EUROPEAN UNION AGENCY FOR RAILWAYS

Moving Europe towards a sustainable and safe railway system without frontiers.

Vojkan Stefanovic
Senior IT Manager (Common Components)
Team Leader General IT
RailNetEurope
Austria Campus 3
Jakov-Lind-Straße 5
1020 Vienna,
Austria

Valenciennes, 28/08/2025

Compliance Test Results – New Common Interface (CI)

Dear Vojkan Stefanovic,

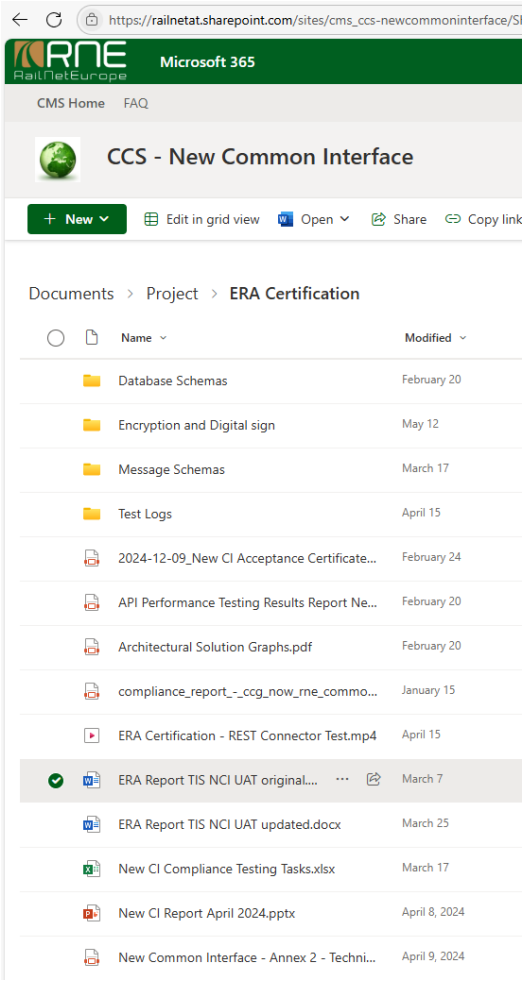
I confirm and certify herewith that the TAF TSI functionality implemented by the company RailNetEurope (Austria), within the project "New Common Interface (CI)" is compliant with the appropriate provisions of chapter "4.2.11 Networking & Communication" (and in particular of "4.2.11.5. Common Interface") of the TAF-TSI Regulation (EU) 2021/541 and its *new future* Technical Document TD-104 "Common interface" (version 4.0.0).

In this activity the company RailNetEurope implemented the new version of the TAF TSI Common Interface function. To enable the ERA verification, RailNetEurope has sent on 08/11/2024 to ERA technical descriptions for the New Common Interface version 3.0.6 and has set up for ERA a Common Interface test environment (see Annexes).

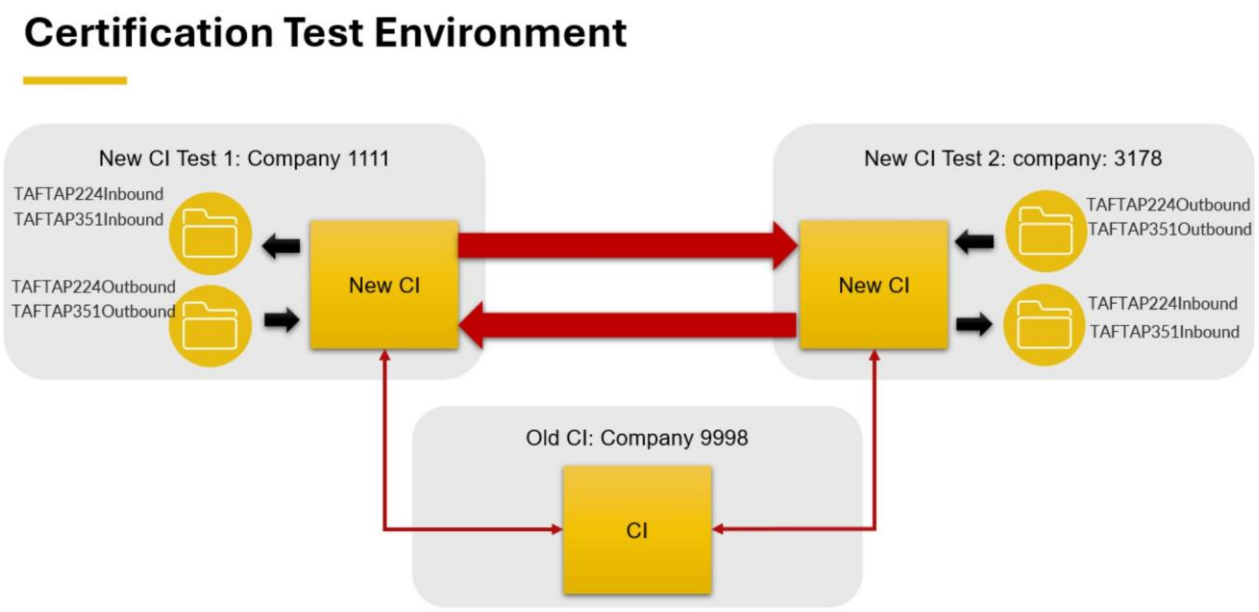
Mickael VARGA
European Union Agency for Railways
Operational Data Unit
Project Officer
Email: mickael.varga@era.europa.eu

Annex 1 – New Common Interface testing technical descriptions

They were made available by RNE to ERA and could be found here: [CCS - New Common Interface - Documents](#).



Annex 2 – New Common Interface test environment



Access to the file systems of above Common Interface instances was done with the help of Win SCP version 6.3.7.

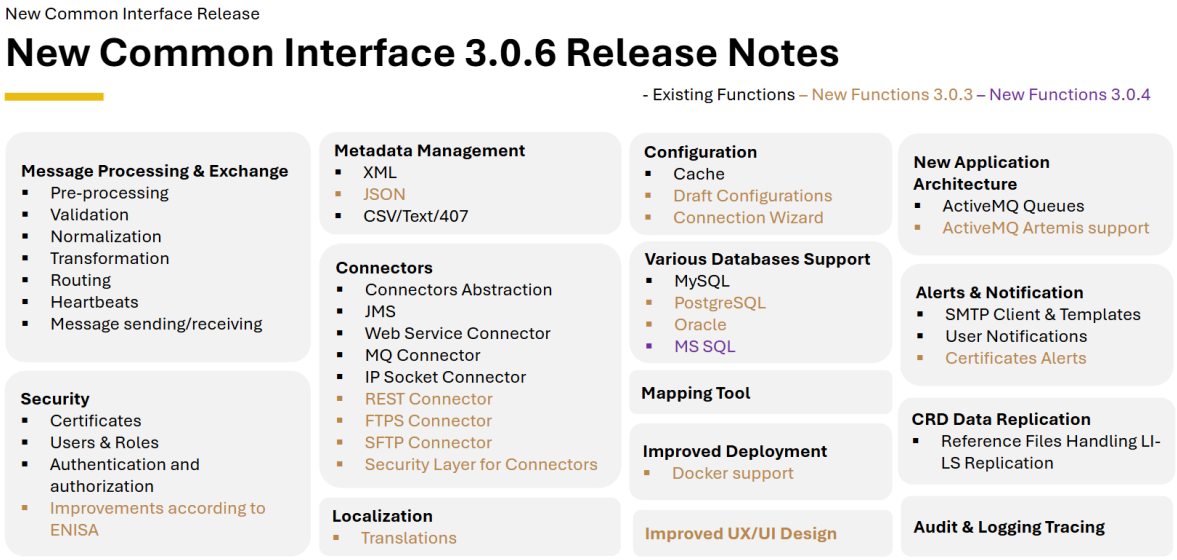
Old Common Interface
WEB User Interface access: https://10.244.35.225:8443/

New Common Interface Test 1
WEB User Interface access: https://10.244.35.205:8443/

New Common Interface Test 2
WEB User Interface access: https://10.244.35.206:8443/

Annex 3 – Peer to peer communication and interoperability

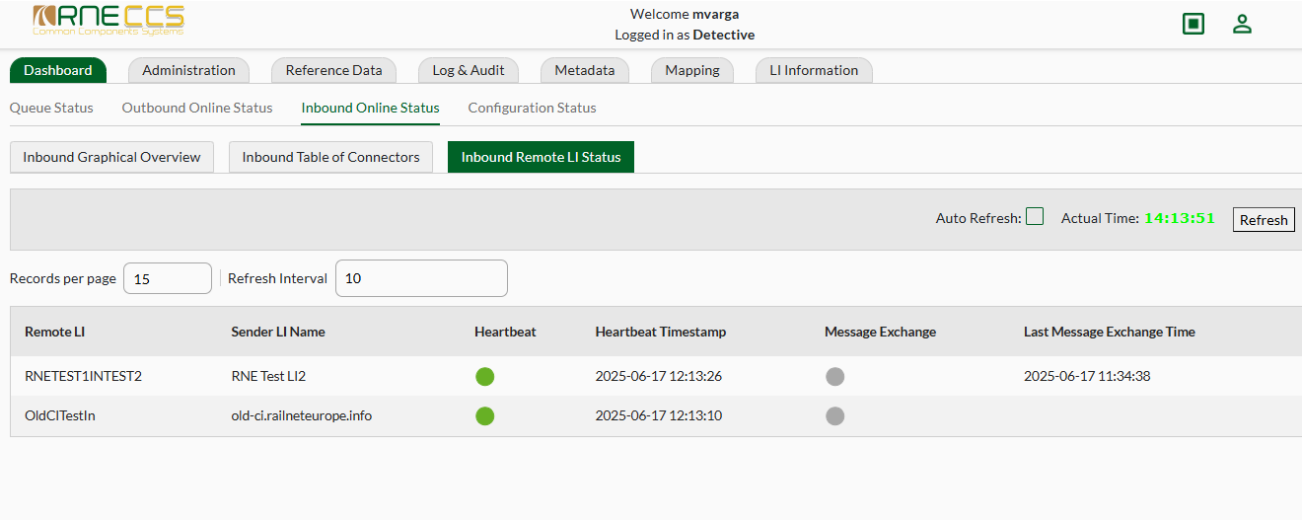
The New Common Interface ensures rail interoperability through its existing and new functions as summarised below:



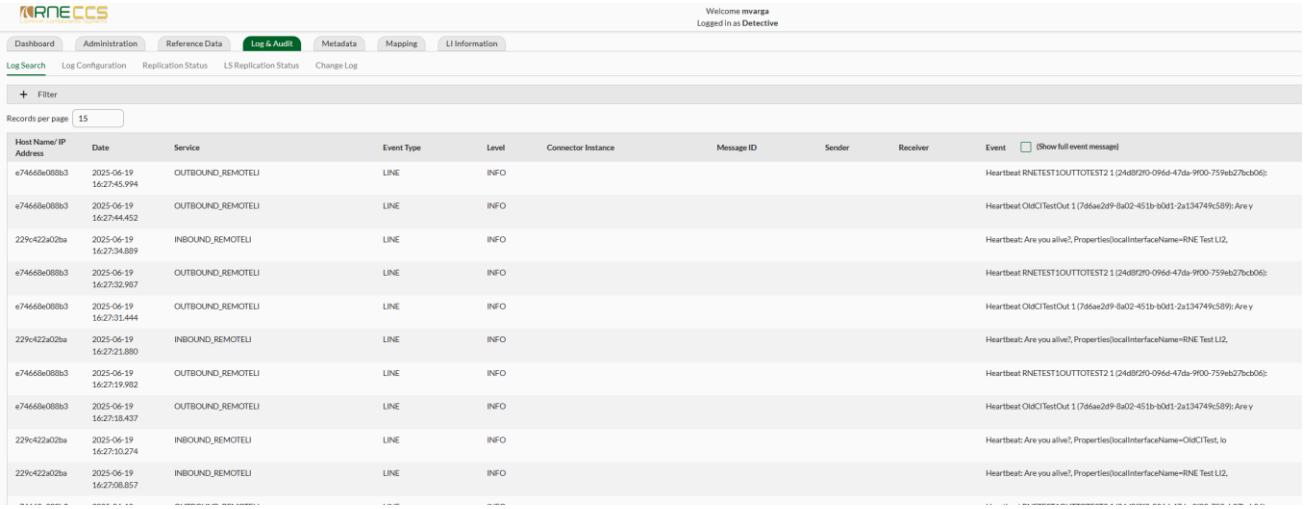
It allows – in line with the provisions of of "4.2.11.5. Common Interface" of the TAF-TSI Regulation (EU) 2021/541 – the peer to peer communication between 2 CI local instances / database as it is capable to handle

- message formatting of outgoing messages according to the metadata,
- signing and encryption of outgoing messages,
- addressing of the outgoing messages,
- queuing of messages,
- authenticity verification of the incoming messages,
- decryption of incoming messages,
- conformity checks of incoming messages according to metadata,
- handling the single common access to various databases.

With above message formatting and conformity checks, the CI ensures European rail interoperability. It is also important to mention that a CI can only perform all these actions when it is secured that a connection with the other CI instance is available (heartbeat is available):

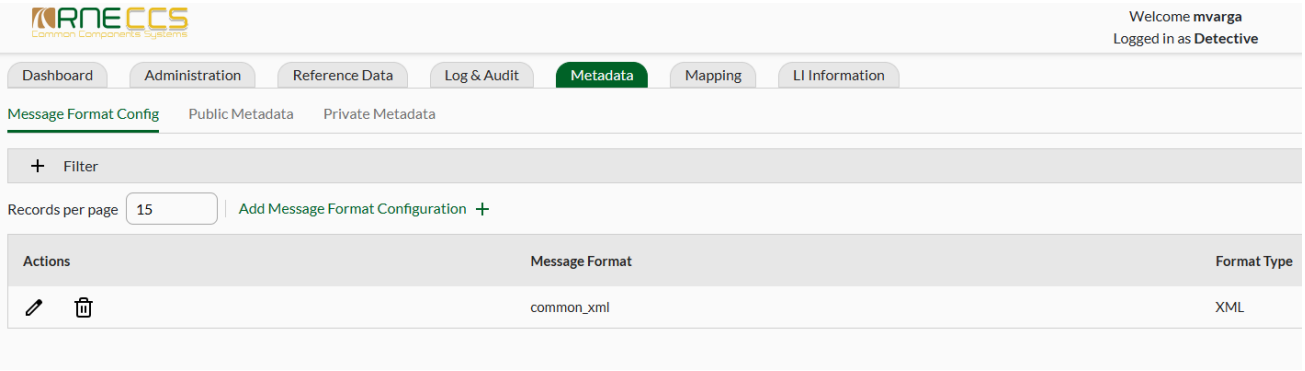


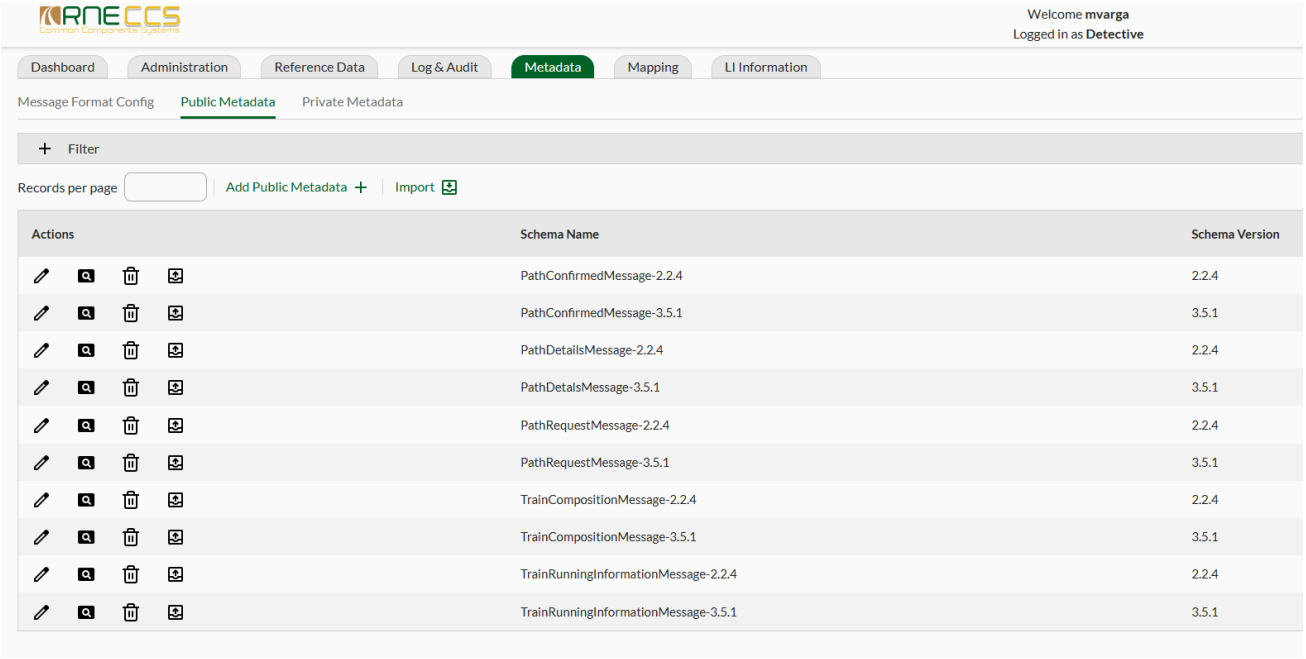
Above heartbeat is configurable and can be monitored within the Common Interface.

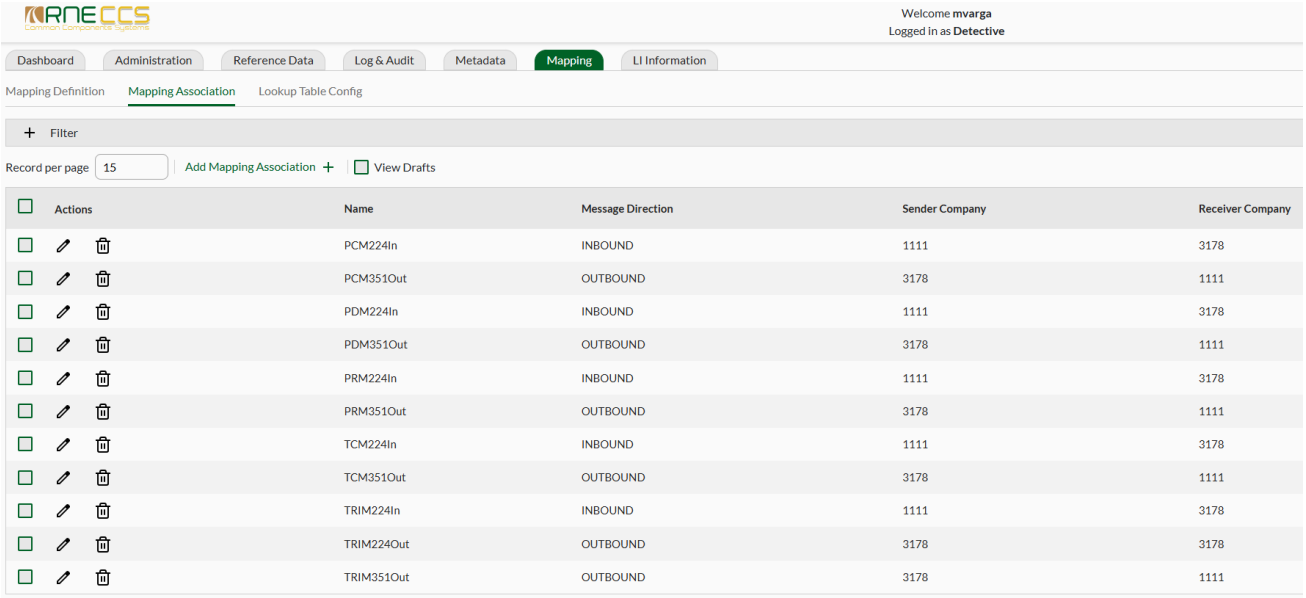


1. message formatting

The outbound message formatting can be configured within the following CI menus:







Above information (together with other CI definitions and operational information data) is stored in a dedicated CI database – see DDL (data definition language) excerpt:

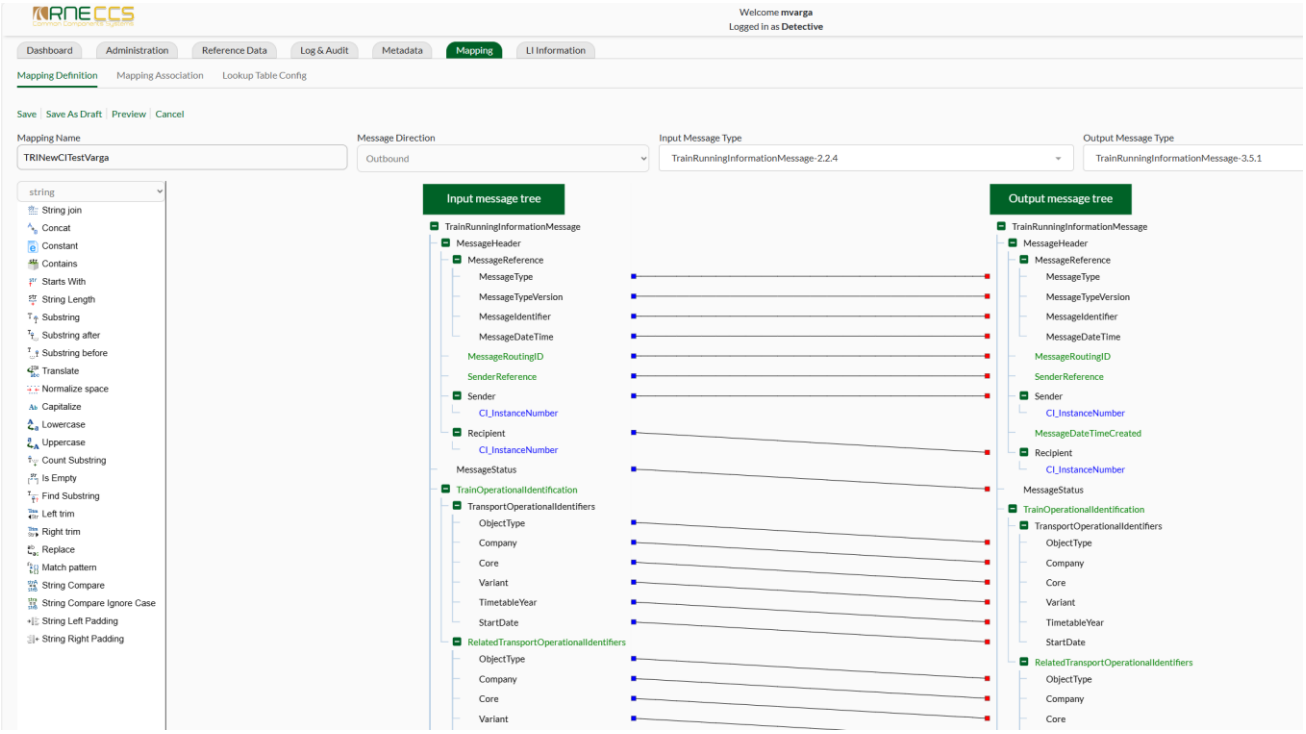
```
CREATE TABLE IF NOT EXISTS `processing_reports` (
  `created_time` datetime(6) NOT NULL,
  `id` binary(16) NOT NULL,
  `draft` bit(1) DEFAULT NULL,
  `effective_time` datetime(6) DEFAULT NULL,
  `operator_id` binary(16) DEFAULT NULL,
  `replaced_time` datetime(6) DEFAULT NULL,
  `uneffective_time` datetime(6) DEFAULT NULL,
  `common_msg_type_enum` tinyint DEFAULT NULL,
  `connector_instance` varchar(255) DEFAULT NULL,
  `connector_type_enum` tinyint DEFAULT NULL,
  `end_date` date DEFAULT NULL,
  `first_group_enum` tinyint DEFAULT NULL,
  `last_time` int DEFAULT NULL,
  `legacy_msg_type_enum` tinyint DEFAULT NULL,
  `msg_direction_enum` tinyint DEFAULT NULL,
  `receiver_company_id` binary(16) DEFAULT NULL,
  `remote_li_name` varchar(255) DEFAULT NULL,
  `report_type_enum` tinyint DEFAULT NULL,
  `second_group_enum` tinyint DEFAULT NULL,
  `sender_company_id` binary(16) DEFAULT NULL,
  `start_date` date DEFAULT NULL,
  PRIMARY KEY (`id`,`created_time`),
  CONSTRAINT `processing_reports_chk_1` CHECK ((`common_msg_type_enum` between 0 and 2)),
  CONSTRAINT `processing_reports_chk_2` CHECK ((`connector_type_enum` between 0 and 6)),
  CONSTRAINT `processing_reports_chk_3` CHECK ((`first_group_enum` between 0 and 5)),
  CONSTRAINT `processing_reports_chk_4` CHECK ((`legacy_msg_type_enum` between 0 and 2)),
  CONSTRAINT `processing_reports_chk_5` CHECK ((`msg_direction_enum` between 0 and 1)),
  CONSTRAINT `processing_reports_chk_6` CHECK ((`report_type_enum` between 0 and 3)),
  CONSTRAINT `processing_reports_chk_7` CHECK ((`second_group_enum` between 0 and 5))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE IF NOT EXISTS `queue_management` (
  `created_time` datetime(6) NOT NULL,
  `id` binary(16) NOT NULL,
  `draft` bit(1) DEFAULT NULL,
  `effective_time` datetime(6) DEFAULT NULL,
  `operator_id` binary(16) DEFAULT NULL,
  `replaced_time` datetime(6) DEFAULT NULL,
  `uneffective_time` datetime(6) DEFAULT NULL,
  `common_msg_type` varchar(255) DEFAULT NULL,
  `connector` varchar(255) DEFAULT NULL,
  `failure_from` varchar(255) DEFAULT NULL,
  `failure_to` varchar(255) DEFAULT NULL,
  `msg_id` varchar(255) DEFAULT NULL,
  `private_msg_type` varchar(255) DEFAULT NULL,
  `queue_depth` bigint DEFAULT NULL,
  `queue_name_enum` tinyint DEFAULT NULL,
  `receiver_id` binary(16) DEFAULT NULL,
  `sender_company_id` binary(16) DEFAULT NULL,
  PRIMARY KEY (`id`,`created_time`),
  CONSTRAINT `queue_management_chk_1` CHECK ((`queue_name_enum` between 0 and 3))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE IF NOT EXISTS `remote_li_info` (
  `created_time` datetime(6) NOT NULL,
  `id` binary(16) NOT NULL,
  `draft` bit(1) DEFAULT NULL,
  `effective_time` datetime(6) DEFAULT NULL,
  `operator_id` binary(16) DEFAULT NULL,
  `replaced_time` datetime(6) DEFAULT NULL,
  `uneffective_time` datetime(6) DEFAULT NULL,
  `additional_sender_company_id` binary(16) DEFAULT NULL,
```

Outbound messages can be transformed through a dedicated mapping tool which can store mappings according to the various types and the direction of the message. This tool can generate output files.

The schema validation of the output files occurs during the message processing. During runtime, all outbound messages are validated against the defined XSD schemas. Any validation errors that occur during this process are logged and reported accordingly.



2. signing and encryption

The CI can sign and encrypt outbound messages according to the provision of the ERA Technical Document 104 using a private key derived from the TLS certificate.

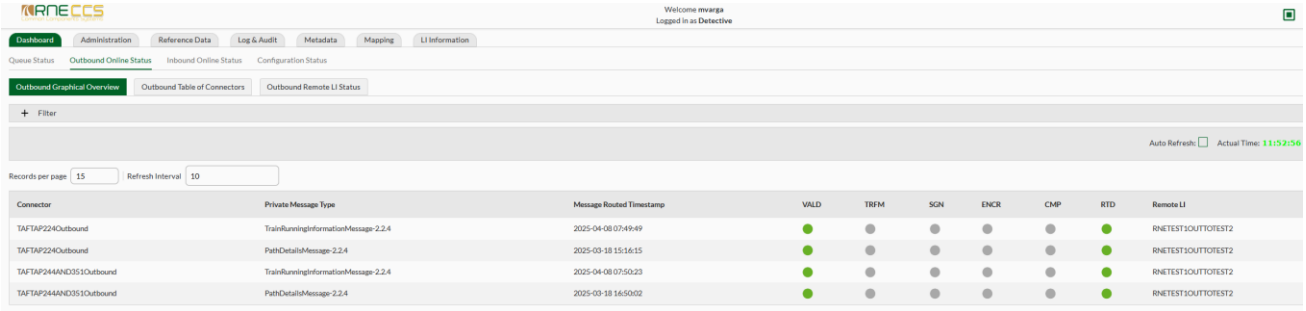
The output below is an example from the CI system that is handling outbound requests and heartbeats for communication between networked components.

The first section includes a DEBUG log, indicating an outbound request is being sent. The message is compressed , encrypted, signed, and encoded in UTF-8. The sender and receiver have designated aliases.

The next sections are INFO logs related to heartbeat messages. These are “Hey, are you alive?” (heartbeat) checks between different components, confirming communication is active.

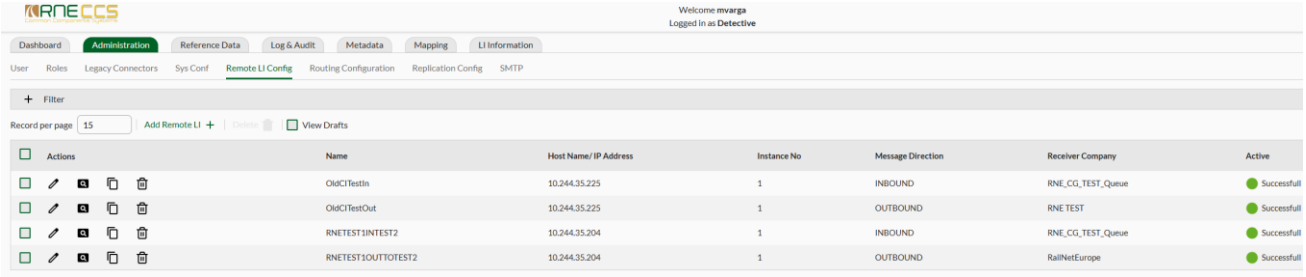

```
2025-05-12T09:03:26.521Z DEBUG 1 --- [n(908)-10.0.1.5] c.r.n.c.o.r.i.s.OutboundConnectorImpl    : Send OutboundRequest(messageLIHost=10.244.35.204, messageIdentifier=3226be84-04e3-406a-a0b
signature=MOILI7pIZlrNJM1BowkO/C/w8KZz+k6Va695L27kwLdzbw/1SG/vORUwGNzrcm0mkkT6I9sLk/qruRaNb0jJ8ARmeDl4JfjNAyAAFCv9czEYU+QE/ugW1K2s38jmw95kPYaefQmAqVS6VE0hO4amby510AWOC/6RrhRZgjRLIxsoXDkYsY
+XKgHwkNj3SXH8OLnf/NNdb6UO3LDBgD4ypjfDGQH4RFJWEP1yEh3XAETjCM1HJKyODoYzHnHjy+QMaDIiawhM36RBmhSyfVnxrlqwbWJDX9cw41gctLmCMy/qkqlnL7Z0qrsMH55r/tpeg7pbRARcABPfnocLhPSWobV1Mn
+ViXid1na2HiMJg6zcphANY8P78sGpvQrusrP56tL2QQ8gwIYXCQrBsvzH0BuV5Cpe83bW9aKeXft7+mWWl6GK7V9elrMrb8JmfSvZ9zE2Sf90Wb3gffn3iZGjsUc6Q58r3Dw6ldgsVHWJSMjZ1SP5P/rH6WZn1v8bJp6pAwxiyveOXDEablvCXB0DSr
message=eJwB2Asn9IxJs2TFQuDi6dEdhk9/bZSV9Pz4gcKbkn4di5TrJVfdG49rWyGYrqIlpMP3ZKL++BdvzBtq4ujZQIEXfh17qqb5mM/bkmdlKZLL+eBzxhZV2HKk2Hi3Ab8uqrjcZGI0eEm
+SpsAXCEasZJf9i7QJvZ8IepziPrmMYSaRdV6Qs7kTj50LDt5RTCi70XZn2urgF4xmu7wg1FeMb/T99Kps4b1188c7c6d9HYRbtvj23eETcEK/YNA5pmm1PaCcquH4csJWMtEi/qqBZ/xRk1pkQSq7BDgDBKU0ND4/mzVpPgxAJqfp45Ahwt8ifEkvfr
Sy1o40giVYwp0KwEwpnAldrBZRPbYCfAl5nY1mAVKEvjVPviYmDKsY4H/s8NR8t69B13D2FU48+b9SOX27VkVGqmk/IrtUM75jW/RVo5cTbAM+qm3NmAGNa29s1IMcJwnbjOhIg8hhkELqMLtoiTBDZMytpJMkUT5eiPODN
+esWsXI1gawduBYccQXZ2MMFtaNbw08wMuSLuzFg2xGag8DYy1tGqETK6xdJ1t9SM5mtOSJOFkHwVb9oNe672XUKXBwpUduGaYz6e99SLvW7xiByrJxTGknKUqUK87V8TgH45LdsSmc1+t0ZBT6gJMhNAvaAGLquHkXJS+fCkMpbDO4M56Cyq4K+arRa
+69IMyJjLnqybK3WDrgpOPkNSpR4abVWg85WccqnzDFxPwq8gbvZgUp4vik670toNTXAIWHLPqDdsFgvSLwVFQW6Nw1LbGahgN2CDAamSZ/NydxUxwcciJ/VHUzryTm53VSeoH4kdfvZBUa4UJ8SWntws/MJnCw2Wj6H/s8bSpsaFiFziF5efFJ5xVP
+uX4/H/jOkjSMg8ncserUnvJE6unGZ3IBRJwYuy02tigHQtAEyfGFWCs3wS0BPIAH4qmaSbZkZzFggmNOH/Ze3qFuQMAvWtzblYwv75f4/azaasL3SiIcUd/A60nLkWovH5eLe44AsZAk5twpH4gjdFToJx8LtKxa8KC1jwqJl1F7/IPhpfREWQL+kjk
+rqVqFTwiGwh7ATNFuKKqktBmeyRsFUhd1hVTojdr/H00NZpxjtIgWxzN/fnkCsM4UE9oR6STJgQ/qJSu8zR+NT71IHz43C1q6x5VptuTEtTIfVIKnXrAjtpcWariekZmrthFjcJv/1I1Lum3ytkufrpERRiRuCvxudmd9B6M375rSgL0fa1DvEY
+D2StjAvCk1CxMDX2dUqs79bnIemMxSckEB2Q9utoHrJVSgiPs824YSTSxvjFYWIp2ltO161490W3RsdHWHD8DOUOUFEhjyYQWO6L0WbBSsqW03P8oZn4mV/P1PVLx4kTMREa43K0zjtHdPjWAQJME9UY6ojHIDSyMvVhfv1Qj8omFfjZOOqjuI9jAgj
+gUTA6D7nRIV/oMfpTqnYQCynm2anNrt6GmsaEryfw1glh4/MihGVb2LET5kAXuw/tTkefyLDhxm3iOFA4RpCF81TqXrV0/jfzXG4+hsaZC3edMcA3dZ
+UhYqarbzFFg3LE3pfhxic/Tuk0ipd5Dgz6eUQz7ao9nYgCG2NPBWYd0JXJ/IHBhxGvHgRDQ3PU9wONcAQiuEAg3EFiDspJp9hglvXmFK8R/1TB960ta4rPNvRZ3rIG9PzUAujTUOWaQOG3Wp6U0+YjNYdxjBSKGsQjXWy5B5jiZ6D5CJsxg4wVexr50
+AV9sDrfsa8Mz6O3hsNVwc3AikI68zd41BJMAg3bG/bO1XsTEfQuhHDROtfUKc1d7M9ocYKAAzrWHrjuktaVadnJkHZTb7r2zSzp1/SENh8BFB1kRNfH18IDDVu/eShf7uMXh9WccHrFisowDzMK+st36KSLh3OO14QPnpi/XXWapfePVPJn5YsK3Zz
+wSD6ragUz5dpp0KhTnUVIGeMZPpc2Fh9D85gwbVQ4ADYDapnyVbMkN3yF/l1/yZ2mUAoCXvd2E3SThcojx3FHOoer13abqX0WaYXCv6xuAmbyzd2a1MahcDomkK+9p4YZxS2e8IaUse4txek9Iaq5I6GCjjHFnhWzfrpWUKkYMVNV3iUrN4LHpL5Z+N
+0hoguZ1rpR9ztyrqj7PA5AWnIfARlaR8OWPh7N9e5cTol+sLeeA0eP5NMdFdERaKT1XvdaUE2ukVs1fEfADHicXfWCs6vfG4uIdh1KmjYHmPybiXT+ktkh1tJEADxG6HGjGkwC0g09dJcHZuEd0+zeD+401EGLvkR0DESm23QehHlvANKW5e2OaFUVW
+qZbugDZIU5mzh8qSyOwSbhXCVTKrgbOHXUU4Fegnrsbw8wt88OFuYgJsWibH/JLgu7jHRD7MR0yMKWhBdmDHOOqigCZtj8j91iyFB6A5nDH+lF0rGuLTHBeZyJgWv+Myj3moYQbrfakTaXq29YkDxheNOnUY1FCdEKZUUimIhZ47N1X8q8H18rZWNp9
+FjgHXujXxuHs93PsCgbDXXG01UQii2h6k4Toc2ZayLPYgjlPTliTYpB3I5AGdApYXB8wTN715TUNLvBWKzy8DG+61JfRlQnsMxFnvBIVroym2HaU9gCXMHvmCaZdB7ZHpKXC/ejm9yvnhkhTXisgDyISK2wrcOs+nwjPQ91CbzwdnuvAKi2836hnq3l
+7pBg1ttfwzqqyGSgoV53VV5ZNBb6dp2211JvR+rPiDfJabz2CR2G/oemfosbBXK8m2hMorMxVpIKn5Bqs39hF80WsH4JgA1vxyjluA
+V/tTkxCXhkbs13tqTocfkIwxnbTJuRvpGyra6dEBCjmpuG8kA/MMQjqH3+fztWyd0V84H7BnQYARNcjW5Uyzgbsoqq9P2uqr8TBGscDhFnWpAwVtvUgD8pntRKBQ3IUdNA1E5CzoZLKTp21gBjUU0MJa7kMPNGWVzzRh06P1/ZmQadAX8fYRtGyXRK4
+85kdoHBXfxwAYQ5nrZpUY4/tVJyVrkWyW3G7LUaxnsX7Y2rtYKEx/UdbMXoWAjSTrMVuvHkGkjK1VPx4rREKQMNhQDXO+BUwZzZuU0N11cQONDNjlWf1Xv0xgHU6zrzQ==) -> OutboundResponse(responseStatus=ACK, messageIdentifie
msgType=4005, msgTypeVersion=2.2.4, msgDateTime=2025-05-12T09:03:26.413, remoteLIName=RNE Test LI1, remoteLIInstanceNumber=01, messageTransportMechanism=WEBSERVICE)
2025-05-12T09:03:33.873Z  INFO 1 --- [00-759eb27bcb06] c.r.n.c.o.r.impl.HeartbeatServiceImpl    : Heartbeat RNETEST2OUTTOTEST1 1 (24d8f2f0-096d-47da-9f00-759eb27bcb06): Are you alive?, Pro
remoteInterfaceName=RNETEST2OUTTOTEST1, remoteInterfaceInstanceNumber=1, remoteInterfaceHost=10.244.35.203, remoteInterfacePort=443, transportMode=WEBSERVICE) -> [HEART_BEAT_WS_RECEIVED]
2025-05-12T09:03:38.619Z  INFO 1 --- [5b-eb42131e3450] c.r.n.c.o.r.impl.HeartbeatServiceImpl    : Heartbeat OldCITest_Out 1 (55c2f61e-06d7-4e6a-8e5b-eb42131e3450): Are you alive?, Properti
remoteInterfaceName=OldCITest_Out, remoteInterfaceInstanceNumber=1, remoteInterfaceHost=10.244.35.225, remoteInterfacePort=8443, transportMode=WEBSERVICE) -> [HEART_BEAT_WS_RECEIVED]
```

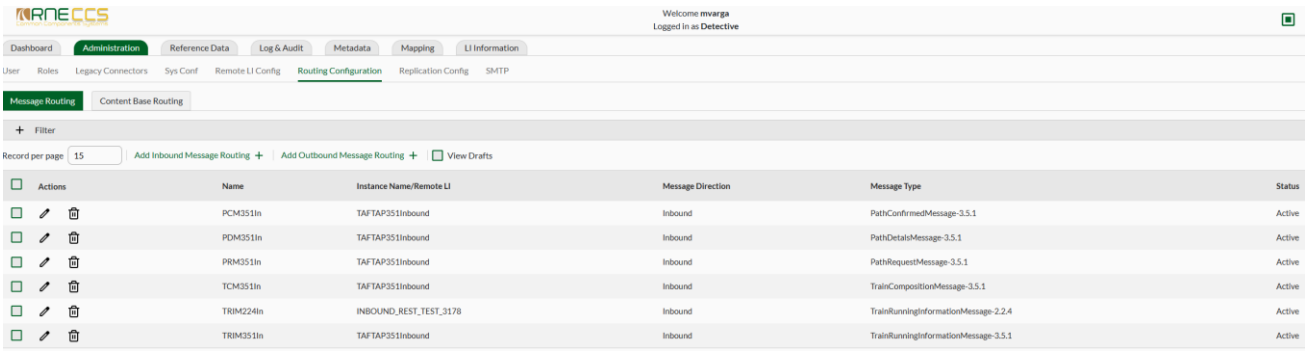

The signing, encryption (and compression) per message type and routing can be defined in the CI:



3. addressing of the outgoing message

The outbound messages can be routed in the CI through the dedicated configurations of the remote CI and for the routing. With this option it is guaranteed that dedicated messages types will be sent to the appropriate remote Cis.

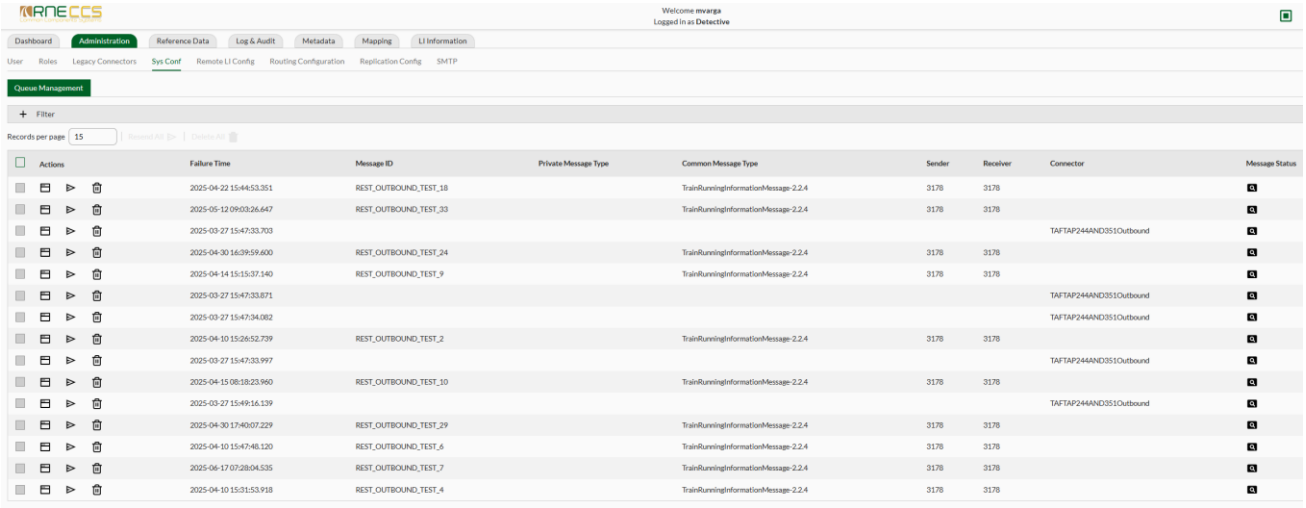




4. queuing of messages

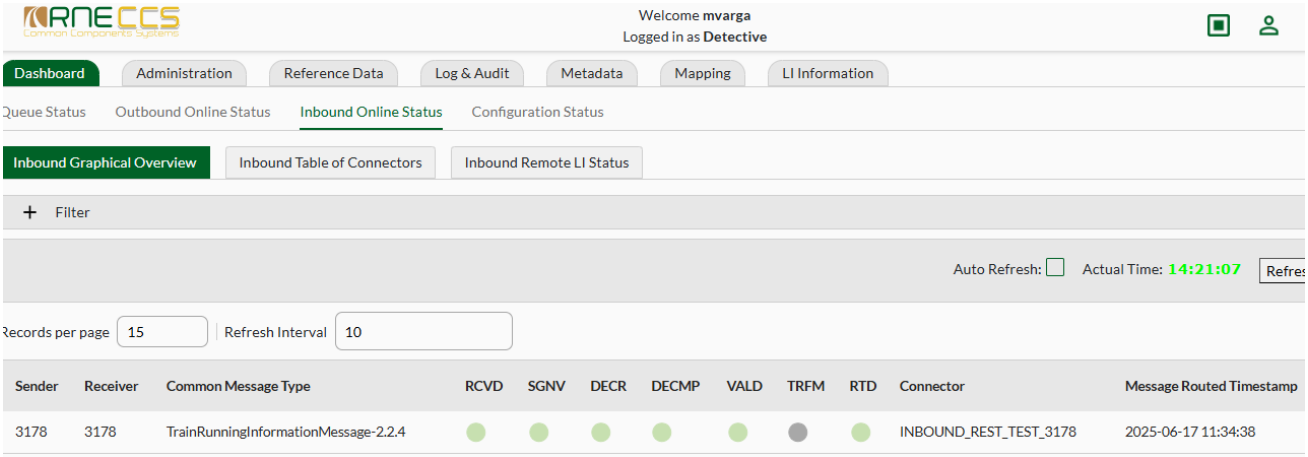
The CI can queue outbound (and inbound) messages according to the provision of the ERA Technical Document 104. During the tests 1004 TrainRunningInformationMessages have been exchanged between the CI local instances “New Common Interface Test 1” and “New Common Interface Test 2”. The CI queuing between the Translation & Validation layer and the Security & transport layer enabled both CI local instances to exchange above 1004 messages in the right sequence, with a processing time from 2025-04-08T07:49:49 to 2025-04-08T07:50:23.17, say within 34 seconds, with 29,5 messages per second (all values documented in LOG files).

Failed messages from the queue are listed with their Error stack trace:



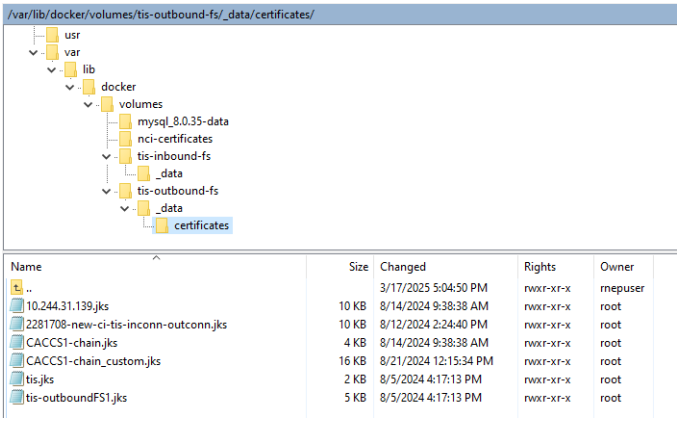
5. authenticity verification of the incoming messages

Through the routing configurations and the certificate provided by the remote CI local instance, the CI can verify the authenticity of the incoming messages.



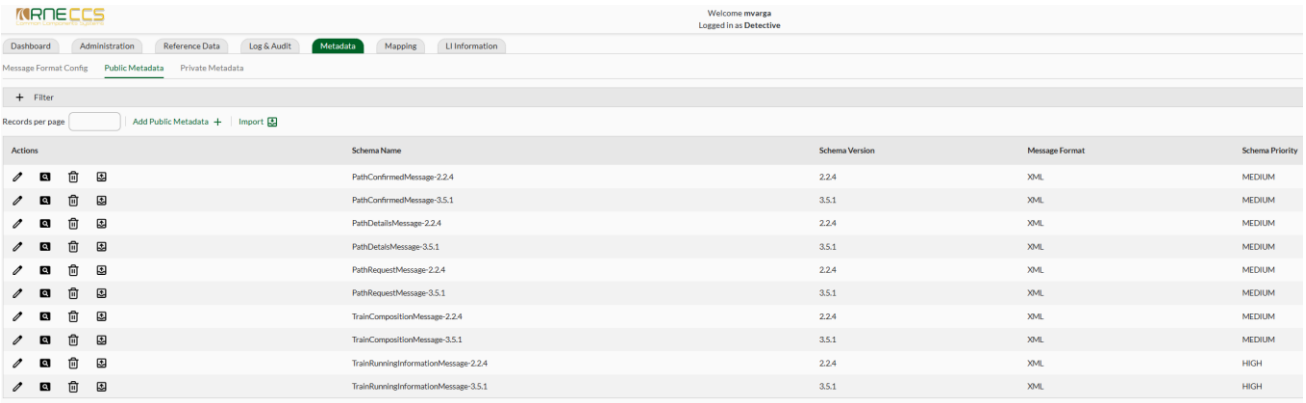
6. decryption of incoming messages

The decryption is performed by using the public key derived from the TLS certificate. The certificates are stored at (as an example) /var/lib/docker/volumes/tis-outbound-fs/_data/certificates at CI level.



7. conformity checks of incoming messages according to metadata

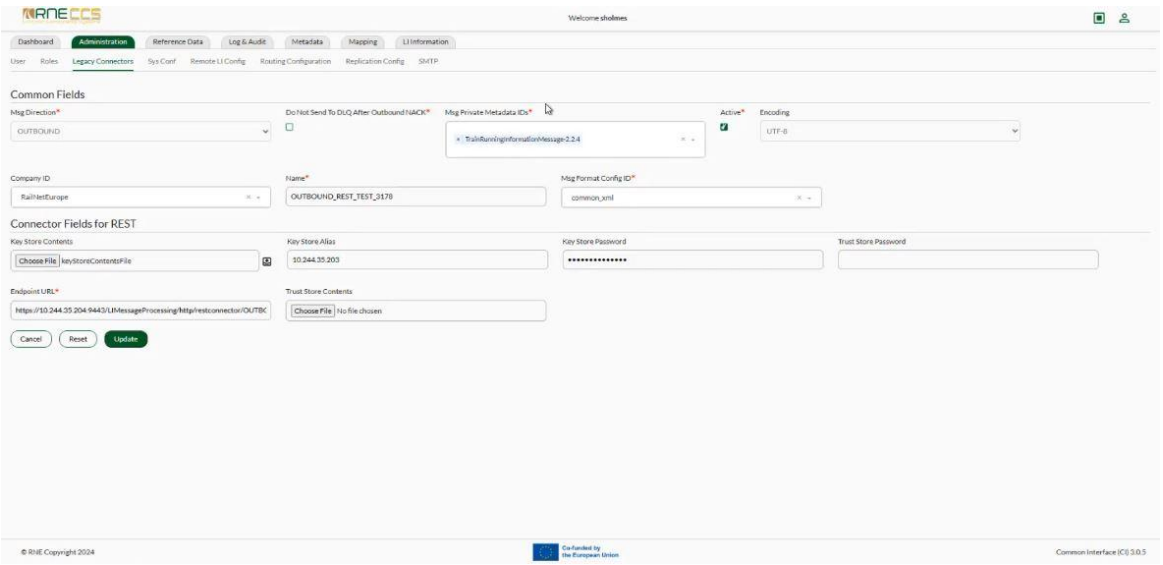
In the same manner as stated in 1 message formatting above, the incoming messages can be checked and validated against pre-defined metadata rules and the XSDs created and stored in a dedicated section of the Common Interface.



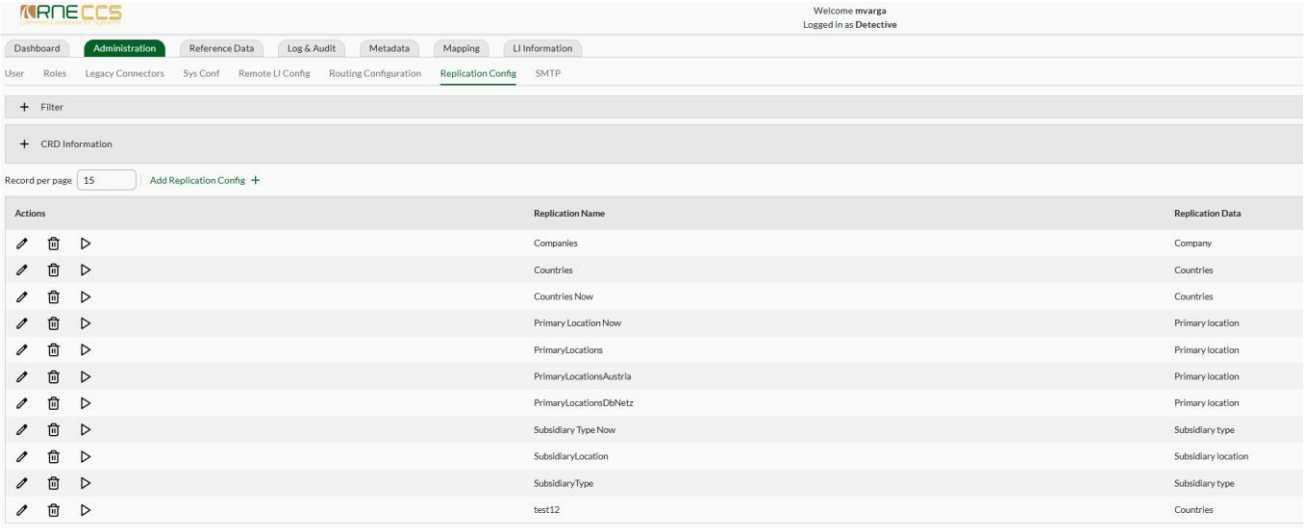
8. handling the single common access to various databases

Replication Web service allows the selection of reference files for transfer to the calling Local Interface (a node in the network interfacing the legacy systems). The Local Interface requesting replication can choose the replication data and get the required data replicated (for example, data belonging to a specific country or company). Specific attributes of reference file can be used for filtering the replication of data including last modified date.

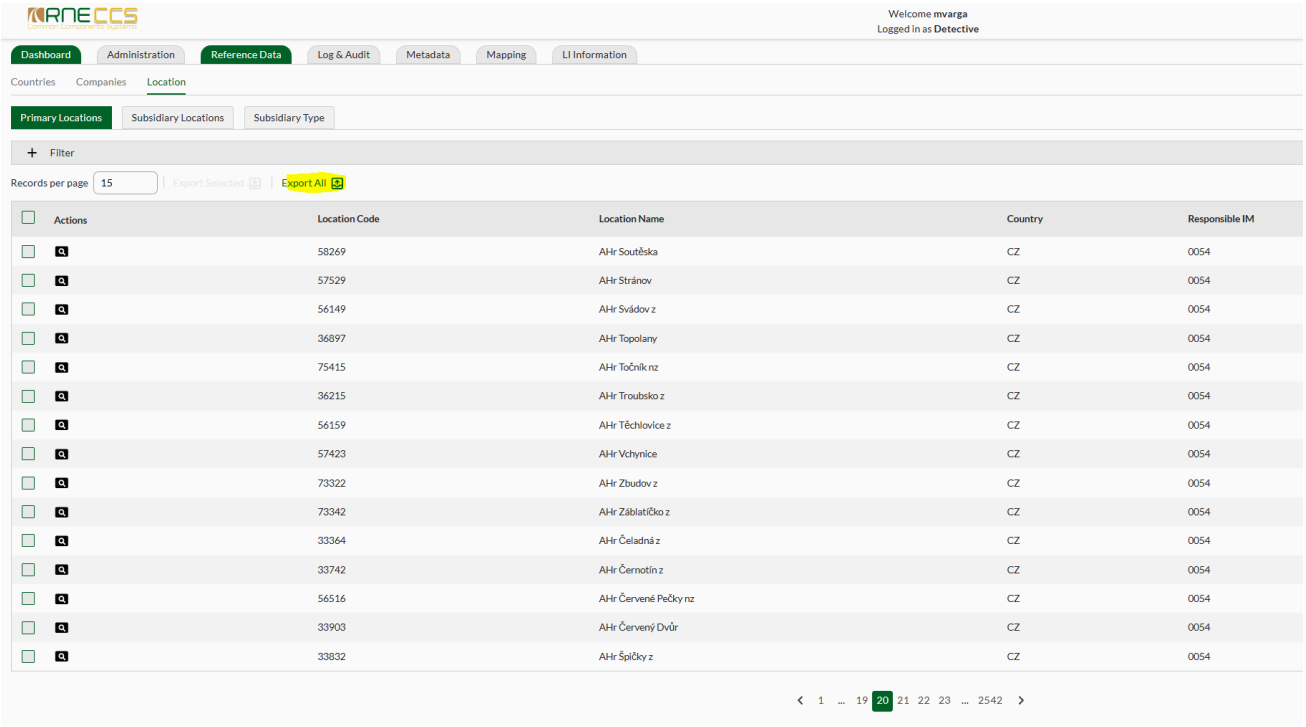
Local Interface can replicate the Reference Data available in the CRD using the CRD Replication Web service (SOAP). Legacy Systems can consume the reference data provided by LI Replication web service.



Rule based replication of databases is also configurable with the Common Interface with filters, predefined schedules and volume.



Export of the replicated data is possible:



In addition, the whole set of legacy connectors (REST included) is available to connect the Common Interface with users' internal systems.