settlement process (e.g. the determination of settlement positions, monitoring of the exchange of payments, etc.) not only for own purposes but also for other ancillary system participants on its RTGS dedicated cash account.		RTGS
SoD	Start of day.		CLM/RTGS
SSS	See securities settlement system.		RTGS
Standing liquidity transfer order	Instruction of a direct participant to transfer regularly a fixed amount (time or event triggered) between different accounts (main cash accounts, dedicated cash accounts) of the same participant.		CLM/RTGS

Term	Definition	Acronym	Source [25]
STP	See straight-through processing.		RTGS
Straight-through processing	The automated end-to-end processing of trades/payment transfers, including the automated completion of generation, confirmation, clearing and settlement of instructions.	STP	RTGS
Sub account	Specific account, belonging to an RTGS dedicated cash account, holding dedicated liquidity to allow the settlement of an ancillary system using the interfaced settlement procedure.		RTGS
Systemic risk	The risk that the inability of one institution to meet its obligations when due causes other institutions to be unable to meet their obligations when due. Such failure may cause significant liquidity or credit problems and, as a result, could threaten the stability of or confidence in markets.		CLM/RTGS
T2	See TARGET2.		CLM/RTGS
T2S	See TARGET2-Securities.		CLM/RTGS
TARGET2	The Trans-European Automated Real-time Gross settlement Express Transfer system, which functions in accordance with Guideline ECB/2007/2 of 26 April 2007 (OJ L 237, 8.9.2007, p. 1).	T2	CLM/RTGS
T2 services	T2 services contains of CLM and RTGS.		CLM/RTGS

Term	Definition	Acronym	Source [25]
TARGET2-Securities	The set of hardware, software and other technical infrastructure components through which the Eurosystem provides the services for central securities depositories and central banks that allow core, neutral and borderless settlement of securities transactions on a delivery versus payment basis in central bank money.	T2S	CLM/RTGS
TARGET	Trans-European Automated Real-time Gross settlement Express Transfer: the Eurosystem's real-time gross settlement system for the euro. The first-generation TARGET system was replaced by TARGET2.		CLM/RTGS
TARGET opening day	A day on which settlement takes place according to the daily processing schedule and according to the published calendar of opening days.		CLM/RTGS
Tasks	Tasks are activities in a task queue which need to be performed.		CLM/RTGS
Technical account	Account used in the context of ancillary systems settlement as intermediary account for the collection of debits/credits.		CLM/RTGS
TIPS	Target Instant Payment Settlement: real-time settlement system for retail payments settled in central bank money.		CLM/RTGS
Transaction Reference Number	A unique reference number used to identify each payment instruction.	TRN	CLM/RTGS
Transit account	(Technical) account maintained in CLM and RTGS component, T2S and TIPS for the processing of liquidity transfers.		CLM/RTGS

Term	Definition	Acronym	Source [25]
TRN	See transaction reference number.		CLM/RTGS
U2A	See user-to-application.		CLM/RTGS
UI	See user interaction.		CLM/RTGS
URD	See user requirements document.		CLM/RTGS
User	A user can be an individual person or technical user interacting with the T2 services.		CLM/RTGS
User interaction	Activity by a user undertaken whilst interacting with the market infrastructure services, either through a graphical user interface or via a local software application.	UI	CLM/RTGS
User requirement	A condition or capability needed by a stakeholder to solve a problem or achieve an objective.		CLM/RTGS
User requirements document	The document setting out the user requirements.	URD	CLM/RTGS
User-to-application	A connectivity mode for the exchange of information through a graphical user interface.	U2A	CLM/RTGS
UTC	See coordinated universal time.		CLM/RTGS
V-shape	Type of transmission of messages meaning the addressed platform takes care of the further routing of messages.		CLM/RTGS
Warehoused payment	Payments submitted up to ten calendar days in advanced. In this case, the payment message is warehoused until the day –time settlement phase with the respective date starts.		CLM/RTGS
XML	See Extensible Mark-up Language.		CLM/RTGS