EUROPEAN RAILWAY AGENCY

INTEROPERABILITY UNIT		
TRANS-EUROPEAN CONVENTIONAL RAIL SYSTEM SUBSYSTEM TELEMATICS APPLICATIONS FOR FREIGTH		
Reference:	**Document Type**:	User acceptance test of the common interface
Version : 1.1		
Date : 04/04/2013		

	Edited by	Reviewed by	Approved by
Name	Mickael VARGA Stefan JUGELT	Kai BRANDSTACK	Denis BIASIN
Position	Project Officer	Head of Sector	Head of Unit
Date & Signat.			

AMENDMENT RECORD

Version	Date	Section number	Modification/description	Author
0.1	13/01/2013	All	First draft	Stefan JUGELT
1.0	18.02.2013	All	Review	Mickael VARGA Stefan JUGELT
1.1	04.04.2013	1, 8	Review	Stefan JUGELT

Important remark:

This Test Report from 2013 contains the results of test performed by ERA to evaluate the compliance of the CCG (now RNE) Common Interface Reference Implementation. Meanwhile some improvements based on ERA recommendation have been implemented in newer releases of the Common Interface.

Table of Contents

Tables

1. INTRODUCTION

1.1 Background to the assignment

The task for the assessment of the TAF TSI implementation was set out by the TAF TSI regulation 62/2006.

According to the chapter 7.1.3. of this regulation:

1. Once the Strategic plan is completed, all activities related to the implementation of the subsystem Telematic Application for Freight have to be justified against this deployment plan. Any proposed non-adherence by a RU or IM should be justified in the implementation dossier submitted to the Member State, to the European Railway Agency and to the EC..

According to this regulation ERA shall be informed and monitor the progress of the development of the TAF TSI.

According to TAF TSI 4.2.14.7 and TAP TSI 4.2.21.7 is the Common Interface mandatory for each actor in order to join the rail interoperability community". This central component is the main component to facilitate the message exchange between the different actors in the telematics applications for rail domain.

ERA has performed these tests on the basis of a written agreement with the CCG. The agreement has granted to ERA the right for a test of the CI on ERA IT-systems.

1.2 Summary

1.2.1 Monitoring of the development of the CI

Currently the railway undertakings and infrastructure managers in Europe are exchanging information's with messages in a legacy format. Those messages are developed in the past by the railway undertakings and standardized by organisations of the railway sector (e.g. UIC). Those messages are not used on a European level, even if they are exchanged between major railway undertakings.

The TAF TSI regulation has changed the requirements for the message exchange between those undertakings. The for the message exchange a set of common used XML-messages was established. The railway undertakings shall be enabled to transform the messages between their legacy systems and the common used messages. For this purpose a so called message converter is needed.

Reference Implementation of the Common Interface was built by the UIC Common Components Group based on TAF TSI — ANNEX D.2: APPENDIX E — COMMON INTERFACE.

To test the capabilities of this tool ERA has made a test campaign to understand the functionalities provided by this tool and to check if the tool is capable to cover the requirements from TAF TSI. This report describes how ERA has done this task between September 2012 and January 2013.

ERA can highly recommend the rail sector to use such a standardised message exchange tool like the Common Interface Reference implementation for telematics applications. The tested common interface - could reduce significantly the efforts to create interoperable rail telematics systems.

2. ABBREVIATIONS AND REFERENCES

2.1 Abbreviations

Table 1 : Abbreviations

Abbreviation	Definition
BP	Basic parameter
CCG	Common components group
CI	Common Interface
CR	Change Request
CRD	Central reference database
EC	European Commission
EIM	European Rail Infrastructure Managers
ERA	European Railway Agency (also referred to as Agency)
RNE	Rail Net Europe
TAF	Telematics Applications for Freight
TSI	Technical Specification for Interoperability
UIC	Union International des Chemins de Fer.
WP	Work group organised by ERA

2.2 Reference Documents

Table 2 : Reference documents

Ref. N°	Document Reference	Title	Last Issue
[1]	Directive 2008/57/EC	Interoperability of the conventional rail system	17.06.2008
[2]	TAF TSI Regulation No 62/2006	Commission regulation on the technical specification for interoperability concerning the technical specification for interoperability relating to the telematic applications for freight subsystem of the trans-European conventional rail system	18.01.2006
[3]	TAP TSI Regulation (EU) No 454/2011	Commission Regulation on the technical specification for interoperability relating to the subsystem 'telematics applications for passenger services' of the trans-European rail system.	05.05.2011

3. OVERVIEW

3.1 Description of the common interface

The common interface is described in [1] as "Common Interface (CI) is a message transformation middleware, which does the transformation from legacy format of messages into common or shared metadata format (XML) and vice versa. The supply consists of two applications namely - Local Instance (LI) and Central Repository Domain (CRD). LI is the name of the Common Interface from a user perspective.

A single LI can be used either by multiple companies or single company. Each LI will optionally have a failover cluster associated with it."

3.2 Timeline of the test

The tests of the CI were planned for the autumn 2012.

4. GOALS

The main goals of the test for the common interfaces were to test the common interface from a point of view of the government responsible for the supervision of the implementation of the TAF TSI.

Therefore ERA has asked the following questions for the tests of the common interface:

- Is the usage of the CI user friendly (e.g. configuration, operation)?
- Is the common interface ready for use in RUs/IM with different sizes (e.g. it runs under comparable conditions in different IT-landscapes)?
- It is able to run under heavy load to fulfil the requirements for a reliable message processing?
- It is possible to use the CI also for the exchange of TAP TSI messages for commercial basic parameters?

It was decided by ERA to test the CI with a user acceptance test to check the capabilities to fulfil the user requirements from the point of view of a railway undertaking.

4.1 User acceptance test

4.1.1 Test of the GUI

The test of the GUI was dedicated to answer to the question if the CI can be used easily. The main functionalities of the CI GUI were tested:

- Set-up of the basic parameters of the CI (e.g. company details)
- Set-up of local interfaces to legacy systems with different transport protocols (e.g. SMTP, file)
- Set-up of different message types (e.g. train running messages, reservation messages)
- Set-up of mappings between private and common message types
- Set-up of different message formats (e.g. text based messages, XML-messages)
- Set-up of remote CI

4.1.2 Test of the CRD

One main component of the CI is the central reference database (CRD). This centrally hosted database store all the reference data used for the CI. The reference data cover the following areas:

- Country reference data

- Company reference data

- Location reference data (Primary location data, Subsidiary location data)

Each CI has to connect to this database to receive regularly updates for the main reference data. It should be tested how the CI can be configured to use the database.

The goals of this test were:

- configure a connection to the Central reference database

- scheduled update of reference data in the local interface

- read and navigate through the downloaded data

4.1.3 Test of several legacy connectors

The testing of the several connectors provided by the CI was one of the essential test cases. It should be tested how to use these legacy connectors to connect the legacy systems of the RU with the CI and transform the legacy messages to the common used messages according to TAF TSI.

It was decided to test the following legacy connectors:

File-system connector: To test the access to the CI through the file system. This test case should be tested to approve a simple method to provide a local legacy interface to the common interface.

JMS connector: the test should ensure the access to the legacy system through a message bus (based on the message bus HornetQ). This legacy connector may be used preferably by large railway undertakings with a large scale IT-infrastructure

SMTP connector: the test should show how the access to the common interface for a small RU without any IT-infrastructure – only equipped with an email address - may be ensured. The sending/receiving of emails to the common interface could provide the simplest method for an access to the CI.

4.2 Loadtest

The load test of the CI should show how the CI can handle a huge amount of data for sending/receiving of messages.
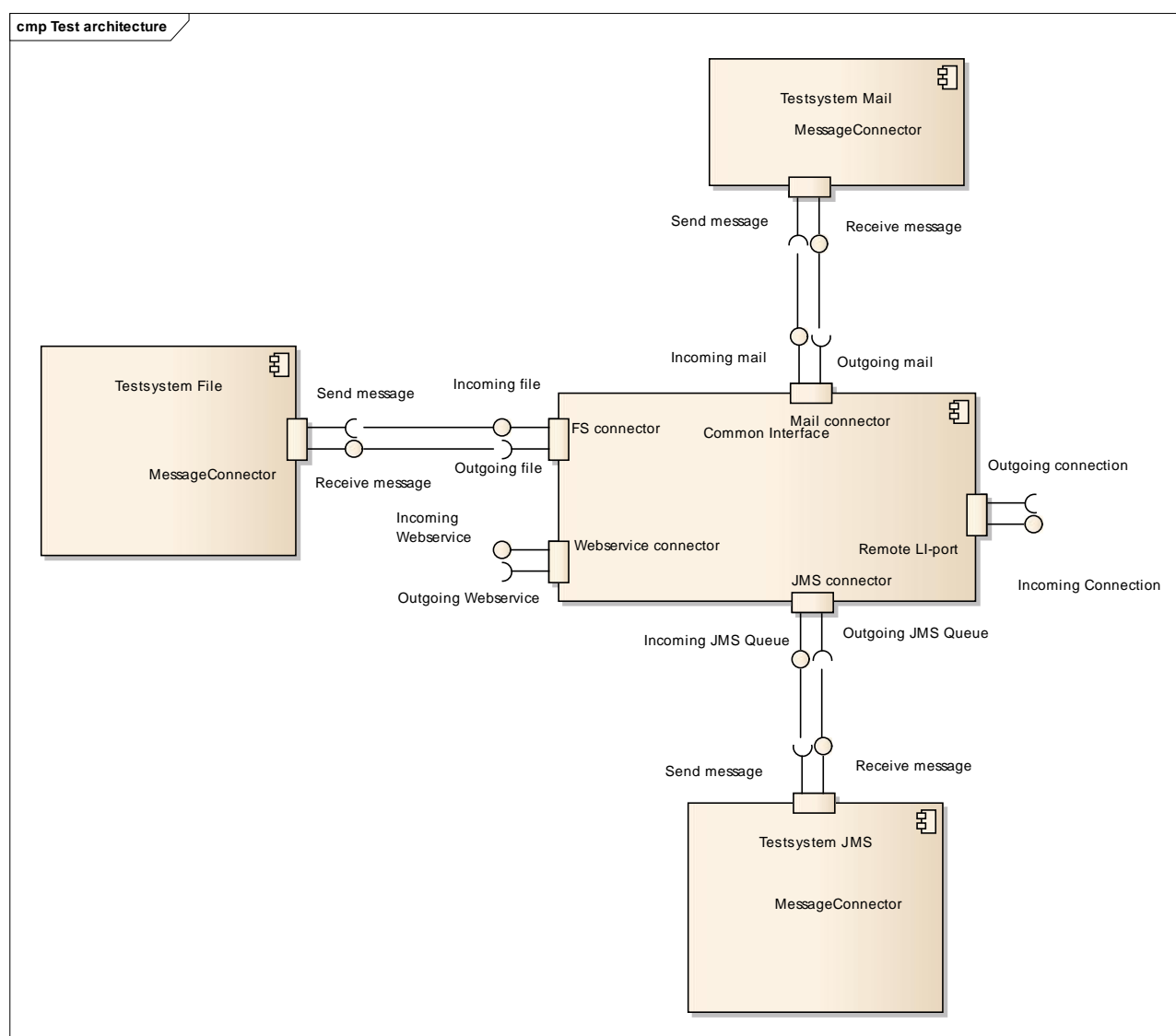
5. TEST ENVIRONMENT

5.1 Test architecture

For the tests the test architecture should support the following functionalities:

- Providing a user interface via web-browser to run the GUI tests
- Providing connectors for different legacy systems (File-system, JMS and mail)

The following drawing shows the test architecture for the tests for the CI:

The test components for the connectors were realized by Mock-Ups written in Java. All tests were created as tests based on the test framework TestNG.

5.2 Hardware

The following hardware was used for the test environment for the CI:

Server:	Virtualized host on a VMWare-Server
Server OS:	Windows 2008 R2 Enterprise Servicepack 1
CPU:	Intel Xeon (E 5504, 2 GHz, 4 processors)
Memory:	4 GB

5.3 Software

The following software was used for the test environment for the CI:

Common interface	version CI 1.1.2, replaced during the test campaign with rel. 1.2
IDE	Eclipse
Testframework	TestNG
Test-code	Test-connectors written in Java (Green mail, Java-code for JMS)
Tools for loadtest	Testing AnyWhere 7.5, Apache JMeter 2.8

6. TESTS

6.1 Testplan

#	Testcase	Purpose	preconditions	Expected results
Configuration				
1	Configuration of remote CI via Webservice	Test the configuration of the CI for the communication with a remote CI. The configuration shall be made with the support of the provided documents for the setup	CI installed and running	exchange of heartbeat messages wi remote CI via Webservice possible
2	Configuration of remote CI via JMS	Test the configuration of the CI for the communication with a remote CI via JMS. The configuration shall be made with the support of the provided documents for the setup	CI installed and running	exchange of heartbeat messages wi remote CI via JMS possible
3	Configure CRD-synchronisation	Test the configuration of the synchronisations of the reference data with the CRD	CI installed and running	Synchronisation set-up successful, connection possible
4	Run CRD- synchronisation manually	Test of the adhoc synchronisation with CRD	CI installed and running, CRD synchronisation configured	Data will be synchronized
5	Run scheduled CRD synchronisation	Test of the schedules synchronisation with CRD	CI installed and running, CRD synchronisation configured	Data will be synchronized
6	Configure FileSystem-legacy connector	Test of configuration of FileSystem-legacy connector	CI installed and running	Configuration of legacy connector possible

| 7 | Configure JMS legacy connector (via HornetQ message bus) | Test of configuration of JMS-legacy connector | CI installed and running | Configuration of legacy connector possible |
| 8 | Configuration SMTP-legacy connector | Test of SMTP-configuration for sending emails via the legacy connector | CI installed and running | Configuration of legacy connector possible |

Communication-Tests

8	Send/Receive TrainRunningMessage as XML-message via FS-LI	Send TrainRunning as XML	CI installed and running, FS-connector configured	Message can be transmitted
10	Send TrainRunningMessage as 407-message/Receive as XML-message via FS-LI	Receive as 407 message, mapping to XML, Sending and receiving as XML, compare original and secondary CSV file for non-matches	CI installed and running, FS-connector configured	Message can be transformed from 407-message to XML and transmitted as XML-message
12	Send/Receive TrainRunningMessage as XML-message via JMS-LI	Send TrainRunning as XML	CI installed and running, JMS-connector configured	Message can be transmitted
13	Send TrainRunningMessage as 407-message/Receive as XML-message via JMS-LI	Receive as 407 message, mapping to XML, Sending and receiving as XML, compare original and secondary CSV file for non-matches	CI installed and running, JMS-connector configured	Message can be transformed from 407-message to XML and transmitted as XML-message
14	Send/Receive TrainRunningMessage as 407-message/Receive as XML-message via SMTP-LI	Receive as 407 message, mapping to XML, Sending and receiving as XML, compare original and secondary CSV file for non-matches	CI installed and running, SMTP-connector configured	Message can be transformed from 407-message to XML and transmitted as XML-message
17	Transform CIS ASOUP messages 200 or 201 into TrainRunningMessage as XML message	To test whether transformation is possible	CI installed and running, ASOUP specs available	

LoadTests

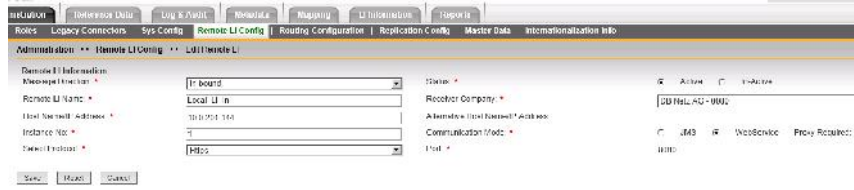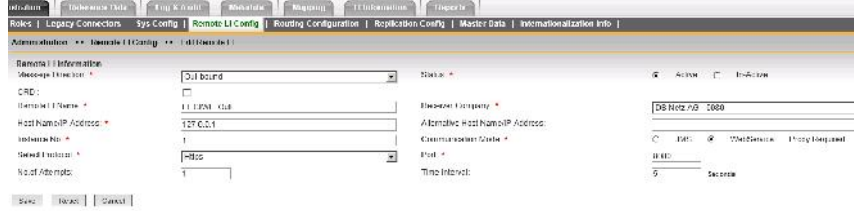| 18 | Send/Receive TrainRunningMessage as XML-message via JMS-LI | To test whether CI can transmit messages under a huge workload | CI installed and running, JMS-connector configured | |
| 19 | Send/Receive TrainRunningMessage as 407-message/Receive as XML-message via JMS-LI | To test whether CI can convert and transmit messages under a huge workload | CI installed and running, JMS-connector configured | |

| 20 | Populate CRD and SQL Server database with location codes | To test whether CI can handle huger XML messages with 60.000 locations | CI installed and running, CRD synchronisation configured and access granted | |

TAP TSI commercial parameters test

21	Configuration for a exchange of PRM reservation data (PRM Availability request, PRM availability reply)		CI installed and running	
22	Send/Receive PRM-availability request via JMS-LI		CI installed and running, JMS-connector configured	
23	Request/Reply of PRM-availability request and PRM-availability-reply via JMS-LI		CI installed and running, JMS-connector configured	

GUI-Test

| 24 | GUI test | To test whether all menus, windows etc are enabled, can be reached/used in logical sequence without pointer device (TAB, shortcut etc) | CI installed and running | |

6.2 Description of the GUI-test

6.2.1 Configuration

6.2.1.1 Test case 1 - Configuration of remote CI via Webservice

Purpose:	Test the configuration of the CI for the communication with a remote CI via Webservice.
Set-up:	CI installed and running
Test procedure:	
Action	**Input**
Login into the CI (http://localhost:8080/LI/logoutUser.action)	
Go to Administration – Remote LI Config	
Define the interface Local_LI_in as follows:	
Define the interface LI_CIWL_Out as follows:	
Go to Log & Audit – Log View	
Open the logfile RemoteLIHeartBeatCheckLog.log	
Check if the logfile contains heartbeat messages	
Test Passed	

6.2.1.2 Test case 2 - Configuration of remote CI via JMS

Purpose:	Test the configuration of the CI for the communication with a remote CI via JMS.
Set-up:	CI installed and running
Test procedure:	
Action	**Input**
Login into the CI (http://localhost:8080/LI/logoutUser.action)	
Go to Administration – Remote LI Config	
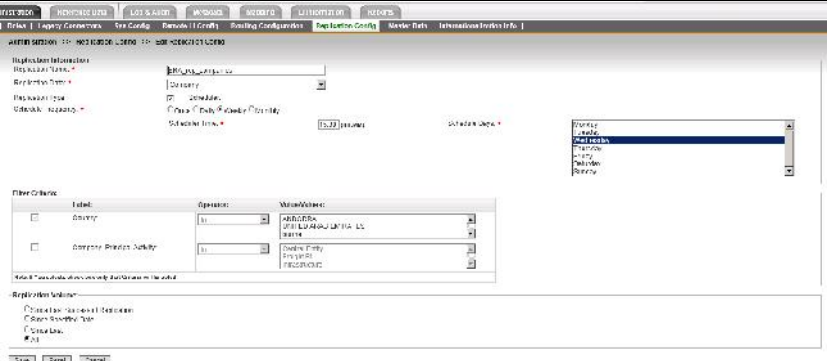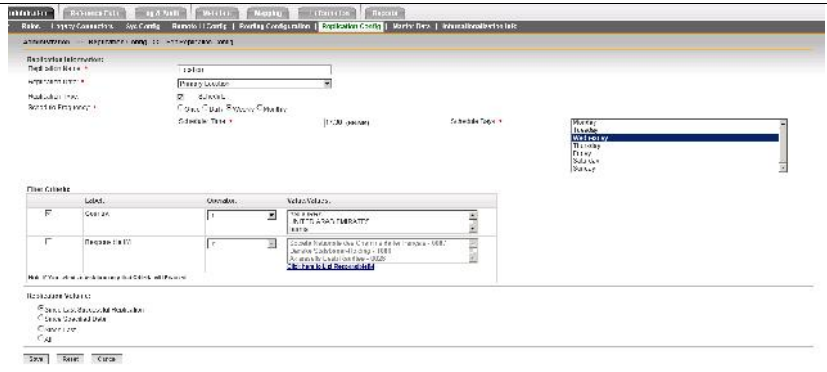
Define the interface Local_LI_in as follows:	
Define the interface LI_CIWL_Out as follows:	
Go to Log & Audit – Log View	
Open the logfile RemoteLIHeartBeatCheckLog.log	
Check if the logfile contains heartbeat messages	
Test Passed	

6.2.2 CRD-configuration

6.2.2.1 Test case 3 Configure CRD-synchronisation

Purpose:	Test of the configuration of the CI for the replication with the central reference database.
Set-up:	CI installed and running
Test procedure:	
Action	**Input**
Login into the CI (http://localhost:8080/LI/logoutUser.action)	
Go to Administration – Replication Config	
Configure the replication for companies as follows (Add replication config):	
Configure the replication for countries as follows (Add replication config):	

Configure the replication for primary locations as follows (Add replication config):	
Test Passed	

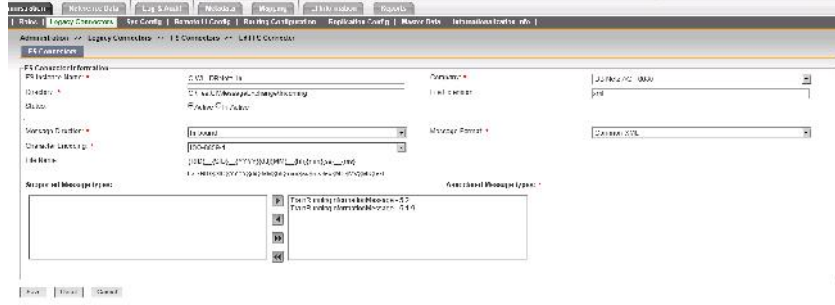#### 6.2.2.2 Test case 4 Run CRD- synchronisation manually
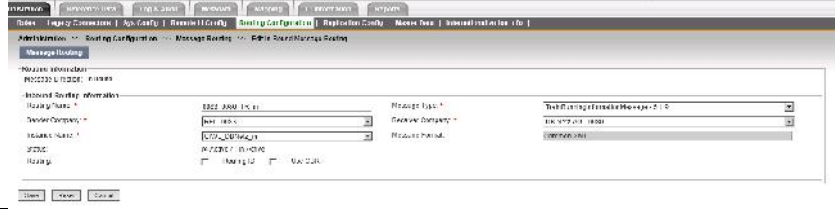
Purpose:	Test of the manual replication with the central reference database.
Set-up:	CI installed and running, configuration according to "Test case 3" successful
Test procedure:	
Action	**Input**
Login into the CI (http://localhost:8080/LI/logoutUser.action)	
Go to Administration – Replication Config	
Click on "Replication now" for the row "countries"	The message "Record Replicated Successfully" appears
Test Passed	

6.2.2.3 Test case 5 Run scheduled CRD synchronisation
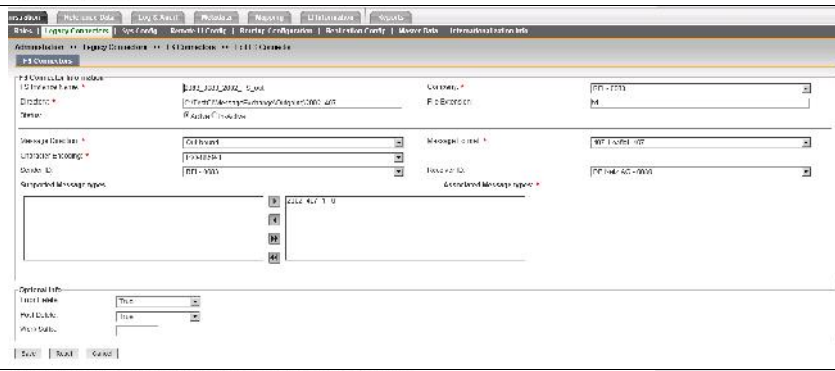
Purpose:	Test of the scheduled replication with the central reference database.
Set-up:	CI installed and running, configuration according to "Test case 3" successful, Wednesday 18h15 MET elapsed
Test procedure:	
Action	**Input**
Login into the CI (http://localhost:8080/LI/logoutUser.action)	
Go to Log & Audit – Replication status	Check if the "replication status" for the columns "countries" and "Primary locations" is "Successful"
Test Passed	
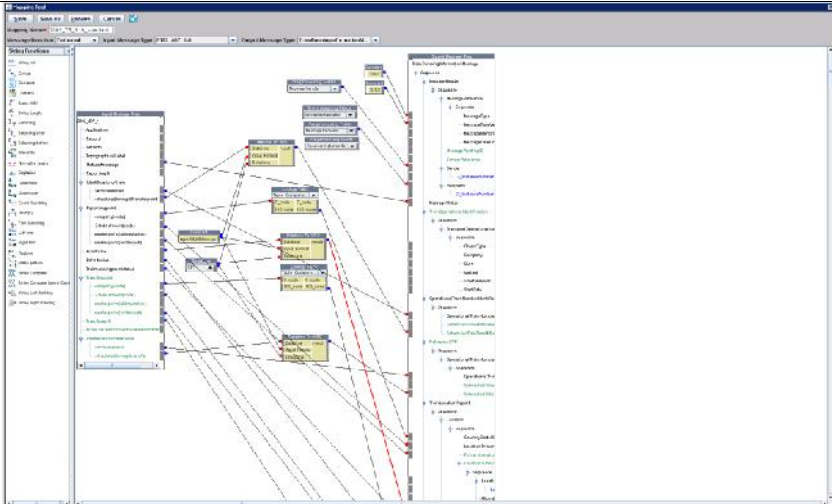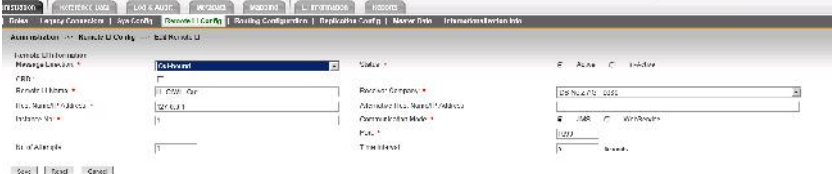
6.2.3   Configuration of connectors

6.2.3.1   Configuration of file system connector

Purpose:	Test of the configuration of a file system connector to a legacy system.
Set-up:	CI installed and running
Test procedure:	
Action	**Input**
Login into the CI (http://localhost:8080/LI/logoutUser.action)	
Go to Administration – Legacy connectors – FS connector	
Configure an inbound file system connector as follows	
Configure an outbound file system connector as follows	
Configure out-bound routing as follows	
Configure in-bound routing as follows	
Go to Mapping – Mapping definition	
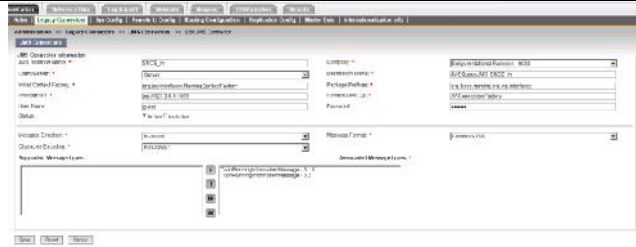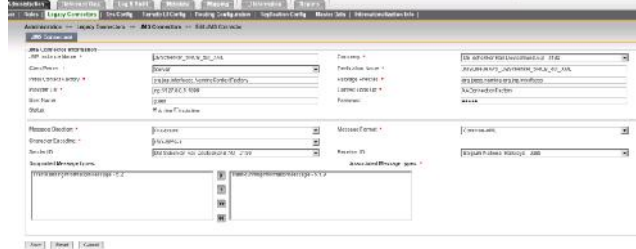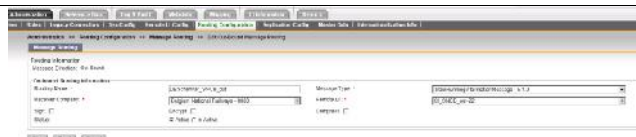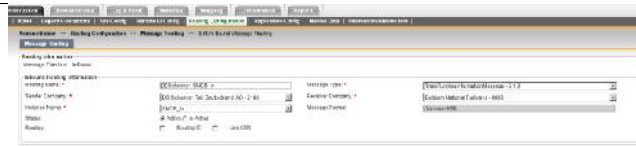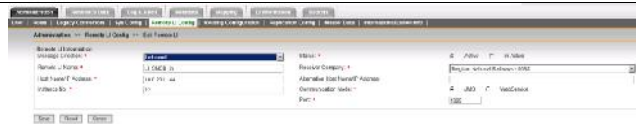
Configure a mapping "2002_TR_519_standard" for the mapping from UIC-407 train run message to Train RunningInformationMessage	
Configure the remote out-bound CI as follows	
Configure the remote in-bound CI as follows	
Go to Administration – Routing configuration	
Configure the outbound mapping association as follows:	
Configure the inbound mapping association as follows:	
Test Passed	

6.2.3.2 Configuration of SMTP connector

Purpose:	Test of the configuration of a SMTP (mail system) connector to a legacy system.
Set-up:	CI installed and running
Test procedure:	
Action	**Input**
Login into the CI (http://localhost:8080/LI/logoutUser.action)	
Go to Administration – Legacy connectors – SMTP connector	

Configure an inbound SMTP connector as follows	
Configure an outbound SMTP connector as follows	
Go to Administration – Routing configuration	
Configure out-bound routing as follows	
Configure in-bound routing as follows	
Go to Mapping – Mapping definition	
Configure the remote out-bound CI as follows	
Configure the remote in-bound CI as follows	
Test Passed	

6.2.3.3 Configuration of JMS connector

Purpose:	Test of the configuration of a JMS connector to a legacy system.
Set-up:	CI installed and running

Test procedure:

Action	Input
Login into the CI (http://localhost:8080/LI/logoutUser.action)	
Go to Administration – Legacy connectors – JMS connector	
Configure an inbound JMS connector as follows:	
Configure an outbound JMS connector as follows:	
Go to Administration – Routing configuration	
Configure out-bound routing as follows	
Configure in-bound routing as follows	
Go to Mapping – Mapping definition	
Configure the remote out-bound CI as follows	
Configure the remote in-bound CI as follows	

Test Passed

6.3 Connection tests

The connection test where executed with simulated legacy systems. The simulation test cases were written in java. For the simulation of the legacy

systems mock objects were created and the test cases have used these mock-objects to execute the tests on the external interfaces of the common interface. The tests were executed within the test framework NGUnit.

6.3.1   Connection test File legacy connector

Purpose:	- Test of the XML-message transport with a file system legacy connector via common interface to a file system legacy connector. - Test of the text-message (UIC 407) transport and transformation to a XML-TrainRunningMessage with a file system legacy connector via common interface to a file system legacy connector.
Set-up:	CI installed and running, configuration of file system connector (see chapter 6.2.3.1) successful The test data are provided as files TrainRunningExample_mod_519.xml and 2002_407.txt
Test procedure:	
Action	**Input**
Navigate to Eclipse	
Load the project "CI_Testing"	
Start the Testcase "Test_FS_CIWL_DB"	
Test Passed	

6.3.2   Connection test SMTP legacy connector

Purpose:	- Test of the text-message (UIC 407) transport and transformation to a XML-TrainRunningMessage with a SMTP legacy connector via common interface to a SMTP legacy connector.
Set-up:	CI installed and running, configuration of SMTP connector (see chapter 6.2.3.2) successful The test data are provided as file 2002_407.txt
Test procedure:	
Action	**Input**
Navigate to Eclipse	
Load the project "CI_Testing"	
Start the Testcase "Test_Mail"	
Test Passed	

6.3.3 Connection test JMS legacy connector

Purpose:	- Test of the XML-message transport with a JMS legacy connector via common interface to a JMS legacy connector.
Set-up:	CI installed and running, configuration of JMS connector (see chapter 6.2.3.3) successful The test data are provided as file TrainRunningExample_mod_519.xml

Test procedure:	
Action	**Input**
Navigate to Eclipse	
Load the project "CI_Testing"	
Start the Testcase "Test_JMS_Schenker_SNCB_XML"	
Test Passed	

Purpose:	- Test of the text-message (UIC 407) transport and transformation to a XML-TrainRunningMessage with a JMS legacy connector via common interface to a JMS legacy connector.
Set-up:	CI installed and running, configuration of JMS connector (see chapter 6.2.3.3) successful The test data are provided as file 2002_407.txt

Test procedure:	
Action	**Input**
Navigate to Eclipse	
Load the project "CI_Testing"	
Start the Testcase "Test_JMS_Schenker_SNCB_txt"	
Test Passed	

6.4 Load tests

The load tests were executed to test the behaviour of the system under load. For this purpose the configuration of the CI was reused. For the load tests the legacy connectors for JMS and for the file system were tested.

The load tests were executed in the following tools:

- Testing Anywhere

- Apache JMeter

6.4.1 Load test with Testing Anywhere

Purpose:	- PERMANENT test for smaller RU for train running messages via file system legacy connector.
Set-up:	CI installed and running, configuration of FS connector (see chapter 6.2.3.1) successful The test data are provided as file TrainRunningExample_mod_519.xml

Test procedure:

Action	Input
Start Testing Anywhere	
Use the following test code	
Start the Test	

Test Passed:
- Test start 18:53h (17 Dec 2012)
- Test end 09:00h (18 Dec 2012)
- Test total duration: 847 minutes

- Message type sent: TrainRunningMessage (TAF TSI) in XML format

- Total messages sent: 8406 messages (failed : 15 messages)
- Throughput : ≈ 9,92 messages/minute

6.4.2 Load test with Apache JMeter

The load test with Apache JMeter was dedicated to the tests with JMS-message transmission with the CI.

Apache JMeter was configured to send XML-messages via JMS to a specific queue for a legacy connector of the CI. The message was transmitted via the CI to a dedicated message queue on the receiver side of the CI. The message was picked-up by Apache JMeter. All messages were logged by Apache JMeter. These logfiles were used for the analysis.

Configuration of the message publisher:
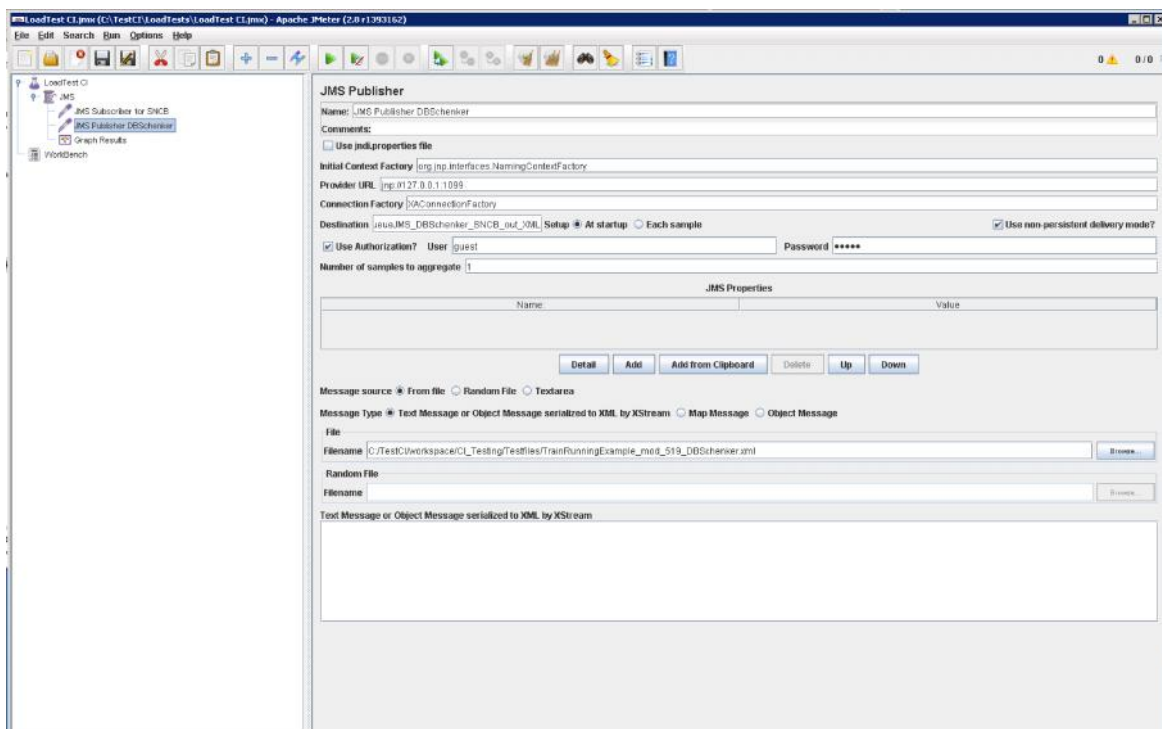
Configuration of the message receiver:
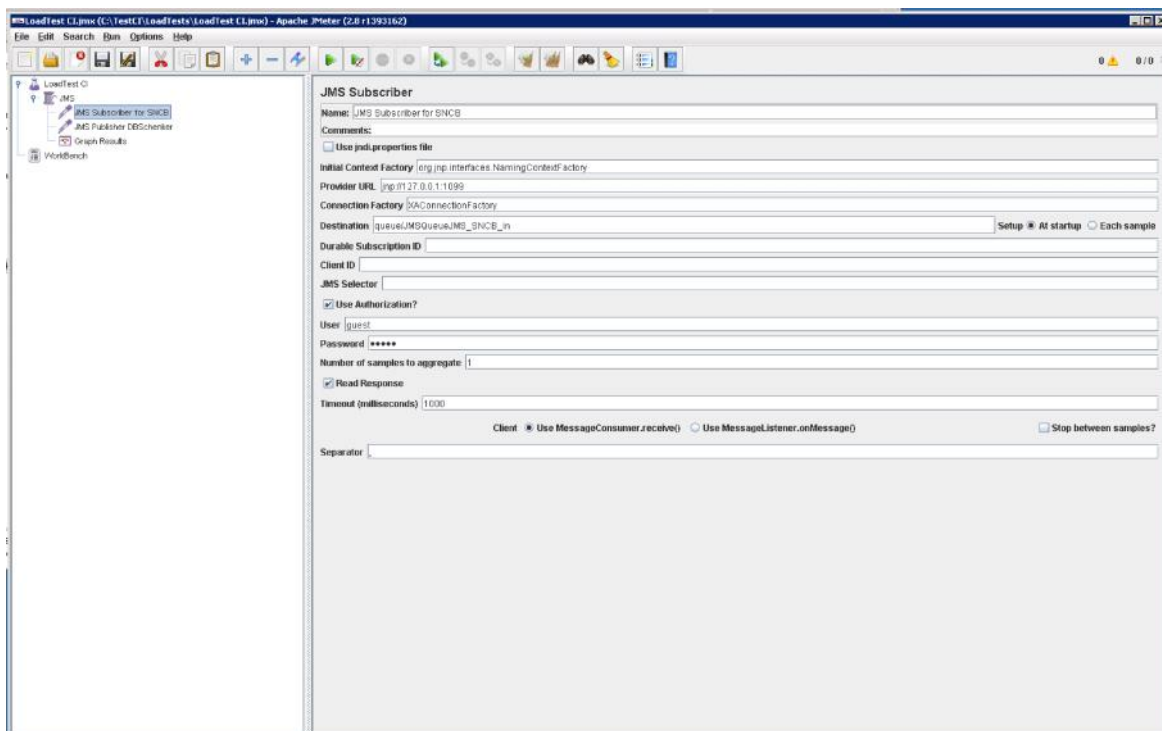
Purpose:	- PERMANENT test for smaller RU for train running messages via JMS legacy connector.
Set-up:	CI installed and running, configuration of JMS connector (see chapter 6.2.3.1) successful

	The test data are provided as file TrainRunningExample_mod_519.xml
Test procedure:	
Action	**Input**
Start Apache JMeter	
Use the following test code	See configuration above
Start the Test	

<div style="background-color:green">

Test Passed:

- Test start 17:28h (20 Dec 2012)
- Test end 20:39h (20 Dec 2012)
- Test total duration: 191 minutes

- 143354 messages transmitted (send and receive)
- 71677 messages sent
- Throughput: 6,26 sent messages/s
- 70800 messages received (878 errors)
- Throughput: 6,18 received messages/s

</div>

6.5 Test with TAP TSI commercial parameters

The purpose of this test scenario was to check the ability of the CI for the message exchange of messages for the TAP TSI commercial basic parameters. In the commercial basic parameters of the TAP TSI the exchange of messages related to timetable information, tariff information and reservation messages is specified. These commercial information's are available in different data formats: EDIFACT for timetable data exchange, text files for tariff information, XML-messages for reservation messages and bit messages for reservation messages.

The goal of this part of the tests is to configure the CI for a message exchange of those messages. This could be a future use case for the usage of the CI to be used for the exchange of TAP TSI commercial messages. The CI is currently not supporting EDIFACT messages or bit messages according to TAP TSI technical document B.5. Therefore the tests are made with XML-messages.

6.5.1 Configuration for TAP TSI B.10 AvailabilityRequest

Purpose:	The purpose of this test was the test of the CI if it is able to handle TAP TSI messages coming from the commercial basic parameters for PRM reservation.
Set-up:	CI installed and running, Message-type for Availability request modified with the TAF TSI MessageHeader-element

	XML-message type for the AvailabilityRequest as private and common (with TAF message header) message type available configuration of JMS connector successful The test data are provided as file B10Example.xml
Test procedure:	
Action	**Input**
Navigate to Eclipse	
Load the project "CI_Testing"	
Start the Testcase "Test_JMS_B10"	
Test Passed:	

6.5.2   Configuration for TAP TSI B.10 AvailabilityReply

Purpose:	The purpose of this test was the test of the CI if it is able to handle TAP TSI messages coming from the commercial basic parameters for PRM reservation.
Set-up:	CI installed and running, Message-type for Availability request modified with the TAF TSI MessageHeader-element XML-message type for the AvailabilityReply as private and common (with TAF message header) message type available configuration of JMS connector successful The test data are provided as file B10ExampleReply.xml
Test procedure:	
Action	**Input**
Navigate to Eclipse	
Load the project "CI_Testing"	
Start the Testcase "Test_JMS_B10Reply"	
Test Passed:	

7. RESULTS

7.1 Configuration tests

ERA has recognized that the configuration of the CI without support of the CCG helpdesk is in some cases difficult. The provided user manual [1] contains exhaustive information about the whole functionality of the CI. For the configuration of dedicated parts of the CI this manual is detailed enough.

ERA has detected only few parts of the provided documentation with not sufficient information:

- for the configuration of the connection to the CRD, the need of SSL-certificate was not explained explicitly
- the importance of function "Reloading cache" in case of configuration was not explained. It has been recognized by ERA, that the change of the configuration will be applied for a running CI, if the appropriate parameters are reloaded in the cache.

7.2 Connection tests

It could be demonstrated that the CI was able to connect to the given legacy connectors. The CI was able to handle the transmitted messages including the mapping between the UIC 407-format and the TAF TSI compliant XML-message.

7.3 Load tests

It could be demonstrated that the CI was able to handle a huge amount of messages with the file system connector and the JMS-legacy connector. It could be shown, that the JMS-based connector is able to handle much more messages than the file system based legacy connector.

The analysis of the data of the tests with the JMeter and the JMS-connector has shown the difference of the processing times for the sending and receiving of the messages by the CI. The load of the messages processed by the CI could be assumed as constant, where the overall picture of the processed messages has shown a huge dispersion of the processing times. The reasons (e.g. garbage collection of the JVM, configuration of HornetQ, load of the test system) for this behaviour have not been analysed in detail.

During the tests no exhaustive or increasing memory consumption of the CI could be discovered.

Example of load test results:
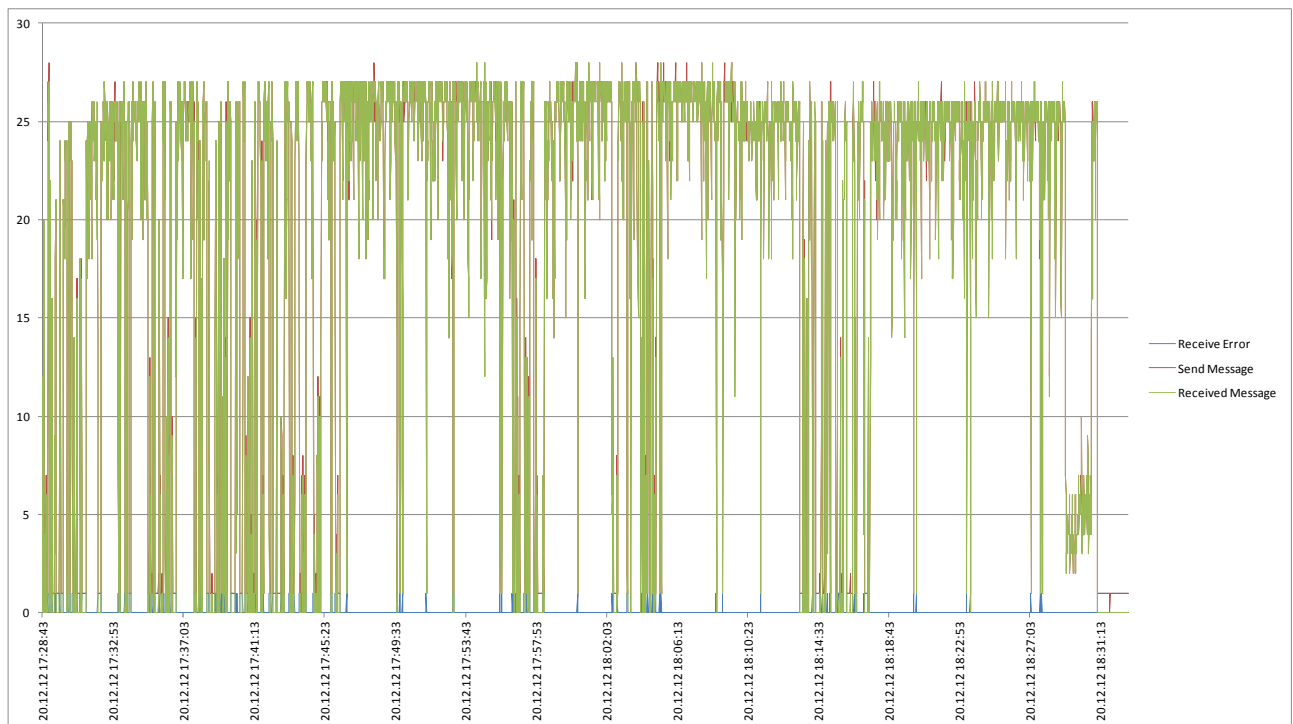
X-axis: time
Y-axis: message count

7.4   **Test of exchange of TAP commercial basic parameters**

It could be shown, that the exchange of TAP TSI XML reservation messages can be made with the common interface. On the basis of the AvailabiltyRequest/Reply according to TAP TSI technical document B.10 it could be shown, that those messages can be exchanged. Precondition for the message exchange is nevertheless the creation of a common message type in line with the requirements of the CI, especially with a TAF TSI compliant message header. The message header of the TAP TSI technical document B.10 is currently defined different and therefore an interoperable message exchange with the CI for those messages is not possible.

8.   CONCLUSIONS

8.1   GUI configuration

During the tests ERA has got a depth insight in the usage of the CI. It was possible to configure the CI for all needed test cases. However the support from the CCG was needed to configure especially the message exchange for the commercial parameters.

For the configuration of the CI ERA have the following remarks:

1. The configuration has to be simplified

    ERA had some difficulties to configure the CI (e.g. legacy connectors, remote LI, mapping). With the provided user documentation of the CI it could not be configured successfully in all cases. With the help of the CCG helpdesk or the CCG project manager for the CI it was possible to configure it finally.

    ERA would recommend to improve the usability of this tool by a guided configuration. If the tool is more used by European IM's and RU's the configuration should be simplified (or alternatively better documented) to support those users with few experience of the configuration of such a tool. ERA proposes to assist the user in the following use cases:

    a. Configuration of the connection to the CRD

    b. Configuration of whole process for outgoing messages, from the definition of the legacy system, routing, mapping until the configuration of the remote LI.

    c. Configuration of whole process for incoming messages, from the definition of the remote LI, of routing and mapping until the configuration of the legacy system.

2. ERA propose to CCG to assist the user with the following measures for the configuration of the CI:

    a. Provision of step-by-step guidelines as help files/documentation

        This documentation shall show the needed steps for the configuration for the use cases above. These step-by-step-guides should show the GUI, the most important parameters and the dependencies to the other parts of the CI-configuration (e.g. message definition, routing).

    b. configuration support with an assistant

A more sophisticated approach to support the users for the configuration could be the provision of a configuration procedure as assistant guided configuration. The assistant should use the already existing dialogs for the configuration of the CI (e.g. message type, legacy connector, remote LI). These dialogs should be linked by the assistant with a transfer of the needed parameters for the following dialog from the previous ones. A logical check of the provided parameters for each step and at the end of the guided configuration could be also helpful.

8.2 Connection tests

For the connection tests ERA has implemented own test procedures in Java. It can be assumed that these procedures will be implemented similar for productive use of the CI.

During the implementation of the test procedures ERA has discovered the problem to detect errors during sending/receiving of messages from legacy systems. The errors are stored by the CI in logfiles, dedicated message queues (e.g. for redelivery) or communicated to the CI operator by email or SNMP.

For ERA it was not clear how legacy systems can implement error detection if there are errors during the sending of messages. It is especially important for the implementation of JMS-based legacy connectors. In such a case the legacy systems cannot react on errors occurred during sending, e.g. via error handling in the legacy system. During a discussion with the CCG about the possible solution for this problem were proposed:

1. Using the SNMP-protocol to transfer error messages from the CI to the legacy system. The legacy system has to implement an error handling for the SNMP messages received from the CI. These messages have to be used for detection of errors during the CI-message processing (e.g. XML-errors).

2. The SMTP functions can be used to inform the responsible IT-staff about errors during the message processing in the CI.

These both proposals can be used for the implementation of the legacy systems.

9. LIST OF ANNEXES

Following documents are annexed to this recommendation:

Table 3 : Annexes to this report

N°	Reference	Title	Version	Date
1	JV10025_LI_User _Manual_v1.1.doc	Common Component Group LI-User Manual	1.1	27.04.2012
2	TestCode_CI.zip	Testcode for the CI		18.02.2013

12. ANNEX

Table 4 : submitted tickets for the CI-bugtracking

N°	Ticket #	Title	Date
1	2013021810000018	XSD-files with more than 1 points in the file name (e.g. yyy.1.2.xsd) are not accepted	18.02.2013
2	2013021810000027	XML-Namespace duplication	18.02.2013