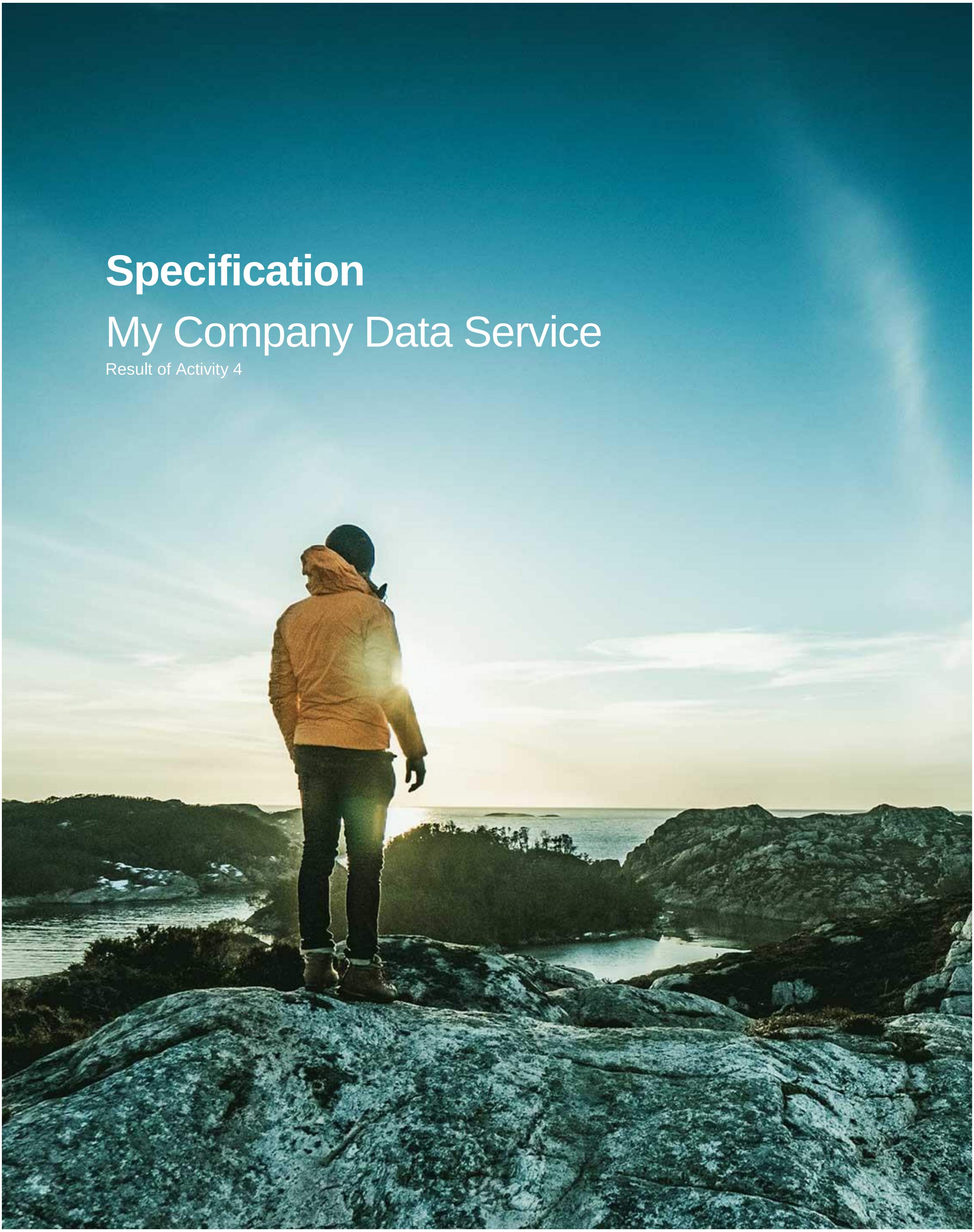


Specification

My Company Data Service

Result of Activity 4

Co-financed by the European Union
Connecting Europe Facility

tietoEVRY

Contents

Co-financed by the European Union
Connecting Europe Facility

Summary

The document describes the My Company Data Service (MCDS). MCDS is a server software for accounting service providers and government authorities to exchange economic transactions data and reports.

In the document you can see MCDS use case descriptions and additional specifications like accounting entry data standard, mappings between accounting data entry and different other standards.

The use case description defines what features the MCDS has. Accounting entry data standard describes what elements an accounting entry must contain. The standard is based on XBRL GL 2015.

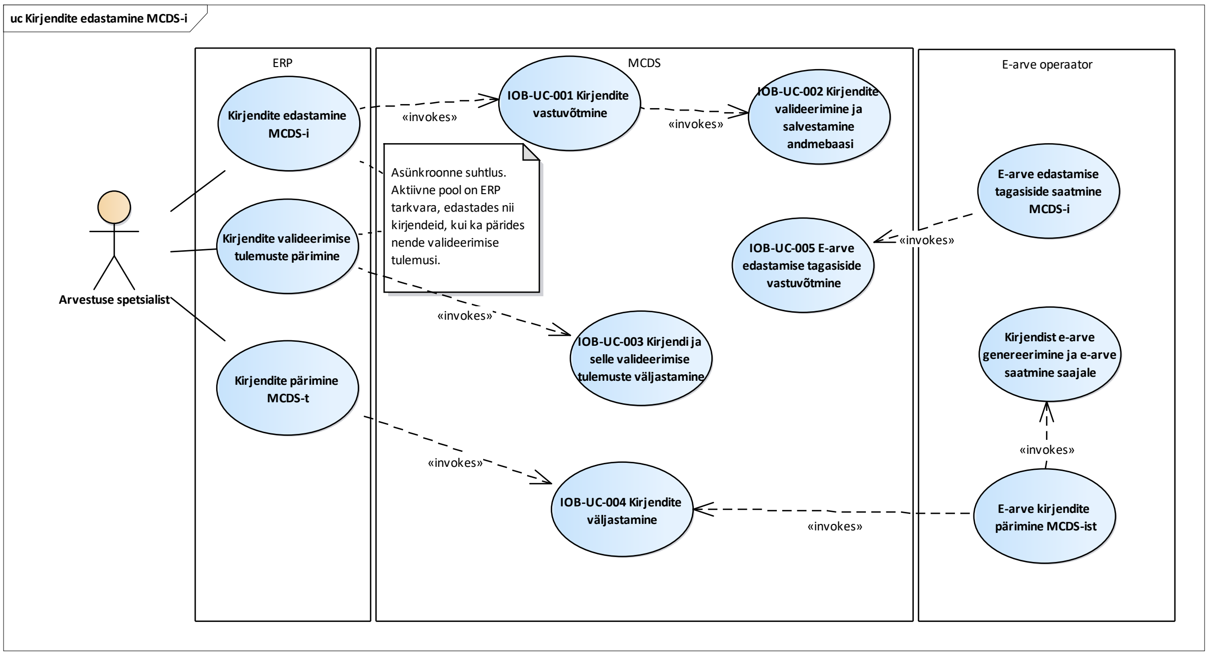
Mappings attached the document helps XBRL GL data users to map their data structures to accounting data entry standard.

Co-financed by the European Union
Connecting Europe Facility

1 Use Case Descriptions

Use case description defines features of the MCDS system.

1.1 Accounting entry management


uc Kirjendite edastamine MCDS-i
ERP
MCDS
E-arve operaator
Arvestuse spetsialist
Kirjendite edastamine MCDS-i
Kirjendite valideerimise tulemuste pärimine
Kirjendite pärimine MCDS-t
IOB-UC-001 Kirjendite vastuvõtmine
IOB-UC-002 Kirjendite valideerimine ja salvestamine andmebaasi
IOB-UC-003 Kirjendi ja selle valideerimise tulemuste väljastamine
IOB-UC-004 Kirjendite väljastamine
IOB-UC-005 E-arve edastamise tagasiside vastuvõtmine
E-arve edastamise tagasiside saatmine MCDS-i
Kirjendist e-arve genereerimine ja e-arve saatmine saajale
E-arve kirjendite pärimine MCDS-ist
«invokes»
«invokes»
«invokes»
«invokes»
«invokes»
«invokes»
«invokes»

Asünkroonne suhtlus. Aktiivne pool on ERP tarkvara, edastades nii kirjendeid, kui ka pärides nende valideerimise tulemusi.

Figure 1: Accounting entries transmission to MCDS

1.1.1 IOB-UC-001 Receiving accounting entries

MCDS includes a web service to transmit accounting entries data. All records transmitted to the system undergo two-level validation. First, the validation of the structure of the xml, then the validation against business rules.
Data transmission service needs the service user to be authenticated.

Input REST Json:
1. Data transmitter ERP ID, 1..1.
2. The data in XBRL GL XML format, Base64 encoded, 1..1. The input XML should contain one or several accounting entries.

Output REST Json:
1. Error code, 1..1.
2. Shipment ID - NEW, 1..1.

Error codes:
000 - The errors are not generated.
001 - Technical error.
002 - XBRL is not valid.
003 - ERP ID invalid or missing.
004 - ERP-l there are no rights for at least one input entry to transmission to MCDS.
005 - XML does not contain the data owner registry code.

Co-financed by the European Union
Connecting Europe Facility

Flow	Steps
Basic Path	1 The system checks for the presence of an ERP ID among the elements of the classification "ERP". In case, if the ERP ID value of the classification of elements from not found, there will be error 003. 2 The system will validate incoming XML against the XBRL GL version “XBRL Global Ledger 2015. In case, if the XML does not validate, get error 002 and use case ends, because of the rights check, what requires a valid XML for getting the organization identifier from the XML. In case, if in the XML is not a block gl- bus:organizationIdentifiers, where GL-bus:organizationDescription == "business_reg_number" and GL-bus: organizationIdentifier != NULL, an error 005 occures. Value "business_reg_number" indicates that the bus:organizationIdentifier element contains business registry code. It must be accounting entry owner registry code. 3 The system checks the validity of ERP authorizations in the system. ERP have the right to transmit the data, if the received input entry any case applies to the following condition for permission table: permission.erp = classifier_item.id AND classifier_ item.code == ERP ID AND ((classifier_ item.valid _from <> NULL AND classifier_ item.valid _until <> NULL AND classifier_ item.valid _from <= current date AND current date <= classifier_ item.valid _until) OR (classifier_ item.valid _from == NULL AND classifier_item.valid_until == NULL) OR (classifier_ item.valid _from <> NULL AND classifier_ item.valid _until == NULL AND classifier_ item.valid _from <= current date)) AND at least one gl-bus_organizati onIdentifier == organization.regnr AND in the record XML permission.id_orga == organization.id AND

Co-financed by the European Union
Connecting Europe Facility

	(permission.business _area == NULL OR permission.business_area == record in XML gl-bus: sourceApplication) AND permission.report_type == NULL. In the XML in the input, the system checks each record separately, and if the above condition does not apply, error 004 occurs and the use case ends. 4 The system saves the shipment in the MCDS database, table package. 5 The system returns the XML sender service output data. 6 System invokes "IOB-UC-002 Validation of entries and storage in the database". 7 The end.

1.1.2 IOB-UC-002 Validation of entries and storage in the database

All records submitted to the MCDS pass validation. The validation result is an information that whether the entry is in accordance with the applicable rules corrupted or okay. In addition, the system stores
in MCDS database also all entries remarks. These are the errors and warnings, which is due to defined rules.

Input REST Json:
1. Shipment ID, 1..1.

Output REST Json:
1. Error code, 1..1.

Error codes:
000 - No errors occurred.
001 - Technical error.

Flow	**Steps**
Basic Path	1 The system checks the received XML against compliance with the business rules, which are described in the document of business rules. One entry represents XBRL GL XML gl- cor:entryHeader block contents + the prior information is in the upper half of the. This means that when checking an entry, the system generates as many separate xmls from the original xml as there are gl-cor:entryHeader blocks in the original xml. One xml entry contains only one gl-cor: entryHeader block. 2 The system saves each validated entry into the separate record in the database table accounting_entry, saving:

Co-financed by the European Union
Connecting Europe Facility

	• Field accounting_entry_uuid, element gl- cor:entryHeader/@id; • Field entry_type, element gl-cor_entriesType; • Field target_application, element gl-bus:targetApplication; • Field accounting_entry_business_area, element gl-bus:sourceApplication. In addition, the system adds an entry to the accounting_entry_detail table for each entryDetail block, indicating the date contained in the gl-cor: postingDate element in the accounting_entry_detail.posting_date field. 3 The system saves a summary of the validation results in table validation_result. In addition, the accounting_entry.valid field in the record table indicates TRUE if the record was validated and FALSE if errors were found. In addition, the system enters the name and version number of the rules for which the record was checked in the accounting_entry.rule field of the record. 4 The end.

1.1.3 IOB-UC-003 Output of the record and its validation results

The MCDS outputs the results of the record validation. No input parameters are required (except ERP ID). If the input is not sent, outputs the system last calendar month data, you query is entitled to see.

Input REST Json:
1. ERP ID, 1..1.
2. Shipment (XBRL GL Record XML) ID, 0..1.
3. Start time of sending, 0..1.
4. End time of sending, 0.. 1.
5. Record identifier in MCDS, XBRL element gl-cor_uniqueID, 0..1.

Output REST Json:
1. Error code, 1..1.
2. Shipment, 0..n.
2.1. Shipment ID, 1.
2.2. Shipment delivery time.
2.3. Item, 0.. n.
2.3.1. Record ID, 1..1 (gl-cor_uniqueID).
...
2.3.n. Error, 0..n.
2.3.n.1. Error code, 1..1.
2.3.n.2. Error text, 1..1.
2.3.n.3. Rule code, 1..1.
2.3.n.4. Name and version of the rules, 1..1.
2.3.n + 1. Invoice, 0..1.

Co-financed by the European Union
Connecting Europe Facility

2.3.n + 1.1. Invoice id, 1..1.
2.3.n + 1.2. Invoice xml, 1..1.
2.3.n + 1.3. Invoice creation time, 1..1.
2.3.n + 1.4. Invoice status, 0.. 1.
2.3.n + 1.4.1. Status, 1..1.
2.3.n + 1.4.2. Message, 0..1.
2.3.n + 1.4.3. Creation time, 1..1.

Error codes:
000 - No errors occurred.
001 - Technical error.
002 - ERP ID invalid.
003 - ERP does not have the rights to use the requested data.

Flow	**Steps**
Basic Path	1 The system queries database records, requesting all data for the current month (according to accounting_entry.created) from the accounting_entry table, which the heir (ERP) has the right to see in the absence of input parameters. ERP has the right to see the entry, if: permission.erp = classifier_item.id AND classifier_ item.code == ERP ID AND ((classifier_ item.valid _from <> NULL AND classifier_ item.valid _until <> NULL AND classifier_ item.valid _from <= current date AND current date <= classifier_ item.valid _until) OR (classifier_ item.valid _from == NULL AND classifier_item.valid_until == NULL) OR (classifier_ item.valid _from <> NULL AND classifier_ item.valid _until == NULL AND classifier_ item.valid _from <= current date)) AND at least one gl-bus_organizationIdentifier == organization.regnr AND in the record XML permission.id_orga == organization.id AND

	permission.business _area == NULL AND (Permission.business _area == NULL OR permission.business_area == kirjendi XML GL-bus: so urceApplication) AND permission.report _type == NULL. 2 In the case, if the query to the given input parameters, filters the system out the data of the following elements of: - Shipment ID = package.package _uuid. - Record ID (gl-cor_uniqueID) = accounting_ entry.entry _uuid. - package.created is within the time period in the input. 3 The system outputs accounting entries (table accounting_entry) and accounting entry comments (table validation_result) data. 4 The end.

1.1.4 IOB-UC-004 Output of entries

The MCDS issues the records according to the parameters specified by the requester. The requester must have the appropriate authorization to obtain the data.

Input REST Json:
1. ERP ID, 1..1.
2. Record owner code (registry code), gl-bus_organizationIdentifier, 1..1.
3. Entry creation time, MCDS's recording time by, 0..1.
4. Entry creation end time, MCDS's recording time by, 0..1.
5. Entry type, gl-cor_entriesType, 0..1.
6. Entry identifier(s) in MCDS, XBRL element gl-cor_uniqueID, 0..n.

If the input parameters are not sent, then the system outputs all current month information.

Output REST Json:
1. Error code, 1..1
2. Entry, 0.. n.
2.1. Record ID.
...
(entries data in the full composition of the entry elements)
...

Output encoded to Base64 encoding.

Error codes:
000 - The errors are not generated.
001 - Technical error.
002 - ERP ID invalid.

Co-financed by the European Union
Connecting Europe Facility

003 - ERP does not have the rights to use the requested data.

Flow	**Steps**
Basic Path	1 The system checks whether the ERP ID (classifier_ item.code) in classifier "ERP" (classifier.code) exists, and the current (classifier_item.valid_from, classifier_item.valid_until). In case, if the ERP ID missing or not valid, there occurs an error 002. 2 The system queries the database of records, querying all entries for the current month that the heir (ERP) has the right to see in the absence of input parameters. ERP has the right to see the record if the ERP ID is associated with an organization whose organization.id is in the accounting_entry.id_orga field via the permission table. The current month means that the accounting_entry.created_date is within the current month. 3 In the case, if the query to the given input parameters, filters the system at the input shown: - entrys owner Code (reg. number) == organization.regnr. - accounting_ entry.created remains at the input indicated start time and end time. - Record type == classifier_ item.code (gl-cor_entriesType), where accounting_entry.type == classifier_item.id. - Entry identifier == accounting_entry_uuid (gl-cor_uniqueID). 4 The system checks the ERP permissions to use the requested data. The following condition applies to the existence of rights: permission.erp = classifier_item.id AND classifier_ item.code == ERP ID AND ((classif ier_ item.valid _from <> NULL AND classifier_ item.valid _until <> NULL AND classifier_ item.valid _from <= current date AND current date <= classifier_ item.valid _until) OR (classifier_ item.valid _from == NULL AND classifier_item.valid_until == NULL) OR (cla ssifier_ item.valid _from <> NULL AND classifier_ item.valid _until == NULL AND

	classifier_ item.valid _from <= current date)) AND at least one gl-bus_organizationIdentifier == organization.regnr AND in the record XML permission.id_orga == organization.id AND (permission.business _area == NULL OR permission.business_area == record in XML gl-bus: sourceApplication) AND permission.report _type == NULL. In case, if the absence of rights of all of the filtered accounting entries, there will be error 003 and use case ends. 5 The system returns accounting entries, which part of the ERP has rights. 6 The end.

1.1.5 IOB-UC-005 Receiving e- invoice transmission feedback

The system receives e-invoice operator feedback on the success or failure of e-invoice generation and transmission.

Input REST Json:
1. The EPP ID 1..1 (goes to HTTP headers).
2. Feedback, 1.. n.
2.1. E-invoice entry UUID, 1..1.
2.2. E-invoice in XML, which is generated by the e-invoice operator, 1..1.
2.2. E-invoice status, 1..1 (values: 'FAILED_TO_COMPILE', 'TRANSMITTED', 'TRANSMISSION_FAILED').
2.3. E- invoice generation time (timestamp with time zone), 1..1.
2.4. Comment, 0..1, case, if the sending has failed, put into element error messages.

Output REST Json:
1. Error code, 1..1. In case of success gives 204 Completed and *body is* emty.

Error codes:
000 - The errors are not generated.
001 - Technical error.
002 - Inlet shown UUID is not found.
003 - ERP ID invalid or missing.
004 - ERP-I there are no rights for at least one input entry transmission into MCDS.
00 5 - The ERP has lack of rights to the e-invoice feedback transmission.

Flow	**Steps**
Basic Path	1 The system takes the input data.

	2 The system checks whether the ERP ID is available in the classifier ERP and the case, if not there will be error 003 and the system work is interrupted. 3 If a query performing ERP has a classifier ERP attribute, which corresponds to the condition classifier_ attribute.code == 'e invoicing_o perator ' and the corresponding classifier_attribute_value.value <> 'YES' or the attribute is not, you will get error 005 and use case ends. 4 In the case, if the input shown entry UUID does not exist, there will be error 002 and no feedback entries will be stored in the database. 5 The system adds a new record in the invoice table for each feedback in the input, saving the generated invoice in the xml field invoice.invoice, in the invoice.generationTime field from the input e-invoice generation time and in the invoice_status table the status and message in the feedback record. 6 The end.

1.2 Data publishing

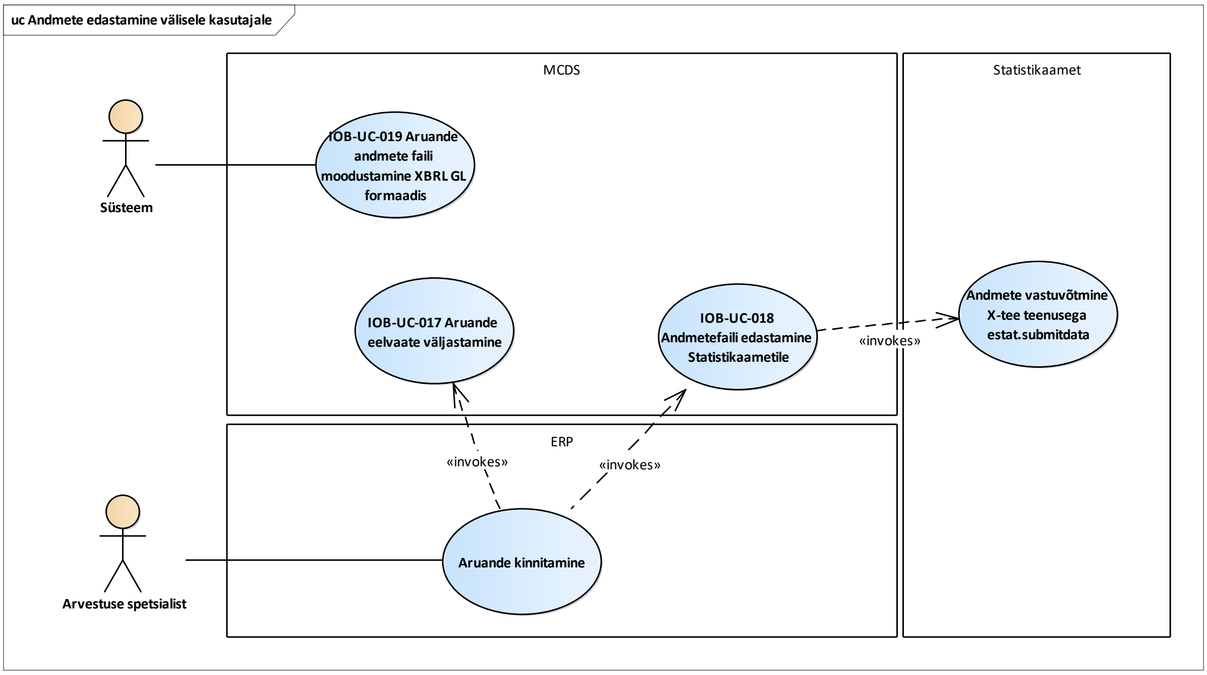

uc Andmete edastamine välisele kasutajale
MCDS
Statistikaamet
Süsteem
IOB-UC-019 Aruande andmete faili moodustamine XBRL GL formaadis
IOB-UC-017 Aruande eelvaate väljastamine
IOB-UC-018 Andmetefaili edastamine Statistikaametile
«invokes»
Andmete vastuvõtmine X-tee teenusega estat.submitdata
ERP
«invokes»
«invokes»
Aruande kinnitamine
Arvestuse spetsialist

Figure 2: Data transfer to an external user

1.2.1 IOB-UC-017 Output preview of the report
The system outputs all the reports, which correspond to the input shown in the data, and which the ERP has available rights to see.

Input REST Json:

1. ERP identifier, 1..1.
2. Organization reg. nr, 1..n.
2. In what status to output reports (possible values "GENERATED", "ERROR", "CANCELED", "SUBMITTED", NULL), 0..1.
3. Report type, values from classification "REPORT_TYPE", 1..1.
4. Report start date, 0..1.
5. Report submission deadline, 0..1.

Output REST Json:
1. Error code, 0..1.
2. XBRL GL reports, 0..n.
2.1. report ID, 1..1.
2.2. status of the report, 1..1.
2.2. report type ID, 1..1.
2.3. deadline for submission of the report, 1..1.
2.4. records of the content of the report, 1..n. The elements are issued separately from the *endpoint,* the input of which is the report ID.

Error codes:
000 - The errors are not generated.
001 - Technical error.
002 - Report type id invalid.
003 - ERP ID invalid or missing.
004 - ERP does not have rights to view the report.

Flow	**Steps**
Basic Path	1 The system verifies whether the ID from the ERP according classifier_ item.code classifier ERP is available. If not, error 003 occurs and the use case is aborted. 2 The system checks whether the ERP ID in the input corresponds to an identifier in the permission.erp column and if so, whether the permission.id_orga in the same record corresponds to the reg of the organization in the input. number in the field organization.regnr. If no matching records are found, error 004 occurs and the use case is aborted. 3 The system checks whether the input of the report on the code of the classifier "REPORT_TYPE" exists. If not, error 002 occurs and the usage history stops working. 4 The system checks whether the ERP has right to view reports. It means that there is a query performing ERP corresponding permission entry, which REPORT_TYPE == input to the report type. In case, if does not, there will be error 004 and use case ends. 5 The system requests from the table report, based on the data shown in the input, report entries that meet the following conditions: - latest report_status.status == status.

Co-financed by the European Union
Connecting Europe Facility

	- report.id_orga == organization reg. no.organization.id - classifier_ item.code == report type, classifier.code == "REPORT_TYPE". - start date of the report <= report.deadline <= end date of the report. 6 The system outputs a list of found reports with the following columns: - report ID, report.uuid. - report status, latest report_status.status. - organization reg.nr., organization.regnr. - report type code, classifier_item.code. - report submission deadline, report.deadline. - report content, report.content. 7 The end.

1.2.2 IOB-UC-018 Transmission of data file to Statistics Estonia

Data submitted to the Statistical Office through the X-Road service.

Since the data sent to the SA requires the sender x-path sub-system information (subsystem name and, to the security server corresponding permissions to do. SA has data for receiving X-road service called estat.submitdata. This is the so-called mailbox or sent to the XBRL GL with the corresponding The xml file is attached as an attachment, the WSDL is available from RIHA (https://www.riha.ee/Infos%C3%BCystemid/Vaata/estat) but you can also ask SA, which has also created a client application that can data estat.submitdata to the service, it can be used as a sample if necessary (https://koodivaramu.eesti.ee/statistikaamet/xgate-client).
Input REST Json:
1. ERP ID, 1..1.
2. XBRL GL XML format report ID, 1..1.
3. The method, as a report subject implement, 1..1 (values of the "submit", and "CANCEL"). Developed two endpoints, of which one shall submit a report, and the second cancels the report.

Output REST Json:
1. Error code, 1..1.
2. Error text, 0..1.

Error codes:
000 - The errors are not generated. Report submitted successfully.
001 - Technical error.
002 - Report not found.

003 - ERP ID invalid or missing.
004 - ERP-l There are no rights to report view and transmission.
005 - The report could not be transmitted over X-Road.

Flow	**Steps**
Basic Path	1 The system verifies whether the ID from the ERP according classifier_ item.code classifier ERP is available. If not, error 003 occurs and the usage history is aborted. 2 The system checks whether the ERP ID in the input corresponds to an identifier in the permission.erp column and if so, whether the permission.id_orga in the same record corresponds to the reg, number of the organization in the input in the field organization.regnr. If no matching records are found, error 004 occurs and the use case is aborted. 3 The system checks if the report ids in the input can be found in the database in the report.uuid column. If not found, error 002 occurs and the use case is aborted. 4 If the method value "CANCEL" is entered in the input, the system marks the status of the report as canceled. The action moves to step 9. 5 If the value of the input method is not "SUBMIT", the operation proceeds to step 9. 6 The system generates an estat.submitdata message for the X-Road service by appending the XBRL GL xml (report.content) as input. 7 The system submits the message to Statistics Estonia via X-Road. 8 The system marks the report as sent in the database (report.status = "FORWARDED TO STATISTICS") if the report was sent successfully and if the transmission failed, error 005 will occur. The system adds an error text element to the output from the X-Road service. 9 The system adds the event that occurred into the event_log table. 10 The end.

1.2.3 IOB-UC-019 Creating report data files in XBRL GL format

The system runs this use case daily at the time indicated in the system parameters, generating all reports that have reached the generation time.

Input - None.

Output:
1. Error codes, 1..1.

Error codes:
000 - No errors occurred.
001 - Technical error.
002 - An element specified by a classifier has a value that does not belong to the elements of the classifier.
003 - Report XML is invalid.

Report attributes in "REPORT_TYPE":

1. xbrli: unit.id = “EUR” - the unit of the report.
2. gl- cor: entriesType - report type. The default is “Other”.
3. gl- cor: language - Report language according to ISO 639. Two-letter code.
4. gl- bus: creator - report generator. For example. “Information MCDS”
5. gl- cor: entriesComment - report comment.
6. gl- bus: sourceApplication - information system name, where the report originated. For example. "Information MCDS".
7. gl- muc: defaultCurrency - The currency used in the report.
8. gl- cor: accountSubType - report used the classification code of the reporting taxonomy based. Used classifications can be several from the required nu mber of the source of the classification attribute. See https://www.stat.ee/sites/default/files/2020-08/palk_toojoud_taksonoomia_20200131.xlsx.
9. gl- cor: sourceJournalID - reference to the source of the data. A report can have multiple source references. For each, the classifier has a value with the same code (classifier_ attribute.code).
10. REP_PERIOD - shows the report period (values “continuous”, “day”, “week”, “month”, “quarter”, “half_year”, “year”).
11. SUBMISSION_DAY - shows a report for submission day period to the next period, or a multi-fifth day after the period of the end of the report submitted.
12. REP_GEN_LAG - The time the report was generated in days after the start of the report period. Indicates the number of days after the reporting period, the beginning of the report ready to be generated.
13. REP_PERIOD_START_DAY - day number, which begins with the reporting period. For example, a value of 1 if the period begins on January 1.

Elements of the generated report, the values of which must be formed by the system:

1. XBRL: identifier.scheme - a report by the registry code to the publisher, the database organization.reg_authority. The default is "http://www.ariregister.rik.ee".
2. xbrli: identifier - registry code of the reporter, in the database organization.regnr.
3. xbrli: instant - date the report was generated. (Report period field).
4. gl-cor: uniqueID - unique identifier of the report. Reg. no. + report date + report time, separated by a hyphen.
5. gl- cor: creationDate - date the report was created.

Flow	**Steps**

Co-financed by the European Union
Connecting Europe Facility

Basic Path	1 The system starts generating reports on a daily basis at the time indicated in the system parameters. 2 The system takes Classifier "REPORT_TYPE" all valid classificator_item records and checks the company reports has been already generated. 3 The system generates a report on the company's accounting entries, if: - arrived the report generation time in accordance to the classification "REPORT_TYPE" item dynamic attribute (table classifier_attribute) code "GEN_PERIOD” start is arrived, given the attribute “REP_GEN_LAG" value. "REP_GEN_LAG" indicates how many days after the start of the period, a report for the previous period must be generated. - The company has a reporting obligation, ie in the organization_report_type table there is an entry that links the organization and the report type and the entry deactivated field has the value FALSE. - the company has no a generated report or the report has the status "CANCELED" or "ERROR" and new records have been added to the system that are part of the report being created. 3 If the observed company does not need to generate a report, the system checks the need to generate a report for the next company until the observed companies run out. Once all companies have been reviewed, the use case ends. 4 The system starts report creation and generates a report header to the following elements: **<? xml version = "1.0" encoding = "UTF-8"?>** **< xbrli: xbrl xmlns: xbrli = " http://www.xbrl.org/2003/instance " xmlns: xlink = " http://www.w3.org/1999/xlink " xmlns: xbrll = " http: / /www.xbrl.org/2003/linkbase " xsi: schemaLocation =" http://www.xbrl.org/int/gl/plt/2015-03-25../ plt / case-cbmut / gl-plt- all-2015-03-25.xsd " xmlns: xsi =" http://www.w3.org/2001/XMLSchema-instance " xmlns: gl-usk =" http://www.xbrl.org/int/ gl / usk / 2015-03-25 " xmlns: gl-cor =" http://www.xbrl.org/int/gl/cor/2015-03-25 " xmlns: iso639 =" http: // www. xbrl.org/2005/iso639 " xmlns: iso4217 =" http://www.xbrl.org/2003/iso4217 " xmlns: gl-plt =" http://www.xbrl.org/int/gl/plt/ 2015-03-25 " xmlns: gl- bus =" http://www.xbrl.org/int/gl/bus/2015-03-25 " xmlns: gl-muc =" http: //www.xbrl. org / int / gl / muc / 2015-03-25 ">** **< xbrll: schemaRef xlink: href = "../ plt /case-cbmut/gl-plt-all-2015-03-25.xsd " xlink: arcrole = " http://www.w3.org/1999/xlink / properties / linkbase " xlink: type =" simple "/>** **< xbrli: context id = " now ">** **< xbrli: entity >**

```
<! - Reporting organization reg.nr, in
the database organization.regnr. scheme attribute reg. no. from the
publisher database organization.reg_authority ->

< xbrli: identifier scheme =
" http://www.ariregister.rik.ee "> 111111111 </ xbrli: identifier >

</ xbrli: entity >

< xbrli: period >

<! - Reporting period. In this case, the date of generation. ->

< xbrli: instant > 2020-11-09 </ xbrli: instant >

</ xbrli: period >

</ xbrli: context >

<! - Report unit of measure from report attributes. ->

< xbrli: unit id = " eur ">

< xbrli: measure > iso4217: EUR </ xbrli: measure >

</ xbrli: unit >

< xbrli: unit id = " NotUsed ">

< xbrli: measure > pure </ xbrli: measure >

</ xbrli: unit >

< gl- cor: accountingEntries >

< gl-cor: documentInfo >

<! - Report Generation Date ->

< gl- cor: entriesType contextRef = " now "> other </ gl-cor:
entriesType >

<! - Unique report id consisting
of reporting agency regnr + report date + time ->

< gl- cor: uniqueID contextRef = " now "> 10002456-2017-05-31-09-36-
51 </ gl-cor: uniqueID >

<! - Report language from report attributes. It is independent of
the rapporteurs and the location from the report. ->

< gl- cor: language contextRef = " now "> iso639: et </ gl-cor:
language >

<! - The report generation date. ->
```

```
< gl- cor: creationDate contextRef = " now "> 2017-06-05 </ gl-cor:
creationDate >

<! - Report generator id or MCDS name, e.g. TietoEVRY -MCDS, if
the MCDS service providers are TietoEVRY. Data on report attributes. -
>

< gl- bus: creator contextRef = " now "> TietoEVRY -MCDS </ gl-bus:
creator >

<! - Report comment on report attributes. ->

< gl- cor: entriesComment contextRef =
" now "> Empty Comment </ gl-cor: entriesComment >

<! - Reporting dates for the range, which is calculated from the
previous report to the date of the final + 1 day until the
current period until the end. The length of the period is shown in
the report attributes. ->

< gl- cor: periodCoveredStart contextRef = " now "> 2017-05-01 </ gl-
cor: periodCoveredStart >

< gl- cor: periodCoveredEnd contextRef = " now "> 2017-05-31 </ gl-
cor: periodCoveredEnd >

<! – ERP system or information system name. In this case Tieto-
MCDS. The information is located in the report's attributes. ->

< gl- bus: sourceApplication contextRef = " now "> Data-MCDS </ gl-
bus: sourceApplication >

<! - Currency used. Located in report properties. ->

< gl- muc: defaultCurrency contextRef = " now "> iso4217: eur </ gl-
muc: defaultCurrency >

</ gl- cor: documentInfo >

< gl- cor: entityInformation >

< gl- bus: organizationIdentifiers >

<! - Registry code of the database reporter (entries owner), in
the database organization.regnr. ->

< gl- bus: organizationIdentifier contextRef = " now "> 10002456 </ gl-
bus: organizationIdentifier >

<! - Registry code of the database reporter (record owner) code issuer,
in the database organization.reg_authority. ->

< gl- bus: organizationDescription contextRef = " now "> AA3 Näidis
OÜ </ gl-bus: organizationDescription >

</ gl- bus: organizationIdentifiers >
```

Co-financed by the European Union
Connecting Europe Facility

	</ gl- cor: entityInformation > 5 The system finds all company records that meet the following conditions: **- one or more gl- cor: entryDetail / gl-cor: postingDate is within the report period. Period begins on REP_PERIOD_START_DAY and ends on REP_PERIOD_START_DAY-1 in the current period.** - entry gl- cor: entryHeader block the gl-cor: sourceJournalID value is the same, that at least one of the generated report to the same code attribute, such as the labor costs of a report event of both "jc"; - entries all gl- cor: accountSubType values included in the report parameters indicated in the GL-cor: accountSubType code attribute values included or entry uses only values from the taxonomy of the report. 6 The system checks to see if all gl-cor: accountSubID values in the records belong to the classification shown in the l-cor: accountSubType element. If any of the gl-cor: accountSubID values do not belong to the classification shown in the gl-cor: accountSubType element, error 002 occurs, the system adds the control result to the report table, and the usage history continues. If there is no corresponding classification in the MCDS database, the system does not check this classification. 7 The system adds the found entryHeader blocks into the report XML. 8 The system adds a new record to the report table in the database, indicating the code of the generated report from the classifier "REPORT_TYPE" in the report_type field. In the report.deadline field, the system enters the period indicated by the "REPORT_TYPE" item (attribute "REP_PERIOD") and the date of the period (attribute "SUBMISSION_DAY"). report.content. 9 The system adds a new entry to the report_status table with the appropriate status, either "GENERATED" if the data was correct or "ERROR" if there were errors in the use of the classifiers. 10 The system adds table report_entry items that bind the report going kirjendid recorded statement. 11 The system will validate the report XML validity. 12 The system moves to generate a new report, as are more companies, which need the report to generate. 13 The end.

1.3 Administrator view

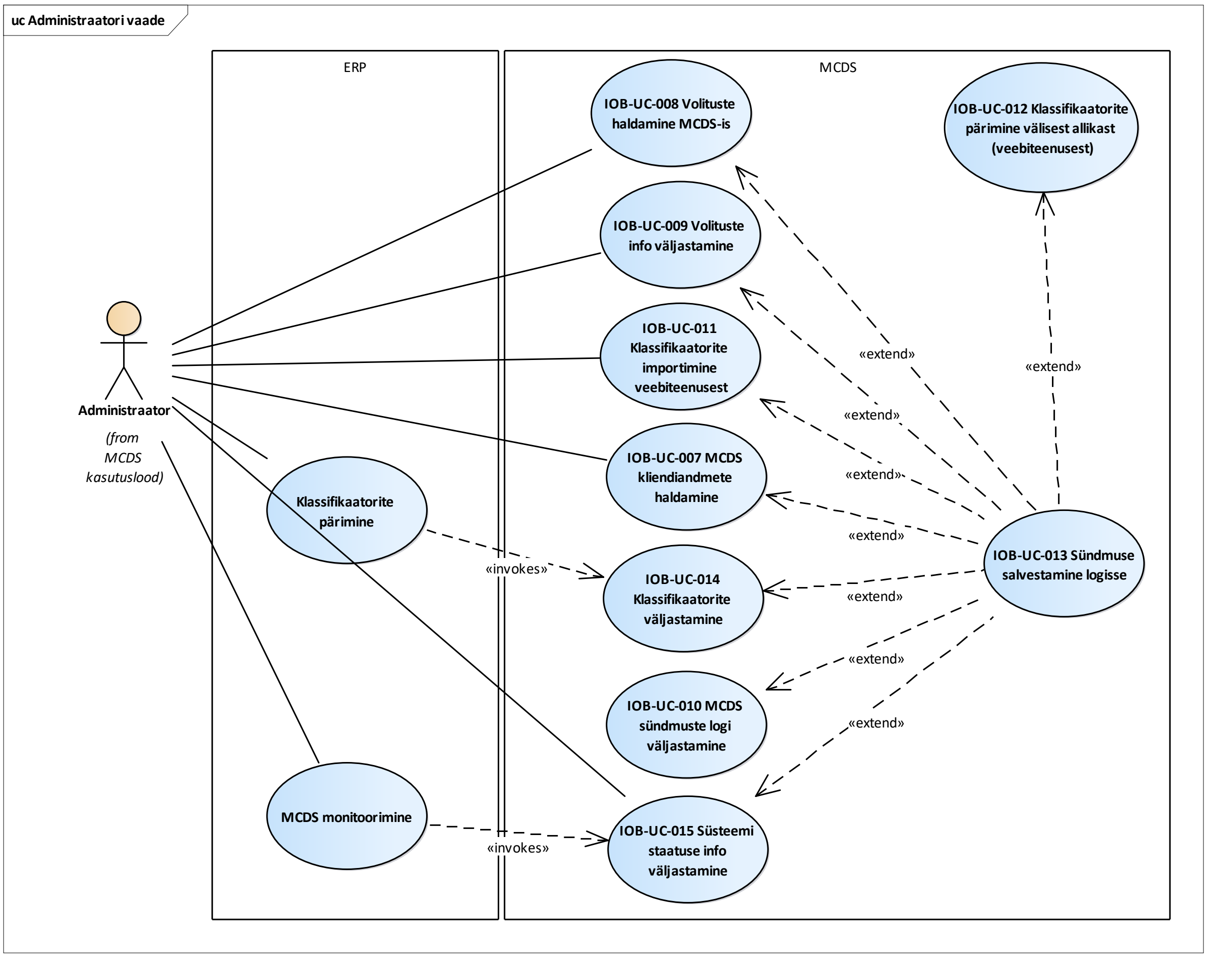

uc Administraatori vaade
ERP
MCDS
Administraator
(from MCDS kasutuslood)
IOB-UC-008 Volituste haldamine MCDS-is
IOB-UC-009 Volituste info väljastamine
IOB-UC-011 Klassifikaatorite importimine veebiteenusest
IOB-UC-007 MCDS kliendiandmete haldamine
IOB-UC-014 Klassifikaatorite väljastamine
IOB-UC-010 MCDS sündmuste logi väljastamine
IOB-UC-015 Süsteemi staatuse info väljastamine
IOB-UC-012 Klassifikaatorite pärimine välisest allikast (veebiteenusest)
IOB-UC-013 Sündmuse salvestamine logisse
Klassifikaatorite pärimine
MCDS monitoorimine
«invokes»
«invokes»
«extend»
«extend»
«extend»
«extend»
«extend»
«extend»
«extend»
«extend»
«extend»

Figure 3: Administrator view

1.3.1 IOB-UC-007 MCDS customer data management

MCDS must include a web service for handling customer data. MCDS customers are divided into two:
1) ERP systems that create records in a database. One ERP system can be linked to multiple data owners or companies whose accounting is in the ERP.
2) the data owners or companies, whose accounting information the system contains.
3) Consumers of reports. These are public authorities to which data owners are required to report, either in the form of micro data or in the form of aggregated data.

ERP systems, data is managed in MCDS as a classification, for which the code is "ERP". Data own personal data, and the recipients of the report data is managed through the use case.

Input, REST Json:
1. Data of the client who is a legal person, 1..n. To be saved to table "organization"
1.1. Person reg. number, 1..1.
1.2. Name, 1..1.
1.3. Address, 1..1.
1.4. Contact details, 1..1.
1.5. For which reports is the company accountable, 0..n, values from the classification "REPORT_TYPE". Saved as a relationship in the table organization_report_type.

Co-financed by the European Union
Connecting Europe Facility

1.6. Deactivate, values "true" and "false".

Output:
1. Error code, 1..n.

Error codes:
HTTP 500, code 001, info: "Technical error".
HTTP 400, code: 003, info: "Report type {reportTypeCode} not found"
004, info: "ERP ID invalid or missing."
005, information: "ERP does not have the right to manage customer data."

Flow	**Steps**
Basic Path	1 The system checks whether in the input shown ERP ID exists in the classifier "ERP", and the case, when does not get the error 004, and use ends. 2 The system checks whether the ERP ID shown in the input is related to the 'manager' role in the system parameters, and if not, error 005 occurs and the use case ends. 4 In the case, if the client does not exist in the database, adds the system a new record. 5 If the same reg. number already exists in the database, the system updates the original record based on the input data by activating or deactivating the organization by adding the current time to the field deactivated. 6 With an update, the contents of the organization_report_type table are aligned with the list of reports shown in the input. Associated and non-existent relationships are added by the system as new records. There are no inputs, but the relationships that exist in the organization_report_type table are deactivated by typing the current time and date of the query in the organization_report_type.deactivated field. 7 If the report shown in the input does not exist in the classification "REPORT_TYPE", error 003 occurs. When checking the existence of the code, the system does not check the classification data validity of the classification item. 8 The end.

1.3.2 IOB-UC-008 Permission management in MCDS

MCDS includes a permission management service, through which it is possible to deliver and retrieve the permissions of an ERP company to MCDS.
Every ERP has the right to access the data of specific companies. This access is stored as permission in the MCDS database.

Co-financed by the European Union
Connecting Europe Facility

Company data (records) are divided into areas (eg sales invoices, purchase invoices, sales invoices2, etc.). Access to a specific ERP is also restricted by these types. Permission management should be a REST service that can be used programmatically or from the command line. The user of the service must be authenticated. The name and password of the user who has permission to change the permissions, are stored in the system settings.

Input REST Json:
1. Permission, 1.. n.
1.1. ERP code, 1..1.
1.2. Company reg. number for which the authorization is granted, 0..1.
1.3. Activity for which the authorization is established, e.g. "sale", "purchase", etc., 0..1.
1.4. The report type ID with a view given the mandate, 0..1 permission.report_type into store classifier_item.id, which corresponds to the report type ID-s shown in the code.
1.5. Deactivate, values TRUE and FALSE, 1..1. The current time of the query goes to the Timestamp field.

Output:
1. Error code, 1..1.

Error codes:
000 - The errors are not generated.
001 - Technical error.
002 - Deactivable permission not found.
003 - Report not found.
004 - Authorization client not found.
005 - ERP code invalid or missing.
006 - ERP does not have the right to manage permissions.

Flow	Steps
Basic Path	1 The system checks whether the ERP ID is in the classifier "ERP" existing and valid, and, if not there will be error 005 and use case ends. 2 The system checks whether the ERP ID shown in the input is related to the 'manager' role in the system parameters, and if not, error 006 occurs and the use case ends. 3 The system gets from input permission information Json file and adds the permissions contained in it to the permission table according to the following steps. Only permissions that Deactivate == FALSE and do not previously exist in the permission table are added. Those permissions that are deactivated in the database are activated by the system if the Deactivate == FALSE in input. 4 The system deactivates all permissions, which is at the input shown as deactivated. The system uses the combination permission.erp + permission.regnr + permission.business_area +

Co-financed by the European Union
Connecting Europe Facility

	permission.report_type as the key. In case, if deactivable permission is not found, there will be error 002. 5 If the report code shown in the input is not found among the elements of the "REPORT_TYPE" classification, error 003 occurs. The validity and versions of the classification elements are not taken into account in the check, as the authorization may be changed for expired reports. 6 The system searches in the input given customer reg. numbers from field regnr of the organization table. If the company is not found, error 004 occurs. 7 The system logs all deactivations and additions, and those related errors and warnings in the table event_log. 8 The system returns an error code that occurred during operation. The use case ends when the first error occurs. 9 The end.

1.3.3 IOB-UC-009 Output of authorization information

Use case output of permission information from MCDS. The service user must be authenticated and have the role of "manager".

Input, REST Json:
1. ERP ID, 0..1.
2. Customer reg. No., 0..1.
3. Field of activity, 0..1.
4. Type of report, 0..1.

Output, REST Json:
1. Power of attorney, 0..n.
1.1. ERP code, 1..1.
1.2. Customer regnr., 0..1.
1.3. Activity ID, 0..1.
1.4. Report type code, 0..1.
1.5. Permission is active.
2. Error code, 0..n.

Error codes:
000 - No errors occurred.
001 - Technical error.
003 - ERP code to mandate the data does not exist.
004 - Client regnr does not exist in the permissions.
005 - ERP code invalid or missing.
006 - ERP does not have the right to manage authority.

Flow	**Steps**

Basic Path	1 The system checks whether the ERP ID exists in classifier "ERP" and is valid, and, if not there will be error 005 and the use case ends. 2 The system checks whether the ERP code in the input is related to the "manager" role in the system parameters, and if not, error 006 occurs and the use case ends. 3 The system filters the contents of the permission table according to the fields shown in the input: - permission.erp. - permission.regnr. - permission.business _area. - permission.report _type. 4 If the classifier_item.id does not exist in the permission table according to the ERP code shown in the input, error 003 occurs. 5 If the id_orga corresponding to the client regnr does not exist in the permission table, error 004 occurs. 6 The system outputs a Json file with all found permission entries. 7 The system returns error codes. 8 The end.

1.3.4 IOB-UC-010 MCDS event log output

The recorded event log entries output of MCDS.

Input, REST Json:
1. Start time of the event, 1..1.
2. End time of the event, 1..1.
3. Is the action succeeded, 0..1, the possible values are TRUE and FALSE.
4. Type of event, 0..n.

Output, Json, all events found for the condition:
1. Events, 0.. n.
1.1. Event ID, 1..1.
1.2. Type of event, 1..1.
1.3. ERP code, 1..1.
1.4. Was the event a success?, 1..1.
1.5. Event at the time, 1..1.
1.6. Notice, 1..1.
2. Error code, 1..1.

Error codes:
000 - The errors are not generated.

Co-financed by the European Union
Connecting Europe Facility

001 - System error.
002 - No events matching the input criteria were found.
005 - ERP code invalid or missing.
006 - ERP does not have the right to use the service.

Flow	**Steps**
Basic Path	1 The system checks whether the ERP ID in classifier "ERP" is existing and valid, and, if not there will be error 005 and use case ends. 2 The system checks whether the ERP ID in the input is related to the role "manager", and if not, there will be error 006 and use case ends. 3 The system takes a data table event_log all log entries, which correspond to the input parameters, and outputs the query submitter JSON format. 4 In the case, if the items are not found, there will be error 002. 5 The end.

1.3.5 IOB-UC-012 Querying classifications from an external source (web service)

The classification manager (e.g. Statistics Estonia) has web services for sharing classifications. MCDS is set to import automatically from Statistics Estonia. The import is started by the MCDS automatically with the set period.

Classifier APIs:
1) Classifications of Statistics Estonia:
https://estat.stat.ee/codelists/availablecodelists/.
2) Query for one classification of Statistics Estonia:
https://estat.stat.ee/codelists/codelist/RTK2T2013ap/.
3) Elements of the classification of Statistics Estonia from a certain date:
https://estat.stat.ee/codelists/codelist/RTK2T2013ap/2020-01-01.

Input REST Json:
1. Classification code.
2. Date, which at the desired data.

Output REST Json:
1. Error code, 0..1.

Error codes:
000 - The errors are not occurred.
001 - Technical error.

To save data received from an external service to the MCDS database:

- CodeListCode – to store to the field "classifier.code".
- CodeListName - to store to the field "classifier.name".

- Code - to store to the field "classifier_ item.code".
- ParentCode - to store to the field "classifier_item.id_upper_classifier_item".
- Name_EN - to store to the field "classifier_item_text.name" and as an additional identifier classifier_item_text.language = "ET".
- Name_EN - to store to the field "classifier_item_text.name" and as an additional attribute classifier_item_text.language = "EN".
- ValidFromDate - to store to the field "classifier _item.valid_from.".
- ValidUntilDate - to store to the field "classifier _item.valid_until ".

Flow	**Steps**
Basic Path	1 The system automatically starts importing classifiers via the configured classifier web services after a period set in the system. 2 The system requests a list of classifications from an external source. 3 The system will import all classifications, classification elements and rewrites the existing same name classifications (the classifier is not versioned in the process). 4 The end.

1.3.6 IOB-UC-013 Recording events to the log

The system stores in the system log all events that are reflected in the event type classification "SYN_TYPE". The event logging request must be implemented in all use cases, that generate the events shown in the classifier.

Input, REST Json:
1. The event, 1..1:
1.1. Event type among the elements of the classification "EVENT_TYPE", 1..1.
1.2. ERP ID, 1..1.
1.3. Event-related record IDs, 1..1, format "< table >: < record ID>".
1.4. Whether the event was successful, 1..1, the values TRUE and FALSE.
1.5. Error messages, 1..1.

Output:
1. Error code, 1.. n.

Error codes:
000 - The errors are not generated.
001 - Technical error.
002 - Unknown event type.
003 - Unknown ERP ID.

Flow	**Steps**

Basic Path	1 The system checks the compliance of the event type and ERP ID (classifier_item.code) in the input with the classifications "EVENT_TYPE" and "ERP" also taking into account the validity period of the classification elements (valid_from, valid_until). The system generates error 002 ("SYN") or 003 ("ERP") if the value in the input is not included in the classifier. 2 The system adds an event on the table event_log: - Event ID - the system generates the next ID; - event_ log.created = current time. - event_log.event_type = event type from input; - event_log.erp = ERP ID from input. - event_log.related_records = ID of the records related to the event from the input. - event_log.erp = classifier_item.id corresponding to the ERP ID. If the event is initiated by a system, the ERP ID must contain the string "SYS". - event_log.success = event success (TRUE or FALSE). - event_log.message = error messages from input. 3 The system returns error codes. 4 The end.

1.3.7 IOB-UC-014 Output of classifications

The output of classifications is an MCDS service that allows the service user to query all valid elements of a classification. When using the service, ERP does not have to look for classifiers from different sources.
The service input is the classifier code and the classifier version. If the version of the input cannot be given, returns the service classification of all versions and those associated with the elements.

Input REST Json:
1. ERP ID, 1..1.
2. Classification code, 0..1.
3. Classification version, 0..1.

Output REST Json:
1. Classifier, 0..n:
1.1. Code, 1..1 - Classification code or abbreviation.
1.2. Name, 1..1 - Name of the classifier.
1.3. Description, 0..1 - Description of the classifier.

1.4. ownerRegNr, 0..1 - Registration number of the authority managing the classification.
1.5. version, 0.. n - Classification version:
1.5.1. code, 1..1 - Classification version code.
1.5.2. name, 0.. 1 - Name of the classification version.
1.5.3. description, 0.. 1 - Description of the classifier version.
1.5.4. select, 0.. 1 - Indication whether the version of
the classification is currently valid.
1.5.5. validFrom, 0.. 1 - Start of validity of the classification version.
1.5.6. validUntil, 0.. 1 - Classification version validity end date.
1.5.6.1. Items, 0..n - Classification element:
1.5.6.1.1. code, 1..1 - Classification element code.
1.5.6.1.2. name, 0..1 - Name of the classifier element.
1.5.6.1.3. validFrom, 0..1 - Start of validity of the classification element.
1.5.6.1.4. validUntil, 0..1 - Classification element expire date.
1.5.6.1.5. seqNr, 0..1 - Sequence number of the classifier element.
1.5.6.1.6. upperClassifierItemCode, 0..1 - Code of the classifier element to which this element belongs.
1.5.6.1.7. classifierAttributes, 0..n - Classifier identifier
1.5.6.1.7.1. code, 1..1 - Code name indicating the additional identifier.
1.5.6.1.7.2. name, 0.. 1 – Additional attribute designation.
1.5.6.1.7.3. value, one.. 1 – Additional attribute value.
1.5.6.1.7.4. format, 1..1 - Additional attribute format.
1.5.6.1.8. links, 0.. n - Relation of a classifier element to another element:
1.5.6.1.8.1. linkedClassifierCode, 1..1 - Linked classification code.
1.5.6.1.8.2. linkedClassifierItemCode, 1..1 - Linked classifier element code.
1.5.6.1.8.3. linkType, 0.. 1 - Link type.
1.5.6.1.8.4. Description, 0.. 1 - Binding nature description.
2. Error code, 0..n.

Error codes:
000 - No errors occurred.
001 - Technical error.
002 - Input is not valid.
005 - ERP code invalid or missing.
006 - ERP does not have the right to use the service.

Flow	**Steps**
Basic Path	1 The system checks whether the ERP ID is in classifier "ERP" existing and valid, and, if not there will be error 005 and use case ends. 2 The system receives the request from the ERP system and checks the existence of the input data in the database. 3 If no classification code is specified in the input, the system output a list of classifications (columns: classifier.code, classifier.name, classifier.description, organization.name). 4 If the classification corresponding to the input is found in the database, the data of the classification corresponding to the input

	and related elements are transmitted to the ERP system. If the corresponding classification is not found, error 002 occures. 5 The end.

1.3.8 IOB-UC-015 System status information output

In this use case, the system status information is issued.

Input:
1. Start date of events, 1..1
2. End date of events, 1..1.

Output:
1. Error code, 1..1.
2. System event statistics, 0..1.
2.1. Number of successful events, 1..1.
2.2. Number of failed events, 1..1.
2.3. Number of systemic errors, 1..1.
3. Current status of the system, 0..1.

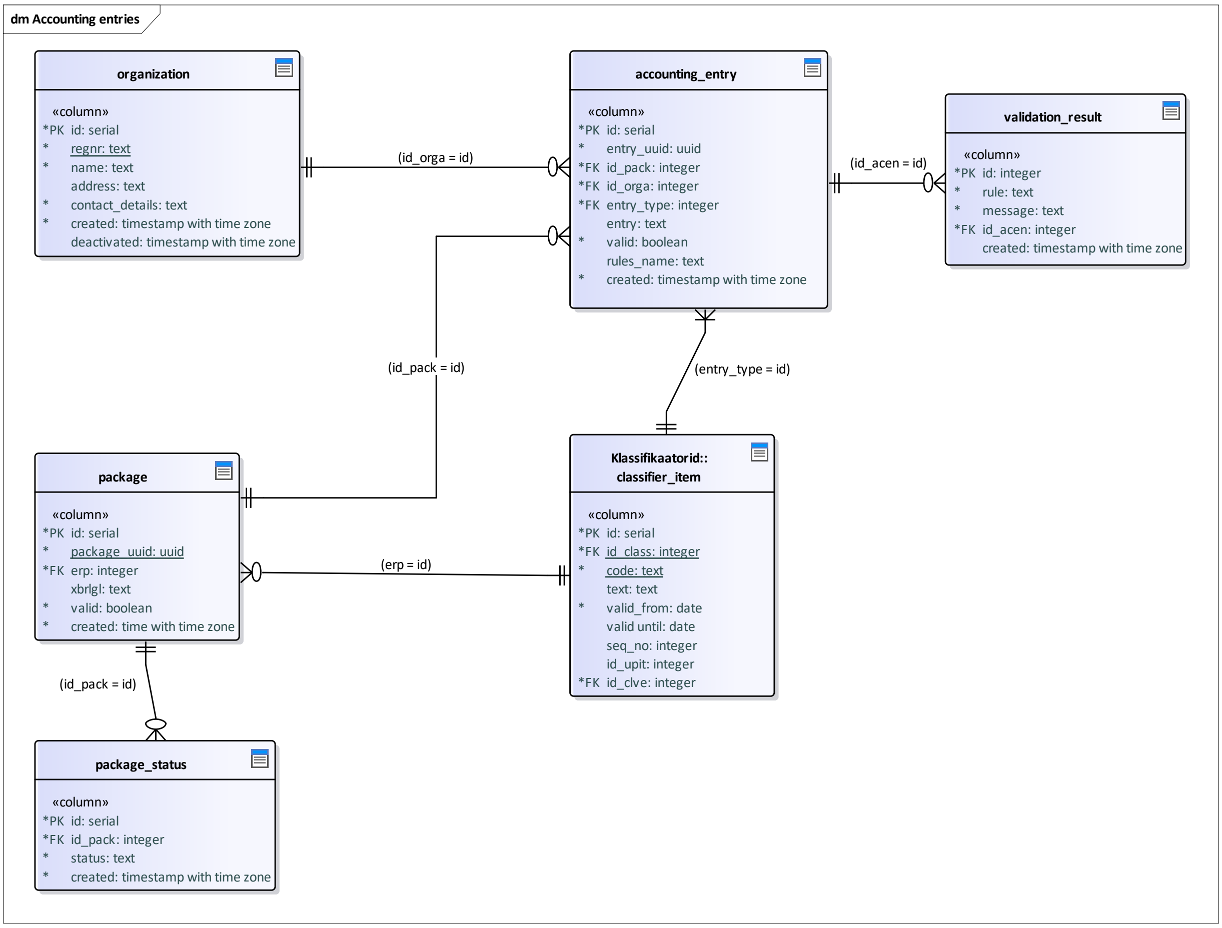
Error codes:
000 - The errors are not generated.
001 - Technical error.
002 - Events not found.
005 - ERP code invalid or missing.
006 - ERP does not have the right to use the service.

Flow	**Steps**
Basic Path	1 The system checks whether the ERP ID is in classifier "ERP" existing and valid, and, if not there will be error 005 and use case ends. 2 The system checks whether the ERP ID is related to the role of "manager", and if not, there will be error 006 and use case ends. 3 The system compiles statistics on events that occurred in the given date range based on the information in the event_log table: - Number of successful events = number of entries, event_log.success is TRUE. - Number of failed events = number of entries, event_log.success is FALSE. - Number of system errors = number of entries where event_log.event_type is "SYSTEM_ERROR". - Current system status = status string from the file "MCDS status.txt".

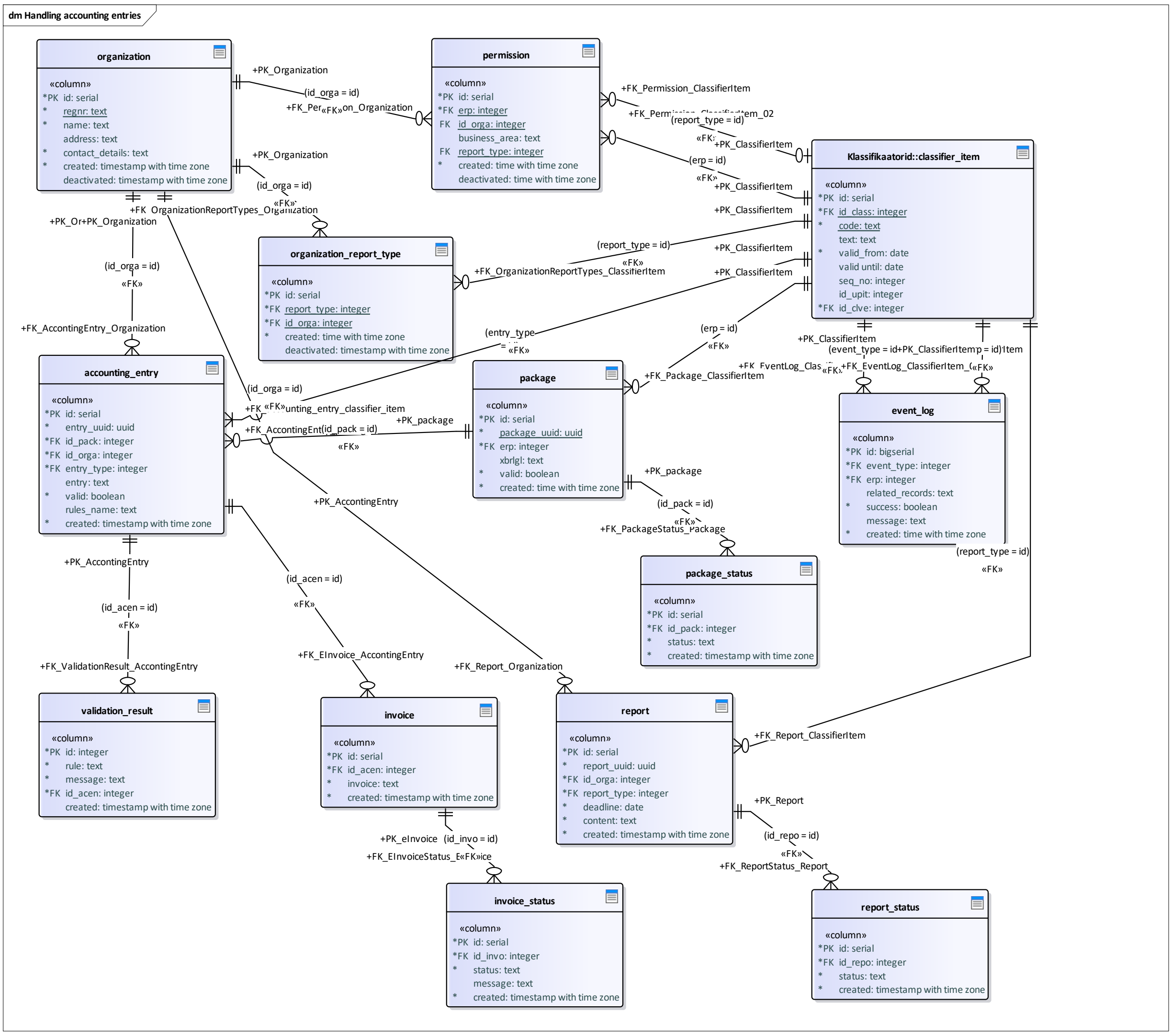
	4 The system returns the compiled statistics or error code 002 if no events were found. 5 The end.

2. MCDS Data Model

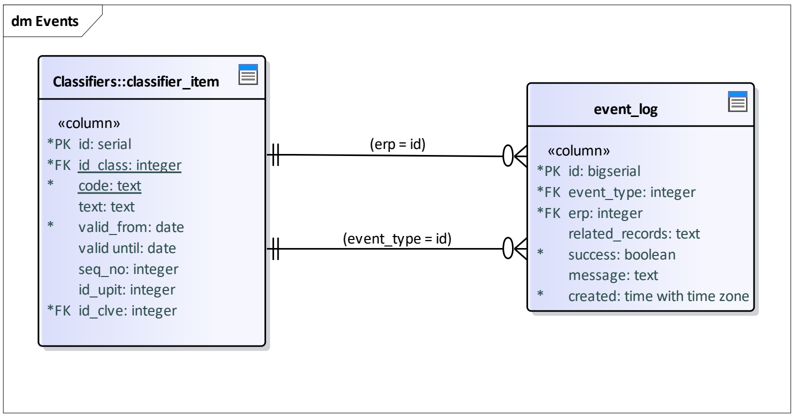
2.1. Accounting entries



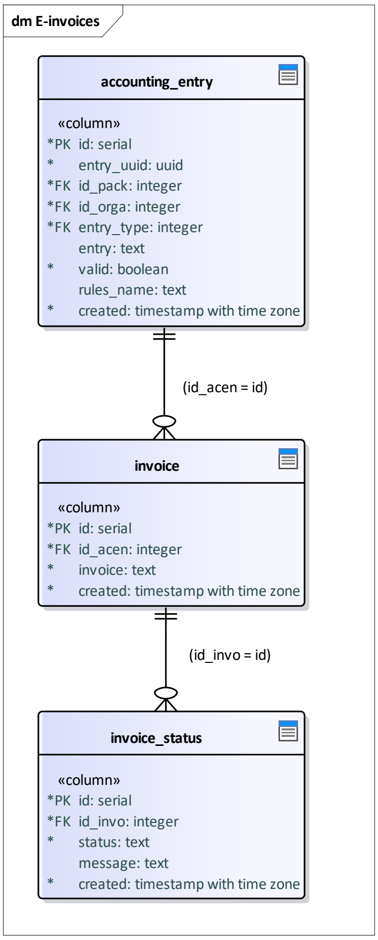
2.2. Handling accounting entries



2.3. Event log

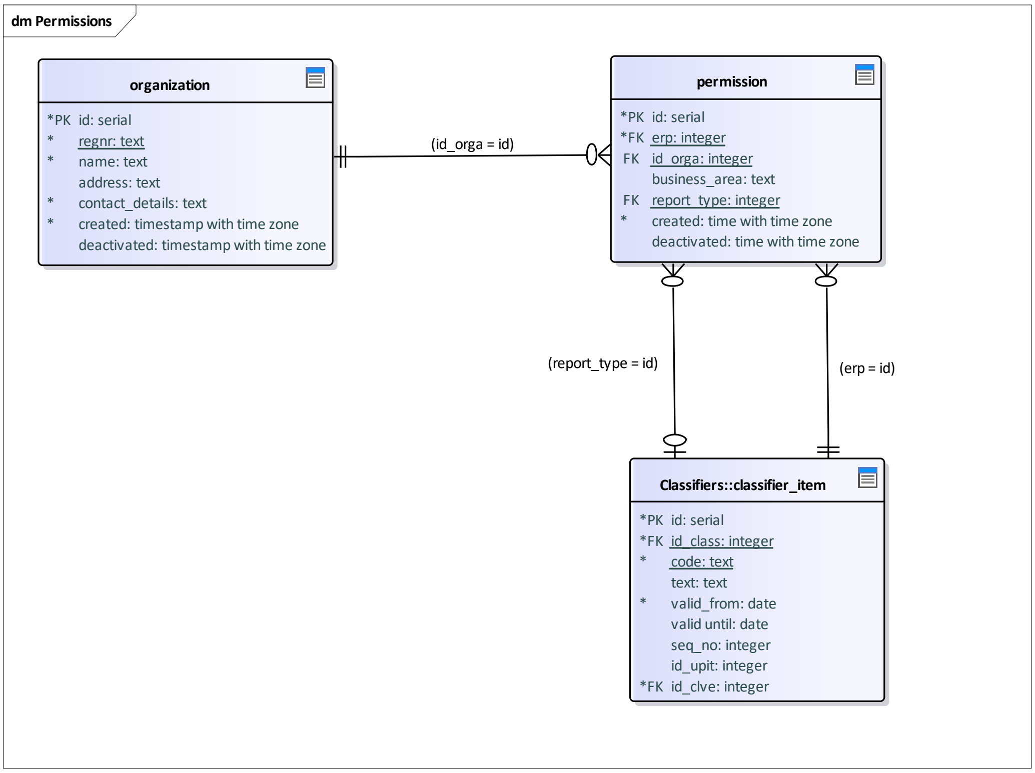


2.4. E-invoices

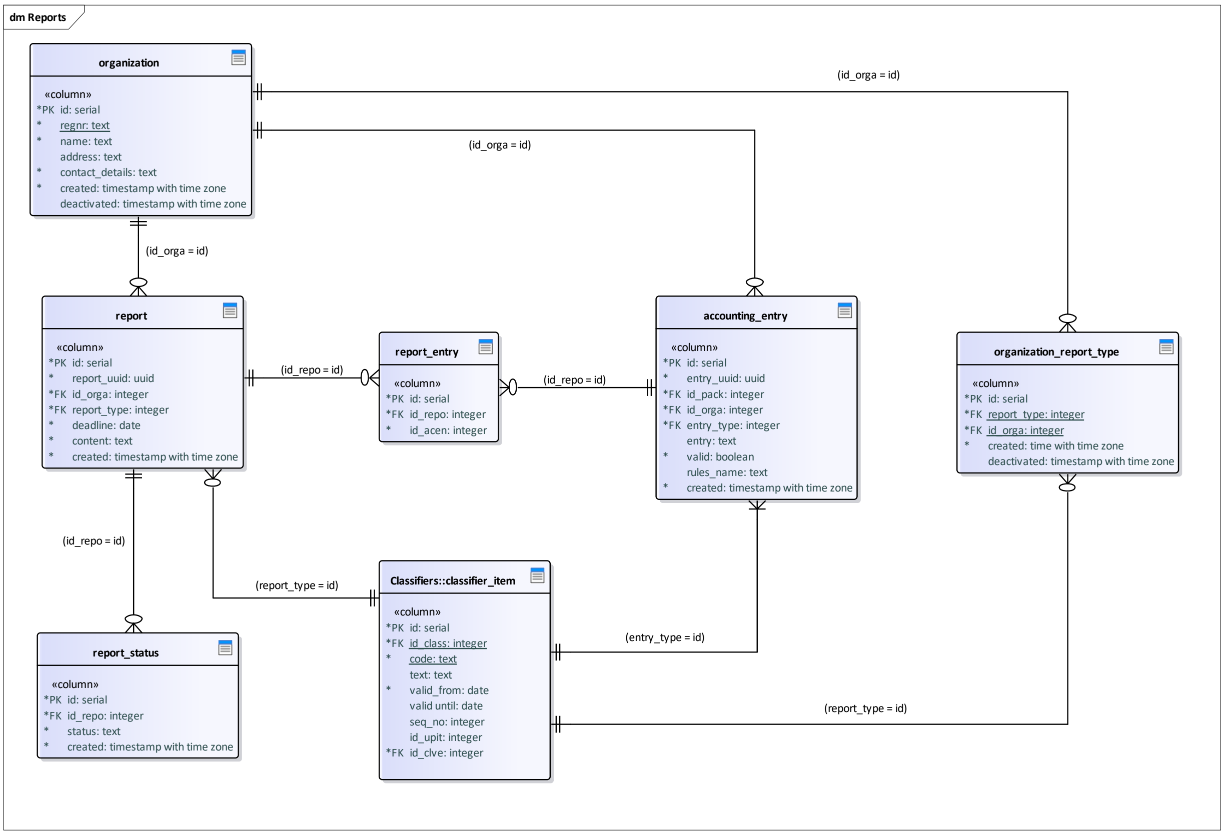


Co-financed by the European Union
Connecting Europe Facility

2.5. Permissions



2.6. Reports



Annex I: XBRL GL accounting entry data standard - final version

XBRL GL accounting entry data standard - final version.docx

Annex II: The mapping between EN UBL 2.1 e-invoice and accounting entry data standard – final version

The mapping between EN UBL 2.1 e-invoice and XBRL GL accounting entry data standard - final version.docx

Annex III: The mapping between Tax and Customs Board KMD and accounting entry data standard - final version

The mapping between Tax and Customs Board KMD and accounting entry data standard - final version.xlsx

Annex IV: The mapping between Tax and Customs Board TSD and accounting entry data standard - final version

The mapping between Tax and Customs Board TSD and accounting entry data standard - final version.xlsx

Annex V: MCDS User Guide - final version

MCDS User Guide – final version.docx