

T2S-0654

Annex 2: General Functional Specifications

Draft sections provided for CR-654 – Penalty Mechanism (updated with CR-715, CR-727 and CR-730)

*Updates of existing GFS sections are shown as underlined text when it is new, or as crossed text when it is removed.*

*For new GFS sections included, the text is not shown underlined because everything is new. Whether a sections is new is market at the beginning of the section in green.*

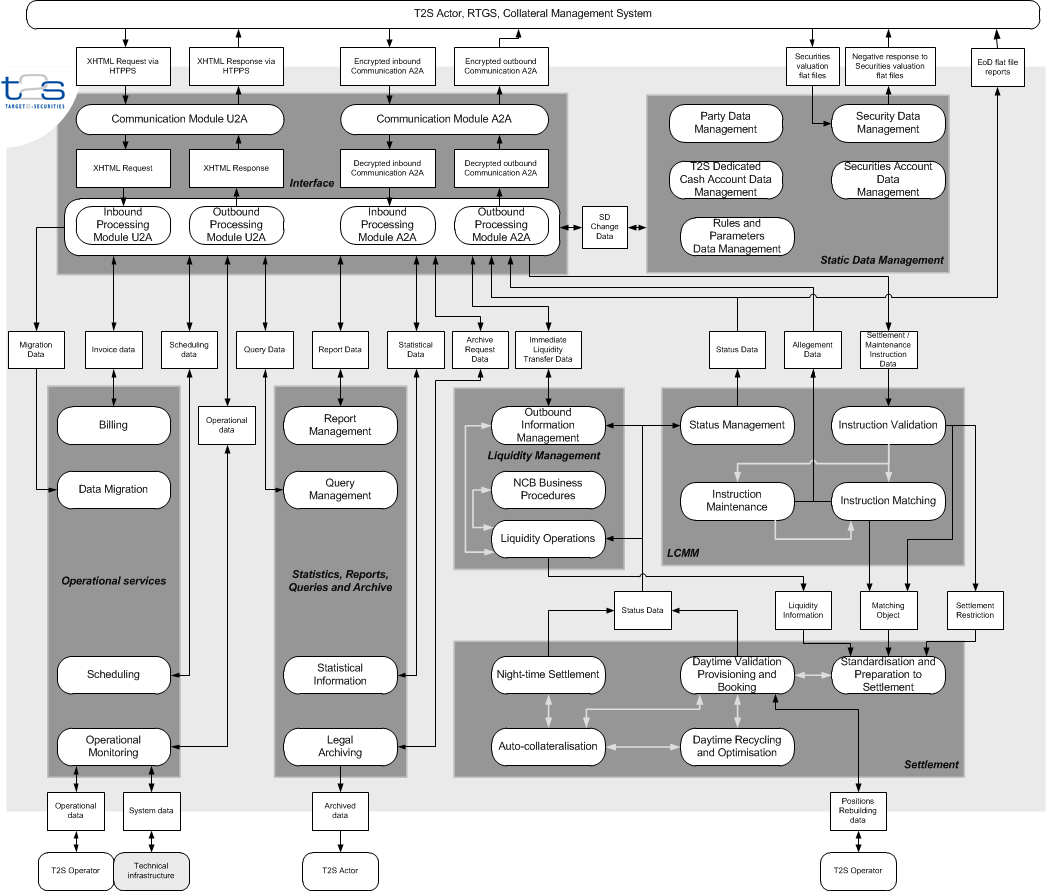
## 2.2 Overall high level diagram

The high-level functional diagram provides an overview of T2S at domains level. It refers to a functional conceptual representation, irrespective of the technical choices to be retained for the implementation of the aforementioned domains.

The following elements are presented on the diagram:

* Domains;
* Modules;
* Data-flows exchanged between the modules and with the outside of the platform.

Only a limited number of flows figures on the diagram in order to limit the complexity of the representation. The exhaustive list of detailed flows can be found in the domains and modules descriptions of chapter 3.

~~~~



…

### 2.3.3 Lifecycle Management and Matching

The *LCMM* domain deals with instructions received through the Interface domain. It is responsible for (i) the validation and matching of settlement instructions, before they are submitted to the *Settlement* domain, and (ii) the management and execution of maintenance instructions, and (iii) the computation and management of Penalties for failed settlement instructions.

This domain is also in charge of checking the possible impact of static data changes on pending instructions, managing the revalidation and the consequences of such impact when relevant, while keeping tracks of the changes in the lifecycle of instructions. The services provided by this domain are available continuously during the whole day T2S operating hours, with the exception of the maintenance window.

The *LCMM* domain encompasses ~~four~~ five modules:

…

#### Penalty Mechanism **(new section)**

The Penalty Mechanism module is responsible for the daily calculation and preparation for reporting of Penalties for failed settlement instructions as well as the complementary features, which are necessary for T2S Actors in this context. Every business day, the Penalty Mechanism module analyses all the failed *Settlement Instruction*s from previous business day in order to compute the necessary Penalties. This module also handles requests from CSDs to modify an existing *Penalty*, as well as the recalculation of *Penalties* that have been modified or impacted by a static data change. This module prepares the information, including the aggregation of amounts (daily or monthly), for the ad-hoc reports for *Penalties*.

…

## 3.2 Interface

### 3.2.4 Inbound Processing Module

#### 3.2.4.3 Description of the functions of the module

##### **3 – Check for U2A Availability**

….

After receiving the *Inbound business data* flow the function checks, if the U2A request has to be rejected due to restricted availability for the dedicated functionality. This is relevant for:

* Liquidity Transfers

– Between Business date change and Change of business date completed and

– During cycle 2 / preparation of sequence y

* Securities position related Queries during sequences with impact on positions
* Balance Queries **{T2S.14.080}** during night-time sequence.
* Penalty queries and modifications during the processing of Penalties.

….

#### 3.2.4.3 Description of the input/output of the module

…

| FLOW | IN/OUT | DESCRIPTION | FROM | TO |
| --- | --- | --- | --- | --- |
| .... |  |  |  |  |
| Inbound LCMM message | OUT |  |  | LCMM: Instruction validation |
| Inbound Penalty Modification message | OUT |  |  | LCMM: Penalty Mechanism |
| Incoming Instruction approval request | OUT |  |  | LCMM: Instruction validation |
| Incoming Instruction approval request | OUT |  |  | LCMM: Penalty Mechanism |
| .... |  |  |  |  |

### 3.2.5 Outbound Processing Module

#### 3.2.5.4 Description of the input/output of the module

…

| FLOW | IN/OUT | DESCRIPTION | FROM | TO |
| --- | --- | --- | --- | --- |
| .... |  |  |  |  |
| Message data | IN |  | LCMM: Status Management |  |
| Outbound Penalty Modification message | IN |  | LCMM: Penalty Mechanism |  |
| Approval request response | IN |  | LCMM: Penalty Mechanism |  |
| .... |  |  |  |  |

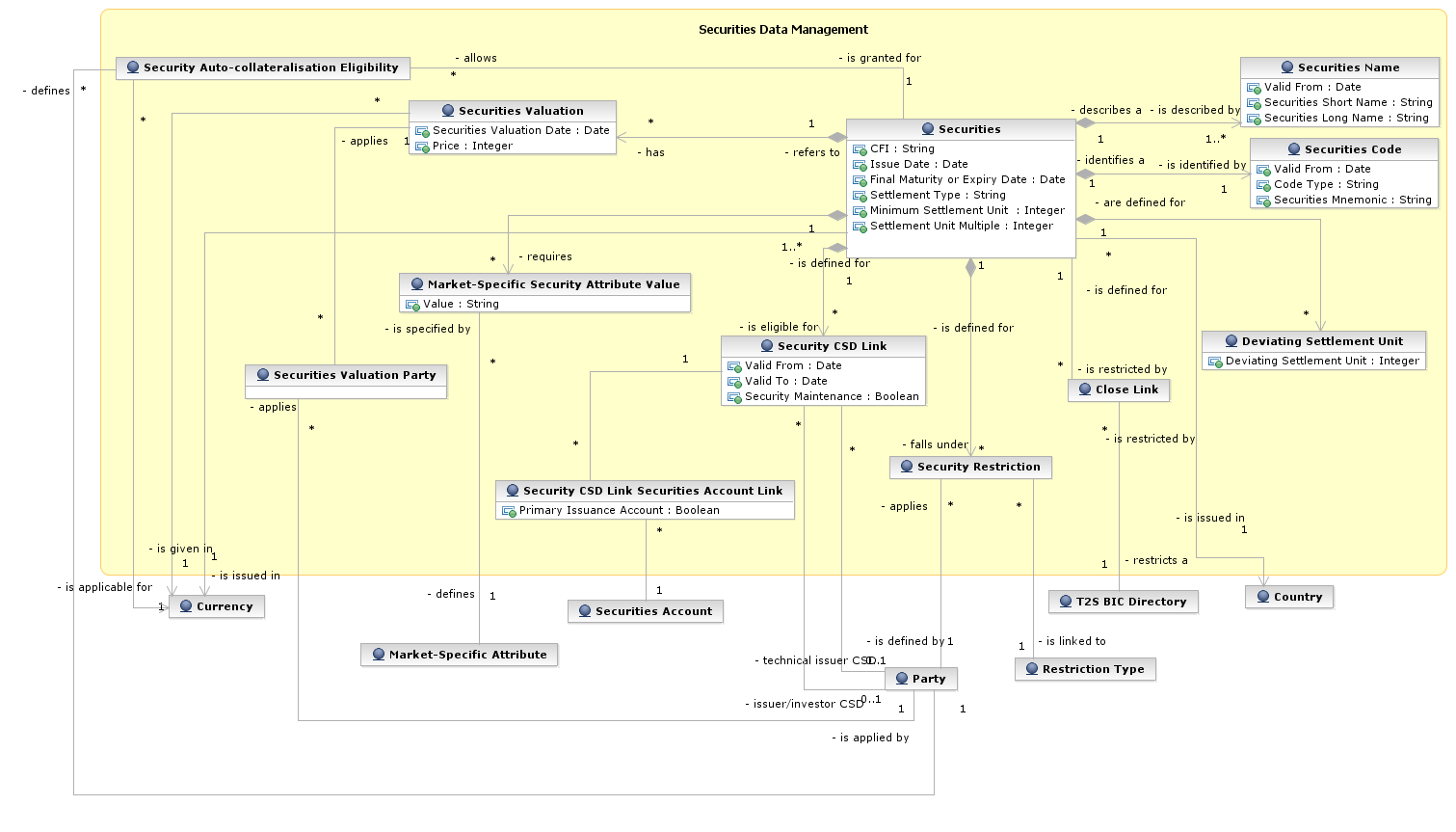
## 3.3. Static Data Management

### Securities Data Management

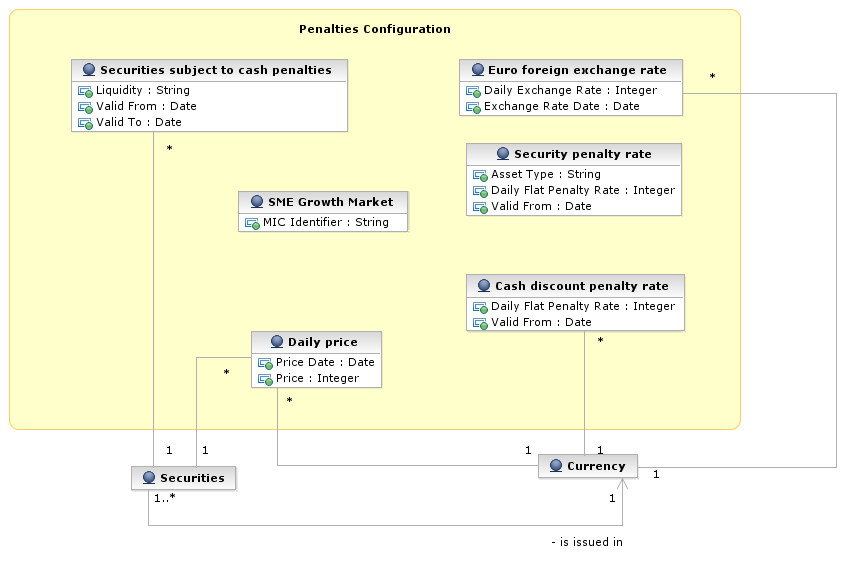
#### Data model of the module

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.DMD.1.1*** |

The following diagram depicts the conceptual data model for Securities Data Management {T2S.16.370}. The scope of securities reference data is limited to the information that T2S requires for settlement.



The following diagram depicts the conceptual data model for *Penalties* configuration related to Securities.



#### Description of the module

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.DEM.1.1*** |

This module provides the services to manage securities reference data, their eligibility for settlement in issuer and investor CSDs and settlement restrictions. It supports the maintenance of close links, valuation prices and the eligibilities of securities in specific currencies for auto-collateralisation offered by NCBs or payment banks in T2S {T2S.16.370}. The functions described below allow the authorised T2S system users (i.e. CSD system administrators) to input their own securities and to access and maintain them, i.e. to create new securities or to update or delete already existing securities as well as managing the cash penalties configuration. When updating a security, each change is applied either as a revision or as an historic item, according to what is specified in the relevant request.

#### Description of the entities

**1 – Securities**

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.ENT.1.1*** |

This entity specifies all securities reference data that do not require a data history, i.e. all the attributes having only one valid value for a given security, regardless of the point in time taken into account {T2S.16.380}.

| ATTRIBUTE | DESCRIPTION |
| --- | --- |
| CFI | Classification of the security according to ISO/DIS 10962:2015(E) standards. |
| Issue Date | Date from which settlement is allowed for the security. |
| Final Maturity or Expiry Date | Final maturity or expiry date of the security. |
| Settlement Type | Type of settlement foreseen for the security. The exhaustive list of possible values is as follows:   * Units * Nominal |
| Minimum Settlement Unit | Minimum unit of the security in accordance with the value specified in the Settlement Type attribute. |
| Settlement Unit Multiple | Settlement Unit multiple of the security in accordance with the value specified in the Settlement Type attribute. |

The entity Securities Name requires a data history. Each security has at least one corresponding item in the Securities Name entity. In addition, each security has one corresponding item in the Securities Code entity.

Each security inherits a country identifier for the country of issuance through the relationship with the Country entity and inherits a currency identifier for its currency of issuance [[1]](#footnote-1) from the Currency entity. Furthermore, it may also be linked to one or many Deviating Settlement Units (see below), Security Restrictions [[2]](#footnote-2) {T2S.16.510}, Market-Specific Attributes [[3]](#footnote-3) (see section [3.3.11.9 [} 180]](#_Ref79848F70C56823C1EE554EF12EF9EE58)) and to one or many combinations of Parties (CBs or payment banks) offering auto-collateralisation and Currencies eligible for auto-collateralisation {T2S.06.720} {T2S.16.899} {T2S.16.900}. Furthermore, each instance of Securities is linked to one or several issuers or technical issuer CSDs and can be linked to one or many investor CSDs by using Security CSD link [[4]](#footnote-4) {T2S.16.460} {T2S.16.470} {T2S.16.480} {T2S.16.710} {T2S.16.935} and one or many parties for the definition of the relevant Close Links [[5]](#footnote-5) {T2S.06.720} {T2S.16.690}. Finally, each security is linked to all its valuation data (see below).

**2 – Securities Code**

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.ENT.2.1*** |

This entity defines the identifiers of a security from a business perspective. The ISIN, based on ISO 6166 standard {T2S.16.420}, uniquely identifies a security. The ISIN does not change during the lifetime of a security, however the model supports the association of more than one occurrence to a single security, which allows managing exceptional cases of input errors.

The model allows only one ISIN to be valid at a given point in time.

| ATTRIBUTE | DESCRIPTION |
| --- | --- |
| Valid From | Starting date of validity for the securities code. |
| Code Type | Code type assigned to the security. Currently, only ISIN (as defined by ISO 6166 standard) is foreseen. |
| Securities Mnemonic | Actual value for the securities code, i.e. the ISIN for the security. |

The Securities Code entity inherits the Securities Identifier attribute from the Securities entity to link the security with its associated ISIN.

**3 – Securities Name**

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.ENT.3.1*** |

This entity specifies the Securities Long Name and Securities Short Name of a security on a time line basis {T2S.16.390} as a security name may change in time owing, for example, to the renaming of the issuer. Therefore, several names may exist for a security, although only one name can exist for a security at any given point in time, requiring T2S to store a Securities Name on a timeline basis {T2S.16.410}.

| ATTRIBUTE | DESCRIPTION |
| --- | --- |
| Valid From | Starting date of validity for the security name. |
| Securities Short Name | Short description of the security (FIDS) according to ISO 18774 standard {T2S.16.400}. |
| Securities Long Name | Long description of the security, according to ISO 18773 Part 1 and Part 2 {T2S.16.400}. |

The Securities Name entity inherits a securities identifier attribute from the Securities entity to link the security name to the underlying security.

**4 – Deviating Settlement Unit**

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.ENT.4.1*** |

This entity specifies the odd lots, in which securities are eligible for settlement. Every security has a multiple settlement unit. A multiple of that defines the standard lot sizes eligible for settlement on condition of being equal or greater than the minimum settlement unit. However, securities exist that have several odd lot sizes outside of the standard settlement multiple. Therefore, T2S stores such deviations from the standard multiple in this entity to perform validations on whether T2S can settle the quantity or nominal amount for the security specified in the Settlement Instruction {T2S.16.500}.

| ATTRIBUTE | DESCRIPTION |
| --- | --- |
| Deviating Settlement Unit | Deviating settlement unit for a security. |

The Deviating Settlement Unit entity inherits the Security Identifier attribute from the Securities entity to link a deviating settlement unit to the underlying security.

**5 – Securities Valuation**

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.ENT.5.1*** |

This entity stores dirty prices for securities, with the haircut already deducted, for the valuation of securities positions for auto-collateralisation in T2S. Both CBs and payment banks offering auto-collateralisation, provide prices for the securities each has identified as eligible for auto-collateralisation {T2S.16.520}.

| ATTRIBUTE | DESCRIPTION |
| --- | --- |
| Securities Valuation Date | Date for which the valuation applies. |
| Price | Price of the security. |

The Securities Valuation entity inherits a currency identifier attribute for the currency of the valuation price from the Currency entity. The entity also inherits a securities identifier attribute from the Securities entity to link a valuation price to the underlying security. Finally, each Securities Valuation is linked to all the parties applying the price, i.e. either a CB (euro or non-euro) or a payment bank.

**6 – Securities subject to cash penalties**

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.ENT.6.1*** |

This entity defines the scope of securities subject to cash penalties . {T2S.16.930}.

| ATTRIBUTE | DESCRIPTION |
| --- | --- |
| Liquidity | It specifies whether a security is liquid or not. The exhaustive list of possible values is as follows:   * Liquid * Illiquid |
| Valid From | Starting date of validity for the configuration of the penalties. |
| Valid To | Ending date of validity for the configuration of the penalties. |

The Securities subject to cash penalties entity inherits the Security Identifier attribute from the linked Securities entity.

**7 – SME Growth Market**

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.ENT.7.1*** |

This entity is a catalogue of Market Identifier Codes (MIC Identifiers) corresponding to trading venues (Place of Trade. {T2S.16.970}. The catalogue is maintained by the T2S Operator only. **{T2S.16.980}**

| ATTRIBUTE | DESCRIPTION |
| --- | --- |
| MIC Identifier | It specifies the Market Identifier Code according to ISO 10383. |

**8 – Security penalty rate**

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.ENT.8.1*** |

This entity stores the *Security penalty rates* according to the asset type. {T2S.16.990}. The maintenance of security penalty rates is performed by T2S Operator only. **{T2S.16.1000}**

| ATTRIBUTE | DESCRIPTION |
| --- | --- |
| Asset Type | It specifies the type of the asset related to the security penalty rate. |
| Daily flat penalty rate | It specifies the value for the security penalty rate. |
| Valid From | Starting date of validity for the security penalty rate. |

**9 – Cash discount penalty rate**

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.ENT.9.1*** |

This entity stores the *Cash Discount Penalty Rates* for each currency. {T2S.16.1010}. The maintenance of cash discount penalty rates related to Euro currency is performed by T2S Operator only based on values provided by the European Central Bank. **{T2S.16.1020}**

For other currencies, maintenance is performed by the Central Bank responsible for the currency.

| ATTRIBUTE | DESCRIPTION |
| --- | --- |
| Daily flat penalty rate | It specifies the value for the cash discount penalty rate (i.e. discount rate of the currency). |
| Valid From | Starting date of validity for the cash discount penalty rate. |

The *Cash Discount Penalty Rate* entity inherits a currency identifier attribute for the currency of the rate from the Currency entity.

**10 – Euro Foreign Exchange Rate**

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.ENT.10.1*** |

This entity stores the *Euro Foreign Exchange Rates* against the Eurofor each currency. {T2S.16.1030}. The maintenance of Euro foreign exchange rates is performed by the European Central Bank according to the official exchange rates. **{T2S.16.1040}**

| ATTRIBUTE | DESCRIPTION |
| --- | --- |
| Daily Exchange Rate | It specifies the exchange rate for the linked currency against the Euro. |
| Valid From | Starting date of validity for the Euro foreign exchange rate. |

The *Euro Foreign Exchange Rate* entity inherits a currency identifier attribute for the currency of the rate from the Currency entity.

**11 – Daily Price**

|  |  |
| --- | --- |
| Reference Id | ***SDMG.SEC.ENT.11.1*** |

This entity stores the *Daily Prices* to be used for the daily calculation of cash penalties.{T2S.16.1050}. The maintenance of daily prices is performed by the Securities Maintinag Entity responsible for the security the price applies to. **{T2S.16.1060}**

| ATTRIBUTE | DESCRIPTION |
| --- | --- |
| Price Date | Date for which the daily price applies. |
| Price | Price of the security. |

The *Daily Price* entity inherits a currency identifier attribute for the currency of the rate from the Currency entity. The entity also inherits a securities identifier attribute from the Securities entity to link a daily price to the underlying security.

…

## 3.4 Lifecycle Management and Matching

### 3.4.1 General Introduction

…

The services provided by this domain are available continuously during the whole T2S settlement day, with the exception of the maintenance window.

Additionally, the *Lifecycle Management and Matching* (LCMM) domain is responsible for:

The detection, calculation, recalculation, maintenance and preparation for reporting of *Penalties*.

The detection, calculation, recalculation and preparation for reporting of *Penalties* are triggered by business events (e.g. Penalties Eligibility Event, Penalties Calculation Event or Penalties Recalculation Event). The service for modifying *Penalties* is available continuously during the whole T2S settlement day, with the exception of the maintenance window and during the calculation, recalculation and preparation for reporting of *Penalties*.

The *LCMM* domain comprises ~~four~~ five modules:

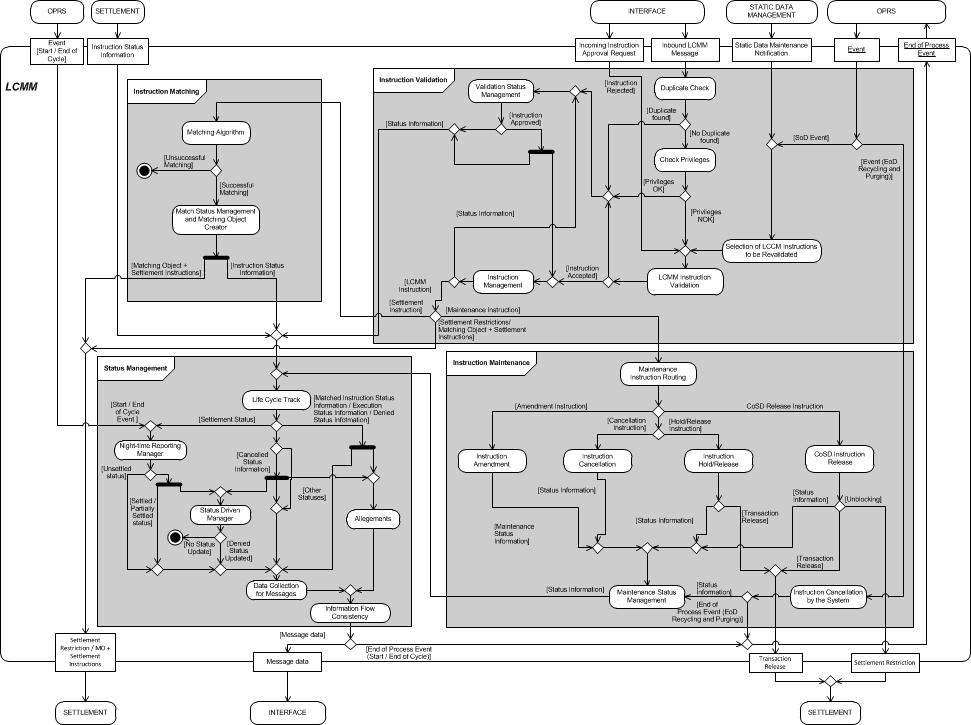
The Instruction Validation module (IVA);

* The Instruction Matching module (IMH);
* The Instruction Maintenance module (IMA);
* The Status Management module (STM).
* The *Penalty Mechanism* module(PEM).

The Instruction Validation module...

…needed to the relevant T2S Parties, via *Interface* domain.

The Penalty Mechanism module is responsible for the daily calculation and preparation for reporting of Penalties for failed settlement instructions as well as the complementary features, which are necessary for T2S Actors in this context. Every business day, the Penalty Mechanism module analyses all the failed *Settlement Instruction*s from previous business day in order to compute the necessary Penalties. This module also handles requests from CSDs to modify an existing *Penalty*, as well as the recalculation of *Penalties* that have been modified or impacted by a static data change. This module prepares the information, including the aggregation of amounts (daily or monthly), for the ad-hoc reports for *Penalties.*





### Dynamic data managed by the domain

#### List of dynamic data managed by the LCMM domain

The LCMM domain manages data related to:

* LCMM Instructions: Those whose purpose is instructing the receipt or delivery of securities with or without payment, the movement of securities within a Securities Account from one position to another, the movement of cash within a Dedicated Cash Account from one balance to another, as well as those whose purpose is the cancelation or modification of one of the former instructions.

The LCMM Instructions managed by the domain and described below in this chapter are the following:

* *Settlement Instructions;*
* *Settlement Restrictions;*
* *Cancellation Instructions;*
* *Amendment Instructions;*
* *Hold/Release Instructions.*
* *Penalties:* Dynamic data for the Penalty Mechanism needed for the detection, calculation, recalculation, management and reporting of Penalties.

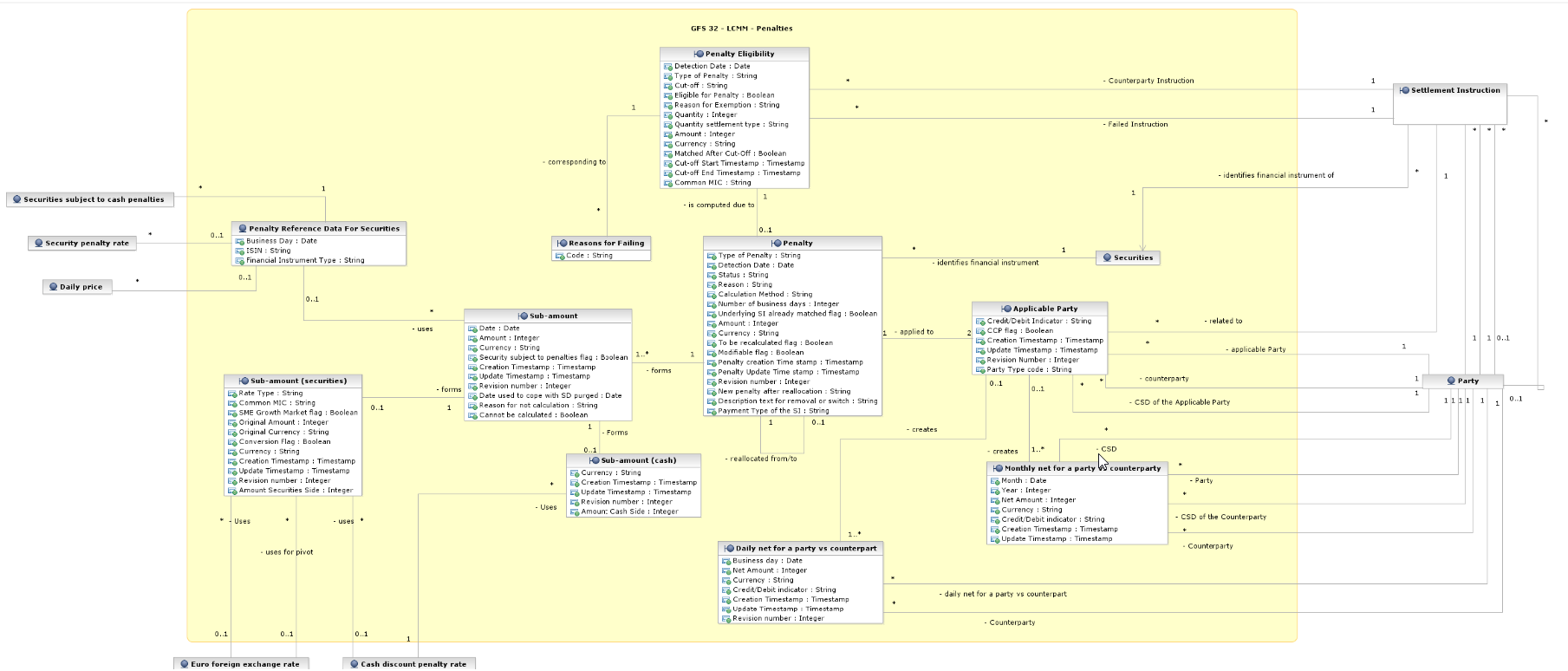
For the definitions of the attributes of the entities described in the following chapter, please refer to the HTML data model.

#### Description of the data related to ~~all~~ LCMM Instructions

…

#### Description of the data related to Penalties **(new section)**

The following diagram shows the conceptual data model for the Penalty detection and the Penalty preparation for calculation and re-calculation.



***Penalty Eligibility***

This entity includes the information needed to identify whether a failed *Settlement Instruction* is eligible for a penalty or not, depending on its ISO transaction code and, also: the reason(s) for settlement failure in case of Settlement Fail Penalties (SEFPs), or the Acceptance Timestamps in case of Late Matching Fail Penalties (LMFPs).

Each Penalty Eligibility instance is linked to the following entities:

* *Settlement Instruction* entity whose possible penalisation is analysed, for the following relationships: failed instruction and counterparty instruction;
* *Penalty,* when the value of the attributeEligible for a Penalty *is “True”;*
* *Reason(s) for Failing of the underlying Settlement Instruction,* in case of being eligible for aSettlement Fail Penalty (SEFP).

***Reasons for Failing***

This entity stores the reason for failing that a *Settlement Instruction* had at the processing of its relevant cut-off. It is linked to the *Penalty Eligibility* to which it corresponds.

***Penalty***

It stores the information about the current version of the *Penalty*. In case recalculations or modifications occur, previous values will be stored in the corresponding audit trail object for this entity.

There are two types of Penalties:

* “SEFP” - Settlement Fail Penalty: For settlement instructions that i) are matched before the relevant cut-off of a given business day and that ii) having reached their Intended Settlement Date (ISD), fail to settle on that business day.
* “LMFP” - Late Matching Fail Penalty: For settlement instructions that are matched in T2S after the relevant cut-off of their Intended Settlement Date (ISD).

Each Penalty is linked to the following entities:

* *Penalty Eligibility* that identifies the need for its calculation;
* The *Penalty* entity itself in case of reallocation, in order to identify, through the association “reallocated from/to”, the two *Penalties* involved when a penalty reallocation request is executed, i.e. the original *Penalty* and the new *Penalty* after reallocation.
* The *Securities* that identifies the financial instrument involved;
* The two *Applicable Parties* it applies to: one link for the Failing Party and a second one for the non-failing party (i.e. the two Counterparties of the *Penalty)*;
* Each and every *Sub-amount* calculatedthat forms the *Penalty*.

***Applicable Party***

This entity stores the specific data corresponding to either the DEBIT side (failing) or the CREDIT side (non-failing) of the *Penalty*. It stores the information about the current version of the *Applicable Party*, previous values will be stored in the corresponding audit trail object for this entity.

It is linked to the following entities:

* The *Penalty* to which the *Applicable Party* applies;
* The *Party* for the following relationships: applicable Party (i.e.: the Failing Party if “DEBIT”, the non-Failing Party if “CREDIT”), CSD of the applicable Party, Counterparty and CSD of the counterparty;
* The *Settlement Instruction* related to the *Applicable Party*, i.e.: the failing *Settlement Instruction* for the Failing Party, and the non-failing *Settlement Instruction* for the non-Failing Party.
* The *Daily Net for a party vs counterparty* for the Party and Counterparty in a given business day;
* The *Monthly Net for a party vs counterparty* created for the Party and Counterparty within a given month.

***Sub-amount***

This entity stores the specific data corresponding to the *Sub-amount(s)* that forms/form a given *Penalty*.

It stores the information about the current version of the *Sub-amount,* previous values will be stored in the corresponding audit trail object for this entity together with the Sub-amount (securities) and Sub-amount (cash) that form the *Sub-amount*.

There will be a *Sub-amount* per business day for which the *Penalty* applies. Therefore:

* In case of a “SEFP”, there will be a single *Sub-amount*; whereas
* In case of a “LMFP”, there will be as many *Sub-amounts* as business days when the *Penalty* applies.

The number of *Sub-amounts* will be indicated in the *Penalty* with the value of the attribute Number of business days.

Each Sub-amount is linked to the following entities:

* The unique *Penalty* the *Sub-amount* relates to;
* The applicable Penalty Reference Data for Securities used in the calculation of the Sub-amount;
* The Sub-amount *(securities)* that forms the *Sub-amount* in case the attribute Calculation Method of the related *Penalty* is “SECU”, “MIXE” or “BOTH”;
* The Sub-amount *(cash)* that forms the *Sub-amount* in case the Calculation Method of the related *Penalty* is “CASH” or “BOTH”.

***Sub-amount*** ***(securities)***

This entity stores specific data of the Sub-amount it forms, i.e. the part of the *Sub-amount* that is based on the quantity of securities failed to be delivered by the underlying instruction. It exists when the Calculation Method of the related *Penalty* is “SECU”, “MIXE” or “BOTH”.

Each Sub-amount (securities) is linked to the following entities:

* The *Sub-amount* it forms;
* The applicable Cash Discount Penalty Rate if used (i.e. only when the Calculation Method of the *Penalty* is “MIXE”);
* The applicable Euro Foreign Exchange Rate if used (i.e. only when value of the Original Currency differs from the value of the Currency, and hence the attribute Conversion Flag is “True”).

***Sub-amount (cash)***

This entity stores specific data of the Sub-amount it forms, i.e. the part of the *Sub-amount* that is based on the amount of cash failed to be delivered by the underlying instruction. It exists when the Calculation Method of the *Penalty* is “CASH” or “BOTH”.

Each Sub-amount (cash) is linked to the following entities:

* The *Sub-amount* it forms;
* The applicable Cash Discount Penalty Rate.

***Penalty Reference Data for Securities***

This entity stores specific information of the *Sub-amount,* i.e. the information from *Static Data* related to the *Security* of the underlying instruction (*Security Subject to penalties* and related *Security* penalty data) and for the corresponding business day.

It is linked to the following entities:

* The *Sub-amount* it applies to;
* The *Securities subject to Penalties* it refers to;
* The Security Penalty Rate used;
* The Daily Price used.

***Daily net for a party vs counterparty***

This entity stores the specific data corresponding to the net amount, due or entitled, from a Party to a given Counterparty for a given business day and currency.

It stores the information about the current version of the *Daily net for a party vs counterparty,* previous values will be stored in the corresponding audit trail object for this entity.

It represents the result of aggregating the Debit amounts due by the party to a counterparty, and the Credit amounts to be received by the party from the counterparty, on a given business day.

Each *Daily net for a party vs counterparty* is linked to:

* The *Applicable Party* that creates it;
* The *Party* entity for the following relationships: Party and Counterparty.

***Monthly net for a party vs counterparty***

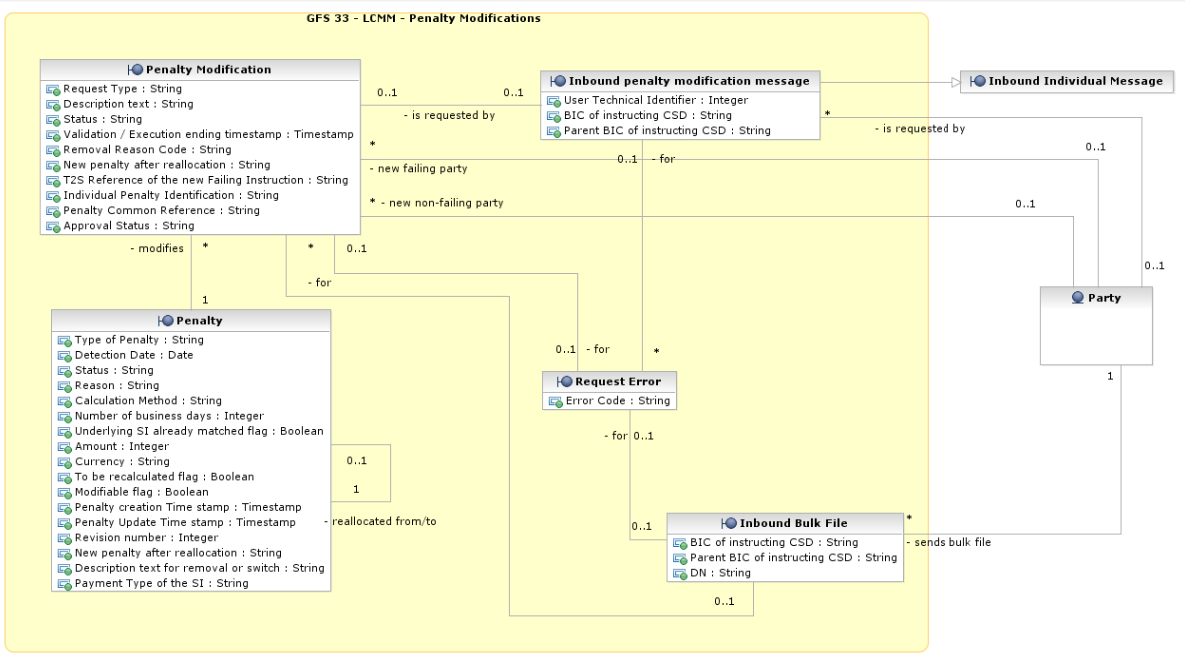
This entity stores the specific data corresponding to the monthly net amount, due or entitled, from a Party to a given Counterparty in a given currency.

It represents the result of aggregating the Debit amounts due by the party to a counterparty, and the Credit amounts to be received by the party from the counterparty, on a given month.

Each *Monthly net for a party vs counterparty* is linked to:

* The *Applicable Party* that creates it
* The *Party* entity for the following relationships: Party, CSD of the party, Counterparty and CSD of the counterparty.

The following diagram shows the conceptual data model for Penalty Modification Requests:



***Penalty Modification***

This entity stores information concerning *Penalty Modification* instructions. It is a specialisation of the LCMM Instruction entity, with an additional attribute for storing the current Status of the request (validated, executed or rejected).

Each *Penalty Modification* is linked to the following entities:

* The *Penalty* whose modification is requested;
* Either the *Inbound penalty modification message* by means of which a penalty modification can be requested via U2A; or the *Inbound Bulk File* by means of which one or several *Penalty Modification(s)* can be also requested via Middleware;
* The *Request error(s)* that can arise when a *Penalty Modification* cannot be executed.

***Inbound Penalty Modification Message***

This entity stores information concerning the *Penalty Modification* instructions received by the Inbound Processing module of the Interface domain **{T2S.16.167}**. It is a specialisation of the *Inbound Individual Message* entity (refer to the dynamic data description of the Interface domain for more information).

Each *Inbound Penalty Modification Message* is linked to:

* The *Penalty Modification* whose request it creates;
* The *Request Error* that can arise while validating the Message;
* Its requestor/sending Party

***Inbound Bulk File***

This entity stores the messages that enter into the Penalty Mechanism from the Middleware via bulk files.

Each *Inbound bulk file* is linked to:

* The *Penalty Modification(s)* whose request it creates: each modification requested in the Bulk File creates one instance in *Penalty Modification*;
* The *Request Error* that can arise while validating the Bulk File;
* Its sending Party.

***Request error***

This entity contains information about errors that may arise either while validating *Inbound Penalty Modification Messages*, *Inbound Bulk File*, or *Penalty Modifications*.

Each *Request error* is linked to the following entities:

* The *Inbound Penalty Modification Message* originating the error; or
* The *Inbound Bulk File* originating the error; or
* The *Penalty Modification* originating the error.

3.4.3 Instruction Validation

……

### 3.4.6. Status Management

…

#### 3.4.6.2 Description of the module

…

**Settlement Instruction status attributes**

…

* T2S will identify as “Failing” all incoming Settlement Instructions and Settlement Restrictions having their Intended Settlement Date before or equal to the Current Business Day.

Once a *Settlement Instruction* has been identified as “Failing” upon completion of its cut-off, the *Penalty Mechanism* Module will retrieve the *Settlement Instruction* status and reasons causing the settlement failure in order to identify if the instruction is eligible for a *Penalty*.

…

### Penalty Mechanism **(new section)**

#### Diagram of the module



#### 3.4.7.2 Description of the module

The *Penalty Mechanism* module manages the detection, calculation, recalculation, modification and preparation for reporting of *Penalties*.

The *Penalty Mechanism* module detects and calculates two types of *Penalties:*

* *Settlement Fail Penalties (SEFP):* For *Settlement Instructions* that i) are “*Matched*” before the end of processing of the relevant cut-off of a given business day and that ii) having reached their Intended Settlement Date (ISD), fail to settle on that business day. These *Penalties* are calculated for each business day the instruction fails to settle in T2S (i.e. for each and every business day that an instruction keeps failing in T2S, a new *Penalty* will be computed).
* *Late Matching Fail Penalties (LMFP):* For *Settlement Instructions* that are “*Matched*” in T2S after the relevant cut-off of their Intended Settlement Date (ISD). These *Penalties* are calculated only once per pair of matched instructions (i.e. on the business day when they are matched), but do consider all the previous business days when the instructions failed to settle due to the late matching i.e. due to their late sending.

The recalculation of *Penalties* is the process consisting on calculating again and updating existing *Penalties* (i.e. that have already been calculated on previous business days) after changes of values of the related T2S Static Data or after the execution of a modification requested by a T2S Actor.

The modification of *Penalties* consists on processing requests sent by T2S Actors to remove, re-include, re-allocate or switch existing *Penalties* that are allowed until they reach the end of their appeal period.

The preparation for reporting consists on calculating the aggregation of amounts (daily or monthly) and preparing the information to be included in the corresponding pre-defined reports.

The inputs for this module are:

* *Penalties Eligibility Event;*
* *Calculation Event;*
* *Recalculation Event;*
* *End of Appeal Period Event;*
* *Preparation for reporting Events;*
* *Inbound Penalty Modification Message;*
* *Inbound Penalty Modification via bulk file.*

#### 3.4.7.3 Description of the functions of the module

The *Penalty Mechanism* module is composed of six functions:

* Penalty Eligibility;
* Calculation;
* Selection for Recalculation;
* Penalty Modifications;
* End of Appeal Period;
* Preparation for Reporting*.*

##### **1- Penalty Eligibility**

*Introduction*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.ELI.1.1* |

This function is triggered every T2S settlement day **{T2S.03.365}** {T2S.03.370} {T2S.03.380} by the *Start Cash Penalty Eligibility Event (SCPE)*, received from the *Scheduling* *Module*. It analyses whether a *Settlement Instruction* is eligible either for a “SEFP” or for a “LMFP” *Penalty*, i.e.: if a *Penalty* has to be calculated either for a settlement fail or for a late matching on the previous business day.

The function is divided in the following sub-functions:

* *Eligibility Start*
* *SEFP Penalty Eligibility*
* *LMFP Penalty Eligibility*
* *Calculation Method Identification*
* *Create Penalty*
* *Get and Store Applicable Party Data*
* *Eligibility End*

*Eligibility Start*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.ELI.2.1* |

Upon reception of the *Start Cash Penalty Eligibility (SCPE)* *Event,* the Eligibility Start sub-function sets the Detection Date for all potential *Penalties* as the previous business date {T2S.03.380}. It also defines the Detection Date beginning and ending Timestamps as follows:

* The Timestamp associated with Change of Business Day two days before (start of d-1)
* The Timestamp associated with the previous Change of Business Day (end of d-1)

Then, the sub-function triggers *SEFP Penalty Eligibility* and *LMFP Penalty Eligibility* sub-functions.

*SEFP Penalty Eligibility*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.ELI.3.1* |

The SEFP Penalty Eligibility sub-function retrieves the *Settlement Instructions* which failed to settle during the Detection Date set in the *Eligibility Start*, i.e. those matched *Settlement Instructions* identified as failed by the *Standardisation and Preparation to Settlement* Module at the processing of their cut-off.

Internally generated realignment instructions are out of scope of the T2S Penalty Mechanism **{T2S.22.040}**, hence, *Settlement Instructions* with ISO Transaction Code “REAL” and with T2S Actor reference “empty” are not retrieved.

For each *Settlement Instruction* retrieved, a *Penalty Eligibility* instance is created.

Corporate actions on stock are not subject to Penalties, hence:

* If the *Settlement Instruction* has ISO Transaction Code “CORP”, its *Penalty Eligibility* instance is recorded with Eligible for a Penalty “False” and Reason for Exemption “ISOT” (i.e. ISO Transaction Code) **{T2S.22.050}**, and the process ends for this *Settlement Instruction*.
* Otherwise, the process continues.

To continue the processing of the *Penalty Eligibility* [[6]](#footnote-6)*,* the sub-function retrieves the Statuses and Reasons that caused the settlement failure of the *Settlement Instruction* **{T2S.22.110}** i.e. the “Unsettled” Reason Codes and Hold Statuses that the *Settlement Instruction* had at the end of processing of its cut-off.

Then, the sub-function checks the “Unsettled” Reason Codes and Hold Statuses versus the *Failing Reasons Dictionary* in order to identify whether the *Settlement Instruction* is eligible for a “SEFP” *Penalty* or not {T2S.22.080}, as described in the following – non-exhaustive – table[[7]](#footnote-7):

|  |  |
| --- | --- |
| SETTLEMENT INSTRUCTION’S  “UNSETTLED” REASON CODES and HOLD STATUSES | PENALTY  ELIGIBLE |
| Unsettled with LACK – Lack of Securities | “True” |
| Unsettled with *CLAC –* Counterparty Insufficient Securities | “False” |
| Unsettled with *MONY – Insufficient Money* | “True” |
| Unsettled with *CMON – Counterparty Insufficient Money* | “False” |
| Unsettled and on Party Hold | “True” |
| Unsettled and on CSD Hold | “True” |
| Unsettled and on COSD Hold | “True” |
| Unsettled and on CSD Validation Hold | “True” |
| Unsettled without Holds and with counterpart instruction on Hold | “False” |

Each “Unsettled” Reason Code and Hold Status is analysed one by one[[8]](#footnote-8):

As soon as one “Unsettled” Reason Code or Hold Status is defined as Penalty Eligible “True” in the *Failing Reasons Dictionary*, the sub-function stops checking and:

* Sets to “True” the attribute Eligible for a Penalty in the *Penalty Eligibility* entity,
* Retrieves the data of the Counterparty’s *Settlement Instruction*, which will be the non-failing leg of the *Penalty*,
* Determines the Common MIC (Place of Trade informed equally in both *Settlement Instructions*);
* Calculates and stores the remaining Quantity and/or Amount to be settled at the Cut-off timestamp of the Detection date, by subtracting to the original quantity of securities or to the original amount of the *Settlement Instruction*, the quantity of securities or the amount of cash settled at the Cut-off; and
* Sends the *Penalty Eligibility* to the *Calculation method identification* sub-function to continue with the process.

In case that after analysing all the “Unsettled” Reason Codes and all Hold Statuses of the *Settlement Instruction,* none of them was defined as Penalty Eligible “True” in the *Failing Reasons Dictionary,* then the sub-function:

* Sets to “False” the attribute Eligible for a Penalty in the *Penalty Eligibility* entity,
* Sets the attribute Reason for Exemption in the *Penalty Eligibility* entity to:
* Either “RECO” if all “Unsettled” Reason Codes were defined in the *Failing Reasons Dictionary*,
* Or “REPR” if at least one of the “Unsettled” Reason Codes was missing in the *Failing Reasons Dictionary*.
* Ends the process for this *Settlement Instruction*.

*LMFP Penalty Eligibility*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.ELI.4.1* |

The LMFP Penalty Eligibility sub-function retrieves the *Settlement Instructions* that i) have an Intended Settlement Date equal to or before the Detection Date set in the *Eligibility Start* and ii) were matched on the previous business day (i.e. those with Matching Timestamp between the Detection Date beginning and ending Timestamps).

Internally generated realignment instructions are out of scope of the *T2S Penalty Mechanism* **{T2S.22.040}**, hence *Settlement Instructions* with ISO transaction code “REAL” and with T2S Actor reference “empty” are not retrieved.

For each *Settlement Instruction* retrieved, in order to determine if it is eligible for a “LMFP” *Penalty*, the sub-function checks:

1. If it was matched at a point in time when it was no longer possible to settle it on that business day (i.e. on the Detection Date). For this purpose, the sub-function checks if the *Settlement Instruction* fulfils any of the following:

* Either the *Settlement Instruction*’s *Reason History* has a “LATE” Reason Code; or
* The Matching Timestamp is after the start of the last cut-off of that business day (i.e. after the “EFOP” Event Start Time timestamp of the Detection Date);

In case the *Settlement Instruction* fulfils any of the two conditions, a *Penalty Eligibility* instance is created with Matched After Cut-Off flag “True”

1. Provided the previous check is not met (the instruction was matched at a point in time when it was still possible to settle it on the Detection Date), the sub-function checks:

* If the Intended Settlement Date of the *Settlement Instruction* is before the Detection Date.

In case the *Settlement Instruction* fulfils this condition, a *Penalty Eligibility* instance is created with Matched After Cut-Off flag “False”;

In case none of the checks is met, no instance is created in *Penalty Eligibility* for the *Settlement Instruction* (i.e. the instruction did not match late at all) **{T2S.22.160}**,**{ T2S.22.170}**.

For each *Penalty Eligibility* instance created, the sub-function determines its Eligible for Penalty attribute as follows:

* If the underlying *Settlement Instruction* has ISO transaction code “CORP”, the Eligible for Penalty is set to “False” and the Reason for Exemption to “ISOT” (i.e. ISO Transaction Code) **{T2S.22.050}.**
* Otherwise, the counterparty’s *Settlement Instruction* is retrieved through the *Matching Object* and:
* If the underlying *Settlement Instruction* was matched in T2S, the Validation Ending Timestamp of both *Settlement Instructions* is compared:
  + In case the underlying *Settlement Instruction* has a later Validation Ending Timestamp than the one of its counterparty, then the Eligible for Penalty is set to “True”;
  + Otherwise (i.e. the underlying *Settlement Instruction* has an earlier Validation Ending Timestamp than the one of its counterparty), then the Eligible for Penalty is set to “False” and the Reason for Exemption to “RECO” (reason for failing is associated to the counterparty instruction because it entered later) **{T2S.22.160}**.
* If the underlying *Settlement Instruction* was sent as already matched, it has the same Validation Ending Timestamp as its counterpart´s, therefore, the sub-function analyses the Securities Movement Type:
  + In case the underlying *Settlement Instruction* is “DELI”, then the Eligible for Penalty is set to “True”;
  + Otherwise (i.e. the underlying *Settlement Instruction* is “RECE”), then the Eligible for Penalty is set to “False” and the Reason for Exemption to “RECO” (reason for failing is associated to the counterparty instruction because it entered later) **{T2S.22.170}**.

For each *Penalty Eligibility* instance with Eligible for a Penalty “True”, the sub-function

* Retrieves and stores the Common MIC (Place of Trade informed equally in the *Settlement Instruction* and in its counterpart´s), the original Settlement Quantity and the Matched Settlement Amount i.e. at the moment of the matching; and
* Triggers the *Calculation method identification* sub-function to process them.

*Calculation method identification*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.ELI.5.1* |

The Calculation method identification sub-function determines the Calculation Method that has to be applied to calculate the *Penalty*.

The possible values for the Calculation Method are the following:

* *“SECU”:* The *Penalty* is based on the quantity of securities failed to be delivered, the price and the penalty rate of the relevant Asset Type;
* *“MIXE”:* The *Penalty* is based on the quantity of securities failed to be delivered, the price of the security and the discount rate of the relevant Currency;
* *“CASH”: Penalty* is based on the amount of cash failed to be delivered and the discount rate of the relevant Currency;
* *“BOTH”: Penalty* is the sum of “*SECU”* and “*CASH”*. I.e. the sum of:
* The *Penalty* based on the quantity of securities failed to be delivered, the price and the penalty rate of the relevant Asset Type (“*SECU”*); and
* The *Penalty* based on the amount of cash failed to be delivered and the discount rate of the relevant currency (“*CASH”*).

The Calculation Method is determined based on the Instruction Type of the underlying *Settlement Instruction* that generated the *Penalty Eligibility* instance, according to the following logic **{T2S.22.120} {T2S.22.121} {T2S.22.130} {T2S.22.140} {T2S.22.210} {T2S.22.220} {T2S.22.230} {T2S.22.231}**:

|  |  |
| --- | --- |
| INSTRUCTION TYPE | CALCULATION METHOD |
| **PFOD** (“DPFOD” or “CPFOD”)   * Quantity = 0 | “CASH” |
| **FOP** (“DFOP” or “RFOP”)   * Payment Type = “FREE” and Quantity > 0 | “SECU” |
| **DVP**   * Securities Movement Type = “DELI”, Quantity > 0 ,   Payment Type = “APMT” and Credit/Debit Code = “CRDT” | “SECU” |
| **DWP or RWP**   * Either Securities Movement Type = “DELI”, Quantity > 0 ,   Payment Type = “APMT” and Credit/Debit Code = “DBIT”;   * Or Securities Movement Type = “RECE”, Quantity > 0 ,   Payment Type = “APMT” and Credit/Debit Code = “CRDT” | “BOTH” |
| **RVP\***   * Securities Movement Type = “RECE” , Quantity > 0 ,   Payment Type = “APMT” and Credit/Debit Code = “DBIT” | ”MIXE” |

\* According to **{T2S.22.232},** in the case of a RVP received as an already matched instruction, the *Penalty* Calculation Method applied should be “SECU”. However, in case of already matched instructions, the *LMFP Penalty Eligibility* sub-function analyses the Securities Movement Type of the *Settlement Instruction*, and in case it is “DELI”, the *Settlement Instruction* is deemed eligible for a Penalty; otherwise, i.e. it is “RECE”, the *Settlement Instruction* is deemed as not eligible for a Penalty. Hence, the *Calculation Method identification* sub-function won’t have to determine the Calculation Method for a *Penalty* whose underlying *Settlement Instruction* is RVP and sent as already matched.

Once the Calculation Method is determined, the sub-function triggers the *Get Penalty Data* sub-function.

*Create Penalty*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.ELI.6.1* |

The sub-function creates the corresponding *Penalty* with Status *“PEND”* associated to the *Penalty Eligibility,* and assigns it a Common Penalty ID (composed of 15 characters).

Additionally, the following information already retrieved by the previous sub-functions is stored in the *Penalty:*

* Type of Penalty (LMFP or SEFP)
* Detection Date
* Calculation Method (SECU, CASH, MIXE or BOTH)
* Security
* Underlying SI Already Matched flag (True or False)

Then, the sub-function determines the attributes Payment Type of the SI and, when possible, the Currency of the *Penalty* as follows **{T2S.22.060}:**

* In case the underlying *Settlement Instruction* Payment Type Code is “APMT” (i.e. against payment), then the Payment Type of the SI is recorded as “APMT” and the Currency is set with the same value as the Currency of the Amount of the *Settlement Instruction;*
* In case the underlying *Settlement Instruction* Payment Type Code is “FREE” (i.e. free of payment), then the Payment Type of the SI is recorded as “FREE” and the Settlement Type of the Security is checked to determine the Currency:
* If the Settlement Type is “FAMT” (i.e. Face amount), the Currency of the *Security* is retrieved from Static Data and: if it is a T2S Settlement Currency (i.e. “EUR” or “DKK”), the Currency of the *Penalty* is set with the same value as the Currency of the *Security;* otherwise, the Currency of the *Penalty* is set to “EUR”;
* If the Quantity Settlement Type is “UNIT” (i.e. Units), the Currency of the *Penalty* is not determined and left empty until it is determined by the Calculation function.

Once all activities of this sub-function are finished, the *Get and Store Applicable Party Data* sub-function is triggered.

*Get and Store Applicable Party Data*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.ELI.7.1* |

The sub-function determines the Failing Party and the Non-Failing Party of the *Penalty*, i.e.: the *Party* debited with the *Penalty*, and the *Party* credited with the *Penalty*, respectively.

* In case of a “LMFP” with Underlying SI Already Matched flag “True”, the *Instructing Party* of the underlying *Settlement Instruction* will be both the Failing and the Non-Failing Party **{T2S.22.180}.**
* Otherwise, the sub-function checks the Depositories of the underlying *Settlement Instruction* for which the *Penalty* is computed*:*
* In case the Depositories involved in the *Settlement Instruction* are both CSDs in T2S, the Failing Party will be the account owner of the *Securities Account* of the instruction, whereas the Non-Failing Party will be the account owner of the *Securities Account* of the counterparty {T2S.22.090} **{T2S.22.180}.**
* In case of External CSD settlement scenarios (either Delivering Depository is an External CSD or Receiving Depository is an External CSD), T2S will consider the “External CSD” as the relevant securities account owner and, consequently, as the Failing Party or the Non-Failing Party **{T2S.22.100} {T2S.22.190}**.

With the information above, the sub-function generates two *Applicable Party* instances (i.e.: one for the Failing Party and another one for the Non-failing Party) and, for each one of them, retrieves and stores its necessary attributes, i.e.:

* Individual Penalty ID: Composed of 16 characters (i.e.: F+Common Penalty ID for the Failing Party; N+Common Penalty ID for the Non-Failing Party).
* The CSD of the Applicable Party (of the Failing Party or of the Non-Failing Party[[9]](#footnote-9)).
* Credit / Debit indicator: “Debit” for the Failing Party or “Credit” for the Non-Failing Party.
* CCP flag: It indicates whether the *Party* is a CCP or not, as defined in the relevant *Attribute Domain* **{T2S.13.490}**.
* T2S Reference of the related *Settlement Instruction* (of the Failing Party or of the Non-Failing Party). In case the *Penalty* is a “LMFP” with Underlying SI Already Matched flag “True”, and the *Instructing Party* is both the Failing and the Non-failing party, T2S assumes that the Delivering leg is the *Settlement Instruction* of the Failing Party and the Receiving leg is the *Settlement Instruction* of the Non-Failing Party **{T2S.22.180}**.

*Eligibility End*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.ELI.8.1* |

Once the *SEFP Penalty Eligibility* and *LMFP Penalty Eligibility* sub-functions have processed all the corresponding *Settlement Instructions* and all *Penalties* and their *Applicable Parties* have been created, this sub-function checks if there are instances in Penalty Eligibility from previous business days (i.e. with Detection Date before the previous business day) with Reason for Exemption “REPR”. In case there is any, it is sent to *SEFP Penalty Eligibility* sub-function to reprocess it*.*

Once all Penalty Eligibility instances with Reason for Exemption “REPR” have been reprocessed, the sub-function sends the End of ProcessEvent (*Start Cash Penalty Eligibility*) to the *Scheduling* *Module.*

##### **2- Penalty Calculation**

*Introduction*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.CAL.1.1.* |

This function is triggered every T2S settlement day **{T2S.03.370}** {T2S.03.380} by the *Calculation of Penalties (CALC)* Event, received from the *Scheduling* *Module*. Additionally, this function is also triggered by the function *Selection for Recalculation* in order to calculate againexisting *Penalties.*

The Penalty Calculation function calculates the Amount of each *Penalty*.

The Amount of a *Penalty* can be formed by one or several *Sub-amount*s, depending on the Number of business days when the *Penalty* applies:

* In case of “SEFP” the Number of business days is always one (and therefore there is one *Sub-amount*) **{T2S.22.115}**;
* In case of “LMFP” the Number of business days (and therefore *Sub-amount*s) can be one or several ones **{T2S.22.200}**.

Additionally, depending on the applicable Calculation Method (previously identified by the *Penalty Eligibility* function) the Amount of each *Sub-amount* will be formed by:

* Either an Amount Cash Side, based on the amount failed to be delivered, if the Calculation Method of the *Penalty* is “CASH”; or
* An Amount Securities Side, based on the quantity of securities failed to be delivered, if the Calculation Method of the *Penalty* is “SECU” or “MIXE”; or
* Both, an Amount Cash Side based on the amount failed to be delivered and an Amount Securities Side based on the quantity of securities failed to be delivered, if the Calculation Method of the *Penalty* is “BOTH”.

The function is divided in the following sub-functions:

* *Fetch New Penalties to be calculated*
* *Penalty Preparation*
* *Sub-amount Preparation*
* *Calculate Sub-amount*

*Fetch New Penalties to be calculated:*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.CAL.2.1* |

Upon reception of the *Calculation of Penalties (CALC) Event*, this sub-function retrieves all the new *Penalties* (with Detection Date the previous business day and Status “PEND”), which have to be calculated in the current business day. Then it triggers the *Penalty Preparation* sub-function.

*Penalty Preparation:*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.CAL.3.1* |

The sub-function is triggered either by the *Fetch New Penalties to be calculated* sub-function for new *Penalties* to be calculated, or by the function *Selection for Recalculation* for recalculation of existing *Penalties*.

For each *Penalty* to be (re)calculated, this sub-function identifies the Number of business days, and generates the corresponding Sub-amounts:

* For “SEFP” *Penalties*, the Number of business days is always one, i.e. the applicable business day is the Detection date of the *Penalty* **{T2S.22.115}**.
* For “LMFP” *Penalties*, the sub-function counts the applicable business days from the Intended Settlement Date of the underlying *Settlement Instruction*, until the following date:
* The business day before the Detection date of the *Penalty*, if the *Settlement Instruction* was matched before its cut-off (i.e. if the corresponding *Penalty Eligibility* has Matched After Cut-Off flag “False”);
* The Detection date of the *Penalty*, if the *Settlement Instruction* was matched after its cut-off (i.e. if its corresponding *Penalty Eligibility* has Matched After Cut-Off flag “True”) **{T2S.22.200}**.

In order to count the applicable business days, the sub-function considers T2S Settlement Days, i.e. days for which the instruction can settle in T2S. Therefore:

* In case the Payment Type of the SI is “APMT” (against payment instructions), T2S Settlement Days are business days that are not closing days for the relevant Currency of the cash amount of the *Settlement Instruction*;
* In case the Payment Type of the SI is “FREE” (free of Payment instructions), T2S Settlement Days are business days that are not T2S common closing days for all currencies.

Then, the sub-function generates one *Sub-amount* for each of the identified applicable business day(s) of the *Penalty* (i.e. creates one *Sub-amount* withDate equal to the applicable business day, for each of the applicable business days). Depending on the Calculation Method, each *Sub-amount* can be formed by:

* Either one *Sub-amount (securities)* if the Calculation Method of the *Penalty* is “SECU” or “MIXE”; or
* One *Sub-amount (cash)* if the Calculation Method of the *Penalty* is “CASH”;
* One *Sub-amount (securities)* and one *Sub-amount (cash)* if the Calculation Method of the *Penalty* is “BOTH”.

The *Sub-Amounts* of the *Penalty* are sent to the *Sub-amount Preparation* sub-function. Once the *Sub-amount Preparation* and *Sub-amount Calculation* sub-functionshave processed all the *Sub-Amounts* of the *Penalty*:

* If the *Security* is not Subject to Penalties **{T2S.22.010}** in any of the applicable business days (i.e. all the *Sub-amounts* of the *Penalty* have theSecurity subject to penalties flag equal to “False”), the Status of the *Penalty* is updated to “Not computed” (“NCOM”) and, therefore, the *Penalty* will not be reported **{T2S.22.200}** [See Penalties Preparation for Reporting function].
* The sub-function informs the Selection for Recalculation function that the *Penalty* has been recalculated.

Finally, once all the new Penalties have been calculated, the sub-function sends the End of Process Event (*Calculation of Penalties*) to the *Scheduling* *Module*.

*Sub-amount Preparation:*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.CAL.4.1* |

Upon reception of the *Sub-amounts* of a *Penalty*, either from the Penalties Preparation sub-function or from the *Selection for Recalculation* function, this sub-function retrieves all the necessary information for each of them.

First, the sub-function checks if the *Security* is Subject to Penalties for each *Sub-amount* (i.e. for the Date of each *Sub-amount*) and stores the result in the Security subject to penalties flag of the relevant *Sub-amount*.

* For *Sub-amounts* with Security subject to penalties flag is “False” the processing ends*.*
* For each *Sub-amount* with Security subject to penalties flag “True”, the sub-function retrieves and stores the necessary *Penalty Reference Data for Securities* attributes and Static data information to calculate the *Penalty*. Note that in case of LMFPs, if any of the *Sub-amounts* has aDate before the Date used to cope with Static Data purged or Retention Date (e.g.: *Sub-amount* Date is “d-98”, being “d” the current business day), the *Penalty Reference Data* retrieved and stored for calculation is the first available one **{T2S.22.240}** i.e.: data from “d-90”.

The sub-function performs two steps:

* Retrieve Price, Currency and Euro FX
* Determine rate

*Retrieve Price, Currency and Euro FX:*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.CAL.4.2* |

For *Sub-amounts* whose *Penalty* has Calculation Method “SECU”, “MIXE” or “BOTH”, this step performs the following actions:

The step first retrieves the Daily Price of the *Security* for the relevant Date of the *Sub-amount* **{T2S.22.150} {T2S.22.240}**.Then:

In case the Daily Price is not available in Static Data, the *Sub-amount* is flagged as Cannot be calculated “True”.

Otherwise, the step performs different checks depending on the Settlement Type of the *Security*.

* In case it is “UNIT”:
* The step verifies that the Settlement Type of the quantity stored in *Penalty Eligibility* from the underlying *Settlement Instruction* is also “UNIT”. In case the condition is not fulfilled, the relevant *Sub-amount* is flagged as Cannot be calculated “True”, because the Price would be in “UNIT” (Price Amount) whereas the failed quantity would be in “FAMT” (i.e. the Settlement Type of the Security has been updated and the *Penalty* cannot be calculated for this business day).
* The step checks whether the Payment Type of the SI of the *Penalty* is “FREE”, and if so, the Currency of the Penalty is determined as follows: If Currency of the *Daily Price* of the *Security* is a T2S Settlement Currency (i.e. “EUR” or “DKK”), the Currency of the *Penalty* is set with the same value as the Currency of the *Daily Price;* otherwise, the Currency of the *Penalty* is set to “EUR”.
* Finally, in case the *Sub-amount* can be calculated, the step checks if an FX conversion is needed. For this purpose, the process verifies whether the Currency of the *Penalty* is different from the Currency of the *Daily Price.* If they are equal, the Conversion flag of the *Sub-amount (securities) is set to* “False” and the step *Determine rate* is triggered; if they are different, the Conversion flag of the *Sub-amount (securities) is set to “*True” and:
* If the Currency of the *Penalty* is “EUR”, only one *Euro Foreign Exchange Rate* is retrieved and stored for the relevant Currency of the *Daily Price* and Date.
  + In case the *Euro Foreign Exchange Rate* is not available in Static Data, the *Sub-amount* is flagged as Cannot be calculated “True” and the processing ends.
  + Otherwise, the step *Determine rate* is triggered.
* If the Currency of the *Penalty* is not “EUR” (i.e. it is “DKK”), an additional *Euro Foreign Exchange Rate* is necessary to pivot from “EUR” to “DKK”, thus retrieved and stored.
  + In case any of the two *Euro Foreign Exchange Rates* is not available in Static Data, the *Sub-amount* is flagged as Cannot be calculated “True” and the processing ends.
  + Otherwise, the step *Determine rate* is triggered.
* In case it is “FAMT”:
* The step verifies that the Settlement Type of the quantity stored in *Penalty Eligibility* from the underlying *Settlement Instruction* is also “FAMT”.
  + In case the condition is not fulfilled, the relevant *Sub-amount* is flagged as Cannot be calculated “True” because the Price would be in “FAMT” (Coefficient) whereas the failed quantity would be in “UNIT” (i.e. the Settlement Type of the Security has been updated and the *Penalty* cannot be calculated for this business day) and the processing ends.
* In case the above condition is fulfilled and the Sub-amount can be calculated, the step verifies that the Currency of the *Security* is the same as the Currency of the quantity stored in *Penalty Eligibility* from the underlying *Settlement Instruction.*
  + In case the condition is not fulfilled, the relevant *Sub-amount* is flagged as Cannot be calculated “True” (i.e. the Currency of the Security has been updated and the *Penalty* cannot be calculated for this business day) and the processing ends.
* Finally, in case the *Sub-amount* can be calculated, the step checks if a FX conversion is needed. For this purpose, the process verifies whether the Currency of the *Penalty* is different from the Currency of the *Security.* If they are equal, the Conversion flag of the *Sub-amount (securities) is set to* “False” and the step *Determine rate* is triggered; if they are different, the Conversion flag of the *Sub-amount (securities) is set to “*True” and:
* If the Currency of the *Penalty* is “EUR”, only one *Euro Foreign Exchange Rate* is retrieved and stored for the relevant Currency of the *Security* and Date.
  + In case the *Euro Foreign Exchange Rate* is not available in Static Data, the *Sub-amount* is flagged as Cannot be calculated “True” and the processing ends.
  + Otherwise, the step *Determine rate* is triggered.
* If the Currency of the *Penalty* is not “EUR” (i.e. it is “DKK”), an additional *Euro Foreign Exchange Rate* is necessary to pivot from “EUR” to “DKK”, thus retrieved and stored.
  + In case any of the *Euro Foreign Exchange* *Rates* is not available in Static Data, the Sub-amount is flagged as Cannot be calculated “True” and the processing ends.
  + Otherwise, the step *Determine rate* is triggered.

*Determine rate:*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.CAL.4.3* |

Depending on the Calculation Method of the *Penalty* the step performs the following actions:

* For “CASH” and “MIXE” Calculation Methods, the step retrieves the *Cash Discount Penalty Rate* (expressed as a daily flat rate) of the corresponding Currency of the *Penalty* and Date of the *Sub-amount*. Then:
* In case the *Cash Discount Penalty Rate* is not available in Static Data, the *Sub-amount* is flagged as Cannot be calculated “True” and processing ends.
* Otherwise, the *Sub-amount* is sent to the *Calculate Sub-amount sub-function.*
* For “SECU” Calculation Method, the step determines whether the Common MIC (Market Identifier Code) is an SME Growth Market or not. Then, for the relevant Date and Security the sub-function requests Static Data the Financial Instrument Type, Liquidity (if the Financial Instrument Type is equal to “SHRS”) and the *Security Penalty Rate* (for SME or for non-SMEdepending on the SME Growth Market flag) **{T2S.22.145}**:
* In case either the Financial Instrument Type, Liquidity or *Security Penalty Rate* is not available in Static Data, the *Sub-amount* is flagged as Cannot be calculated “True” and the processing ends.
* Otherwise, the *Sub-amount* is sent to the *Calculate Sub-amount sub-function.*
* For “BOTH” Calculation Method, the above described actions for “CASH” and “SECU” are performed.

Once the rate has been determined, the sub-function sends the *Penalty* and all *Sub-amount*s flagged as Cannot be calculated “False” to the *Calculate Sub-amount* sub-function.

*Calculate Sub-amount:*

|  |  |
| --- | --- |
| *Reference Id* | *LCMM.PEM.CAL.5.1.* |

For each *Sub-amount* received from the *Sub-amount Preparation* sub-function, the sub-function calculates the Amount of the *Sub-amount*, applying the steps required for each Calculation Method:

* For “CASH” Calculation Method, the Amount of the *Sub-amount* is the same asAmount Cash side, which is calculated as **{T2S.22.130} {T2S.22.220}**:

Amount(failed to be delivered) \* *Cash Discount Penalty Rate*

* For “SECU” Calculation Method, the Amount of the *Sub-amount* is the same asAmount Securities side, which is calculated in two steps, first the sub-function calculates the Original Amount of *Sub-amount (securities)* and then, if necessary it is converted applying the previously retrieved *Euro foreign exchange rates*. More in detail:
* The Original Amount of *Sub-amount (securities)*is calculated as **{T2S.22.120} {T2S.22.210} {T2S.22.232}**:

*Securities Penalty Rate* \* *Daily* *Price* of the *Security* \* Quantity(failed to be delivered)

* The Amount Securities side is calculated as:
* If the Conversion flag of the *Sub-amount (securities) is* “False”, the Amount Securities side is equal to the Original Amount;
* If the Conversion flag of the *Sub-amount (securities) is* “True”, depending on the number of *Euro foreign exchange rate* stored, the Amount Securities side is:

Original Amount \* First *Euro foreign exchange rate*, if *Sub-amount* Currency is “EUR”; or

Original Amount \* First *Euro foreign exchange rate* \* Second *Euro foreign exchange rate*.

* For “MIXE” Calculation Method, the Amount of the *Sub-amount* is the same asAmount Securities side, which is calculated in two steps, first the sub-function calculates the Original Amount of *Sub-amount (securities)* and then, if necessary it is converted applying the previously retrieved *Euro foreign exchange rates*. More in detail:
* The Original Amount of *Sub-amount (securities)*is calculated as **{T2S.22.121} {T2S.22.231}**:

*Cash Discount Penalty Rate* \* *Daily* *Price* of the *Security* \* Quantity(failed to be delivered)

* The Amount Securities side is calculated as:
* If the Conversion flag of the *Sub-amount (securities) is* “False”, the Amount Securities side is equal to the Original Amount
* If the Conversion flag of the *Sub-amount (securities) is* “True”, depending on the number of *Euro foreign exchange rate* stored, the Amount Securities side is:

Original Amount \* First *Euro foreign exchange rate*, if *Sub-amount* Currency is “EUR”; or

Original Amount \* First *Euro foreign exchange rate* \* Second *Euro foreign exchange rate*.

* For “BOTH” Calculation Method, the Amount of the *Sub-amount* will be the sum of the Amount Securities side and the Amount Cash side, which are calculated as described above for “SECU” and “CASH” Calculation Method respectively **{T2S.22.140}** **{T2S.22.230}**.

Once the Amount of the *Sub-amount* has been calculated, the Amount of the *Penalty* is updated, summing the Amounts of all of its *Sub-amounts*, and the sub-function informs either the *Penalty Preparation* sub-function or the Selection for Recalculation function that the *Sub-amount* and the *Penalty* have been processed.

##### **3- Penalty Modification**

*Introduction*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.MOD.1.1* |

The Penalty Modification function is triggered either by the *Inbound Penalty Modification Messages* or by the *Inbound Bulk Files* sent to LCMM, either from the Interface when sent U2A by the T2S Actor or from the Middleware when sent A2A by the T2S Actor, respectively **{T2S.22.450}.**

This function analyses whether an existing *Penalty* can be modified ex-post (i.e. “removed”, “re-included”, “re-allocated” or “switched”) by the CSD when either its corresponding participant or the CSD itself have been imposed with it, provided the end of the Appeal period of the *Penalty* has not been reached yet **{T2S.22.440}**.

A *Penalty Modification* can have the following Request Type:

* *“Penalty Removal” or “REMO”* **{T2S.22.300}***;*
* *“Penalty Re-inclusion” or “REIN”* **{T2S.22.340}***;*
* *“Penalty Switch” or “SWIC”* **{T2S.22.400}***;*
* *“Penalty Re-allocation” or “RALO”* **{T2S.22.370}***.*

In case the required conditions are fulfilled, the *Penalty* *Modification* is executed and the *Penalty* is updated accordingly; otherwise, the *Inbound Penalty Modification Message* or the *Penalty* *Modification* included in the *Inbound Bulk File* is rejected.

This function is divided in the following sub-functions:

* *Validation of Penalty Modifications sent via Bulk File*
* *Validation of Inbound Penalty Modification Messages and Incoming Instruction Approval Requests*
* *Perform Removal*
* *Perform Re-inclusion*
* *Perform Switch*
* *Perform Re-allocation*.

*Validation of the Penalty Modifications sent via Bulk File*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.MOD.2.1* |

Upon reception of the *Inbound Bulk File,* which maycontain one or several *Penalty* *Modifications*, the sub-function checks if the sender is authorised to perform the action of sending the file on behalf of the instructing CSD. For this purpose the function checks:

* If the BIC and Parent BIC of the Instructing Party*[[10]](#footnote-10)* of the *Inbound Bulk File* correspond to an existing Party defined as “CSD” in T2S Static Data, and if so
* If the Distinguished Name (DN) of the sender, the BIC and Parent BIC of the CSD are defined in the “white list” configuration table stored in the Attribute Domain in T2S Static Data.

In case either the BICs of the Instructing Party do not correspond to a Party defined as “CSD”, or the combination of Distinguished Name (DN), Parent BIC and BIC are not defined in the“white list”, the relevant *Request Error* is stored and an XML response is sent back to the Sender informing about the rejection of the whole Bulk File.

Otherwise, the sub-function checks whether each and every *Penalty Modification* of the *Inbound Bulk File* complies with the business requirements **{T2S.22.310} {T2S.22.320}** **{T2S.22.350}** **{T2S.22.380}** **{T2S.22.410}** **{T2S.22.420}**.

This sub-function rejects the execution of a *Penalty Modification* when one of the conditions in the table below is fulfilled:

|  |
| --- |
| CONDITIONS FOR REJECTION |
| The CSD requesting the *Penalty Modification* with Request Type “REMO”, “REIN” or “SWIC” is not the failing CSD of the *Penalty* |
| The CSD requesting the *Penalty Modification* with Request Type “RALO” is not the same as the CSD of the LMFP (i.e. it is not the CSD of the Instructing Party of the underlying *Settlement Instruction*). |
| The *Penalty Modification* with Request Type “REMO”, “REIN” or “SWIC” does not refer to the Individual Penalty ID of the Failing Party. |
| A Common Penalty ID is provided in the *Penalty Modification* with Request Type “REMO”, “REIN” or “SWIC”. |
| The Individual *Penalty* ID of the *Penalty* provided in the Penalty Modification with Request Type “REMO”, “REIN” or “SWIC” is missing. |
| No *Penalty* exists with the Individual *Penalty* ID provided in the *Penalty Modification* with Request Type “REMO”, “REIN” or “SWIC”. |
| An Individual Penalty ID is provided in the *Penalty Modification* with Request Type “RALO” |
| The Common Penalty ID of the *Penalty* provided in the *Penalty Modification* with Request Type “RALO” is missing |
| No *Penalty* exists with the Common Penalty ID provided in the Penalty Modification with Request Type “RALO” |
| The *Penalty* provided in the *Penalty Modification* is not on its Appeal Period. |
| The *Penalty* provided in the *Penalty Modification* with Request Type “REMO”, “SWIC” or “RALO” does not have Status “Active” (“ACTV”). |
| The *Penalty* provided in a *Penalty Modification* with Request Type “REIN” does not have *Status* “Removed” (“REMO”). |
| The *Penalty* provided in a *Penalty Modification* with Request Type “REIN” is removed because of a previous Re-allocation. |
| A *Penalty Modification* with Request Type “REIN”, “RALO” and “SWIC” has a Reason for removal. |
| A *Penalty Modification* with Request Type “REMO” does not have Reason for removal (“INSO”, “SEMP”, “SESU”, “SUSP”, “TECH”, “OTHR”) or the Reason has an unexpected value. |
| The ‘Description text’ is missing in a *Penalty Modification* with Request Type either “SWIC” or “REMO” and Reason “OTHR”. |
| A *Penalty Modification* with Request Type “REIN”, “RALO” or “REMO” with Reason different from “OTHR” has the ‘Description text'. |
| The Type of the *Penalty* provided in a *Penalty Modification* with Request Type “RALO” is not “LMFP”. |
| The *Penalty* provided in a *Penalty Modification* with Request Type “RALO” has been re-allocated before. |
| The underlying *Settlement Instruction* provided in a *Penalty Modification* with Request Type “RALO” is not an instruction sent as already matched to T2S. |
| In a *Penalty Modification* with Request Type “RALO”, the New Failing Party is neither the delivering nor the receiving party of the underlying *Settlement Instruction* sent already matched. |
| In a *Penalty Modification* with Request Type “RALO” the New Failing Party BIC corresponds to the delivering party of the underlying *Settlement Instruction* but the New Non-Failing Party BIC does not correspond to the receiving party, and vice versa. |
| In a *Penalty Modification* with Request Type “RALO” the T2S reference of the failed Settlement Instruction is neither the delivering nor the receiving *Settlement Instruction* created in T2S. |
| In a *Penalty Modification* with Request Type “RALO” the New Failing Party BIC is equal to the New Non-Failing Party BIC but the T2S Reference of the failed Settlement Instruction is missing. |
| In a *Penalty Modification* with Request Type “RALO” the New Failing Party BIC and the New Non-Failing Party BIC are different but the T2S reference of the failed Settlement Instruction is informed. |
| In a *Penalty Modification* with Request Type “RALO” the New failing Party BIC is missing. |
| In a *Penalty Modification* with Request Type “RALO” the New non-failing Party BIC is missing. |
| In a *Penalty Modification* with Request Type different from “RALO” the New failing Party BIC is informed. |
| In a *Penalty Modification* with Request Type different from “RALO” the New non failing Party BIC is informed. |
| In a *Penalty Modification* with Request Type different from “RALO” the T2S reference of the failed Settlement Instruction is informed. |

In case any of the above described conditions is met, the *Penalty Modification* is rejected (its Status is set to “Rejected”) and the relevant *Request Error* associated to the condition met is stored.

In case none of the above described conditions is met (and the *Penalty Modification* is not rejected), then the *Penalty Modification* is executed (its Status is set to “Executed”), and the *Penalty* is modified as described for each Type of Request in the following sub-functions:

* *Perform Removal:* if the Request Type is “REMO”;
* *Perform Re-inclusion*: if the Request Type is “REIN”;
* *Perform Switch:* if the Request Type is “SWIC”;
* *Perform Re-allocation:* if the Request Type is “RALO”.

Finally, the necessary data informing about the rejection or the execution of the *Penalty Modification* is stored in the Enriched File used to respond to the related *Inbound Bulk File*. The Enriched File will be sent to the Sender of the *Inbound Bulk File* once the Enriched File has the information about the rejection or the execution of all the *Penalty Modifications* included in the *Inbound Bulk File*.

*Validation of Inbound Penalty Modification Messages and Incoming Instruction Approval Requests*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.MOD.3.1* |

The sub-function receives from the Interface domain:

* *Inbound Penalty Modification Messages;*
* *Incoming Instruction Approval Requests*.

Upon reception of an *Inbound Penalty Modification Message*, the sub-function stores it and checks that it complies with the business requirements needed to execute the *Penalty Modification* **{T2S.22.310} {T2S.22.320}** **{T2S.22.350}** **{T2S.22.380}** **{T2S.22.410}** **{T2S.22.420}**.

When the sub-function receives an *Incoming Instruction Approval Request* (approval or revocation) from the Second T2S system user, this sub-function retrieves the referenced *Penalty Modification* (with Approval Status value “Awaiting for Approval”) and:

* In case it is a revocation request, it sets the Approval Status to “Revoked”;
* In case it is an approval request, it retrieves the *Inbound Penalty Modification Message* associated to the *Penalty Modification* and checks again that it complies with the business requirements needed to execute the *Penalty Modification* **{T2S.22.310} {T2S.22.320}** **{T2S.22.350}** **{T2S.22.380}** **{T2S.22.410}** **{T2S.22.420}**.

This sub-function rejects the *Inbound Penalty Modification Message* when one of the conditions in the table below is fulfilled:

|  |
| --- |
| CONDITIONS FOR REJECTION |
| The CSD requesting the *Penalty Modification* with Request Type “REMO”, “REIN” or “SWIC” is not the failing CSD of the *Penalty.* |
| The CSD requesting the *Penalty Modification* with Request Type “RALO” is not the same as the CSD of the “LMFP” (i.e. it is not the CSD of the Instructing Party of the underlying *Settlement Instruction*). |
| The *Penalty Modification* with Request Type “REMO”, “REIN” or “SWIC” does not refer to the Individual Penalty ID of the Failing Party. |
| The *Penalty* provided in the *Penalty Modification* with Request Type “REMO”, “SWIC” or “RALO” does not have Status “Active” (“ACTV”). |
| The *Penalty* provided in a *Penalty Modification* with Request Type “REIN” does not have *Status* “Removed” (“REMO”). |
| The *Penalty* provided in a *Penalty Modification* with Request Type “REIN” is removed because of a previous Re-allocation |
| The Type of the *Penalty* provided in a *Penalty Modification* with Request Type “RALO” is not “LMFP”. |
| The *Penalty* provided in a *Penalty Modification* with Request Type “RALO” has been re-allocated before. |
| The underlying *Settlement Instruction* provided in a *Penalty Modification* with Request Type “RALO” is not an instruction sent as already matched to T2S. |
| In a *Penalty Modification* with Request Type “RALO”, the New Failing Party is neither the delivering nor the receiving party of the underlying *Settlement Instruction* sent already matched. |
| In a *Penalty Modification* with Request Type “RALO” the New Failing Party BIC corresponds to the delivering party of the underlying *Settlement Instruction* but the New Non-Failing Party BIC does not correspond to the receiving party, and vice versa. |
| In a *Penalty Modification* with Request Type “RALO” the T2S reference of the failed Settlement Instruction is neither the delivering nor the receiving *Settlement Instruction* created in T2S already matched. |
| In a *Penalty Modification* with Request Type “RALO” the New non-failing Party BIC is missing. |

In case any of the above described conditions is met, then:

* When it is a validation of an *Inbound Penalty Modification Message,* the relevant *Request Error* associated to the condition met is stored and the information about therejection of the *Outbound Penalty Modification Message* is forwarded to the InterfaceDomain.
* When it is a revalidation due to an *Incoming Instruction Approval Request,* the Approval Status of the related *Penalty Modification* is set to“Rejected”, the relevant *Request Error* associated to the condition met is stored and the information about therejection of the *Approval Request* *Response* is forwarded to the InterfaceDomain.

In case none of the above described conditions is met, then:

* When it is a validation of an *Inbound Penalty Modification Message,* then:
* If it is submitted under Four-Eyes Principle, the *Penalty Modification* is created with Status “Validated” and Approval Status “Awaiting for Approval” and forwards this information to the InterfaceDomain.
* If it is nor submitted under Four-Eyes Principle, the *Penalty Modification* is created and executed (its Status is set to “Executed”), informs InterfaceDomain about the execution, and the *Penalty* is modified accordingly.
* When it is a revalidation due to an *Incoming Instruction Approval Request*, the sub- function sets the Approval Status of the related *Penalty Modification* as “Approved” and informs InterfaceDomain, executes the *Penalty Modification* (its Status is set to “Executed”) and modifies the *Penalty*:

The modification of the *Penalty* is described for each Type of Request in the following sub-functions:

* *Perform Removal:* if the Request Type is “REMO”;
* *Perform Re-inclusion*: if the Request Type is “REIN”;
* *Perform Switch:* if the Request Type is “SWIC”;
* *Perform Re-allocation:* if the Request Type is “RALO”.

*Perform Removal*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.MOD.4.1.* |

The removal of a *Penalty* consists of performing the following actions over it **{T2S.22.330}**:

* Set the Status to “*Removed*” (“REMO”);
* Set the Reason with the value of the Removal Reason Code provided in the *Penalty Modification*, i.e. one of the following*:*
* “'INSO”: *Penalty* was removed because insolvency proceedings are opened against the failing participant;
* “SESU”: *Penalty* was removed because of ISIN suspension from settlement due to a reconciliation issue under Article 65 (2) and (6) of the RTS on CSD Requirements;
* “SUSP”: *Penalty* was removed because of ISIN suspension from trading;
* “SEMP”: *Penalty* was removed because the *Settlement Instructions* involved cash settlement outside the securities settlement system operated by the CSD if, on the respective day, the relevant payment system is closed for settlement;
* “TECH”: *Penalty* was removed because there were technical impossibilities at the CSD level that prevent settlement, such as: a failure of the infrastructure components, a cyber-attack, network problems; or
* “OTHR”: Removed *Penalty* due to a non-standard reason by the CSD.
* In case the Removal Reason Code is “OTHR”, store the Description Text provided in the *Penalty Modification*;
* Set the Amount of the *Penalty* to zero;
* Update the Update Timestamp;
* Set the To be recalculated flag to “False”;
* Flags the *Penalty* for reporting due to its modification but without providing information about its Sub-amounts (Calculation Data) that will not be visible any more.

Once a *Penalty* is removed, it can only be subject to a “Re-inclusion” modification (provided its Appeal Period has not finished yet); any other maintenance request will be rejected.

*Perform Re-inclusion*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.MOD.5.1.* |

The re-inclusion of a *Penalty* consists of performing the following actions over it **{T2S.22.360}**:

* Set the Status to “*Active*” (“ACTV”);
* Set the Reason to “Updated” (“UPDT”);
* Set the Amount of the *Penalty* as the existing one before the *Penalty* was removed;
* Update the Update Timestamp;
* Set the To be recalculated flag to “True”; and
* Select the *Penalty* for reporting due to its modification.

A re-included *Penalty* (provided its Appeal Period has not finished yet) can be subject to any further type of modification request.

*Perform Switch*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.MOD.6.1.* |

The switch of a *Penalty* consists of performing the following actions over it **{T2S.22.430}**:

* Set the Reason to “Switched” (“SWIC”);
* Store the Description Text provided in the *Penalty Modification;*
* Update the Debit/Credit Indicator of both *Applicable Parties* of the *Penalty* as follows:if the value was “Debit” it is updated to “Credit” and vice versa. It is worth mentioning that, as the only attribute updated of the *Applicable Parties* is the Debit/Credit Indicator*,* after the first switch is performed, theIndividual Penalty ID for the (new) Failing Party will start by “N”, and the Individual Penalty ID for the (new) Non-Failing Party will start by “F”.

|  |  |  |
| --- | --- | --- |
| ORIGINAL PENALTY |  | PENALTY SWITCHED |
| Failing Party(“Debit”) | **🡪** | Non-Failing Party (“Credit”) |
| Non-Failing Party(“Credit”) | **🡪** | Failing Party (“Debit”) |

* Updates the Calculation Method when necessary, i.e.:
* If the *Penalty* Calculation Method is “MIXE”, it is updated to “SECU”.
* If the *Penalty* Calculation Method is “SECU”, the Payment Type of the underlying SI is “APTM” and either the Type of Penalty is “SEFP”, or it is “LMFP” with Underlying SI already matched flag “True”, it is updated to “MIXE”;
* Update the Update Timestamp;
* Set the To be recalculated flag to “True”; and
* Select the *Penalty* for reporting due to its modification.

A switched *Penalty* (provided its Appeal Period has not finished yet) can be subject to any further type of modification request.

*Perform Re-allocation*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.MOD.7.1.* |

The re-allocation of a *Penalty* consists of performing a removal of the existing *Penalty* and its substitution by creating a new re-allocated *Penalty* (for the New Failing and New Non-Failing parties), as described below.

The existing *Penalty* is removed **{T2S.22.390}** performing the following actions over it:

* Set the Status to “*Removed*” (“REMO”);
* Set the Reason to “Re-allocated” (“RALO”);
* Set the Amount of the *Penalty* to zero;
* Store the Common Penalty ID of the new *Penalty* re-allocated (created as described below) in the New Penalty After Reallocation;
* Update the Update Timestamp;
* Set the To be recalculated flag to “False”; and
* Select the *Penalty* for reporting due to its modification.

The new re-allocated *Penalty* is created **{T2S.22.370}** with:

* A new Common Penalty ID with 15 characters;
* Status “Active” (“ACTV”);
* Reason “Re-allocated” (RALO);
* Detection Date, Number of Business Days, Currency and Modifiable Flag of the existing *Penalty* (removed as described above);
* The applicable Calculation Method: If the new failing instruction is the receiving leg instead of the delivering one (“RECE” instead of “DELI”) and the Payment Type of the underlying *Settlement Instruction* is “Against Payment” (“APMT”) it is set to “MIXE” (whereas the existing *Penalty* removed was “SECU”); otherwise, it is the same as the one of the existing *Penalty* removed;
* To be recalculated flag “True”;
* Penalty reallocated from equal to the Common Penalty ID of the existing *Penalty* (removed as described above);
* Two new *Applicable Party* instances associated to the new *Penalty* are generated (i.e.: one for the new Failing Party and another one for the new Non-failing Party as provided in the *Penalty Modification*) and, for each one of them, retrieves and stores its necessary attributes, i.e.:
* Individual Penalty ID: Composed of 16 characters (i.e.: F+Common Penalty ID for the new Failing Party; N+Common Penalty ID for the new Non-Failing Party);
* The CSD of the Applicable Party (of the new Failing Party or of the new Non-Failing Party[[11]](#footnote-11));
* Credit / Debit indicator: “Debit” for the new Failing Party or “Credit” for the new Non-Failing Party;
* CCP flag: It indicates whether the *Party* is a CCP or not, as defined in the relevant *Attribute Domain*.
* T2S Reference of the related *Settlement Instruction* (of the Failing Party or of the Non-Failing Party).
* Select the new reallocated *Penalty* for reporting due to its modification.

The *Penalty* removed due to a re-allocation cannot be subject to any *Penalty Modification* request. However, the new re-allocated *Penalty* (provided its Appeal Period has not finished yet) can be subject to any penalty modification request, except to a new “Re-allocation”.

##### **4- Selection for Recalculation**

*Introduction*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.SFR.1.1* |

The Selection for Recalculation function is triggered every T2S settlement day by the *Recalculation of Penalties (RECA)* Event, received from the *Scheduling* *Module* {T2S.03.390} {T2S.03.400}.

Every business day the Static Data Domain provides the Penalty Mechanism Module with the Flow containing the updates (i.e.: insertions, deletions and modifications) occurred since the previous business day deadline (i.e.: since the last file provided with data updates) on the static data used to calculate *Penalties*. The Flow includes *Security Penalty Data* updates, *Cash Discount Penalty Rate* updates, *Euro Foreign Exchange Rate* updates and *List of SME Growth Market* updates.

This function processes the updates informed, selects the *Penalties* affected (directly or via their *Sub-amounts)* and sends them to the Calculation function **{T2S.22.470}**.

The function is divided in the following sub-functions:

* *Updates on Security Subject to Penalties and related data;*
* *Updates on Cash Discount Penalty Rates;*
* *Updates on Euro Foreign Exchange Rates;*
* *Updates on the List of SME Growth Markets;*
* *Recalculation Management.*

*Updates on Security Subject to Penalties and related data*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.SFR.2.1.* |

For each *Security Subject to Penalties and related data* update received in the Flow from Static Data Domain, this sub-function processes it and searches for any affected *Penalty*.

A *Security Subject to Penalties and related data* update contains the following information:

|  |  |
| --- | --- |
| Date | Relevant Business Day[[12]](#footnote-12) to which the updates refer |
| Type of update | Insertion or Modification[[13]](#footnote-13) |
| Security | Relevant *Security* subject to penaltiesto which all the below attributes refer |
| Financial Instrument Type | Possible values: “SHRS”, “SOVR”, “DEBT”, “SECU”, “ETFS”, “UCIT”, “MMKT”, “EMAIL”, “OTHR” |
| Liquidity | Only provided if Financial Instrument Type is “SHRS” |
| Daily Flat Penalty Rate (for non-SMEs) | Applicable *Security Penalty Rate* in case the SME Growth Market flag is “FALSE” |
| Daily Flat Penalty Rate (for SMEs) | Applicable *Security Penalty Rate* in case the SME Growth Market flag is “TRUE” |
| Daily Price (Amount) | Provided if the *Security* has Settlement Type “UNIT” |
| Currency of the Price | Only provided if Price Amount is informed |
| Daily Price (Coefficient) | Provided if the *Security* has Settlement Type “FAMT” |
| Currency of the Security | Only provided if Price Coefficient is informed |

Independently from the type of update received, in order to select a *Penalty*, the function checks that it has:

* The Modifiable Flag “True”;
* The Status different from “REMO”; and
* To be recalculated “False”.

Depending on the possible type of update received, the sub-function performs as follows:

* **Insertion** of a new *Security Subject to Penalties* (and consequent provision of related data):
* The sub-function sets to “True” the To be recalculated of those *Penalties* that have:
* A *Sub-amount* with Security and Date as the ones informed in the update.
* Additionally, the Reason of the Penalty is set to Updated “UPDT” and, if the *Penalty* has Status not Computed “NCOM”, the Status is set to Active “ACTV”.
* **Modification** of the related data of a *Security* *Subject to Penalties* (*Daily Price*, Financial Instrument Type, Liquidity of the *Security* and *Security Penalty Rate*):
* The sub-function selects for recalculation, and sets their Reason to Updated “UPDT”, the *Penalties* that have:
* A Calculation Method different from “CASH”;
* A *Sub-amount* with Security and Date as the ones informed in the update[[14]](#footnote-14).

Once all the updates *Security Subject to Penalties and related data* have been processed, the *Updates on Cash discount penalty rates* sub-function is triggered.

*Updates on Cash discount penalty rates*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.SFR.3.1.* |

For each *Cash Discount Penalty Rate* update received in the Flow from Static Data Domain, this sub-function processes it and searches for any affected *Penalty*.

A *Cash Discount Penalty Rate* update contains the following information:

|  |  |
| --- | --- |
| Date | Relevant Business Day[[15]](#footnote-15) to which the updates refer |
| Currency | Relevant Currency to which all the below attributes refer |
| Type of Update | Insertion, Deletion or Modification |
| Daily Flat Penalty Rate | Applicable *Cash Discount Penalty Rate* |

The sub-function performsin the same way for any type of update received i.e.: insertion, modification or deletion of the *Cash Discount Penalty Rate* used to calculate a *Penalty.*

The sub-function selects for recalculation, and sets their Reason to Updated “UPDT”, the *Penalties* that have:

* The Modifiable Flag “True”;
* The Status different from “REMO”;
* To be recalculated “False”;
* A Calculation Method different from “SECU”; and
* A *Sub-amount* with Currency and Date equal to the ones indicated in the update, as long as the *Sub-amount* flag Security Subject to Penalties is “True”.

Once all the updates on *Cash Discount Penalty Rates* have been processed, the *Updates on Euro Foreign Exchange Rates* sub-function is triggered.

Once the process is finished, the sub-function informs the *Recalculation Management* sub-function.

*Updates on Euro Foreign Exchange Rates*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.SFR.4.1.* |

For each *Euro Foreign Exchange Rate* update received in the Flow from Static Data Domain, this sub-function processes it and searches for any affected *Penalty*.

A *Euro Foreign Exchange Rate* update flow contains the following information:

|  |  |
| --- | --- |
| Date | Relevant Business Day[[16]](#footnote-16) to which the updates refer |
| Currency | Relevant Currency to which all the below attributes refer |
| Type of Update | Insertion, Deletion or Modification |
| Daily Exchange Rate | Applicable *Euro Foreign Exchange Rate* |

The sub-function performsin the same way for any type of update received i.e.: insertion, modification or deletion of the *Euro Foreign Exchange Rates* used to calculate a *Penalty:*

The sub-function selects for recalculation, and sets their Reason to Updated “UPDT”, the *Penalties* that have:

* The Modifiable Flag “True”;
* The Status different from “REMO”
* To be recalculated “False”;
* A Calculation Method different from “CASH”;
* Conversion flag “True”;
* A *Sub-amount* with Currency and Date equal to the ones of the update, as long as the *Sub-amount* flag Security Subject to Penalties is “True”.

Once all the updates on *Euro Foreign Exchange Rates* have been processed, the *Updates on List of SME Growth Markets* sub-function is triggered.

*Updates on List of SME Growth Markets*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.SFR.5.1.* |

For each update in the *List of SME Growth Markets* received in the Flow from Static Data Domain, this sub-function processes it and searches for any affected *Penalty*.

A *List of SME Growth Markets* update contains the following information:

|  |  |
| --- | --- |
| MIC Code | Market Identifier Code |
| Type of Update | Insertion, Deletion or Modification[[17]](#footnote-17) |

Upon any type of update received, the function selects all the *Penalties* that have:

* The Modifiable Flag “True”;
* The Status different from “REMO”;
* To be recalculated “False”;
* Calculation Method “SECU” or “BOTH”; and
* A *Sub-amount* with Common MICequal to the one indicated in the update, as long as the *Sub-amount* flag Security Subject to Penalties is “True”.

Depending on the possible type of update received (insertion and deletion), the sub-function performs as follows:

* **Insertion** of a new SME Growth Market:
* The sub-function sets to “True” the flag SME Growth Market of the affected *Sub-amounts*.
* Additionally, if the Financial Instrument Type of the Security is different from “SOVR” the flag To be recalculated and the Reason of the *Penalty* are set to “True” and to “UPDT” respectively.
* **Deletion** of SME Growth Markets:
* The sub-function sets to “False” the flag SME Growth Market of the affected *Sub-amounts*.
* Additionally, if the Financial Instrument Type of the Security is different from “SOVR” the flag To be recalculated and the Reason of the *Penalty* are set to “True” and to “UPDT” respectively.

Once all the updates on the *List of SME Growth Markets* have been processed, the *Recalculation Management* sub-function is triggered.

*Recalculation Management*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.SFR.5.1.* |

After the previous sub-functions have finished their processing, this sub-function sends:

* The *Penalties* that are either flagged as To be recalculated “True” and that do not have *Sub-amounts* (i.e. it is a new reallocated Penalty), or *Penalties* with Detection Date before the previous business day and Status “PEND” to the sub-function *Penalty Preparation* in theCalculation function;
* The *Penalties* and their *Sub-amount(s),* when the *Penalty* is flagged as To be recalculated “True” to the sub-function *Sub-amount Preparation* in theCalculation function; and
* The *Penalties* and their *Sub-amount(s),* when the *Penalty* is flagged as To be recalculated “False” but has been selected for recalculation by the previous sub-functions, to the sub-function *Sub-amount Preparation* of the Calculation function.

Once the *Calculation* function has processed all the *Penalties*, the sub-function the End of Process Event (*Recalculation of Penalties*) to the *Scheduling* *Module.*

##### **5- Appeal Period End Process**

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.APE.1.1.* |

The Appeal Period End Process function is triggered every T2S settlement day by the *End of Appeal Period Process* *(EAPP)* Event, but it only performs the update of *Penalties* when the Appeal Period of the previous month finishes.

This function is responsible for updating the *Penalties* detected on the previous month that have reached the last day of their corresponding Appeal Period, by setting their Modifiable flag to “False”, which means that:

* These *Penalties* and their *Sub-amounts* will not be selected for recalculation **{T2S.22.470}**; and
* No further modification requests will be accepted on these *Penalties* **{T2S.22.440}**.

Upon reception of the *End of Appeal Period Process* *(EAPP)* Event, this function compares the value of the *Appeal Period End Day* stored in Static Data (13th business day of every month) with the value of the system parameter, *Current Nth business day of the month,* and if they are the same, the function selects all the *Penalties* whose Detection Date is in the previous calendar month and updates their Modifiable flag from “True” to “False”.

Finally this function sends the End of Process Event *(End of Appeal Period Process - EAPP)* to the *Scheduling* *Module.*

##### **6- Preparation for Reporting**

*Introduction*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.PPR.1.1.* |

This function is triggered every T2S settlement day by the *Report Preparation for New Penalties* *(RPDA)* Event, the *Report Preparation for Modified Penalties* *(RPMD)* Event and the *Report Preparation of Monthly Aggregated Amounts* *(SRPM)* Event {T2S.03.430}.

This function is in charge of preparing the information to be reported in the *Daily Penalty List*, the List of Modified Penalties and the Monthly Aggregated Amounts.

|  |  |  |
| --- | --- | --- |
| REPORT NAME | REPORT CONTENT | URD REFERENCES |
| *Daily Penalty List* | It reports per Currency, T2S *Party* and Counterparty all the new *Penalties* computed for the previous T2S settlement day (i.e. with Detection Date equal to “current Business day -1”) | **{T2S.13.320}**  **{T2S.13.340}** |
| *List of Modified Penalties* | It reports per Currency, T2S *Party* and Counterparty, the updates performed, since the previous report, in *Penalties* with Detection Date before “current Business day -1” | **{T2S.13.390}**  **{T2S.13.410}** |
| *Monthly Aggregated Amounts of Penalties* | It reports per Currency and T2S *Party and Counterparty*, the aggregated bilateral net amount of *Penalties* computed for the T2S settlement day of the previous month (i.e. with Detection Date within the previous month). | **{T2S.13.460}**  **{T2S.13.470}** |

Once the Preparation for Reporting function has prepared the information, the relevant report can be generated upon reception of the corresponding Event. For the creation of these reports, T2S relies on Static Data Domain*.* The information specifying which event triggers which report for the data of a party, is stored in the *Report Configuration* in *Static Data*. T2S will create only reports pre-defined in the *Report Configuration*.

The Preparation for Reporting function is divided in the following sub-functions:

* *Preparation for Daily Penalty List*
* *Preparation for List of Modified Penalties*
* *Preparation for Monthly Aggregated Amounts Penalties*

*Preparation for Daily Penalty List Report*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.PPR.2.1.* |

Upon reception of the *Report Preparation for New Penalties* (*RPDA*) Event, this sub-function retrieves and prepares the necessary data for the possible generation of any pre-defined *Daily Penalty List* **{T2S.13.340} {T2S.13.350}**, i.e.:

* The sub-function prepares ad-hoc tables with the necessary information of the *Penalties* with Detection Date equal to “current Business day -1” and Status “ACTV”, their *Applicable Parties, Sub-amounts and* underlying *Settlement Instruction*, as well as the *Daily Nets for a party vs counterpart.*
* For this purpose, the sub-function calculates the *Daily Nets for a party vs counterpart* as follows:
* From all the *Applicable Parties* related to *Penalties* with Detection Date equal to “current Business day -1” and Status “ACTV”, the sub-function creates a *Daily Net for a party vs counterpart* for each pair of“Applicable Party andCounterpart”*.*
* Then,the sub-function adds or subtracts the Amount of the *Penalty* of each *Applicable Party* with the related pair of“Party andCounterparty”*,* i.e. if the Credit/Debit Indicator of the *Applicable Party* is “Debit”, the Amount of the *Penalty* is subtracted to the Net Amount of the *Daily Aggregated Net Amount*; whereas if it is “Credit”, the Amount of the *Penalty* is added to the Net Amount of the *Daily Aggregated Net Amount*.
* If the sign of the resulting Net Amount of the *Daily Aggregated Net Amount* is positive, the Credit/Debit Indicator attribute is set as “Credit”; whereas if it is negative, the Credit/Debit Indicator attribute is set as “Debit”.

Finally, this sub-function sends the End of Process Event (*Report Preparation for New Penalties)* to the *Scheduling* *Module*.

*Preparation for List of Modified Penalties*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.PPR.3.1.* |

Upon reception of the *Report Preparation for Modified Penalties* (*RPMD*) Event, this sub-function retrieves and prepares the necessary data for the possible generation of any pre-defined *List* *of Modified Penalties* **{T2S.13.410} {T2S.13.420}**, i.e.:

* The sub-function prepares ad-hoc tables with the necessary information of the updated *Penalties* (flagged for reporting due to its recalculation or modification, hence they must have Detection Date before “current Business day -1” and Status “ACTV” or “REMO”), their *Applicable Parties* and *Sub-amounts* (only for *Penalties* with Status “ACTV”),as well as the updated *Daily Nets for a party vs counterpart.*
* For this purpose, the sub-function calculates the updated *Daily Net for a party vs counterpart* as follows:
* Selects all the different Detection Dates of the updated *Penalties,* and for each one:
* From all the *Applicable Parties* related to *Penalties* with the relevant Detection Date and Status “ACTV”, the sub-function calculates the new *Daily Net for a party vs counterpart* for each pair of“Party andCounterpart”*.*
* The sub-function adds or subtracts the Amount of the *Penalty* of each *Applicable Party* with the related pair of“Party andCounterpart”*,* i.e. if the Credit/Debit Indicator of the *Applicable Party* is “Debit”, the Amount of the *Penalty* is subtracted to the Net Amount of the *Daily Aggregated Net Amount*; whereas if it is “Credit”, the Amount of the *Penalty* is added to the Net Amount of the *Daily Aggregated Net Amount*.
* If the sign of the resulting Net Amount of the *Daily Aggregated Net Amount* is positive, the Credit/Debit Indicator attribute is set as “Credit”; whereas if it is negative, the Credit/Debit Indicator attribute is set as “Debit”.

Finally, this sub-function sends the End of Process Event (*Report Preparation for Modified Penalties*) to the *Scheduling* *Module*.

*Preparation for Monthly Aggregated Amounts of Penalties Report*

|  |  |
| --- | --- |
| Reference Id | *LCMM.PEM.PPR.4.1.* |

Upon reception of the *Report Preparation of Monthly Aggregated Amounts* *(SRPM)* Event, this function compares the value of the *Appeal Period End Day* stored in Static Data (13th business day of every month) with the value of the system parameter, *Current Nth business day of the month,* and if they are the same, this sub-function retrieves and prepares the necessary data for the possible generation of any pre-defined *Monthly Aggregated Amounts* *of* *Penalties* on the relevant business day defined by the *Monthly Reporting Day* stored in Static Data (14th business day of the month) **{T2S.13.460} {T2S.13.470}**, i.e.:.

* The sub-function prepares ad-hoc tables with the necessary information of the *Penalties* with Detection Date within the previous month and Status “ACTV” and their *Applicable Parties*,as well as the *Monthly Nets for a party vs counterpart.*
* For this purpose, the sub-function calculates the *Monthly Nets for a party vs counterpart* as follows:
* Selects all the different Detection Dates of the previous month*,* and for each one:
* From all the *Applicable Parties* related to *Penalties* with the relevant Detection Date and Status “ACTV”, the sub-function creates a *Monthly Net for a party vs counterpart* for each pair of“Party andCounterparty”*.*
* Then,the sub-function adds or subtracts the Amount of the *Penalty* of each *Applicable Party* with the related pair of“Party andCounterparty”*,* i.e. if the Credit/Debit Indicator of the *Applicable Party* is “Debit”, the Amount of the *Penalty* is subtracted to the Net Amount of the *Monthly Aggregated Net Amount*; whereas if it is “Credit”, the Amount of the *Penalty* is added to the Net Amount of the *Monthly Aggregated Net Amount*.
* If the sign of the resulting Net Amount of the *Monthly Aggregated Net Amount* is positive, the Credit/Debit Indicator attribute is set as “Credit”; whereas if it is negative, the Credit/Debit Indicator attribute is set as “Debit”.
* Additionally, for all the *Monthly Aggregated Net Amount,* the sub-function verifies that their Net Amount is the sameas the sum of theNet Amounts of the *Daily Aggregated Net Amount* forBusiness dayswithin the related month and the samePartyandCounterparty*.*

Finally, this sub-function sends the End of Process Event (*Report Preparation of Monthly Aggregated Amounts*) to the *Scheduling* *Module*.

#### 3.4.7.3 Description of Penalties status transitions



The Penalty Eligibility function creates a *Penalty* with Status Pending ("PEND"). This is an intermediate status until the Calculation function retrieves and processes the *Penalty*.

The Calculation function checks whether the *Security* of the Underlying *Settlement Instruction* is subject to penalties or not on each of the applicable business days of the Penalty:

* If the *Security* is subject to penalties at least in one of the applicable business days, the *Penalty* Status is updated to Active ("ACTV") and without any Reason;
* If the *Security* is not subject to penalties in any of the applicable business days, the *Penalty* Status is updated to Not Computed ("NCOM") and without any Reason.

During the Appeal Period of the *Penalty*, it can be subject to modifications and recalculations.

The Penalty Modification function can update the Status of the *Penalty* depending on the type of the *Penalty Modification* executed:

* In case of a removal, the *Penalty* Status is updated to Removed ("REMO") with the corresponding Reason (“INSO”, “SEMP”, “SESU”, “SUSP”, “TECH” or “OTHR”) indicated in the *Penalty Modification*;
* In case of a re-inclusion (which can only be performed on *Penalties* with Status "REMO" and a Reason different from Re-allocated “RALO”), the *Penalty* Status is updated to Active ("ACTV") with Reason Updated (“UPDT”);
* In case of a switch, the *Penalty* Status remains as Active ("ACTV") but the Reason is changed to Switched (“SWIC”); and
* In case of a re-allocation, the Status of the *Penalty* is updated to Removed ("REMO") with Reason Re-allocated (“RALO”), and a new re-allocated *Penalty* is generated with Status Active ("ACTV") and Reason Re-allocated (“RALO”).

The Selection for Recalculation function can update the Status of the *Penalty* depending on the type of the Static Data update received:

* In case an insertion of Securities subject to penalties impacts a *Penalty* with Status Not Computed ("NCOM"), the *Penalty* Status is updated to Active ("ACTV") with Reason Updated (“UPDT”); and
* In any other case, the *Penalty* Status remains as Active ("ACTV") but the Reason is changed to Updated (“UPDT”).

The statuses depicted in red in the above diagram are possible final statuses. They become final upon the end of their appeal period, i.e.: when the Appeal Process End function updates the *Penalty* Modifiable Flag to “False”, the Status that the *Penalty* has at this point in time becomes its final Status.

#### 3.4.7.4 Description of the Input / Output of the module

| **FLOW** | IN/OUT | **DESCRIPTION** | FROM | TO |
| --- | --- | --- | --- | --- |
| Event (Start Cash Penalty Eligibility) | IN | This event is sent in order to trigger the start of the Penalty Eligibility. | OPRS: Scheduling |  |
| End of Process Event (Start Cash Penalty Eligibility) | OUT | This end of process event is sent when the Penalty Eligibility is completed. |  | OPRS: Scheduling |
| Event (Calculation of Penalties) | IN | This event is sent in order to trigger thecalculation of new *Penalties*. | OPRS: Scheduling |  |
| End of Process Event (Calculation of Penalties Detected) | OUT | This end of process event is sent when thecalculation of new *Penalties* is completed. |  | OPRS: Scheduling |
| Inbound Bulk File | IN | Flat file received for bulk loading *Penalty Modifications* | ICM: A2A Mode |  |
| Inbound Penalty Modification Message | IN | Flow received for communicating *Penalty Modifications* | INTF: Inbound processing |  |
| Incoming Instruction Approval Request | IN | Flow received to update the Approval Status of *Penalty Modification* requests requiring 4 eyes processing. | INTF: Inbound processing |  |
| Penalty Modification Response Flat File | OUT | Flat sent with the responses to the *Penalty Modifications* requested via flat file. |  | ICM: A2A mode |
| Penalty Modification Response U2A | OUT | Flow sent with the response to a *Penalty Modification* requested via U2A. |  | INTF: Outbound processing |
| Event (Recalculation of penalties) | IN | This event is sent in order to trigger the Selection for Recalculation. | OPRS: Scheduling |  |
| Static data Delta File | IN | Flow with the Static Data updates over *Security Penalty Data*,  *Cash Discount Penalty Rates, Euro Foreign Exchange Rates* and in the *List of SME Growth Markets* | SDMG: Penalties |  |
| End of Process Event (Recalculation of penalties) | OUT | This end of process event is sent when the Selection for Recalculation is completed. |  | OPRS: Scheduling |
| Event (End of Appeal Period) | IN | This event is sent in order to trigger the Appeal Period End Process. | OPRS: Scheduling |  |
| End of Process Event (End of Appeal Period) | OUT | This end of process event is sent when the Appeal Period End Process is completed. |  | OPRS: Scheduling |
| Event (Report Preparation for New Penalties*)* | IN | This event is sent in order to trigger the preparation for reporting of new *Penalties* in the *Daily Penalty List*. | OPRS: Scheduling |  |
| Event (Report Preparation for Modified Penalties) | IN | This event is sent in order to trigger the preparation for reporting of updated *Penalties* in the *List of Modified Penalties.* | OPRS: Scheduling |  |
| Event (Report Preparation of Monthly Aggregated Amounts) | IN | This event is sent in order to trigger the preparation for reporting of the *Monthly Aggregated Amounts.* | OPRS: Scheduling |  |
| End of Process Event (Report Preparation for New Penalties) | OUT | This end of process event is sent when the preparation for reporting of new *Penalties* in the *Daily Penalty List* is completed. |  | OPRS: Scheduling |
| End of Process Event (Report Preparation for Modified Penalties) | OUT | This end of process event is sent when the preparation for reporting of updated *Penalties* in the *List of Modified Penalties* is completed. |  | OPRS: Scheduling |
| End of Process Event (Start Report Preparation of Monthly Aggregated Amounts) | OUT | This end of process event is sent when the preparation for reporting of *Monthly Aggregated Amounts of Penalties* is completed. |  | OPRS: Scheduling |

#### 3.4.7.5 Data accessed by the module

| **DATA** | DATA ENTITY | **ACCESS MODE** | COMMENTS |
| --- | --- | --- | --- |
| **STATIC DATA** | | | |
| Static Data | Closing Day | Read | Accessed for checking purposes |
| Securities | Read | Accessed for checking purposes |
| Securities subject to Penalties | Read | Accessed for checking purposes |
| SME Growth Market | Read | Accessed for checking purposes |
| Security Penalty Rate | Read | Accessed for checking purposes |
| Cash Discount Penalty Rate | Read | Accessed for checking purposes |
| Euro Foreign Exchange Rate | Read | Accessed for checking purposes |
| Currency | Read | Accessed for checking purposes |
| Daily Price | Read | Accessed for checking purposes |
| Party | Read | Accessed for checking purposes |
| Party Code | Read | Accessed for checking purposes |
| *Attribute Domain- List of CCPs and “white list”* | Read | Accessed for checking purposes |
| *System parameter- Appeal Period End Day* | Read | Accessed for checking purposes |
| *System parameter-*  *Monthly Reporting Day* | Read | Accessed for checking purposes |
| *System parameter-*  Nth BD of the month | Read | Accessed for checking purposes |
| **DINAMIC DATA** | | | |
| LCMM Instructions | *Settlement Instruction* | Read |  |
| *Matching Object* | Read |  |
| *Reason History* | Read |  |
| *Status History* | Read |  |
| Penalties | *Penalty Eligibility* | Read/Write |  |
| *Penalty* | Read/Write |  |
| *Applicable Party* | Read/Write |  |
| *Failing Reasons Dictionary* | Read |  |
| *Reasons for failing* | Write |  |
| *Penalty Reference Data for Securities* | Read/Write |  |
| *Sub-amount* | Read/Write |  |
| *Sub-amount (securities)* | Read/Write |  |
| *Sub-amount (cash)* | Read/Write |  |
| *Penalty Modification* | Read/Write |  |
| *Inbound Bulk File* | Read/Write |  |
| *Inbound Penalty Modification message* | Read/Write |  |
| *Request Error* | Write |  |
| *Daily net for a Party vs Counterparty* | Read/Write |  |
| *Monthly net for a Party vs Counterparty* | Read/Write |  |

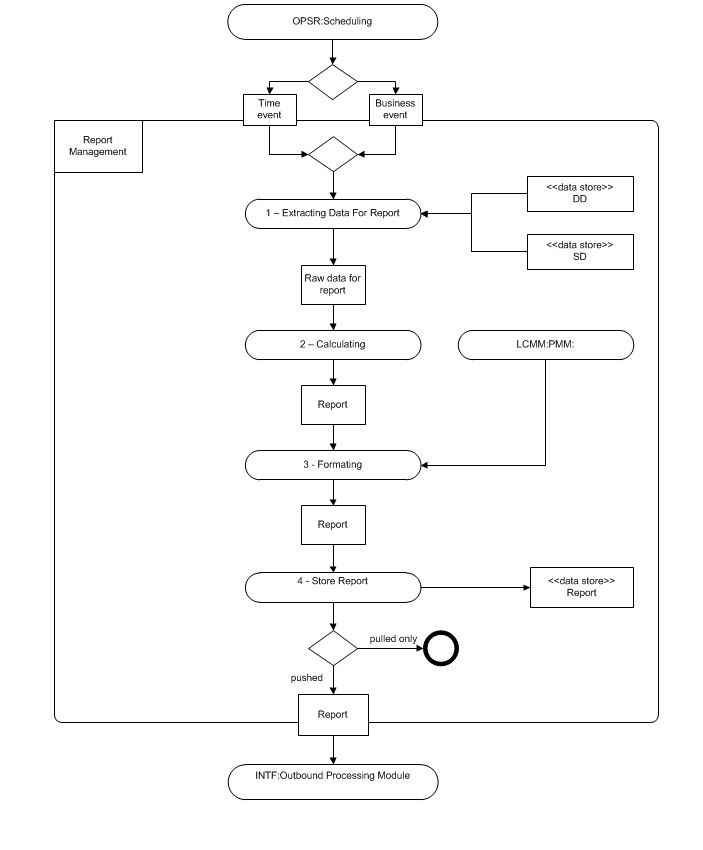
## 3.7 Statistics, Queries, Reports and Legal Archiving

### 3.7.4 Report Management

…

#### 3.7.4.2 Static Functional Description

***Diagram of the module***



## 4.1 List of Use Cases

*…*

### 4.1.5 QU: Queries

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **CRITERIA** | | | **REPRESENTATIVE USE CASE** |
| **COMMUNICATION**  **MODE** | **QUERY CATEGORY** | **QUERY TYPE** |
| … | … | … | … |  |
| 211b | U2A | Dynamic Queries | Itemised Billing Data Query |  |
| 211c | U2A | Penalty | Cash Discount Penalty Rate Query |
| 211d | U2A | Penalty | Daily Price Query |
| 211e | U2A | Penalty | Euro Foreign Exchange Reference Rate Query |
| 211f | U2A | Penalty | Monthly Net Penalty Amount Query |
| 211g | U2A | Penalty | Penalty Details Query |
| 211h | U2A | Penalty | Penalty List Query |
| 211i | U2A | Penalty | Securities Penalty Rate Query |
| 211j | U2A | Penalty | Securities Subject to Cash Penalties Query |
| 211k | U2A | Penalty | SME Growth Market |

### 4.1.6 RE: Reports

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **CRITERIA** | | | | | | **REPRESENTATIVE  USE CASE** |
| **COMMUNICATION**  **MODE** | **INFORMATION**  **BASIS** | **TRIGGERING** | **CLASSIFICATION** | **CATEGORY** | **TYPE** |
| … | … | … | … | … | … | … |  |
| 283 | A2A | Party | Business Event | Delta Report | Statement of transactions | Securities instructions |  |
| 283a | A2A | Penalty | Business Event | Complete Report | *Daily Penalty List* | Penalty |  |
| 283b | A2A | Penalty | Business Event | Complete Report | *List of Modified Penalties* | Penalty |  |
| 284c | A2A | Penalty | Business Event | Complete Report | *Monthly Aggregated Amounts* | Penalty |  |

1. |  |  |
   | --- | --- |
   |  | See section [3.3.11.13 [} 1]](#_Ref4C7FC45BE75C11BC1021C4B9D3AA0C29) for a description of the Country and Currency entities. |

   [↑](#footnote-ref-1)
2. |  |  |
   | --- | --- |
   |  | For each security restriction, a period of validity, a restricting party (i.e. the T2S operator or a CSD) and a restriction type (see section [3.3.11.10 [} 1]](#_RefAC3996E334EE7D1239E7F71DF2EB2997)) must be specified. |

   [↑](#footnote-ref-2)
3. |  |  |
   | --- | --- |
   |  | For each market-specific attribute, a valid value must be specified. The possibility to manage market-specific attributes shall be given not only to issuer CSD but also to investor CSDs, so to make them able to define restriction types based on their securities market-specific attributes. |

   [↑](#footnote-ref-3)
4. |  |  |
   | --- | --- |
   |  | For each Security CSD link, a period of validity, the type of link (i.e. if the CSD is issuer, investor or technical issuer for the relevant security) and a Boolean value indicating if the CSD is responsible for maintaining the security must be specified (for each security one and only one CSD can be responsible for its reference data maintenance including the cash penalties configuration). Moreover, one or many issuance accounts can be defined within a given Security CSD Link. |

   [↑](#footnote-ref-4)
5. |  |  |
   | --- | --- |
   |  | Each Close Link defines a link between the relevant security and a party, i.e. specifies that the security is not eligible as collateral for that party. |

   [↑](#footnote-ref-5)
6. Or to reprocess an existing *Penalty Eligibility* with Reason for Exemption “REPR” sent by the Eligibility End sub-function [↑](#footnote-ref-6)
7. The exhaustive list is provided in the UDFS [↑](#footnote-ref-7)
8. In case a Reason Code is missing in the “Failing reasons Dictionary” an alarm is raised to the T2S Operator [↑](#footnote-ref-8)
9. In case of an External CSD settlement scenario, the CSD of the Applicable Party is the parent CSD of the External-CSD. [↑](#footnote-ref-9)
10. BICs informed in the Header of the Bulk File ( “BIC of the CSD” and “Parent BIC of the CSD” respectively) [↑](#footnote-ref-10)
11. In case of an External CSD settlement scenario, the CSD of the Applicable Party is the parent CSD of the External-CSD. [↑](#footnote-ref-11)
12. It must be a day before the previous business day [↑](#footnote-ref-12)
13. There is no deletion because a *Security* cannot be removed from the *Security Subject to Penalties* in the past [↑](#footnote-ref-13)
14. Given that the update is a modification of the related data of a *Security Subject to Penalties*, the *Sub-amounts* affected would have the Security Subject to Penalties flag set to “True” [↑](#footnote-ref-14)
15. It must be a day before the previous business day [↑](#footnote-ref-15)
16. It must be a day before the previous business day [↑](#footnote-ref-16)
17. In this case a modification is treated as a deletion and an insertion [↑](#footnote-ref-17)