



EUROPEAN COMMISSION
DIRECTORATE-GENERAL FOR MOBILITY AND TRANSPORT
DIRECTORATE-GENERAL FOR ENERGY
SRD - Shared Resource Directorate
SRD.4 - Informatics & Logistics

TACHOnet

XML Message Reference

Schema Version 3.2

Date:	06/04/2018
Version:	3.6
Authors:	Martin Gardiner
Revised by:	Luisa Guedes
Approved by:	Bernardo Martinez
Public:	Limited

1. TABLE OF CONTENTS

Contents

1. TABLE OF CONTENTS	2
1.1. Table of Figures	5
1.2. Document History	6
1.3. Glossary	7
2. BIBLIOGRAPHY	9
3. CHANGES FROM PREVIOUS DOCUMENT VERSIONS	10
3.1. Version 3.0 to 3.1	10
3.2. Version 3.1 to 3.2	10
3.3. Version 3.2 to 3.3	11
3.4. Version 3.3 to 3.4	12
3.5. Version 3.4 to 3.5	12
3.6. Version 3.5 to 3.6	13
3.7. Known Issues	13
4. INTRODUCTION	14
4.1. TCN 1.5 vs. 3.0	14
4.2. Breaking Changes – 1.5 Messages	14
4.2.1. Card Number Length.....	14
4.2.2. Batches	15
4.2.3. Dates	15
4.2.4. Response Status Codes.....	15
4.2.5. Driver Card Status Codes.....	15
4.2.6. Driving Licence Country.....	15
4.3. Breaking Changes - 3.0 Messages.....	15
4.3.1. Common Header and Message Format	15
4.3.2. Message Names.....	16
4.3.3. sentAt Validation	16
4.3.4. Reason Codes	16
4.3.5. Transliteration, Phonex and NYSIIS.....	16
4.3.6. Modify Card Status Workflow	16
4.3.7. TCN_ Receipt Messages	17
4.3.8. Web Services.....	17
4.4. Non-Breaking Changes	17
4.4.1. Batches	17
4.4.2. Synchronous Connections	17
4.4.3. Multiple Points of Contact	17
4.4.4. Heartbeat	18
4.5. Side-by-Deployment	18
5. TCN GLOBAL ARCHITECTURE	19
5.1. Centralised Architecture.....	19
5.2. XML Messaging Framework	20
5.3. Asynchronous and Synchronous Message Patterns	21
5.3.1. Asynchronous End-Points	21
5.3.2. Synchronous End-Points	21

5.4. Multiple Points of Contact	28
5.5. Single Message and Batch Message Modes	28
5.5.1. Single Message Mode	28
5.5.2. Batch Message Mode	29
5.6. Message Structure	31
5.7. Standardisation of XML Messages	32
5.7.1. Schema Version	32
5.7.2. Encoding	32
5.7.3. Namespaces.....	32
5.7.4. Optional Attributes.....	32
5.7.5. Message Header	32
5.7.6. Message Body	33
5.7.7. Identifier Definitions in XML Messages.....	34
5.8. Message Timeout	36
5.8.1. Asynchronous Timeout	36
5.8.2. Synchronous Timeout	37
5.8.3. Batch Timeout.....	37
6. MESSAGE VALIDATION	38
6.1. ISO Country Codes	38
6.2. Status Codes and Status Messages	39
6.2.1. CCS and CIC Responses	39
6.2.2. ICDL Responses.....	39
6.2.3. MCS Responses	39
6.2.4. MCS Acknowledgements.....	40
6.2.5. Invalid Responses.....	40
6.2.6. Status Message.....	40
6.3. Date and Time	40
6.3.1. Date	41
6.3.2. Datetime	41
6.4. Card Status	41
6.4.1. Application and Rejected Statuses	42
6.5. Card Status Transition.....	43
6.5.1. Standard Issue	44
6.5.2. Card Renewal / Replacement	45
6.5.3. Card Suspension / Cancellation.....	46
6.5.4. Card Exchange	47
6.6. Hub Business Validation.....	47
6.6.1. Card Status Card Dates	47
6.6.2. Name String and Search Keys.....	47
6.6.3. Search Status and Content.....	48
7. TCN SERVICES	49
7.1. Messaging Services	49
7.2. Check Issued Card (CIC)	50
7.2.1. Search Mechanism	50
7.2.2. Heartbeat	54
7.2.3. CIC Version Mappings	55
7.2.4. CIC Request Message Sample	61

7.2.5. CIC_Response MessageSample	62
7.3. Issued Card Driving Licence (ICDL)	63
7.3.1. Notifying Member State	64
7.3.2. Responding Member State	64
7.3.3. ICDL Version Mappings	66
7.3.4. ICDL_Request Message Sample	70
7.3.5. ICDL_Response Message Sample	70
7.4. Check Card Status (CCS)	71
7.4.1. CCS Version Mappings	72
7.4.2. CCS_Request Message Sample	78
7.4.3. CCS_Response Message Sample	79
7.5. Modify Card Status (MCS)	80
7.5.1. Automatic Register Updates	80
7.5.2. MCS Message Flow	80
7.5.3. Synchronous and Asynchronous MCS Messages	81
7.5.4. MCS Conversion Mappings	81
7.5.5. MCS_Request Message Sample	89
7.5.6. MCS_Response Message Sample	90
7.5.7. MCS Request Acknowledgement Sample	90
7.5.8. MCS Response Acknowledgement Sample	91
7.6. Error Notification (EN)	91
7.6.1. Error workflowId	91
7.7. EN Message Sample	96

1.1. Table of Figures

Figure 1 - TACHOnet Architecture.....	19
Figure 2 - Standard Message Flow.....	20
Figure 3 – Request-Response SOAP Message Request Example.....	22
Figure 4 – Request-Response SOAP Message Response Example.....	23
Figure 5 – Request-Response SOAP Message Fault Example.....	24
Figure 6 – One-Way SOAP Message Request Example.....	26
Figure 7 – One-Way SOAP Message Response Example.....	26
Figure 8 - Singlecast Message Example.....	29
Figure 9 - Batch Message Flow.....	29
Figure 10 – Singlecast Batch Message Example.....	30
Figure 11 - Broadcast Batch Message Example.....	30
Figure 12 - Message Identifiers.....	34
Figure 13 - Message Timeouts.....	36
Figure 14 - Card Issue Process.....	44
Figure 15 - Card Renewal / Replacement Process.....	45
Figure 16 - Card Suspension / Cancellation Process.....	46
Figure 17 - Card Surrender Process.....	46
Figure 18 - Card Exchange Process.....	47
Figure 19 - Name String / Search Key Example.....	48
Figure 20 - Search Response with Rejected Card Status.....	51
Figure 21 - Search Response with Expired Card Status.....	52
Figure 22 - Search Response with Two Cards.....	53
Figure 23 - Heartbeat Message Example.....	54
Figure 24 - CIC HTTP Request Message Example.....	61
Figure 25 - CIC SOAP Request Message Example.....	61
Figure 26 – CIC HTTP Response Message Example.....	62
Figure 27 - CIC SOAP Response Message Example.....	63
Figure 28 - ICDL Message Usage.....	65
Figure 29 – ICDL HTTP Request Message Example.....	70
Figure 30 – ICDL SOAP Request Message Example.....	70
Figure 31 - ICDL HTTP Response Message Example.....	70
Figure 32 - ICDL SOAP Response Message Example.....	71
Figure 33 - CCS HTTP Request Message Example.....	78
Figure 34 - CCS SOAP Request Message Example.....	78
Figure 35 - CCS HTTP Response Message Example.....	79
Figure 36 - CCS SOAP Response Message Example.....	80

Figure 37 - MCS Message Flow.....	81
Figure 38 - MCS Message Flow 1.5 to 3.0 Version Mapping.....	82
Figure 39 - MCS Message Flow 3.0 to 1.5 Version Mapping.....	85
Figure 40 - MCS HTTP Request Message Example.....	89
Figure 41 - MCS SOAP Request Message Example.....	89
Figure 42 – MCS HTTP Response Message Example.....	90
Figure 43 - MCS SOAP Response Message Example	90
Figure 44 - MCS HTTP Request Acknowledgement Message Example.....	90
Figure 45 - MCS SOAP Request Acknowledgement Message Example.....	91
Figure 46 – MCS HTTP Response Acknowledgement Message Example.....	91
Figure 47 – MCS SOAP Response Acknowledgement Message Example	91
Figure 48 - Error in Request.....	92
Figure 49 - Error in Forwarded Request.....	93
Figure 50 - Error in Response	94
Figure 51 - Error in Forwarded Response	95
Figure 52 - Error Notification HTTP Message Example	96
Figure 53 - Error Notification SOAP Message Example	96

1.2. Document History

Version	Date	Comment
3.0	23/05/2016	Document published
3.1	14/06/2016	Minor spelling corrections and clarifications. XSD 3.1 changes.
3.2	20/06/2016	Internal review.
3.3	03/08/2016	Updated following the June 2016 technical workshops.
3.4	01/02/2017	Updated following the January 2017 technical workshops.
3.5		

Contact:

EC-MOVEHUB-PROJECT@ec.europa.eu (for pre-production questions and issues)
EC-MOVEHUB-TESTING@ec.europa.eu (for testing requests and execution questions)
EC-MOVEHUB-HELPDESK@ec.europa.eu (for production questions and issues)

1.3. Glossary

This table is a list of terms used when referring to the MOVEHUB applications and is used in all MOVEHUB documentation. As such it will contain terms that are not pertinent to this document.

AETR	The European Agreement Concerning the Work of Crews of Vehicles Engaged in International Road Transport
CAB	Change Advisory Board
CCB	Change Control Board
CCL	Check Community Licence The message type exchanged by ERRU to determine the status of a road transport undertaking's Community Licence.
CCS	Check Card Status The message type exchanged by TCN to retrieve details of a specified driver card from another member state.
CGR	Check Good Repute The message type exchanged by ERRU to determine the repute of a transport manager.
CIA	Card Issuing Authority The body responsible for issuing driver cards within a member state.
CIC	Check Issued Cards The message type exchanged by TCN to search for driver cards by name and surname in other member states.
CPC	Certificate of Professional Competency Managers of a road transport undertaking must hold a certificate of professional competence (and be of good repute).
DC	Driver card The tachograph driver card.
DIGIT	Directorate General for Informatics
DL	Driving Licence
DLIA	Driving Licence Authority The body responsible for issuing driving licences within a member state.
EC	European Commission
ENER	Directorate General for Energy
ERRU	European Register of Road Transport Undertakings A network interconnecting the national registers of road transport managers.
EU	European Union
Hub	The central Hub that is hosted at the EC and routes messages between MS.
ICDL	Issued Card Driving Licence The message type exchanged by TCN to inform a driving licence issuing member state that a driver card has been issued against that licence in another member state.

INF	<p>Infringement Notification</p> <p>The message type exchanged by ERRU to inform the member state of establishment that a haulier has committed an infringement in another member state.</p>
KPI	Key Performance Indicator
MCS	<p>Modify Card Status</p> <p>The message type exchanged by TCN to send a driver card status modification to another member state.</p>
MOVE	Directorate General for Mobility and Transport
MOVEHUB	The generic term used to refer to the applications managed by MOVE (ERRU, RESPER, TCN).
MS	<p>Member State</p> <p>Within this document the abbreviation MS refers to all connected countries and not just the 28 European Union member states.</p>
NCP	<p>National Contact Point</p> <p>The national authority to whom all TCN requests via phone / email / post can be directed.</p>
PKI	<p>Public Key Infrastructure</p> <p>The provision and management of digital certificates identification and encryption.</p>
RESPER	<p>Réseau de Permis de Conduire</p> <p>A network interconnecting the national driving licence registers.</p>
RSI	Road Side Inspection
SLA	Service Level Agreement
SPOC	<p>Single Point of Contact</p> <p>The technical connection between the TCN central Hub & MS systems.</p>
SRD	<p>Shared Resource Directorate</p> <p>At the European Commission, a Directorate responsible for providing horizontal services to more than one DG or Directorate-General, including IT and supporting services, in this case to MOVE and ENER</p>
TESTA	<p>Trans European Services for Telematics between Administrations</p> <p>The EU's private network interconnecting all EU Institutions, EU agencies, Member States' administrations and European Economic Area (EEA) countries.</p>
TCN	The abbreviation for TACHOnet and used in this document to mean the TACHOnet network encompassing the HUB and the MS applications.
XSD	XML Schema Definition

2. BIBLIOGRAPHY

- [1] [2016/68] *Commission Implementing Regulation (EU) 2016/68 of 21 January 2016 on common procedures and specifications necessary for the interconnection of electronic registers of driver cards*, 2016.
- [2] [165/2014] *Regulation (EU) No 165/2014 of the European Parliament and of the Council of 4 February 2014 on tachographs in road transport, repealing Council Regulation (EEC) No 3821/85 on recording equipment in road transport and amending Regulation (EC) No 561/2006*, 2014.
- [3] [2017/1503] *COMMISSION IMPLEMENTING REGULATION (EU) 2017/1503 of 25 August 2017 amending Implementing Regulation (EU) 2016/68 on common procedures and specifications necessary for the interconnection of electronic registers of driver cards (Text with EEA relevance)*.
- [4] [2135/98] *COUNCIL REGULATION (EC) No 2135/98 of 24 September 1998 amending Regulation (EEC) No 3821/85 on recording equipment in road transport and Directive 88/599/EEC concerning the application of Regulations (EEC) No 3820/84 and (EEC) No 3821/85*, 1998.
- [5] [TCN/Final/2015] *Tachonet Study 2014 Final Report 1.03*, 26/02/2015, 2015.
- [6] [TCN/Proposal/2015] *Tachonet 3.0 Technical Proposal v1.03*, 2015.
- [7] [TCN/XML/1.50] *Tachonet XML Messaging Reference Guide v1.50*, 2010.
- [8] [TCN/XSD/3.0] *TACHOnet XSD Message Reference v3.0*, 2016.
- [9] [MOVEHUB/Network] *MOVEHUB Networking Guide 2.0*, 2017.
- [10] [TCN/XSD-HTML/2016] *TCN 3.0 XSD Technical Specifications (HTML)*, 2016.
- [11] [TCN/XSD-Word/2016] *TCN 3.0 XSD Technical Specifications (Word)*, 2016.

3. CHANGES FROM PREVIOUS DOCUMENT VERSIONS

3.1. Version 3.0 to 3.1

Section	Change Description
3	Add this section. All subsequent section numbers increase by 1.
All	Harmonize the page footers.
Multiple	Correct spelling mistakes.
7.2.5	Add <i>searchMethod</i> attribute to the message example.
0	Add <i>driverCardNumber</i> and <i>drivingLicenceNumber</i> attributes to the message example.
7.5.2	Remove the <i>otherInformation</i> attribute from the message example.
0	0
0	Modify the <i>Body</i> element to contain the nested <i>CardDetails</i> element.

3.2. Version 3.1 to 3.2

Section	Change Description
All	Redraft ambiguous sentences / paragraphs.
5.5.2	Explanation of MCS batches. Batches are not permitted via synchronous protocols.
5.7.7.3	<i>workflowID</i> is unique across all member states / requests.
5.8	Improve description of timeout calculation.
6.2	NotYetConnected status removed.
6.4	The card status table and descriptions have been updated to match the legislation [2016/68].

3.3. Version 3.2 to 3.3

Section	Change Description
All	Correct spelling mistakes and grammatical errors.
All	Added diagrams & schematics.
4.2	A correlation UUID is no longer returned with the HTTP 202 response.
5.2	Retry on send failure is removed.
5.7.4	Handling of optional attributes updated to match the XSD implementation.
5.8.3	New section – batch timeout description.
6.2	Status codes listed by message type.
6.4.1	New section – null card number.
6.6	New section – Hub validation.
7.2.1	Responding MS will return the latest driver card that was found in the register irrespective of its actual status.
7.2.1	Current and future cards must be returned in the search response.
7.2.3	New section – v1.5 to v3.0 & v3.0 to v1.5 CIC conversion mappings.
7.3.3	New section – v1.5 to v3.0 & v3.0 to v1.5 ICDL conversion mappings.
7.3.47.3.4	Added the person details & search keys to the ICDL request example.
1.1.1	New section – v1.5 to v3.0 & v3.0 to v1.5 CCS conversion mappings.
7.5.1 & 0	New sections – processing for automatic and ‘manual’ MCS register updates.
7.5.3	New section – v1.5 to v3.0 & v3.0 to v1.5 MCS conversion mappings.
7.6.1	New section – missing <i>workflowId</i> .
7.6	Added error flow diagrams.

3.4. Version 3.3 to 3.4

Section	Change Description
All	Correct spelling mistakes and grammatical errors.
4.1	Redrafted the section to emphasise that TCN 3.0 provides a mapping between the versions.
4.2	New section – breaking changes for 1.5 messages.
5.3.1	Asynchronous URLs were corrected.
5.8	Errors in the description of the timeout calculation were corrected.
6.2.5	The description of the Not Available status was corrected.
6.4.1	Start of validity and expiry dates should be 1900-01-01 when the card status is Application or Rejected.
6.5.1	The description of the card issuing status transition has been amended to remove the reference to card number being optional in the response (it is mandatory, see 6.4.1).
6.5.3	The diagram for the Suspended status transition was corrected (Suspended is not a final state).
6.6.2	Diagram updated to use v3.1 messages (not 1.5).
7.2.1	Improved the description of how the results that a search should return and added examples.
7.3.2	Explanation of recording and using ICDL messages updated.
(7.5.2)	Reviewed Register Updates – section deleted; the ‘update pending’ flag was dropped from the proposal.
7.5.3	New section – clarification of synchronous MCS message exchange.

3.5. Version 3.4 to 3.5

Section	Change Description
Various	Miscellaneous spelling and grammar corrections.
(4.4.1) & (7.2.2)	Following consultation with the Data Protection Office these sections concerning CIC wildcard searches have been removed. Wildcard search functionality is not to be implemented.
5.3.2	Updated for typed synchronous messages.
5.8	Message timeout diagram added.
5.8.1 & 5.8.2	A business message is returned with a Timeout status – not an Error Notification.
5.3.1	Removed the incorrect information concerning the asynchronous messaging URL – message type does not form part of the URL.
7.2.1	Updated to explicitly state that wildcard searches are not supported.

3.6. Version 3.5 to 3.6

Section	Change Description
Various	Change yes / no values to true / false in example messages.
5.3	Add a reference to the MOVEHUB Network Guide.
5.3.2	Synchronous URLs corrected.
5.7.6	Updated to clarify the use of requestPurpose and requestSource.
6.2	Timeout and NotAvailable status codes are not valid on responses from a MS to the Hub.
6.2.2	Updated the NotOK status description.
7	Added SOAP examples.
7.2	Corrected Figure 20 and Figure 21 to show CIC response examples (instead of CCS).
7.5.2	Corrected the maximum period of time for the MCS response to be sent. Changed from 7 calendar days to 10 calendar days.

3.7. Known Issues

None.

4. INTRODUCTION

The TCN network exists to enable MS to fulfil the obligations laid down in [165/2014] which provide for the electronic exchange of tachograph card registration data and in [2016/68] which “makes mandatory the interconnection of electronic national registers on driver cards throughout the Union”.

The information in this document provides an explanation of the requirements laid down in the implementing act [2016/68] as amended by [2017/1503].

The version of TCN currently in use by the MS (1.5) was originally implemented as an optional application, available to MS to assist with their obligations under the previous legislation [2135/98], but not mandatory to use. The implementation and use of TCN in the member states has varied widely and so the users of the network are not guaranteed that a counterparty MS uses the application in a consistent or expected manner. With the knowledge that TCN use would become mandatory for all MS, the Commission undertook a study in the latter months of 2014 to discover the usage patterns and issues that MS encountered. The results of this study were published in the TCN Study Report [TCN/Final/2015].

This study led to a number of proposals for the improvement of the TCN network such that all MS (and the Commission) can rely on the implementation and messages being exchanged and used in a consistent manner. These proposals were laid out in the TCN Technical Proposal [TCN/Proposal/2015].

This version of the TCN XML Messaging Reference Guide replaces the TCN 1.5 XML Guide [TCN/XML/1.50] and implements the proposals laid out in the Technical Proposal [TCN/Proposal/2015].

This document should be read in conjunction with the TCN XSD reference [TCN/XSD/3.0] which is published alongside this XML guide.

4.1. TCN 1.5 vs. 3.0

The move from TCN 1.5 to TCN 3.0 is a breaking upgrade; neither will the 1.5 messages work with a 3.0 member state application, nor will the workflow functionality of the Modify Card Status transfer from 1.5 to 3.0. The central hub will map messages from one version to another, but all downstream applications within the MS will need to be upgraded to handle 3.0 messages.

The major functional and message format changes are outlined below. These are also described elsewhere in the body of this document, but they are grouped together here for ease of reference.

4.2. Breaking Changes – 1.5 Messages

The requirements of the 3.0 messages have necessitated some minor functional changes to the existing 1.5 messages in order to make a successful mapping between the two versions of the application.

These changes should not be noticeable during normal operation but will, nonetheless, in some circumstances cause messages to be rejected where previously they were not.

4.2.1. *Card Number Length*

The TCN 1.5 documentation [TCN/XML/1.50] defines card number as alphanumeric length 16, but the XSD does not enforce this. Consequently TCN 1.5 accepts and processes messages with card numbers that are longer and shorter than 16 characters. In order to map between 1.5 and 3.0 messages TCN will enforce length 16 card numbers for 1.5 messages:

- For card numbers <16 the Hub will pad the number with leading zeros and forward the message
- For card numbers >16 the Hub will reject the message

4.2.2. Batches

Currently batches are processed by the hub as they are received. TCN 3.0 will queue batches until late evening and then process them sequentially in order of arrival. Once TCN 3.0 is deployed to production the behaviour of batches will change for member states still using TCN 1.5. The hub will ignore the timeout value specified by the requestor and queue the batches until the following night. They will then be processed overnight and the responses returned.

4.2.3. Dates

In various places TCN 1.5 allows dates to be truncated to just the year (YYYY) or the year and month (YYYY-MM). TCN 3.0 requires valid ISO 8601 dates, see 6.3.1. Additionally, dates with zeros in the day or the month are rejected.

4.2.4. Response Status Codes

The status codes Timeout and NotAvailable are not allowed for member state responses. Only the hub can send those status codes. If a response is sent to the hub with these values in the status code the hub will respond with a TCN_receipt message.

The status code is also validated against the data content of the response – if the status code is ‘Found’ then there must be search results; and conversely, there must be no found data if the search status is ‘NotFound’.

Finally, the hub will no longer contain ‘not yet connected’ member states and so this status code will no longer appear in responses from the hub.

4.2.5. Driver Card Status Codes

The driver card status code is validated against the card status:

- If the card status is HandedOver then the start date must be <= today and the expiry date must be >= today.
- If the card status is Expired then the expiry date must be < today.

4.2.6. Driving Licence Country

The driving licence issuing country attribute must contain a valid UNECE code. Additionally for ICDL messages, the driving licence issuing country attribute must match the ‘to’ attribute in the MS2TCN_IssuedCardDL_Req header.

4.3. Breaking Changes - 3.0 Messages

The previous section detailed the changes to functionality that member states using TCN 1.5 messages will notice when MOVE deploys the TCN 3.0 hub to production. This section details the breaking changes between TCN 1.5 and TCN 3.0 applications.

4.3.1. Common Header and Message Format

The message format has been changed so that all messages have the same generic structure:

<Message Name>

 <Header>

 <Body>

Further, TCN 1.5 messages mixed routing data and business data, e.g. for the ICDL notifications it was necessary to use the *DrivingLicenceIssuingNation* in the body of the message to determine the notification destination. All technical and routing data is now encapsulated in the *Header* element.

The format of the Body element (the business data content) has been modified so that attributes in different messages that contain the same data are named in the same way. Also, the message format has been modified so that the same data structures are reused between message types.

The new Header format is described in 5.7.1 below.

4.3.2. Message Names

The names of the messages have been altered to remove the previous mix of fully named terms and abbreviated terms. More significantly, the direction component of the message name (MS2TCN and TCN2MS) has been removed. There is now a single message format for Request and Response for each workflow; the same message format is used for both the MS2TCN request and the forwarded TCN2MS request, and similarly for response messages there is a single format for both the MS2TCN message and the TCN2MS message.

See 5.3 below for an explanation of the TCN message types.

4.3.3. sentAt Validation

The *Header* element contains a *sentAt* attribute which is populated by the sender with the date and time when the message is sent. TCN will not perform any validation on this attribute data, all messages will be processed as they are received regardless of the value in the *sentAt* attribute. Nonetheless, the accuracy of this attribute is important when troubleshooting connectivity or performance issues between the MS and the Hub.

See 5.7.4 below for an explanation of the message header element.

4.3.4. Reason Codes

TCN 3.0 adds two new attributes to the Body element of the request messages. The attributes will enable the Hub to collect statistics as to how and why the MS use the TCN network. This data can then be shared with the MS to provide a greater understanding of TCN usage and possible opportunities for improvement.

See 5.7.6 below for an explanation of the message body element.

4.3.5. Transliteration, Phonex and NYSIIS

TCN uses a phonetic algorithm to provide keys with which to search for names in a MS tachograph card register. The reason for this is to allow for and mitigate the differences in name spellings between languages. TCN 1.5 uses the Phonex algorithm, however it has been shown that this algorithm is not only slower than some of the alternatives, but that it produces a greater number of false positives.

For TCN 3.0 the Phonex algorithm will be replaced by the NYSIIS algorithm (as used in the other MOVEHUB applications). This will require that MS re-encode the names in their tachograph card register.

Additionally, prior to running the NYSIIS algorithm, the name string is converted to ASCII by 'transliterating' the Latin, Greek or Cyrillic name string. Each UTF-8 character in the name string is converted to an ASCII representation by way of a mapping table. This table has been significantly extended in comparison to that used for TACHOnet 1.5.

See 7.2.1 below for an explanation of the search by name mechanism in the CIC service.

4.3.6. Modify Card Status Workflow

The MCS workflow has caused a lot of confusion for MS and, indeed, a number of MS have not implemented it at all. The primary cause of problems is that this message type requires a CIA to update their card register and to return an immediate response. However, a number of MS are unwilling to automate the process of updating their register and so are unable to return a valid or sensible immediate response.

This workflow has been modified to decouple the notification and associated response and thereby giving the CIA time to carry out any required validation procedures before updating the card register.

See 0 below for an explanation of the MCS service.

4.3.7. TCN Receipt Messages

TCN 1.5 used TCN_Receipt messages to notify a MS of a technical or business failure when that MS either submitted an XML message that was not well formatted or when the that MS was not expecting a response message. Additionally, TCN 1.5 only provided for TCN_Receipt messages to be sent from the Hub to the MS.

TCN 3.0 deprecates the TCN_Receipt message and provides in its place a new message type and a new pattern. The Error_Notification message may be used by both the Hub and the MS to notify the other of any error.

See 7.6 below for an explanation of this message.

4.3.8. Web Services

The Commission currently offers transliteration and Phonex web services (hosted at <https://webgate.ec.testa.eu/move-Hub/tcnws/>). These services receive very little traffic as the MS are not required to populate search requests with the phonetic keys (this is done at the Hub), and using the web service to encode the names in the tachograph card register is likely to be too slow to be practicable. Additionally, the Commission provides a web front end for ad hoc transliteration and phonetic key queries (<https://webgate.ec.testa.eu/move-Hub-apps/CWSConsumer.aspx>).

Therefore, these web services will no longer be offered for TCN 3.0.

The Commission will continue to distribute Java and .Net libraries that MS may incorporate into their own applications.

4.4. Non-Breaking Changes

4.4.1. Batches

Batches are retained in TCN 3.0 but the Hub will ‘de-batch’ any requests that it receives and forward the requests individually to the counterparty MS. This means that MS who neither wish nor need to use batches will not be required to implement any processes to handle the receipt of a batched request.

See 5.5 for an explanation of batches.

4.4.2. Synchronous Connections

TCN 3.0 will introduce the possibility to use synchronous connections between the MS and Hub. Using a synchronous connection a MS can send a request to the Hub and receive the response on the same connection.

Similarly, if a MS uses synchronous protocols within their own environment and they wish the Commission to connect with them (i.e. to send message to the MS) via a synchronous connection, then this will be possible.

The Hub will initially expose a standard SOAP connection. If MS have other connectivity requirements then the EC will, without obligation, examine the possibility of introducing additional protocols.

See 5.3 below for an explanation of the connection options.

4.4.3. Multiple Points of Contact

Using asynchronous connections there is a SPOC to which the Hub forwards all requests and returns all responses. Many MS have requested that there be multiple points of contact (MPOC) within the MS so allowing different administrations to connect to the network directly.

TCN 3.0 provides this capability, but only via synchronous connections to the Hub. There must be a SPOC to which the Hub will forward requests from other MS; however, any request received from a MS over a synchronous connection will use that connection to return the response. Hence, MS may set up connections to TCN from multiple locations.

See 5.4 for an explanation of MPOC.

4.4.4. Heartbeat

TCN 3.0 will use dummy CIC searches to measure the connectivity status of MSs. This will be used to calculate availability statistics as well as the current method of calculating the difference between the number of requests sent and responses received.

See 7.2.2 below for a description of the heartbeat service.

4.5. Side-by-Deployment

It is not feasible for all MS to upgrade their application at the same time. Therefore, the Hub will provide the functionality to map and convert 1.5 messages and workflows to 3.0 (and vice versa) such that MS may upgrade their applications according to their own timetable (subject to [2016/68]) without regard to the status of other MS.

5. TCN GLOBAL ARCHITECTURE

TCN is built on a Hub and spoke architectural model and the network comprises the TCN Central Hub (under the responsibility of the EC) and the MS systems (acting as the spokes under the responsibility of the respective MS). In this model the MS is the owner of their own tachograph register and the EC Hub facilitates the exchange of data between the MS.

The Hub is the single interface for the exchange of requests and responses for the TCN services.

5.1. Centralised Architecture

The Hub application brokers the exchange of messages between the MS. The protocols and points of contact may differ between MS, but all messages are routed through the Hub.

MSs need to implement a system which shall act as the SPOC for all request messages that are forwarded to that MS. This system will be responsible for receiving the XML messages from the Hub and must have high availability, reliability and scalability.

MS may choose to receive responses through the same application and route messages internally behind the SPOC. Alternatively, MS may choose to send requests to the Hub from multiple applications over synchronous messaging protocols. The responses to the requests will be returned to the sender on the synchronous response, see 5.3.2 below.

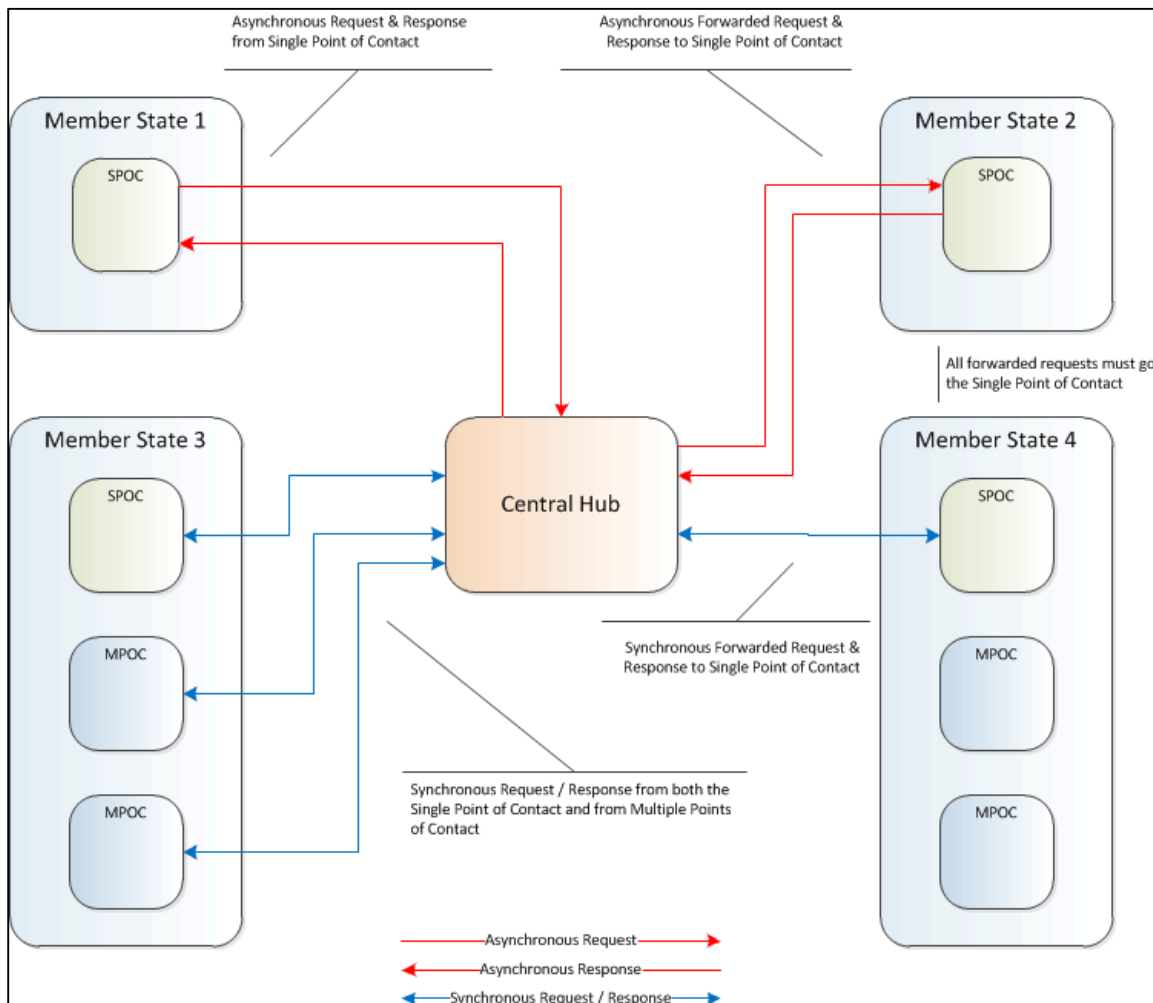


Figure 1 - TACHOnet Architecture

5.2. XML Messaging Framework

The Hub:

- receives a request from a MS
- forwards / broadcasts the request to the necessary counterparty MS
- collects the individual response(s)
- prepares the consolidated response based on the individual response(s) from the MS
- forwards the final response to the requesting MS.

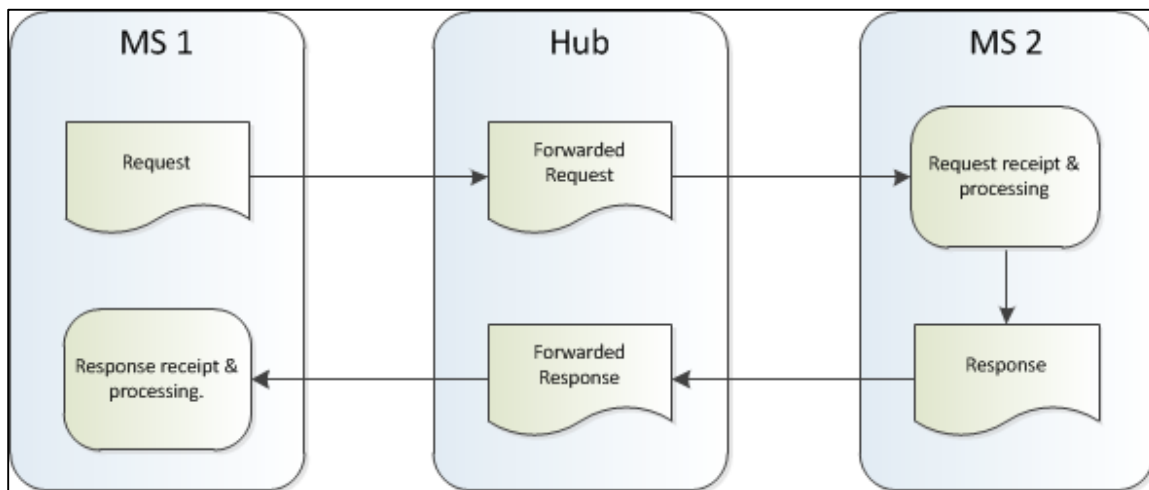


Figure 2 - Standard Message Flow

The following rules or implementation constraints must be observed when implementing the Hub and the MS.

- All MS must be connected to the TESTA network to be able to exchange messages.
- Every MS (through their SPOC) is in charge of ensuring 24/7, high-availability 98% uptime. Response time of less than 10 seconds, for every request coming as an XML message from the Hub, see [2016/68] Annex VI.
- If the request / response message received from a MS has an invalid format then the Hub will not process the message. After recording the details of an invalid message, the Hub will notify the sending MS that an invalid message has been received (via an EN message, assuming that the sender can be determined from the content of the malformed message).
- When a MS sends, via XML over HTTPS (i.e. asynchronously), request(s) or response(s) to the Hub, the Hub will answer with the HTTP '202' status code, (e.g. "HTTP/1.1 202"). Similarly, a HTTP '200' or HTTP '202' status code must be returned when the Hub posts successfully to the MS.
- When a MS sends a message via a synchronous connection the response XML will be returned in the body of the response; there will be no correlation token.
- Each MS (as well as the Hub) must be designed to cope with potential communication and server problems (e.g. 'HTTP 500' returned by the Hub, final response not received from the Hub within time, timeout); or if an XML message (request or response) has not been acknowledged with the HTTP '200' or '202' status code. If the Hub is not able to send the message (request or response) to the destination MS it will send a message with an appropriate status code to the message originator.
- The Hub provides two separate URL addresses, one for request messages and one for response messages. This helps the Hub to prioritise the messages and the MS are requested to implement the same pattern.

5.3. Asynchronous and Synchronous Message Patterns

TCN 1.5 only provides for the exchange of messages via asynchronous messages which are exchanged as XML over HTTP (the request is received and the connection closed; the response is returned on a new connection). This message exchange pattern dates from the original specifications for TCN over 10 years ago and was designed to use the simplest available exchange mechanism (enabling easy implementation and integration in diverse technical environments at MS).

However, messaging patterns between heterogeneous systems have developed since the debut of TCN and this pattern does not always easily integrate into the application architecture at the member states. Specifically, the more recent application integration patterns are based on synchronous messaging whereby the response is returned on the same connection that initiated the request.

TCN 3.0 will provide for the exchange of messages over synchronous connections as well as the existing asynchronous connection. All connections between the Hub and MS must use HTTP as the underlying transport; the Commission will not open the firewall to other protocols.

Initially the two options for connections between the MS and the Hub will be:

- Asynchronous: XML over HTTP
- Synchronous: SOAP 1.2 over HTTP

Additionally, MOVE will consider, without obligation, requests from member states to implement additional protocols such that their applications can be more easily integrated with the Hub (e.g. JSON, ReST, OData).

The requirements and methodology to connect to the ERRU network can be found in the MOVEHUB Networking Guide [MOVEHUB/Network].

5.3.1. *Asynchronous End-Points*

Messages exchanged asynchronously with the Hub will operate in TCN 3.0 in the same fashion as the current TCN 1.5. The XML will be POSTed to the Hub URL and the XML response message will be POSTed to the MS SPOC.

The URLs exposed by the Hub will change for TCN 3.0 to allow the Commission more granular control over the management of the Hub. The URL will be:

<https://webgate.ec.testa.eu/move-hub/tcn/http/request/<memberstate>/btshttreceive.dll>

<https://webgate.ec.testa.eu/move-hub/tcn/http/response/<memberstate>/btshttreceive.dll>

Where *<memberstate>* is the two letter ISO country code of the sender. The *from* attribute in the message header will be validated against the URL used to send the message to the hub; i.e. they must be the same ISO code.

5.3.2. *Synchronous End-Points*

Member States that wish to exchange messages with the hub over a synchronous protocol **must** use 2-way SSL. For asynchronous connections the hub will already have registered the URL to which the response is to be sent. However, for synchronous protocols the response is returned to the originator of the connection – so the hub must be assured of the requestor's identity and this is done by requiring the client certificate.

The hub exposes two web services: a request-response (2-way) service for normal business messages and a 1-way service for Error Notification messages. The synchronous messaging endpoints have a similar convention to the asynchronous URLs:

<https://movehub.ec.testa.eu/move-hub/tcn/soap/one-way/<memberstate>/tcn.svc>

<https://movehub.ec.testa.eu/move-hub/tcn/soap/reqres/<memberstate>/tcn.svc>

Where `<memberstate>` is the two letter ISO country code of the sender. As with the asynchronous messaging service, the `from` attribute in the message header will be validated against the URL used to send the message to the hub; i.e. they must be the same ISO code.

The WSDL for each service can be obtained directly from the service itself:

<https://movehub.ec.testa.eu/move-hub/tcn/soap/oneWay/<memberstate>/tcn.svc?wsdl>

<https://movehub.ec.testa.eu/move-hub/tcn/soap/reqres/<memberstate>/tcn.svc?wsdl>

5.3.2.1. 2-Way Synchronous Messages

Data will be exchanged as typed XML in the SOAP body. The 2-way web service is to send business requests to the hub and to receive the business response. If there is a failure during the processing of the request an Error Notification will be returned in the soap Fault channel.

An example of a CCS request is:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <CheckCardStatus_Request xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
      <Header version="3.2"
        technicalId="1d061fe6-5dd1-4325-bed3-3f6f388a538e"
        workflowId="782dec67-f3ac-48f9-9fc6-9d0cc0123f9b"
        sentAt="2016-01-01T00:00:00Z"
        timeoutValue="2016-01-01T00:00:20Z"
        from="IE"
        to="UK"/>
      <Body businessCaseld="CIC Request Example"
        requestPurpose="Control"
        requestSource="RSI">
        <SearchCriteria cardNumber="C6235B5410196509"/>
      </Body>
    </CheckCardStatus_Request>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure 3 – Request-Response SOAP Message Request Example

And the corresponding response:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <CheckCardStatus_Response xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
      <Header version="3.2"
        technicalId="37b0929f-eb16-4347-8291-b42285ef5d77"
        workflowId="782dec67-f3ac-48f9-9fc6-9d0cc0123f9b"
        sentAt="2016-01-01T00:00:09Z"
        from="UK"
        to="IE"/>
      <Body businessCaselId=" CIC Request Example ">
        <SearchCriteria cardNumber="C6235B5410196509"
          statusCode="Found">
          <SearchResults>
            <CardHolderDetails familyName="Creighton-Ward"
              firstName="Penelope"
              dateOfBirth="1939-12-24"
              placeOfBirth="London"/>
            <CardDetails cardNumber="C6235B5410196509"
              cardStatus="Replaced"
              cardIssuingAuthority="VOSA, PO Box 343, Swansea SA1 2YS"
              cardStartOfValidityDate="2015-09-30"
              cardExpiryDate="2020-09-29"
              cardStatusModifiedDate="2015-09-02T13:42:09Z"
              validForDriving="Yes"
              temporaryCard="No"/>
            <DrivingLicenceDetails drivingLicenceNumber="CREIG312249P99IR"
              drivingLicenceIssuingCountry="UK"
              drivingLicenceStatus="Valid"
              drivingLicenceIssuingDate="1986-09-30"
              drivingLicenceExpiryDate="2019-09-29"/>
          </SearchResults>
        </SearchCriteria>
      </Body>
    </CheckCardStatus_Response>
  </soapenv:Envelope>

```

Figure 4 – Request-Response SOAP Message Response Example

And the fault (in place of a response):

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <s:Fault>
      <faultcode>s:CustomError</faultcode>
      <faultstring xml:lang="fr-BE">CustomError</faultstring>
      <detail>
        <ErrorNotification xmlns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
          <Header version="3.2"
            technicalId="d6d266b2-bfd9-4f72-8647-3ce600d9c4ce"
            workflowId="782dec67-f3ac-48f9-9fc6-9d0cc0123f9b"
            sentAt="2017-08-30T11:53:32Z"
            from="EU"
            to="IE"/>
          <Body businessCaselId="CIC Request Example"
            statusCode="DuplicateRequest"
            statusMessage="Request message already exists">
            <OriginalMessage> &lt;?xml version="1.0" encoding="UTF-
            8" ?>&lt;CheckCardStatus_Request xmlns="https://webgate.ec.testa.eu/move-
            Hub/tachonet/3.2">&lt;Header version="3.0" technicalId="1d061fe6-5dd1-4325-bed3-
            3f6f388a538e" workflowId="782dec67-f3ac-48f9-9fc6-9d0cc0123f9b" sentAt="2016-01-
            01T00:00:00Z" timeoutValue="2016-01-01T00:00:20Z" from="IE" to="UK" />&lt;Body
            businessCaselId="CIC Request Example" requestPurpose="Control"
            requestSource="RSI" SearchCriteria
            cardNumber="C6235B5410196509" />&lt;/OriginalMessage>
          </Body>
        </ErrorNotification>
      </detail>
    </s:Fault>
  </s:Body>
</s:Envelope>

```

Figure 5 – Request-Response SOAP Message Fault Example

The operations and types available in the 2-way web service are:

Operation	Request & Response Messages	Schema / Type
CheckIssuedCards_Request	TCN_CheckIssuedCards_Request_InputMessage	CheckIssuedCards_Request
	TCN_CheckIssuedCards_Request_OutputMessage	CheckIssuedCards_Response
CheckCardStatus_Request	TCN_CheckCardStatus_Request_InputMessage	CheckCardStatus_Request
	TCN_CheckCardStatus_Request_OutputMessage	CheckCardStatus_Response
IssuedCardDrivingLicence_Request	TCN_IssuedCardDrivingLicence_Request_InputMessage	IssuedCardDrivingLicence_Request
	TCN_IssuedCardDrivingLicence_Request_OutputMessage	IssuedCardDrivingLicence_Response
ModifyCardStatus_Request	TCN_ModifyCardStatus_Request_InputMessage	ModifyCardStatus_Request
	TCN_ModifyCardStatus_Request_OutputMessage	ModifyCardStatus_Acknowledgement
ModifyCardStatus_Response	TCN_ModifyCardStatus_Response_InputMessage	ModifyCardStatus_Response
	TCN_ModifyCardStatus_Response_OutputMessage	ModifyCardStatus_Acknowledgement
Fault	ErrorNotificationFault_FaultMessage	ErrorNotification

5.3.2.2. 1-Way Synchronous Messages

The 1-Way web service is available for either the Hub or a MS to send an Error Notification to the other when the business message pattern is complete, i.e. if the response is invalid.

Section 7.6 provides a description of the ErrorNotification message and how it is used within the message exchange. Figure 48 and Figure 49 show the use of the ErrorNotification message when a request is invalid. For synchronous messaging these ErrorNotification messages are returned on the soap fault channel.

Conversely, Figure 50 and Figure 51 depict errors in the response message. In these cases the sender of the response considers the message exchange complete and for synchronous messaging the protocol connection is closed. If the response message is invalid the recipient of the message needs a method to inform the responder of the error. For synchronous messaging this is provided by the 1-way web service; an ErrorNotification message is sent and no response (other than the technical http 200 / 202 and an empty body) is received.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ErrorNotification xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
      <Header version="3.2"
        technicalId="5a4b12de-272a-41ed-a27c-37622d9a2528"
        workflowId="782dec67-f3ac-48f9-9fc6-9d0cc0123f9b"
        sentAt="2016-01-01T00:00:00Z"
        from="IE"
        to="EU"/>
      <Body businessCaselId="CIC Request Example"
        statusCode="InvalidFormat"
        statusMessage="XML not well-formed."/>
      <OriginalMessage>
        &lt;?xml version="1.0" encoding="UTF-8" ?>&lt;CheckCardStatus_Request
        xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">&lt;Header version="3.0"
        technicalId="1d061fe6-5dd1-4325-bed3-3f6f388a538e" workflowId="782dec67-f3ac-48f9-9fc6-
        9d0cc0123f9b" sentAt="2016-01-01T00:00:00Z" timeoutValue="2016-01-01T00:00:20Z"
        from="IE" to="UK"/>&lt;Body businessCaselId="CIC Request Example"
        requestPurpose="Control" requestSource="RSI"
        cardNumber="C6235B5410196509"/>&lt;/Body&gt;
      </OriginalMessage>
    </ErrorNotification>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure 6 – One-Way SOAP Message Request Example

And the corresponding response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body/>
</soapenv:Envelope>
```

Figure 7 – One-Way SOAP Message Response Example

Operation	Request & Response Messages	Schema / Type
ErrorNotification	TCN_ErrorNotification_InputMessage	ErrorNotification
	TCN_ErrorNotification_OutputMessage	None

5.3.2.3. *Member State Implementation*

MS wishing to exchange synchronous messages with the hub must implement the same interface as the hub. T

5.4. Multiple Points of Contact

From its inception Tachonet has operated on the principle that there is a single point of contact (SPOC) for all communication between member states, both technically and administratively, and this function is fulfilled by the competent authority (there may be multiple CIAs in a member state, but they all operate under the mandate of a SPOC).

The technical implementation of this is that there is a single URL to which TCN messages are sent and if any further national routing is required to direct a message to a specific CIA (or other administration) within the MS then it is managed within the national network and transparent to the Hub.

MOVE have received requests to extend this model such that the competent authority and inspectorate authority (and any other authorised administration) in a member state may have separate TCN end points – messages will be directed to a different URLs for CIA traffic and enforcement traffic. Once consideration is given to the abandonment of a SPOC to allow two endpoints it is only a small conceptual step to go from two endpoints to multiple endpoints.

This requirement fits with the synchronous messaging pattern (see 5.3.2 above) as the response is returned on the same HTTP connection, so by definition the response is returned to whoever initiated the request.

The assumption is that multiple endpoints only apply to response messages as:

- All requests forwarded to a country are for the CIA, enforcement authorities do not receive TCN requests.
- If a member state has multiple CIAs then neither the requesting member state nor the Hub will know to which CIA a request needs to be routed, this will still need to be managed behind the SPOC and within the national network.

Therefore, all requests forwarded to a MS will still be routed via a single URL, and responses via the asynchronous messaging pattern will also be sent to a single URL.

5.5. Single Message and Batch Message Modes

MS may send messages to, and receive responses from the Hub either as a single request / response or as a batch of multiple requests / responses in the same message. There is no difference in the message format between these two modes of exchange, simply the number of business elements within the *Body* of the message.

5.5.1. *Single Message Mode*

The majority of messages exchanged within TCN today are single messages – the *Body* of the message contains a single search or notification which is forwarded to the counterparty MS(s). Such messages are processed immediately upon receipt.

```

<?xml version="1.0" encoding="UTF-8"?>
<CheckCardStatus_Request xmlns="https://webgate.ec.testa.eu/move-Hub/TCN/3.2">
  <Header version="3.2"
    technicalId="5bd3f8d4-8d4a-4df0-80a7-2b6c12d57397"
    workflowId="52c17dff-d784-4b20-94f1-1921ca3ab5ef"
    sentAt="2016-01-01T00:00:00Z"
    timeoutValue="2016-01-01T00:00:20Z"
    from="IE"
    to="UK"/>
  <Body businessCaseld="2016-01-01 File 1" requestPurpose="Exchange" requestSource="CIA">
    <SearchCriteria cardNumber="3098902370000j30"/>
  </Body>
</CheckCardStatus_Request>

```

Figure 8 - Singlecast Message Example

5.5.2. *Batch Message Mode*

Batches are only of interest (and concern) to the requesting MS. It is the requestor that decides whether or not they wish to submit a batch of requests / notifications. If the Hub receives a batch request it will queue the request until the evening / night so that the processing overhead of batches does overwhelm either the Hub or the responding MS(s). Further, the batch will be separated into individual requests and these will be forwarded one at a time to the counterparty MS. Therefore, no MS is required to implement batch handling logic if they do not want to use batches – no MS will receive a batched request from the Hub.

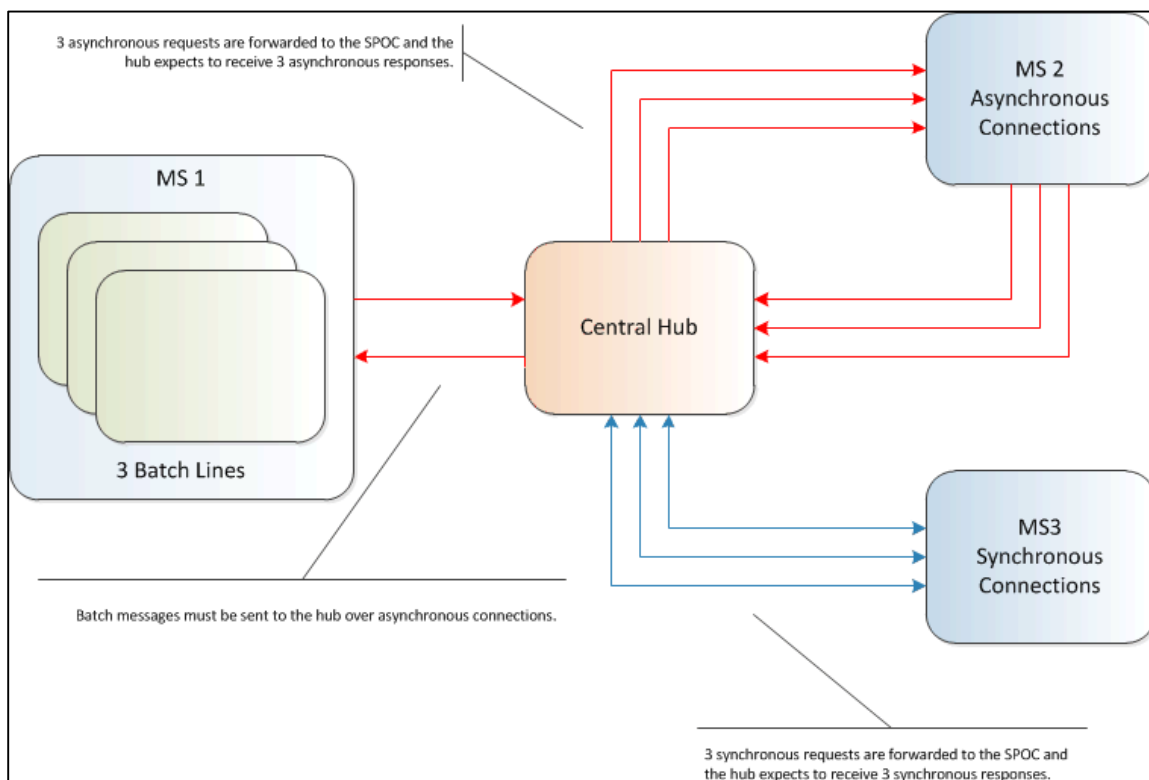


Figure 9 - Batch Message Flow

Up to 100 searches / notification may be batched into a single request. This is the same message type as above but batching two searches into a single request to the Hub:

```

<?xml version="1.0" encoding="UTF-8"?>
<CheckCardStatus_Request xmlns="https://webgate.ec.testa.eu/move-Hub/TCN/3.2">
  <Header version="3.2"
    technicalId="5bd3f8d4-8d4a-4df0-80a7-2b6c12d57397"
    workflowId="52c17dff-d784-4b20-94f1-1921ca3ab5ef"
    sentAt="2016-01-01T00:00:00Z"
    timeoutValue="2016-01-01T00:00:20Z"
    from="IE"
    to="UK"/>
  <Body businessCaselId="2016-01-01 File 1" requestPurpose="Exchange" requestSource="CIA">
    <SearchCriteria cardNumber="3098902370000j30"/>
    <SearchCriteria cardNumber="3098772270000j31"/>
  </Body>
</CheckCardStatus_Request>

```

Figure 10 – Singlecast Batch Message Example

MS YY will receive two individual search requests: one with `<SearchCriteria cardNumber="3098902370000j30"/>` and the other with `<SearchCriteria cardNumber="3098772270000j31"/>`.

Batch requests received by the Hub will be queued and then processed, in order of receipt, between 20:00 UTC and 04:00 UTC each day. Batch requests arriving after 04:00 will be held until the queue is opened at 20:00 later that day. Any batches arriving after 20:00 will be processed in turn up to 04:00 the following day. Any batches that are not processed before the 04:00 cut-off will be held until the subsequent processing window.

It is possible to batch multiple broadcast CIC searches into a single request:

```

<?xml version="1.0" encoding="UTF-8"?>
<CheckIssuedCards_Request xmlns="https://webgate.ec.testa.eu/move-Hub/TCN/3.2">
  <Header version="3.2"
    testId="a"
    technicalId="51640325-e4a0-46ef-bc59-73f5f1c25e90"
    workflowId="2a8d178b-6218-46fb-a3c1-ddf78978aba4"
    sentAt="2016-01-01T00:00:00Z"
    timeoutValue="2016-01-01T00:00:20Z"
    from="UK"
    to="IE"/>
  <Body businessCaselId="TB-IR File 1"
    requestPurpose="Exchange"
    requestSource="CIA">
    <SearchedCriteria familyName="Parker"
      firstName="Aloysius"
      dateOfBirth="1965-09-30"
      placeOfBirth="London"
      familyNameSearchKey="PARCAR"
      firstNameSearchKey="ALAS">
      <DrivingLicence drivingLicenceNumber="AP19650930"
        drivingLicenceIssuingCountry="UK"/>
    </SearchedCriteria>
    <SearchedCriteria familyName="Creighton-Ward"
      firstName="Penelope"
      dateOfBirth="1939-12-24"
      familyNameSearchKey="CRAGTANWAD"
      firstNameSearchKey="PANALAP">
      <DrivingLicence drivingLicenceNumber="PCW19650930"
        drivingLicenceIssuingCountry="UK"/>
    </SearchedCriteria>
  </Body>
</CheckIssuedCards_Request>

```

Figure 11 - Broadcast Batch Message Example

But in these cases MS should be aware of the potentially large final responses that they may receive.

As can be seen in the above examples, the destination MS is set in the *Header/@to* attribute and the multiple requests / notifications are set within the *Body* element. It follows, therefore, that a batch of messages can only be targeted at the same MS within a single batch message; the exception being a batch of CIC searches that is broadcast to all MS (i.e. *to ZZ*).

All messages (except EN) may be exchanged in batch mode. For MCS messages the acknowledgements to the MCS requests will be returned in a corresponding batch. However, the MCS responses will be returned individually from the responding country.

Batches are not permitted using a synchronous connection.

5.6. Message Structure

The structure of every XML message is the same and described in the below table:

Element Name	Description
Root element	The root element of each XML message gives the name, namespace and version of the message.
Header	There is always a Header node giving non-business information about the current workflow (such as workflow and message identifiers, the ISO country codes of the counterparties, sending and expiration timestamps, etc).
Body	There is always a Body node giving the business information of the current workflow. Such business information consists of one or more elements containing relevant data.

The elements used in XML messages define only the structure of the message and contain no data; all the data is contained in attributes of the elements. The naming conventions for elements and attributes are:

Node Type	Description
Element	The naming standard for elements is Pascal cased, e.g.: <ul style="list-style-type: none"> • <i>Header</i> • <i>SearchCriteria</i> • <i>DrivingLicenceDetails</i>
Attribute	The naming standard for attributes is camel cased, e.g.: <ul style="list-style-type: none"> • <i>from</i> • <i>workflowId</i> • <i>driverCardNumber</i>

The XML Schema Definition (XSD) of all the XML messages is supplied separately in an electronic format. The namespace of the TCN XSD specifications is <https://webgate.ec.testa.eu/move-Hub/TCN/3.2> and must be specified as an *xmlns* attribute in the root element of every XML message.

The version of the XSD specification and resulting XML messages will be detailed in the namespace and version attributes of the message root. The version number (n.m) is a defined and fixed value in every release of XML Schema Definition file (XSD).

When sending and receiving an XML message, both the Hub and the MS must check whether the XML message is well formed and conforms to XML syntax rules.

5.7. Standardisation of XML Messages

5.7.1. *Schema Version*

The version of the XSD specification and resulting XML messages will be detailed in the namespace and version attributes of the message root. The version number (n.m) is a defined and fixed value in every release of XML Schema Definition file (XSD).

5.7.2. *Encoding*

All messages will be exchanged in UTF-8. If the encoding directive (`<?xml version="1.0" encoding="UTF-8"?>`) is missing from the message UTF-8 will be enforced by the Hub. If the encoding specifies something different (e.g. `<?xml version="1.0" encoding="UTF-16"?>`) then it will be discarded and UTF-8 will be enforced.

Any errors resulting from enforcing UTF-8 encoding will be returned to the sender via an EN message.

5.7.3. *Namespaces*

The namespace of the TCN messages is <https://webgate.ec.testa.eu/move-Hub/TCN/3.2> and this is declared in the XSDs as the default and target namespaces:

```
<xs:schema xmlns="https://webgate.ec.testa.eu/move-Hub/TCN/3.2"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="https://webgate.ec.testa.eu/move-Hub/TCN/3.2"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="3.2">
```

And in the TCN messages as the default namespace:

```
<CheckCardStatus_Request xmlns=https://webgate.ec.testa.eu/move-Hub/TCN/3.2
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

MS are free to declare their own namespace prefixes as long as the namespace conforms to the published XSD version. MS must also be prepared to receive messages with custom namespace definitions, e.g.

```
<ms1:CheckCardStatus_Request xmlns:ms1=https://webgate.ec.testa.eu/move-Hub/TCN/3.2
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

5.7.4. *Optional Attributes*

Optional attributes may be omitted from the message, if they are present they must have valid content. If an optional attribute is present and contains content then it will be validated against the schema.

All string attributes in the Tachonet schemas are declared as the XSD *xs:token* type and have a minimum length of one. Therefore, an optional attribute with any of the following contents is invalid against the XSD:

- Zero-length string
- Single space (#x20)
- Multiple spaces (#x20)

5.7.5. *Message Header*

All information needed to route all messages should be found in the *Header* and not in the *Body* of the XML message. The header format for all messages is identical.

The contents of the *Header* are:

Attribute Name	Description
version	The official version of the XML specifications will be detailed in the namespace defined in the message XSD and in the version attribute of the Header element of any XML message. The version number ('n.m') will be defined as a fixed value in every release of the XML Schema Definition file (.xsd). The current version number is '3.2'.
testId	Optional id for testing, see 5.7.7.1 below
technicalId	A UUID uniquely identifying each individual message, see 5.7.7.2 below.
workflowId	The WorkflowId is then used to associate all messages in the same workflow, see 5.7.7.3 below.
sentAt	The date and time (UTC) that the message was sent.
timeoutValue	This is an optional date and time attribute in UTC format. This value will be set by the Hub and will inform the responding MS of the time when the request will be timed out at the Hub.
from	The ISO 3166-1 Alpha 2 country code of the originating (requesting) MS sending the message or 'EU' for messages originating from the Hub (e.g. EN messages & CIC broadcast responses).
to	The ISO 3166-1 Alpha 2 letter country code of the MS to which the message is sent; i.e. responding MS. See 5.3 above.

5.7.6. *Message Body*

All business information that is exchanged between administrations is contained within the *Body* element of the message. The Body element of the request messages has the same three attributes:

Attribute Name	Description
businessCaseId	A reference number given by the originating Member State, see 5.7.7.4 below.
requestSource	The function of the administration / authority from which the request originates. Possible values are: CIA, RSI, Hub, Other
requestPurpose	The business purpose of the request. Possible values are: Issue, Exchange, Control, Heartbeat, Other

The requestSource and requestPurpose attributes are used by the hub to gather statistics on the source and purpose of Tachonet requests. They are populated by the sender as follows:

requestSource	
CIA	The request was sent from the Card Issuing Authority.
RSI	The request was sent from an authorised enforcement or control administration (e.g. the police).
Hub	The message was sent from the hub (e.g. Error Notification or Heartbeat message).
Other	A catch all value to provide for any scenario not covered above.

requestPurpose	
Issue	The request was sent as part of the CIA's normal card issuing process.
Exchange	The request was sent as part of the CIA's normal card exchange process.
Control	The request was sent as part of the enforcement administration's normal control or inspection process.
Heartbeat	The request was sent by the hub as a Heartbeat search.
Other	A catch all value to provide for any scenario not covered above.

5.7.7. Identifier Definitions in XML Messages

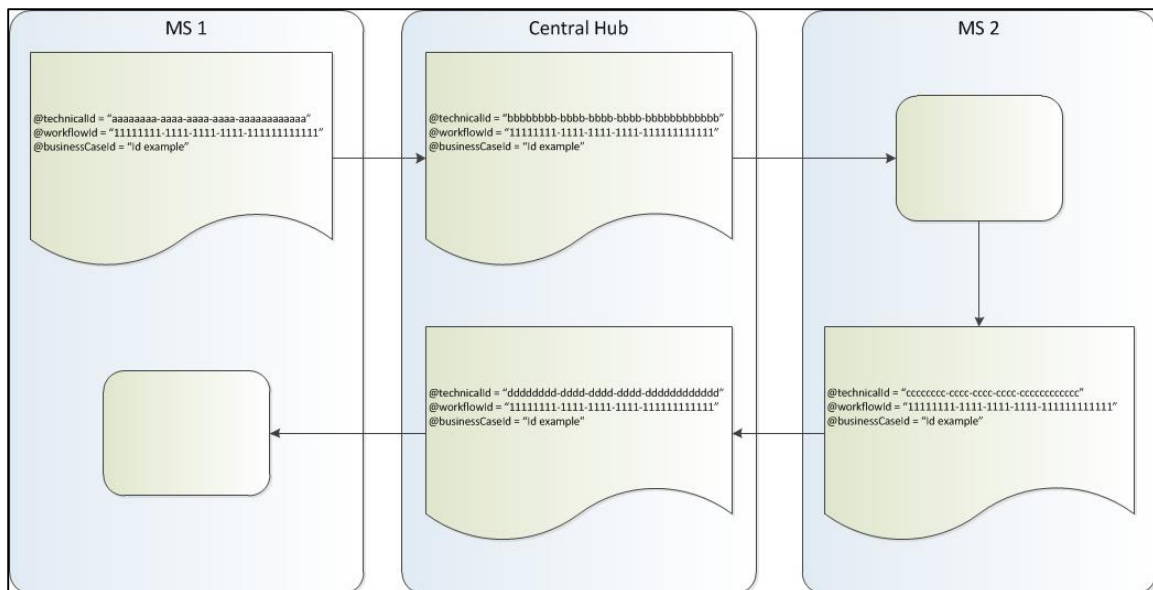


Figure 12 - Message Identifiers

As the exchange of XML messages between the MS and the Hub may be asynchronous, there are four attributes used to identify messages and workflows.

The UUID hexadecimal digits can be in lowercase or uppercase. The Hub does not guarantee that the case of hexadecimal digits is preserved when forwarding requests or returning responses. Therefore, MS must ensure that their applications are case agnostic when handling UUID attributes.

The implementation of a UUID algorithm will be dependent on the programming language being used. MS should choose a language that implements, at minimum, version 4 of the UUID standard¹².

5.7.7.1. *testId*

This is an optional identifier (length 8) for testing purposes only. The originator of the test will populate the identifier and the Hub will return the same identifier in the associated response. Any content in this attribute does not affect the processing of a request and responding MS do not need to populate this attribute.

5.7.7.2. *technicalId*

This is a UUID uniquely identifying each individual message, every message created by either the MS or the Hub will have a new *technicalId*. The sender of the message generates a UUID and populates this attribute. This data is not used in any business capacity; it's purpose is to uniquely identify a message when troubleshooting problems.

MS applications should generate this identifier as close as possible to the actual dispatch of a message to avoid potential bugs allowing duplicate identifiers to be used. The purpose of this identifier is lost if a MS emits messages with the same UUID.

5.7.7.3. *workflowId*

The *workflowId* is a UUID and is generated by the requesting MS and this identifier is then used to associate and correlate subsequent messages in the workflow. The *workflowId* is present in all messages and must be the same for related messages. This identifier is not considered as business data, is a technical attribute used by the Hub and MS applications to associate all the individual messages that comprise a complete workflow.

The Hub validates that, for each request received, the *workflowId* is unique across all member states and all requests.

5.7.7.4. *businessCaseId*

The *businessCaseId* identifier is an attribute of the body element of every message. It is the same length as a UUID canonical string representation³, but free-format. This identifier is similar to the *workflowId* in that it associates the individual messages in a workflow. However, it differs from the *workflowId* in a number of areas:

- Both the *businessCaseId* and the *workflowId* allow responses to be correlated with the originating request; the former is intended as a business data identifier and the latter as a technical data identifier.
- The *businessCaseId* is a free format string (rather than enforcing a UUID) allowing the requesting MS to create identifiers with business meaning, e.g. combinations of card application type, date, handling officer, file number, etc. Alternatively, a UUID may be used in this attribute.
- When the Hub receives new requests the *workflowId* is validated for uniqueness (across all MS and all message types). The *businessCaseId* is not validated and it is the responsibility the requesting MS to ensure that, if they intend to make use of the *businessCaseId*, then their application generates suitable identifiers.
- As the Hub does not enforce the uniqueness of the *businessCaseId* MS applications must be able to process messages that have the same *businessCaseId* values as messages from the same or other MS(s).

¹ https://en.wikipedia.org/wiki/Universally_unique_identifier.

² <https://tools.ietf.org/html/rfc4122>.

³ Length 36 - 32 hexadecimal digits and four hyphens.

5.8. Message Timeout

A request / notification message to the Hub always has a timeout value attached to it (apart from EN messages), and the mechanism of this timeout depends on whether a synchronous or asynchronous messaging pattern is being used.

All messages have a maximum request-response timeout of 20 seconds at the originating MS and 10 seconds at the responding MS. The calculation and management of the timeout is dependent on the messaging pattern: asynchronous or synchronous.

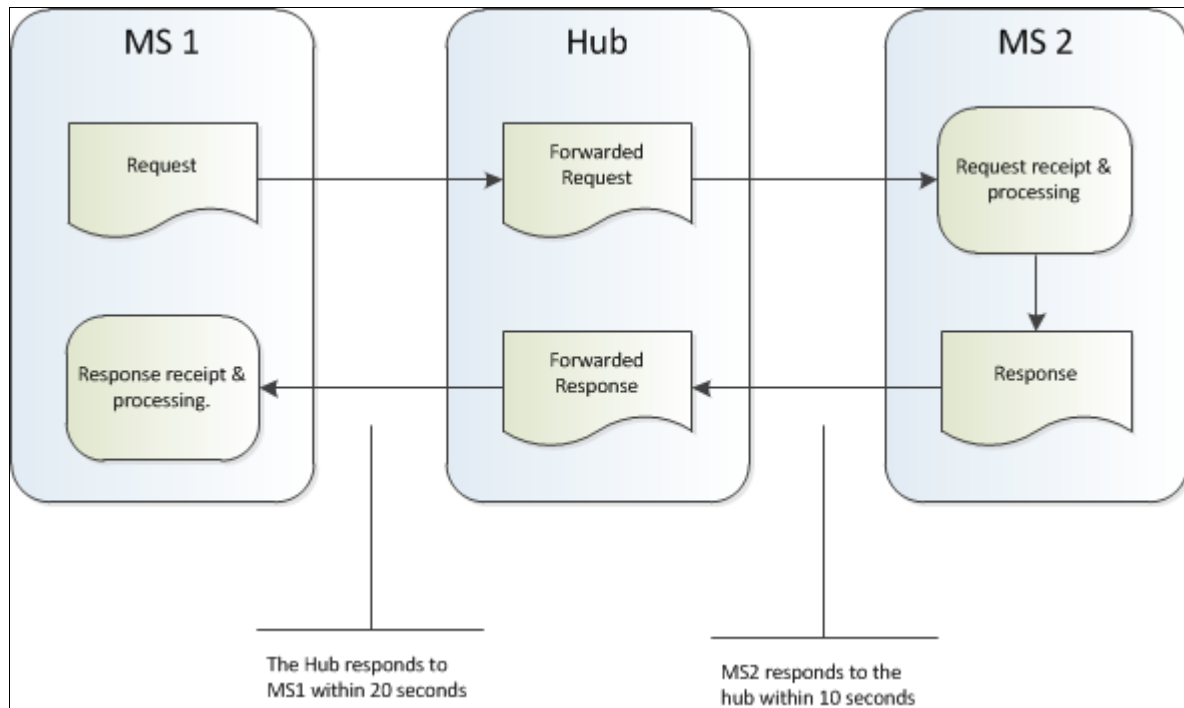


Figure 13 - Message Timeouts

TCN 1.5 allowed the MS to set the *timeout* value in the request sent to the Hub, this is no longer supported – the Hub enforces the timeout. Further, the timeout in the request forwarded to the responding member state was a duration defined in seconds. For TCN 3.2 the timeout calculation will be based on the time that the Hub receives the message and the value in the forwarded request will be an absolute value in ISO 8601 format (see 6.3.2).

5.8.1. *Asynchronous Timeout*

When an asynchronous message is received the Hub will return a HTTP 202 response and record the time that the message was received. The Hub will use the received at time to calculate the time that the originating MS is expecting a response.

The Hub will forward the message to the counterparty MS(s) and populate the Header attribute timeout with a date time value notifying the responding MS by what time they must respond to the Hub. The Hub waits for responses until this time is reached, and when this time is reached the Hub no longer accepts anymore responses and begins the process of returning the final response to the originating MS.

If all the response(s) is / are received before the timeout then the Hub completes the workflow and returns an ‘early’ response to the originating MS.

If no responses have been received the Hub will return a response message at 20 seconds.

5.8.2. *Synchronous Timeout*

The same 20 second timeout applies to synchronous messages and will be enforced on the synchronous connection made to the Hub; either the Hub will have returned an 'early' response message within the 20 second timeout, or a response with a *Timeout* status will be returned.

5.8.3. *Batch Timeout*

A batch request may be sent to the Hub only via an asynchronous HTTP connection and when received the Hub will return a HTTP 202 response. The batch will then be queued until the following overnight processing window (see 5.5.2).

When the batch is processed the Hub will forward the individual batch messages to the counterparty MS(s) and populate the Header attribute timeout with a date time value notifying the responding MS by what time they must respond to the Hub (i.e. 'now' plus 10 seconds).

When the last batch line has arrived at its timeout the batched response is built and returned to the original sender.

6. MESSAGE VALIDATION

When sending and receiving an XML message, the Hub and the CIA applications must check whether it is a "Well Formed" XML document (i.e. a document that conforms to the XML syntax rules) and must validate it against its XML Schema definition (XSD).

6.1. ISO Country Codes

The country codes used within the TCN messages conform to the ISO 3166-1 Alpha 2 standard. The following table gives the ISO codes for countries that are potential Tachonet members. The Connected column gives the current status of these countries. This is a change to TCN 1.5 in which the codes used were the UNECE codes.

These are the codes that are accepted by the Hub for the *To* and *From* attributes in the message Header (countries connected with version 1.5 at the time of the drafting of the document).

n.b. GB is not accepted as a code for message routing, but it is still a valid value for any business data content of the message, such as the driving licence issuing country.

ISO Code	Country	Connected
AD	Andorra	Yes
AL	Albania	No
AM	Armenia	Yes
AT	Austria	Yes
AZ	Azerbaijan	Yes
BA	Bosnia and Herzegovina	Yes
BE	Belgium	Yes
BG	Bulgaria	Yes
BY	Belarus	No
CH	Switzerland	Yes
CY	Cyprus	Yes
CZ	Czech Republic	Yes
DE	Germany	Yes
DK	Denmark	No
EE	Estonia	Yes
ES	Spain	Yes
FI	Finland	Yes
FR	France	Yes
GE	Georgia	Yes
GR	Greece	Yes
HR	Croatia	Yes
HU	Hungary	Yes
IE	Ireland	Yes
IS	Iceland	Yes
IT	Italy	Yes
KZ	Kazakhstan	No
LI	Lichtenstein	Yes

ISO Code	Country	Connected
LT	Lithuania	Yes
LU	Luxembourg	Yes
LV	Latvia	Yes
MC	Monaco	Yes
MD	Moldova	Yes
ME	Montenegro	No
MK	FYRoM	Yes
MT	Malta	Yes
NL	Netherlands	Yes
NO	Norway	Yes
PL	Poland	Yes
PT	Portugal	No
RO	Romania	Yes
RS	Serbia	Yes
RU	Russia	No
SE	Sweden	Yes
SI	Slovenia	Yes
SK	Slovakia	Yes
SM	San Marino	No
TJ	Tajikistan	No
TM	Turkmenistan	No
TR	Turkey	No
UA	Ukraine	Yes
UK	United Kingdom	Yes
UZ	Uzbekistan	No

6.2. Status Codes and Status Messages

The status codes available for use depend on the type of response being created. In the following tables the Timeout and NotAvailable status codes are only valid for responses that are sent by the Hub to a MS. Responses received from a MS with these status codes are rejected.

6.2.1. CCS and CIC Responses

Attribute Value	Description
Found	At least one matching card / driver found by the responding MS.
NotFound	No matching card / driver found by the responding MS.
Timeout	A valid response message has not been received from the responder within the timeout period.
NotAvailable	The responding MS is not available within TACHOnet at the time of the request.

6.2.2. ICDL Responses

Attribute Value	Description
OK	The responding MS has received and processed the notification without issue. Member States which do not register the ICDL notifications should reply OK, unless an error occurs in the message exchange.
NotOK	The responding MS was unable to process the notification (for example the driving licence is expired / withdrawn / does not have an HGV category).
NotFound	The notified driving licence was not found in the driving licence register.
Timeout	A valid response message has not been received from the responder within the timeout period.
NotAvailable	The responding MS is not available within TACHOnet at the time of the request.

6.2.3. MCS Responses

Attribute Value	Description
Updated	The notified status change has been recorded in the tachograph card register.
NotUpdated	The notified status change has been rejected.
NotFound	No matching card found by the responding MS.
Timeout	A valid response message has not been received from the responder within the timeout period.

6.2.4. *MCS Acknowledgements*

Attribute Value	Description
OK	The MCS message (request or response) has been received.
Timeout	A valid acknowledgement message has not been received from the responder within the timeout period.
NotAvailable	The responding MS is not available within TACHOnet at the time of the request.

6.2.5. *Invalid Responses*

Any processing failure, either business or technical, will be notified to the concerned parties via the EN message (with the e). This is a change to TCN 1.5 in which a failure status code could be returned in the business response message.

The status code attribute of this message may contain the following values.

Attribute Value	Description
InvalidData	The XML message is well-formed but the data fails business validation.
InvalidFormat	The format of the message received does not conform to the message schema or the XML is not well-formed.
Timeout	The corresponding XML response message has not been received within time.
ServerError	The corresponding XML response message has not been successfully processed due to a server problem.
NotAvailable	The responding MS has been temporarily disconnected from the Hub.
ResponseNotCorrelated	A MS or the Hub has received a response message which cannot be matched to a request.
DuplicateRequest	A request with the same <i>workflowId</i> has been previously received.
DuplicateResponse	A response with the same <i>workflowId</i> has been previously received.
Other	For any other error.

6.2.6. *Status Message*

Attached to every status code attribute, there's always a corresponding *statusMessage* attribute that can be used to specify an optional message giving more detailed information about the status code value.

Failure status codes should always be accompanied by an explanatory status message.

Any data in this attribute should be entered in English.

6.3. Date and Time

TCN uses ISO 8601 date and time formats.

6.3.1. *Date*

Calendar date representations are ISO 8601 compliant and constrained by the XSD to be [YYYY]-[MM]-[DD].

- [YYYY] indicates a four digit year, 0000 through 9999.
- [MM] indicates a two digit month of the year, 01 through 12.
- [DD] indicates a two digit day of that month, 01 through 31.

e.g.:

- 2016-04-23

6.3.2. *Datetime*

Time representations are also ISO 8601 compliant and use the 24-hour clock. Time values in the messages will always have a date associated with them – there are no attributes that have a time value without a date. The *date* and *time* elements of the *datetime* value are separated by ‘T’.

The TCN messages use the extended time format which is [hh]:[mm]:[ss].

- [hh] indicates a two digit hour between 00 and 24 (where 24 is only used to notate midnight at the end of a calendar day).
- [mm] indicates a two digit minute between 00 and 59.
- [ss] indicates a two digit second between 00 and 60 (where 60 is only used to notate an added leap second).

6.3.2.1. *UTC*

All datetime values in the TCN messages will be in UTC, not in local time. The ISO 8601 timezone designator for UTC is ‘Z’. Therefore, all date time values in the TCN messages are length 20 (19 datetime characters plus the Z timezone designator), e.g.:

- 2016-04-23T13:15:00Z

6.4. Card Status

The following are the allowable values for card status attributes.

Attribute Value	Description
Application	The CIA has received an application to issue a driver card. This information has been registered and stored in the database with the generated search keys.
Approved	The CIA has approved the application for the tachograph card.
Rejected	The CIA did not approve the application.
Personalised	The tachograph card has been personalised.
Dispatched	The National Authority has dispatched the driver card to the relevant driver or delivering agency.
Handed Over	The National Authority has handed over the driver card to the relevant driver.
Confiscated	The driver card has been taken from the driver by the competent authority.
Suspended	The driver card has been taken temporarily from the driver.
Withdrawn	The CIA has decided to withdraw the driver card. The card has been permanently invalidated.
Surrendered	The tachograph card has been returned to the CIA, and declared no longer needed.
Lost	The tachograph card has been declared lost to the CIA.
Stolen	The tachograph card has been reported stolen to the CIA. A stolen card is considered lost.
Malfunctioning	The tachograph card has been reported as malfunctioning to the CIA.
Expired	The period of validity of the tachograph card has expired.
Replaced	The tachograph card, which has been reported lost, stolen or malfunctioning, has been replaced by a new card. The data on the new card is the same, with the exception of the card number replacement index, which has been increased by one.
Renewed	The tachograph card has been renewed because of a change of administrative data or the validity period coming to an end. The card number of the new card is the same, with the exception of the card number renewal index, which has been increased by one.
InExchange	The CIA that issued a driver card has received a notification that the procedure to exchange that card for a driver card issued by the CIA of another Member State has started.
Exchanged	The CIA that issued a driver card has received a notification that the procedure to exchange that card for a driver card issued by the CIA of another Member State has completed.

Rows highlighted in green are statuses that are valid for driving.

Rows highlighted in red are statuses that are invalid for driving.

6.4.1. Application and Rejected Statuses

For the CIC response the *cardNumber* attribute is mandatory. However, a card number is only assigned to the application once it has been approved. Theoretically, a name search could find a 'card' that has a status of Application or Rejected, and in this case there will not be a card number. In this event a dummy data should be returned in the response:

- The card number will be all zeros
- The start of validity date should be 1900-01-01
- The expiry date should be 1900-01-01

n.b. there are no historical CIC responses with a card status of Application in the Tachonet Hub logs; therefore in all the messages exchanged to date there has not been a response which would have a missing card number).

CIC responses also have a mandatory *cardNumber* attribute but the issue does not arise as a card number must have been provided in the request.

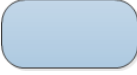

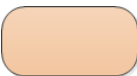


6.5. Card Status Transition

The card status flowchart presented in the previous version of the XML Guide [TCN/XML/1.50] is not correct. It presents an inaccurate view of:

- The flow from one status to another
- The allowable statuses in MCS messages
- The distinction between the status on the driver card and the status in a MCS message

The following sections are provided to illustrate the understanding that MOVE currently has concerning the implementation of card statuses and the possible status transitions between different card statuses.

The key for the diagrams is:

Symbol	Description
	The statuses within the blue shapes occur within the CIA as part of the card creation / issues process. These statuses are not valid for driving.
	The statuses within the green shapes indicate that the card is being sent to or is held by the driver. These statuses are valid for driving.
	The statuses within the red shapes are the final state of the driver card. Once a card has reached this state it cannot transition to another state. These statuses are not valid for driving.
	Unbroken arrows indicate that the process has caused a change in the status of the same card.
	Broken arrows indicate that the process has caused a change in the status of a different card.

6.5.1. Standard Issue

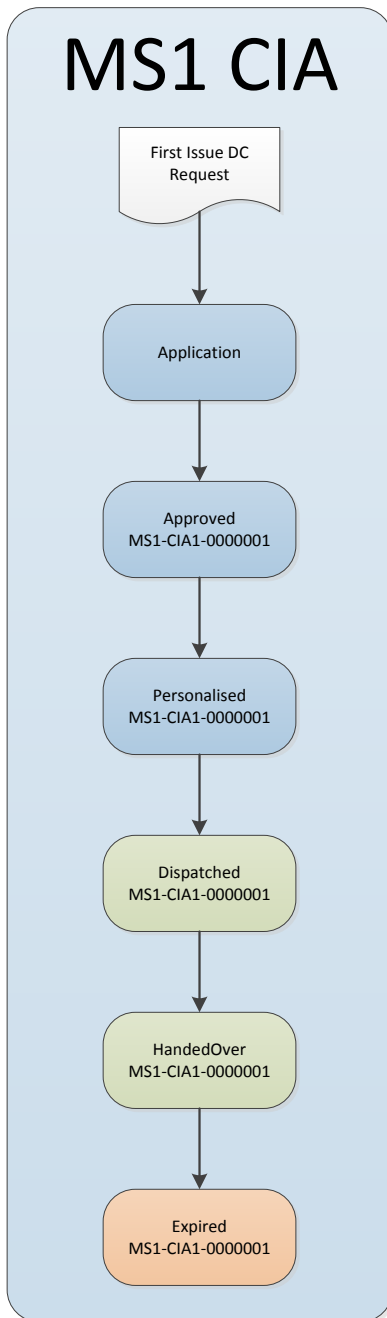


Figure 14 - Card Issue Process

The standard issues process begins with the CIA receiving the application and supporting documentation from the driver, either in paper format or via a web application depending on the process in place within each member state.

Once the card reaches its expiry date, if there is no renewal application from the driver then the card status transitions to Expired and the card's life is at an end.

For the diagrams in the following sections the application process and associated statuses (Application, Approved, Personalized, Dispatched, HandedOver) have been collapsed into a single shape with a name of IssueProcess.

6.5.2. *Card Renewal / Replacement*

Once the card is issued the driver may, before the expiry date, apply to have the card renewed (see [165/2014] Chapter V Article 28). When the card is renewed then the card renewal index (the 16th and final digit in the card number) is incremented by 1. In the example below the card number changes from MS1-CIA1-0000001 to MS1-CIA1-0000002.

The driver may also declare the card lost, stolen or malfunctioning (see [165/2014] Chapter V Article 29) and apply for a replacement card. When the card is replaced then the card replacement index (the 15th and penultimate digit in the card number) is incremented by 1. In the example below the card number changes from MS1-CIA1-0000002 to MS1-CIA1-0000012.

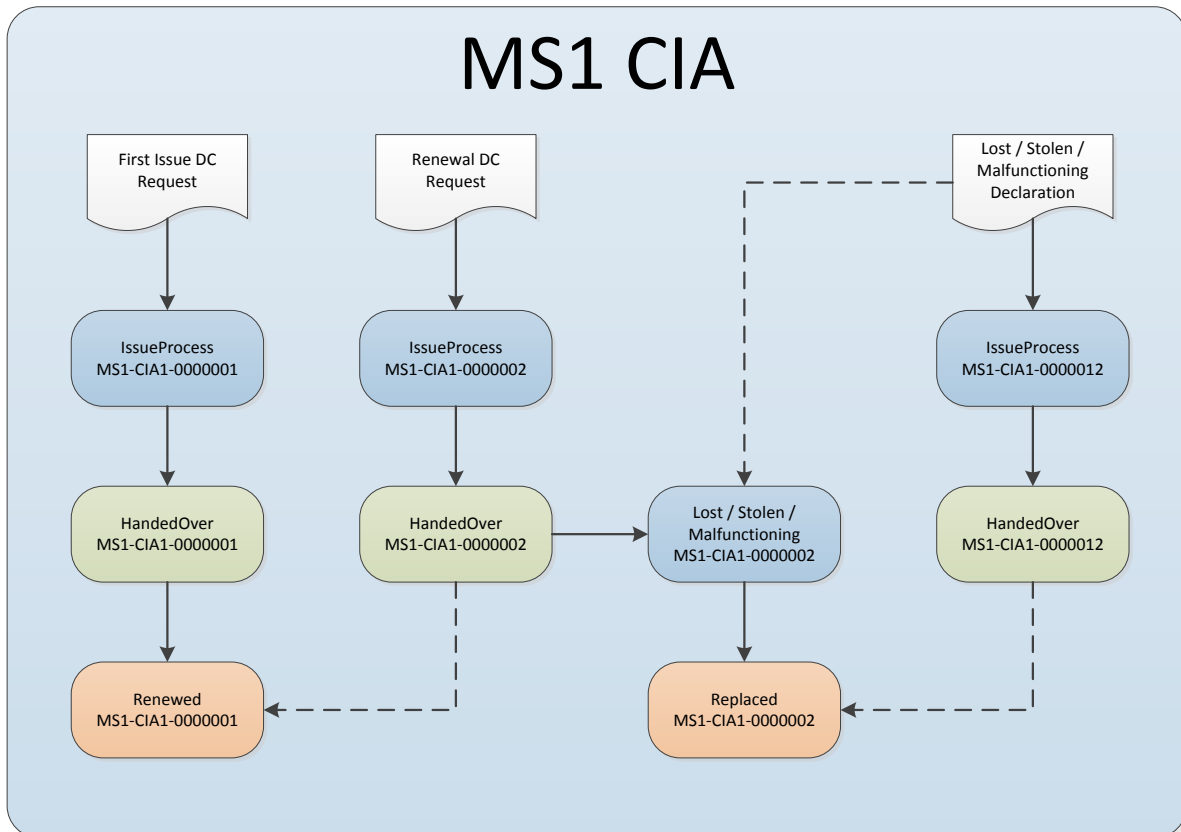


Figure 15 - Card Renewal / Replacement Process

6.5.3. *Card Suspension / Cancellation*

The driver card may reach the end of its life via a number of different routes, initiated either by the driver, a CIA or an enforcement body. The CIA, competent authority or inspectorate may, with a valid reason ([165/2014] Chapter V Article 26.7), withdraw the card for a period of time. When the card is returned to the driver its status must be reset to HandedOver.

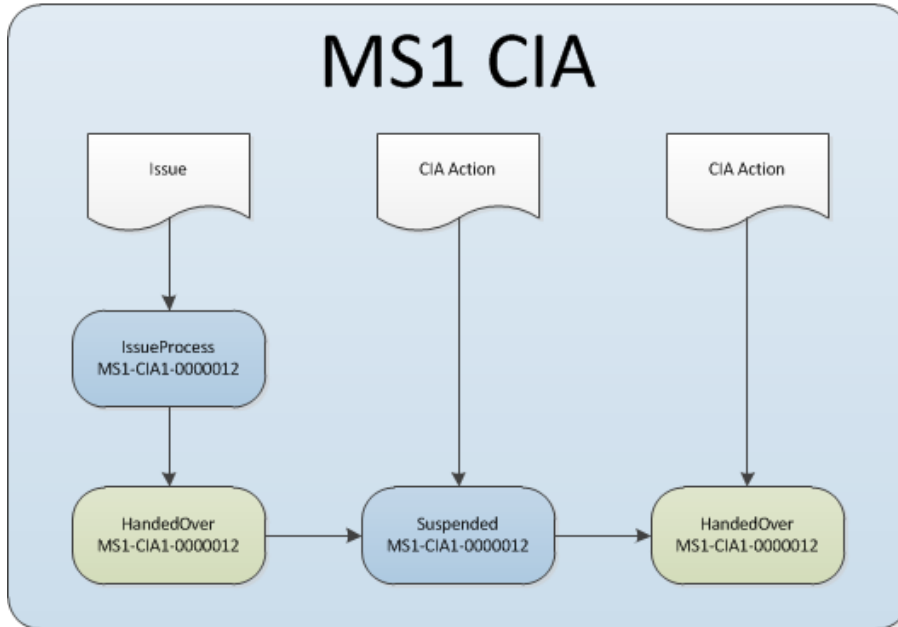


Figure 16 - Card Suspension / Cancellation Process

Alternatively the card may be surrendered by, or permanently removed from, the driver.

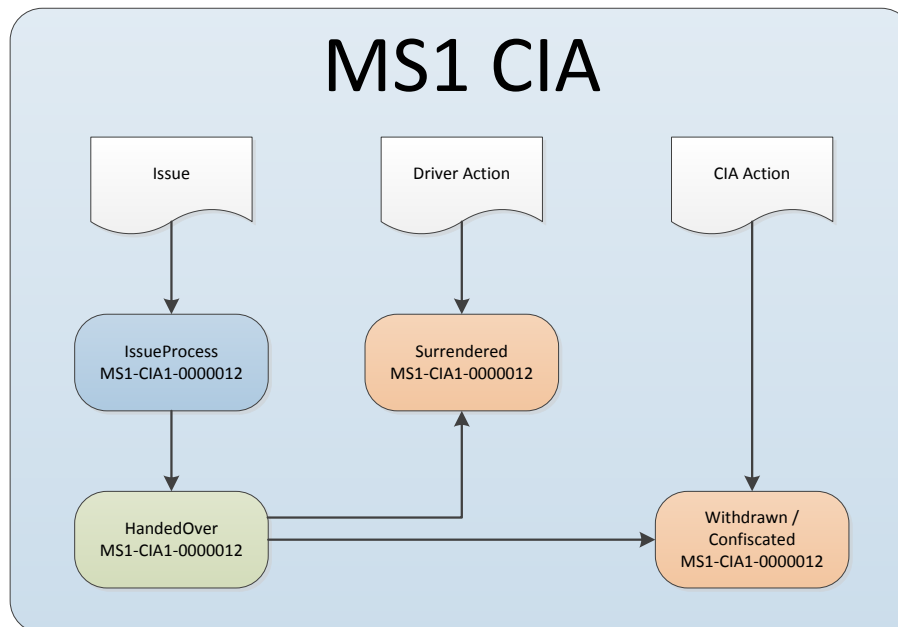


Figure 17 - Card Surrender Process

6.5.4. *Card Exchange*

Finally, the driver may take up normal residence in a new member state and request the CIA of the new member state to issue a new driver card in exchange for his old card.

Whilst the exchange process is in progress his old driver card remains valid. When the new driver card is available the driver must surrender his old driver card and it becomes invalid for driving.

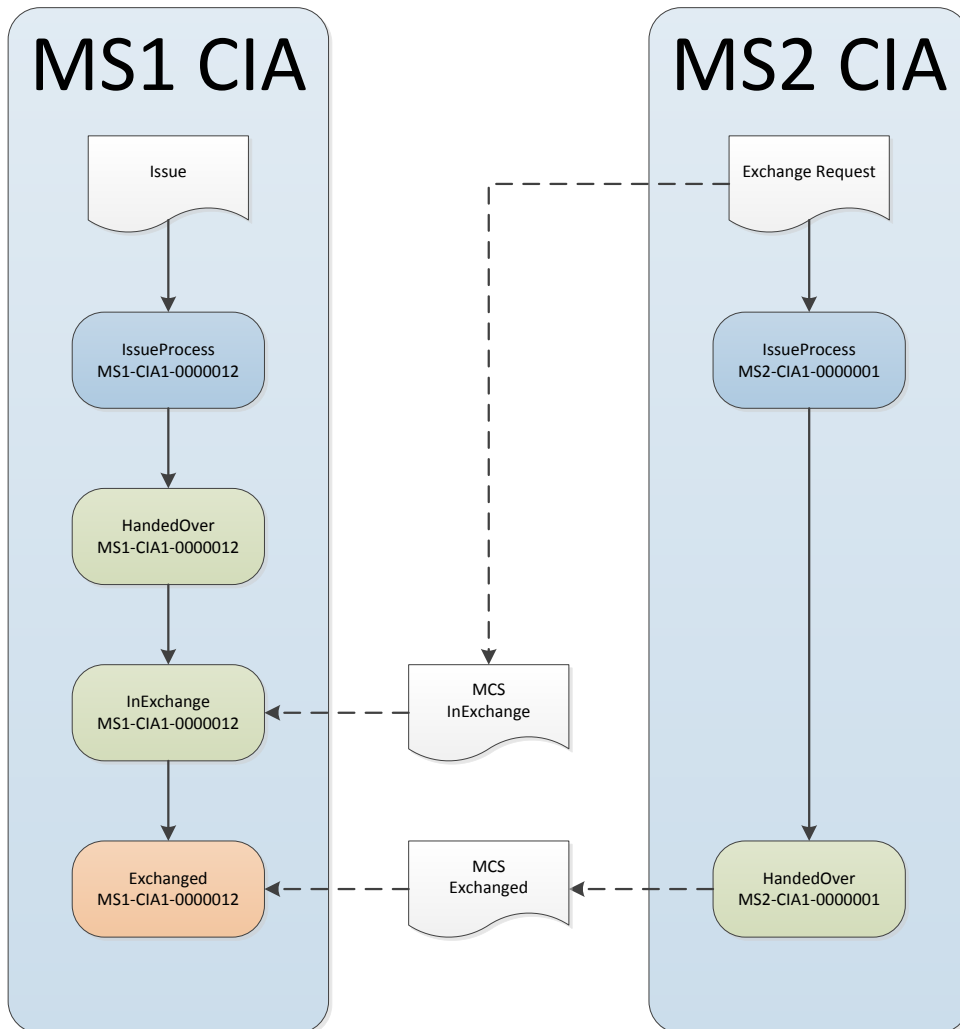


Figure 18 - Card Exchange Process

6.6. Hub Business Validation

Elementary business validation will be performed on the contents of messages passing through the Hub.

6.6.1. *Card Status Card Dates*

The *StartOfValidityDate* and *ExpiryDate* attributes in the CIC and CCS response messages must be logically consistent with the *CardStatus* attribute. For example if the status is *HandedOver* then the start date must be \leq today and the expiry date must be \geq today; or if the status is *Expired* then the expiry date must be $<$ today.

6.6.2. *Name String and Search Keys*

If a CIC response contains a found card then the phonetic key of the found name string must match the phonetic key in the original request (subject to the search method used, see 7.2.1 below).

If the search key in the response does not match the search key in the request then the response will be rejected and an Invalid Message sent to the responder. In the example below the NYSIIS search key for 'Smith' is 'SNATH' and not 'SNAT'.

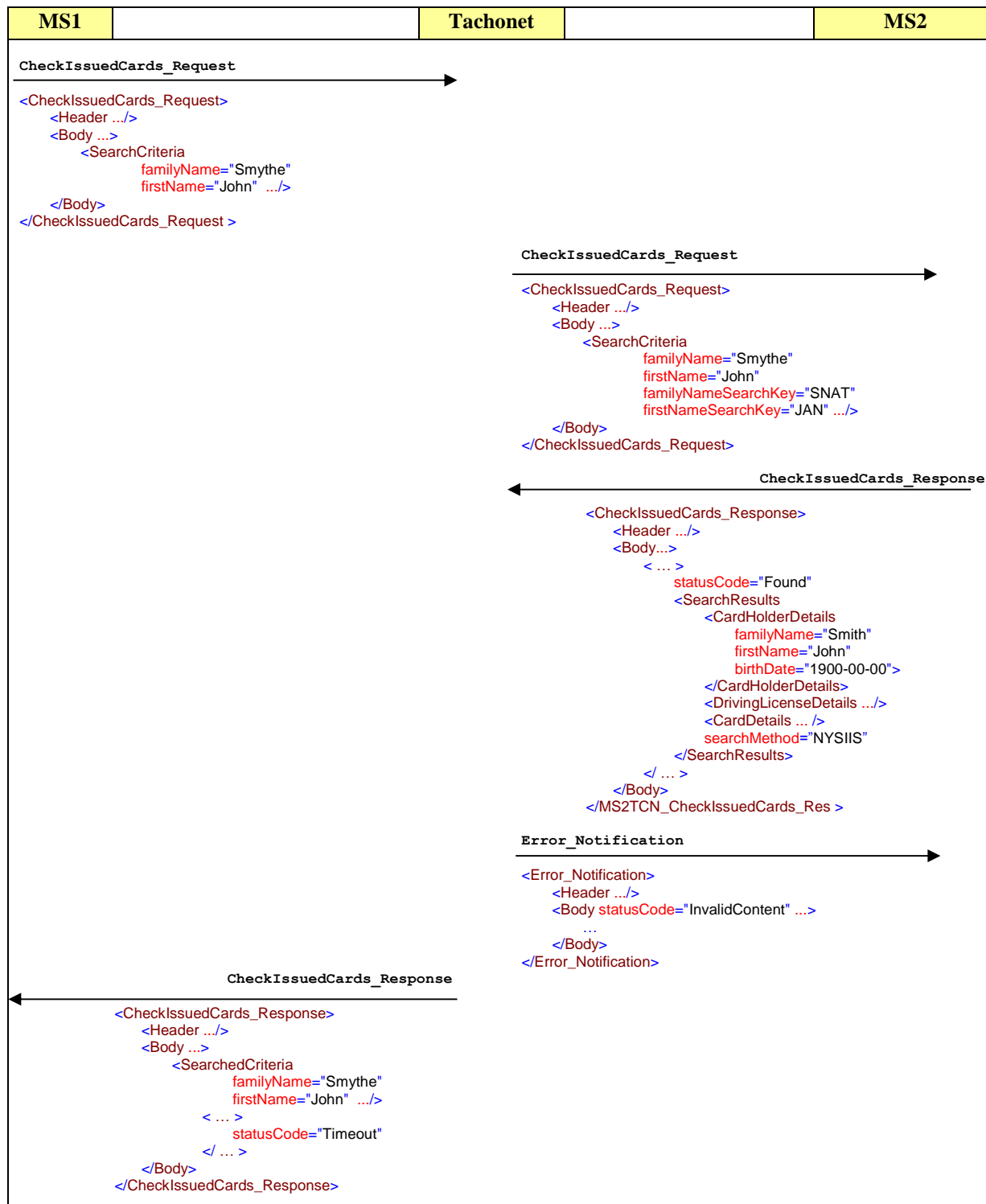


Figure 19 - Name String / Search Key Example

6.6.3. *Search Status and Content*

The status code in the CIC and CCS responses must be logically consistent with the messages content. So, a response with a status of 'found' must contain search results and, conversely, a response with a search status of 'not found' must not contain any search results.

7. TCN SERVICES

The TCN network provides four services to CIAs and other authorised administrations to enable them to fulfil the requirements laid down in [165/2014], [2016/68] and [2017/1503].

A brief description of the TCN messages, given in the following sections, focuses on the business and semantic content of the messages. The technical description of the messages can be found in the HTML and Word documentation ([TCN/XSD-HTML/2016] and [TCN/XSD-Word/2016]) that are distributed alongside this document.

7.1. Messaging Services

Message Name	Description	Reference
CheckIssuedCards_Request	Search for issued tachograph cards by driver name.	7.2
CheckIssuedCards_Response	Search by driver name results.	
IssuedCardDrivingLicence_Request	Notification of a driver card issued against a foreign driving licence.	0
IssuedCardDrivingLicence_Response	Driving licence issuing MS responds to the issued card notification.	
CheckCardStatus_Request	Search for issued tachograph cards by card number.	0
CheckCardStatus_Response	Search by card number results.	
ModifyCardStatus_Request	Notification to a card issuing authority that the status of a tachograph card has changed in another MS.	0
ModifyCardStatus_Response	The card issuing MS advises the notifying MS of the status of the card in the register.	
ModifyCardStatus_Acknowledgement	An acknowledgement of a modify card status notification or response.	
Error_Notification	A message received (by either the Hub or a MS) contains errors.	7.6

The TCN messages are:

Version 1.5	Version 3.0
MS2TCN_CheckIssuedCards_Req	CheckIssuedCards_Request
TCN2MS_CheckIssuedCards_Req	
MS2TCN_CheckIssuedCards_Res	CheckIssuedCards_Response
TCN2MS_CheckIssuedCards_Res	
MS2TCN_CheckCardStatus_Req	CheckCardStatus_Request
TCN2MS_CheckCardStatus_Req	
MS2TCN_CheckCardStatus_Res	CheckCardStatus_Response
TCN2MS_CheckCardStatus_Res	
MS2TCN_IssuedCardDL_Req	IssuedCardDrivingLicence_Request
TCN2MS_IssuedCardDL_Req	
MS2TCN_IssuedCardDL_Res	IssuedCardDrivingLicence_Response
TCN2MS_IssuedCardDL_Res	
MS2TCN_ModCardStatus_Req	ModifyCardStatus_Request
TCN2MS_ModCardStatus_Req	
MS2TCN_ModCardStatus_Res	ModifyCardStatus_Response
TCN2MS_ModCardStatus_Res	
	ModifyCardStatus_Acknowledgement
TCN_Receipt	Error_Notification

7.2. Check Issued Card (CIC)

The CIC service allows a MS to search the tachograph card register in all other countries to determine if a driver has already been issued with a driver card in another country.

The CIC service may be used in either singlecast or broadcast modes, i.e. the requesting MS may direct the Hub to forward the search request to a single counterparty MS or, alternatively, to all MS.

A search request that is required to be broadcast to all other MS is denoted by entering “ZZ” in the *Header/@to* attribute.

The aggregated response that is returned to the requesting MS will have “EU” in the *Header/@from* attribute.

7.2.1. Search Mechanism

A search request message sent by a MS in the CIC workflow has the following relevant information to search for the details of a driver:

- First name (optional)
- Family name (mandatory)
- Date of birth (mandatory)
- Place of birth (optional)
- Driving licence number (optional)
- Driving licence issuing country (optional)

The Hub adds the **first** First Name Search Key and Family Name Search Key to the Request message and sends the search request message to the counterparty MS(s).

Responding MSs must use the date of birth and NYSIIS keys included in the request as the primary search mechanism to find matching records in their database. The search of the database must result in an exact match in order to produce a ‘found’ response – if the optional first name is absent from the search request it must also be absent in the database in order to produce a match. An absent first name does not denote a wildcard search.

Member states may also, optionally, use a customised local search mechanism to enhance the primary search. This secondary mechanism may, for example, use string pattern techniques or use the optional data if it is present in the request.

The responding MS returns the most recent card found in the register. This card may either be not yet issued (e.g. application or rejected); a current, valid driving card; or it may have another status showing that the card has reached end of life (e.g. Expired, Confiscated, Exchanged, etc).

```
<?xml version="1.0" encoding="UTF-8"?>
<CheckIssuedCards_Response xmlns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="37b0929f-eb16-4347-8291-b42285ef5d77"
    workflowId="782dec67-f3ac-48f9-9fc6-9d0cc0123f9b"
    sentAt="2016-01-01T00:00:00Z"
    from="UK"
    to="IE"/>
  <Body businessCaselId="CIC Request Example">
    <SearchCriteria familyName="Creighton-Ward"
      firstName="Penelope"
      dateOfBirth="1939-12-24"
      familyNameSearchKey="CRAGTANWAD"
      firstNameSearchKey="PANALAP">
      <MemberState memberStateCode="UK" statusCode="Found">
        <SearchResults searchMethod="NYSIIS">
          <CardHolderDetails familyName="Creighton-Ward"
            firstName="Penelope"
            dateOfBirth="1939-12-24"
            placeOfBirth="London"/>
          <CardDetails cardNumber="0000000000000000"
            cardStatus="Rejected"
            cardIssuingAuthority="VOSA, PO Box 343, Swansea SA1 2YS"
            cardStartOfValidityDate="1900-01-01"
            cardExpiryDate="1900-01-01"
            cardStatusModifiedDate="2010-09-02T13:42:09Z"
            validForDriving="false"
            temporaryCard="false"/>
          <DrivingLicenceDetails drivingLicenceNumber="CREIG312249P99IR"
            drivingLicenceIssuingCountry="UK"
            drivingLicenceStatus="Valid"
            drivingLicenceIssuingDate="1986-09-30"
            drivingLicenceExpiryDate="2029-09-29"/>
        </SearchResults>
      </MemberState>
    </SearchCriteria>
  </Body>
</CheckIssuedCards_Response>
```

Figure 20 - Search Response with Rejected Card Status

```

<?xml version="1.0" encoding="UTF-8"?>
<CheckIssuedCards_Response xmlns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="37b0929f-eb16-4347-8291-b42285ef5d77"
    workflowId="782dec67-f3ac-48f9-9fc6-9d0cc0123f9b"
    sentAt="2016-01-01T00:00:00Z"
    from="UK"
    to="IE"/>
  <Body businessCaselId="CIC Request Example">
    <SearchCriteria familyName="Creighton-Ward"
      firstName="Penelope"
      dateOfBirth="1939-12-24"
      familyNameSearchKey="CRAGTANWAD"
      firstNameSearchKey="PANALAP"/>
    <MemberState memberStateCode="UK" statusCode="Found">
      <SearchResults searchMethod="NYSIIS">
        <CardHolderDetails familyName="Creighton-Ward"
          firstName="Penelope"
          dateOfBirth="1939-12-24"
          placeOfBirth="London"/>
        <CardDetails cardNumber="C6235B5410196509"
          cardStatus="Expired"
          cardIssuingAuthority="VOSA, PO Box 343, Swansea SA1 2YS"
          cardStartOfValidityDate="2004-09-30"
          cardExpiryDate="2009-09-29"
          cardStatusModifiedDate="2009-09-30T03:42:09Z"
          validForDriving="false"
          temporaryCard="false"/>
        <DrivingLicenceDetails drivingLicenceNumber="CREIG312249P99IR"
          drivingLicenceIssuingCountry="UK"
          drivingLicenceStatus="Valid"
          drivingLicenceIssuingDate="1986-09-30"
          drivingLicenceExpiryDate="2019-09-29"/>
      </SearchResults>
    </MemberState>
  </SearchCriteria>
</Body>
</CheckIssuedCards_Response>

```

Figure 21 - Search Response with Expired Card Status

Further, if a driver card exists in the register but it is not yet active (i.e. the cardStartOfValidityDate is in the future) the responder must return the future card as well as the card immediately prior to the future card as described above (if a prior card exists).

```

<?xml version="1.0" encoding="UTF-8"?>
<CheckIssuedCards_Response xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2" >
  <Header version="3.2"
    technicalId="f148c54c-488f-42bc-bae1-8eb09e8f7a3f"
    workflowId="446940dc-eb68-4462-9424-4893ce051fdb"
    sentAt="2016-01-01T00:00:00Z"
    from="UK"
    to="IE"/>
  <Body businessCaselId="CIC Request Example">
    <SearchCriteria familyName="Creighton-Ward"
      firstName="Penelope"
      dateOfBirth="1939-12-24"
      familyNameSearchKey="CRAGTANWAD"
      firstNameSearchKey="PANALAP">
      <MemberState memberStateCode="UK" statusCode="Found">
        <SearchResults searchMethod="NYSIIS">
          <CardHolderDetails familyName="Creighton-Ward"
            firstName="Penelope"
            dateOfBirth="1939-12-24"
            placeOfBirth="London"/>
          <CardDetails cardNumber="C6235B5410196509"
            cardStatus="HandedOver"
            cardIssuingAuthority="VOSA, PO Box 343, Swansea SA1 2YS"
            cardStartOfValidityDate="2015-09-30"
            cardExpiryDate="2020-09-29"
            cardStatusModifiedDate="2015-09-02T13:42:09Z"
            validForDriving="true"
            temporaryCard="false"/>
          <DrivingLicenceDetails drivingLicenceNumber="CREIG312249P99IR"
            drivingLicenceIssuingCountry="UK"
            drivingLicenceStatus="Valid"
            drivingLicenceIssuingDate="1986-09-30"
            drivingLicenceExpiryDate="2029-09-29"/>
        </SearchResults>
        <SearchResults searchMethod="NYSIIS">
          <CardHolderDetails familyName="Creighton-Ward"
            firstName="Penelope"
            dateOfBirth="1939-12-24"
            placeOfBirth="London"/>
          <CardDetails cardNumber="C6235B5410196509"
            cardStatus="Personalised"
            cardIssuingAuthority="VOSA, PO Box 343, Swansea SA1 2YS"
            cardStartOfValidityDate="2020-09-30"
            cardExpiryDate="2025-09-29"
            cardStatusModifiedDate="2010-09-02T13:42:09Z"
            validForDriving="false"
            temporaryCard="false"/>
          <DrivingLicenceDetails drivingLicenceNumber="CREIG312249P99IR"
            drivingLicenceIssuingCountry="UK"
            drivingLicenceStatus="Valid"
            drivingLicenceIssuingDate="1986-09-30"
            drivingLicenceExpiryDate="2029-09-29"/>
        </SearchResults>
      </MemberState>
      <MemberState memberStateCode="UZ" statusCode="NotAvailable"/>
    </SearchCriteria>
  </Body>
</CheckIssuedCards_Response>

```

Figure 22 - Search Response with Two Cards

7.2.2. *Heartbeat*

The Hub currently uses the messages flowing between MS (and so through the Hub) to assess the availability of the MS and to provide reporting statistics. This method is unreliable when few messages are flowing and can provide false assurances of MS availability or false alerts of MS unavailability.

To improve the monitoring of the TCN network the Hub will emit heartbeat messages and monitor the responses. These heartbeat messages are simply CIC search requests that are 'from' the EU and contain search criteria that will result in a not found response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<CheckIssuedCards_Request xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="afbacb8a-b313-4daa-881b-7b7a65204b69"
    workflowId="446940dc-eb68-4462-9424-4893ce051fdb"
    sentAt="2016-01-01T00:00:00Z"
    timeoutValue="2016-01-01T00:00:20Z"
    from="EU"
    to="IE"/>
  <Body businessCaseId="Heartbeat 2016-01-01T00:00:00Z"
    requestPurpose="Heartbeat"
    requestSource="Hub">
    <SearchCriteria familyName="Heartbeat"
      firstName="Heartbeat"
      dateOfBirth="1900-01-01"
      familyNameSearchKey="HARTBAT"
      firstNameSearchKey="HARTBAT"/>
  </SearchCriteria>
</Body>
</CheckIssuedCards_Request>
```

Figure 23 - Heartbeat Message Example

7.2.3. *CIC Version Mappings*

n.b. In all the mapping tables that follow the path to each attribute contains only the final element name and the attribute name; or just the attribute name. In a number of cases defining the full path from the root of the messages to the attribute makes the tables difficult to read.

7.2.3.1. *v1.5 to v3.0*

The requesting Member State (MS1) sends a v1.5 MS2TCN_CheckIssuedCards_Req message to the Hub. This is converted (as necessary) to a v3.0 CheckIssuedCards_Request message and forwarded to the responding Member State (MS2).

MS2TCN_CheckIssuedCards_Req	CheckIssedCards_Request	Notes
n/a	businessCaseId	GUID.NewGuid()
n/a	requestPurpose	"Other"
n/a	requestSource	"Other"
SearchedDriver/Surname	SearchCriteria/familyName	
SearchedDriver/FirstName	SearchCriteria/firstName	
SearchedDriver/BirthDate	SearchCriteria/dateOfBirth	
SearchedDriver/DrivingLicenseNumber	DrivingLicence/drivingLicenceNumber	
SearchedDriver/DrivingLicenseIssuingNation	DrivingLicence/drivingLicenceIssuingCountry	Convert UNECE to ISO
SearchedDriver/IssuingMemberStateCode	n/a	Used to populate Header/to
n/a	SearchCriteria/placeOfBirth	Optional – empty
n/a	SearchCriteria/familyNameSearchKey	Populated by Hub with NYSIIS search key
n/a	SearchCriteria/firstNameSearchKey	Populated by Hub with NYSIIS search key

MS2 replies with a v3.0 CheckIssuedCards_Response message that is converted at the Hub to a v1.5 TCN2MS_CheckIssuedCards_Res message and forwarded to MS1.

CheckIssuedCards_Response	TCN2MS_CheckIssuedCards_Res	Notes
businessCaseId	n/a	Ignored / lost
SearchCriteria/familyName	SearchedDriver/Surname	
SearchCriteria/firstName	SearchedDriver/FirstName	
SearchCriteria/dateOfBirth	SearchedDriver/BirthDate	
SearchCriteria/placeOfBirth	n/a	Ignored / lost
n/a	SearchedDriver/DrivingLicenseNumber	Taken from the MS2TCN_CheckIssuedCards_Req
n/a	SearchedDriver/DrivingLicenseIssuingNation	Taken from the MS2TCN_CheckIssuedCards_Req
n/a	SearchedDriver/IssuingMemberStateCode	Taken from the MS2TCN_CheckIssuedCards_Req
n/a	SearchedDriver/SearchStatusCode	Calculated by the Hub
n/a	SearchedDriver/SearchStatusMessage	Optional – empty
SearchCriteria/familyNameSearchKey	n/a	Ignored / lost
SearchCriteria/firstNameSearchKey	n/a	Ignored / lost
MemberState/memberStateCode	MemberState/MemberStateCode	
MemberState/statusCode	MemberState/MSstatusCode	
MemberState/statusMessage	MemberState/MSstatusMessage	
CardHolderDetails/familyName	DriverDetails/Surname	
CardHolderDetails/firstName	DriverDetails/FirstName	
CardHolderDetails/dateOfBirth	DriverDetails/BirthDate	
CardHolderDetails/placeOfBirth	DriverDetails/PlaceOfBirth	

CheckIssuedCards_Response	TCN2MS_CheckIssedCards_Res	Notes
DrivingLicenceDetails/drivingLicenceNumber	DrivingLicenseDetails/DLNumber	
DrivingLicenceDetails/drivingLicenceIssuingCountry	DrivingLicenseDetails/DLIssuingNation	Convert ISO to UNECE
DrivingLicenceDetails/drivingLicenceStatus	DrivingLicenseDetails/DLStatus	
DrivingLicenceDetails/drivingLicenceIssuingDate	DrivingLicenseDetails/DLIssueDate	
DrivingLicenceDetails/drivingLicenceExpiryDate	n/a	Ignored / lost
n/a	CardDetails/AdditionalCardStatus	Optional – empty
CardDetails/cardIssuingAuthority	CardDetails/CIA	
CardDetails/cardStatusModifiedDate	CardDetails/StatusModifiedAt	
CardDetails/cardNumber	CardDetails/CardNumber	
CardDetails/cardStatus	CardDetails/CardStatus	
CardDetails/cardStartOfValidityDate	CardDetails/StartOfValidityDate	
CardDetails/cardExpiryDate	CardDetails/ExpiryDate	
CardDetails/validForDriving	CardDetails/CardValidity	
CardDetails/temporaryCard	n/a	Ignored / lost
WorkshopDetails/*	n/a	Ignored / Lost
n/a	MSContactInfo/*	Optional – empty

7.2.3.2. v3.0 to v1.5

The requesting Member State (MS1) sends a v3.0 CheckIssuedCards_Request message to the Hub. This is converted (as necessary) to a v1.5 TCN2MS_CheckIssuedCards_Req message and forwarded to the responding Member State (MS2).

CheckIssuedCards_Request	TCN2MS_CheckIssuedCards_Req	Notes
businessCaseld	n/a	Ignored / lost
requestPurpose	n/a	
requestSource	n/a	
SearchCriteria/familyName	SearchedDriver/Surname	
SearchCriteria/firstName	SearchedDriver/FirstName	
SearchCriteria/dateOfBirth	SearchedDriver/BirthDate	
SearchCriteria/placeOfBirth	n/a	Ignored / lost
SearchCriteria/familyNameSearchKey	SearchedDriver/SurnameSearchKey	Populated by the Hub with Phonex search key
SearchCriteria/firstNameSearchKey	SearchedDriver/FirstNameSearchKey	Populated by the Hub with Phonex search key
DrivingLicence/drivingLicenceNumber	SearchedDriver/DrivingLicenseNumber	
DrivingLicence/drivingLicenceIssuingCountry	SearchedDriver/DrivingLicenseIssuingNation	Convert ISO to UNECE

MS2 replies with a v1.5 MS2TCN_CheckIssuedCards_Res message that is converted at the Hub to a v3.0 CheckIssuedCards_Response message and forwarded to MS1.

MS2TCN_CheckIssuedCards_Res	CheckIssuedCards_Response	Notes
n/a	businessCaseId	Taken from the CheckIssuedCards_Request
SearchedDriver/Surname	SearchCriteria/familyName	
SearchedDriver/FirstName	SearchCriteria/firstName	
SearchedDriver/BirthDate	SearchCriteria/dateOfBirth	
n/a	SearchCriteria/placeOfBirth	Optional – empty
SearchedDriver/DrivingLicenseNumber	n/a	Ignored / lost
SearchedDriver/DrivingLicenseIssuingNation	n/a	Ignored / lost
SearchedDriver/SurnameSearchKey	SearchCriteria/familyNameSearchKey	Validated by Hub and Phonex key replaced with NYSIS key
SearchedDriver/FirstNameSearchKey	SearchCriteria/firstNameSearchKey	Validated by Hub and Phonex key replaced with NYSIS key
Header/from	MemberState/memberStateCode	Convert UNECE to ISO
SearchedDriver/SearchStatusCode	MemberState/statusCode	
SearchedDriver/SearchStatusMessage	MemberState/statusMessage	
DriverDetails/Surname	CardHolderDetails/familyName	
DriverDetails/FirstName	CardHolderDetails/firstName	
DriverDetails/BirthDate	CardHolderDetails/dateOfBirth	
DriverDetails/PlaceOfBirth	CardHolderDetails/placeOfBirth	
DrivingLicenseDetails/DLNumber	DrivingLicenceDetails/drivingLicenceNumber	“1.5 Member State”
DrivingLicenseDetails/DLIssuingNation	DrivingLicenceDetails/drivingLicenceIssuingCountry	Convert UNECE to ISO or “QQ” if unable to determine.
DrivingLicenseDetails/DLStatus	DrivingLicenceDetails/drivingLicenceStatus	

MS2TCN_CheckIssuedCards_Res	CheckIssuedCards_Response	Notes
DrivingLicenseDetails/DLIssueDate	DrivingLicenceDetails/drivingLicenceIssuingDate	
n/a	DrivingLicenceDetails/drivingLicenceExpiryDate	Optional – empty
CardDetails/AdditionalCardStatus	n/a	Ignored / lost
CardDetails/CIA	CardDetails/cardIssuingAuthority	
CardDetails/StatusModifiedAt	CardDetails/cardStatusModifiedDate	
CardDetails/AdditionalStatusModifiedAt	n/a	Ignored / lost
CardDetails/CardNumber	CardDetails/cardNumber	
CardDetails/CardStatus	CardDetails/cardStatus	
CardDetails/StartOfValidityDate	CardDetails/cardStartOfValidityDate	
CardDetails/ExpiryDate	CardDetails/cardExpiryDate	
CardDetails/CardValidity	CardDetails/validForDriving	
n/a	CardDetails/temporaryCard	Optional – empty
n/a	WorkshopDetails/*	Choice – empty

7.2.4. *CIC Request Message Sample*

```
<?xml version="1.0" encoding="UTF-8"?>
<CheckIssuedCards_Request xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="afbacb8a-b313-4daa-881b-7b7a65204b69"
    workflowId="446940dc-eb68-4462-9424-4893ce051fdb"
    sentAt="2016-01-01T00:00:00Z"
    from="IE"
    to="ZZ"/>
  <Body businessCaselId="CIC Request Example"
    requestPurpose="Issue"
    requestSource="CIA">
    <SearchCriteria familyName="Creighton-Ward" firstName="Penelope" dateOfBirth="1965-09-30"/>
    <DrivingLicence drivingLicenceNumber="CREIG312249P99IR" drivingLicenceIssuingCountry="UK"/>
  </SearchCriteria>
</Body>
</CheckIssuedCards_Request>
```

Figure 24 - CIC HTTP Request Message Example

```
Content-Type: text/xml;charset=UTF-8
SOAPAction: "Call_CIC_Workflow"

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:CheckIssuedCards_Request>
      <ns:Header version="3.2"
        technicalId="6767d62c-497c-4090-9045-09ec48775748"
        workflowId="3678b769-7714-48bf-b67a-5966d63984d4"
        sentAt="2018-04-05T13:14:58Z"
        from="IE"
        to="ZZ"/>
      <ns:Body businessCaselId="CIC Request Example" requestPurpose="Issue" requestSource="CIA">
        <ns:SearchCriteria familyName="Creighton-Ward" firstName="Penelope" dateOfBirth="1965-09-30"/>
      </ns:Body>
    </ns:CheckIssuedCards_Request>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure 25 - CIC SOAP Request Message Example

7.2.5. *CIC Response MessageSample*

```

<?xml version="1.0" encoding="UTF-8"?>
<CheckIssuedCards_Response xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2" >
  <Header version="3.2"
    technicalId="f148c54c-488f-42bc-bae1-8eb09e8f7a3f"
    workflowId="446940dc-eb68-4462-9424-4893ce051fdb"
    sentAt="2016-01-01T00:00:00Z"
    from="EU"
    to="IE"/>
  <Body businessCaselId="CIC Request Example">
    <SearchCriteria familyName="Creighton-Ward"
      firstName="Penelope"
      dateOfBirth="1939-12-24"
      familyNameSearchKey="CRAGTANWAD"
      firstNameSearchKey="PANALAP">
      <MemberState memberStateCode="AD" statusCode="NotFound"/>
      <MemberState memberStateCode="AL" statusCode="NotFound"/>

      <!-- Multiple other countries -->

      <MemberState memberStateCode="SE" statusCode="NotFound"/>
      <MemberState memberStateCode="SI" statusCode="NotFound"/>
      <MemberState memberStateCode="SK" statusCode="NotFound"/>
      <MemberState memberStateCode="SM" statusCode="NotAvailable"/>
      <MemberState memberStateCode="TJ" statusCode="NotAvailable"/>
      <MemberState memberStateCode="TM" statusCode="NotAvailable"/>
      <MemberState memberStateCode="TR" statusCode="NotAvailable"/>
      <MemberState memberStateCode="UA" statusCode="NotFound"/>
      <MemberState memberStateCode="UK" statusCode="Found">
        <SearchResults searchMethod="NYSIIS">
          <CardHolderDetails familyName="Creighton-Ward"
            firstName="Penelope"
            dateOfBirth="1939-12-24"
            placeOfBirth="London"/>
          <CardDetails cardNumber="C6235B5410196509"
            cardStatus="HandedOver"
            cardIssuingAuthority="VOSA, PO Box 343, Swansea SA1 2YS"
            cardStartOfValidityDate="2015-09-30"
            cardExpiryDate="2020-09-29"
            cardStatusModifiedDate="2015-09-02T13:42:09Z"
            validForDriving="true"
            temporaryCard="false"/>
          <DrivingLicenceDetails drivingLicenceNumber="CREIG312249P99IR"
            drivingLicenceIssuingCountry="UK"
            drivingLicenceStatus="Valid"
            drivingLicenceIssuingDate="1986-09-30"
            drivingLicenceExpiryDate="2019-09-29"/>
        </SearchResults>
      </MemberState>
      <MemberState memberStateCode="UZ" statusCode="NotAvailable"/>
    </SearchCriteria>
  </Body>
</CheckIssuedCards_Response>

```

Figure 26 – CIC HTTP Response Message Example

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <CheckIssuedCards_Response xmlns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
      <Header version="3.2"
        technicalId="bb298efe-08c5-42f4-8856-f8310c712d1e"
        workflowId="17913211-4d63-4a9c-9a59-b1343cd54394"
        sentAt="2018-04-05T12:00:40Z"
        timeoutValue="2018-04-05T12:00:40Z"
        from="EU"
        to="IE"/>
      <Body businessCaseId="CIC Request Example">
        <SearchCriteria familyName="Creighton-Ward"
          firstName="Penelope"
          dateOfBirth="1965-09-30"
          familyNameSearchKey="CRAGTANWAD"
          firstNameSearchKey="PANALAP">
          <MemberState memberStateCode="AT" statusCode="NotFound"/>
          <MemberState memberStateCode="AL" statusCode="NotFound"/>

          <!-- Multiple other countries -->

          <MemberState memberStateCode="SE" statusCode="NotFound"/>
          <MemberState memberStateCode="SK" statusCode="NotAvailable"/>
          <MemberState memberStateCode="SI" statusCode="NotFound"/>
          <MemberState memberStateCode="RS" statusCode="NotFound"/>
          <MemberState memberStateCode="TJ" statusCode="NotFound"/>
          <MemberState memberStateCode="TM" statusCode="NotFound"/>
          <MemberState memberStateCode="TR" statusCode="NotFound"/>
          <MemberState memberStateCode="UA" statusCode="NotFound"/>
          <MemberState memberStateCode="UK" statusCode="Found">
            <SearchResults searchMethod="NYSIIS">
              <CardHolderDetails familyName="Creighton-Ward"
                firstName="Penelope"
                dateOfBirth="1965-09-30"
                placeOfBirth="London"/>
              <CardDetails cardNumber="C6235B5410196509"
                cardStatus="HandedOver"
                cardIssuingAuthority="VOSA, PO Box 343, Swansea SA1 2YS"
                cardStartOfValidityDate="2015-09-30"
                cardExpiryDate="2020-09-29"
                cardStatusModifiedDate="2015-09-29T00:00:00Z"
                validForDriving="true"/>
              <DrivingLicenceDetails drivingLicenceNumber="CREIG312249P99IR"
                drivingLicenceIssuingCountry="UK"
                drivingLicenceStatus="Valid"
                drivingLicenceIssuingDate="1982-09-30"/>
            </SearchResults>
          </MemberState>
          <MemberState memberStateCode="UZ" statusCode="NotFound"/>
        </SearchCriteria>
      </Body>
    </CheckIssuedCards_Response>
  </s:Body>
</s:Envelope>

```

Figure 27 - CIC SOAP Response Message Example

7.3. Issued Card Driving Licence (ICDL)

After having issued a card to a driver who has provided a foreign driving licence with the card application, the card issuing MS must inform the driving licence issuing country that a card has been issued against a driving licence issued in that country.

The purpose of this message is to prevent fraud by ensuring that a driver who has received a card in a 'foreign' country does not still possess an active driver card if he returns to his 'home' country and requests a card using a national driving licence. 'Foreign' and 'home' in this context is defined in terms of the driving licence issuing country, not the applicant's nationality.

It is important to note that receipt and processing of ICDL notifications has no relationship to the driving licence register. If a CIA has access to the driving licence register then additional validation

may be performed when an ICDL notification is received (i.e. is the driving licence still valid); but it is not a requirement.

7.3.1. Notifying Member State

The responsibility of the notifying (requesting) Member State is to simply issue an ICDL notification (and handle the associated response) when a driver card has been issued to an applicant with a foreign driving licence. The ICDL notification is sent to the CIA of the country that issued the driving licence (assuming that that country is connected to Tachonet).

7.3.2. Responding Member State

The processing of received ICDL notifications is dependent on the business logic that a member state wishes to implement when issuing driver cards. Part of the processing of issuing driver cards is ensuring that the applicant does not hold additional valid driver cards issued in another Member State; this search has two basic forms:

- For applicants with a foreign driving licence CIAs must make a CIC search for 100% of the applications.
- For applicants with a national driving licence CIAs must make a CIC search for at least 30% of the applications – if ICDL data is recorded. The ICDL data is searched using the driving licence number; by using the NYSIIS keys; and by using any applicable custom name search algorithm.
- For applicants with a national driving licence CIAs must make a CIC search 100% of the applications – if ICDL data is not recorded.

The benefits of recording the received ICDL notifications and then validating applications with national driving licences against this data are:

- Reduced network traffic (only a minimum of 30% of applications with a national driving licence need to be checked via a CIC search)
- The ability to continue issuing applications with a national driving licence when the Tachonet Hub or counterparty Member State is unavailable (but the requirement to check 30% of the applications remains).

The following diagram represents the ICDL notification and driver card application workflows.

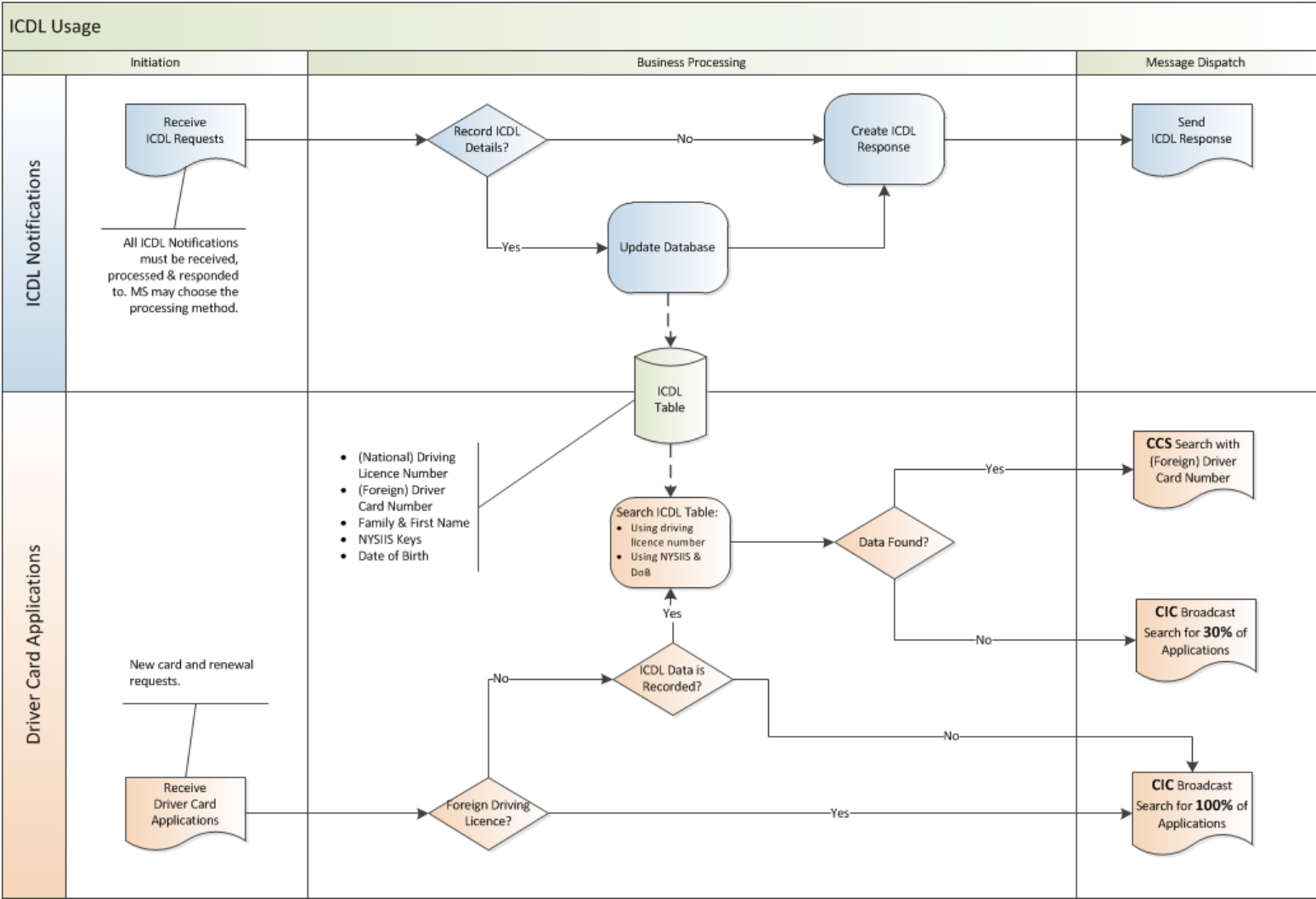


Figure 28 - ICDL Message Usage

7.3.3. *ICDL Version Mappings*

7.3.3.1. *v1.5 to v3.0*

The requesting Member State (MS1) sends a v1.5 MS2TCN_IssuedCardDL_Req message to the Hub. This is converted (as necessary) to a v3.0 IssuedCardDrivingLicence_Request message and forwarded to the responding Member State (MS2).

MS2TCN_IssuedCardDL_Req	IssuedCardDrivingLicence_Request	Notes
n/a	businessCaseId	GUID.NewGuid()
n/a	requestPurpose	"Other"
n/a	requestSource	"Other"
IssuedCard/IssuedCardNumber	IssuedCard/driverCardNumber	
IssuedCard/IssuingMemberStateCode	Header/from	Convert UNECE to ISO
IssuedCard/DrivingLicenseNumber	IssuedCard/drivingLicenceNumber	
IssuedCard/DrivingLicenseIssuingNation	Header/to	Convert UNECE to ISO
n/a	IssuedCard/familyName	"1.5 Member State"
n/a	IssuedCard/firstName	Optional – empty
n/a	IssuedCard/dateOfBirth	"01/01/0001"
n/a	IssuedCard/placeOfBirth	Optional – empty
n/a	IssuedCard/familyNameSearchKey	Optional – empty
n/a	IssuedCard/firstNameSearchKey	Optional – empty

MS2 replies with a v3.0 IssuedCardDrivingLicence_Response message that is converted at the Hub to a v1.5 TCN2MS_IssuedCardDL_Res message and forwarded to MS1.

IssuedCardDrivingLicence_Response	TCN2MS_IssuedCardDL_Res	Notes
businessCaseId	n/a	Ignored / lost
IssuedCard/driverCardNumber	IssuedCard/IssuedCardNumber	
Header/from	IssuedCard/IssuingMemberStateCode	Convert ISO to UNECE
IssuedCard/drivingLicenceNumber	IssuedCard/DrivingLicenseNumber	
Header/to	IssuedCard/DrivingLicenseIssuingNation	Convert ISO to UNECE
IssuedCard/statusCode	IssuedCard/IssuedCardDLStatusCode	
IssuedCard/statusMessage	IssuedCard/IssuedCardDLStatusMessage	
n/a	MSCContactInfo/*	Optional – empty

7.3.3.2. v3.0 to v1.5

The requesting Member State (MS1) sends a v3.0 IssuedCardDrivingLicence_Request message to the Hub. This is converted (as necessary) to a v1.5 TCN2MS_IssuedCardDL_Req message and forwarded to the responding Member State (MS2).

IssuedCardDrivingLicence_Request	TCN2MS_IssuedCardDL_Req	Notes
businessCaseId	n/a	Ignored / lost
requestPurpose	n/a	Ignored / lost
requestSource	n/a	Ignored / lost
IssuedCard/driverCardNumber	IssuedCard/IssuedCardNumber	
Header/from	IssuedCard/IssuingMemberStateCode	Convert ISO to UNECE
IssuedCard/drivingLicenceNumber	IssuedCard/DrivingLicenseNumber	
IssuedCard/familyName	n/a	Ignored / lost
IssuedCard/firstName	n/a	
IssuedCard/dateOfBirth	n/a	
IssuedCard/placeOfBirth	n/a	
IssuedCard/familyNameSearchKey	n/a	
IssuedCard/firstNameSearchKey	n/a	

MS2 replies with a v1.5 MS2TCN_CheckIssuedCards_Res message that is converted at the Hub to a v3.0 CheckIssuedCards_Response message and forwarded to MS1.

MS2TCN_IssuedCardDL_Res	IssuedCardDrivingLicence_Response	Notes
n/a	businessCaseId	Taken from the IssuedCardDrivingLicence_Request
IssuedCard/IssuedCardNumber	IssuedCard/driverCardNumber	
IssuedCard/IssuingMemberStateCode	Header/from	Convert UNECE to ISO
IssuedCard/DrivingLicenseNumber	IssuedCard/drivingLicenceNumber	
IssuedCard/IssuedCardDLStatusCode	IssuedCard/statusCode	
IssuedCard/IssuedCardDLStatusMessage	IssuedCard/statusMessage	

7.3.4. *ICDL Request Message Sample*

```
<?xml version="1.0" encoding="UTF-8"?>
<IssuedCardDrivingLicence_Request xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="5f30cebf-2bf9-421c-99d6-d8fa47553617"
    workflowId="2ba5f2c4-8417-4f3a-96a7-102737311ec4"
    sentAt="2016-01-01T00:00:00Z"
    from="UK"
    to="IE"/>
  <Body businessCaselId="ICDL Request Example" requestPurpose="Issue" requestSource="CIA">
    <IssuedCard driverCardNumber="C6235B5410196509"
      drivingLicenceNumber="CREIG312249P99IR"
      familyName="Creighton-Ward"
      firstName="Penelope"
      dateOfBirth="1965-09-30"
      familyNameSearchKey="CRAGTANWAD"
      firstNameSearchKey="PANALAP"/>
  </Body>
</IssuedCardDrivingLicence_Request>
```

Figure 29 – ICDL HTTP Request Message Example

```
Content-Type: text/xml;charset=UTF-8
SOAPAction: "Call_ICDL_Workflow"

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:IssuedCardDrivingLicence_Request>
      <ns:Header version="3.2"
        technicalId="00435273-965a-4d70-aae1-6a2798572ced"
        workflowId="9a21ed26-1d72-4def-9710-a8d549113008"
        sentAt="2018-04-05T13:12:51Z"
        from="IE"
        to="UK"/>
      <ns:Body businessCaselId="ICDL Request Example" requestPurpose="Issue" requestSource="CIA">
        <ns:IssuedCard driverCardNumber="C6235B5410196509"
          drivingLicenceNumber="CREIG312249P99IR"
          familyName="Creighton-Ward"
          dateOfBirth="1965-09-30"/>
      </ns:Body>
    </ns:IssuedCardDrivingLicence_Request>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure 30 – ICDL SOAP Request Message Example

7.3.5. *ICDL Response Message Sample*

```
<?xml version="1.0" encoding="UTF-8"?>
<IssuedCardDrivingLicence_Response xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="af0ba6ec-c7e0-452c-886f-1fe3ff65da82"
    workflowId="2ba5f2c4-8417-4f3a-96a7-102737311ec4"
    sentAt="2016-01-01T00:00:12Z"
    from="IE"
    to="UK"/>
  <Body businessCaselId="ICDL Request Example">
    <IssuedCard driverCardNumber="C6235B5410196509"
      drivingLicenceNumber="CREIG312249P99IR"
      statusCode="OK"/>
  </Body>
</IssuedCardDrivingLicence_Response>
```

Figure 31 - ICDL HTTP Response Message Example

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <IssuedCardDrivingLicence_Response xmlns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
      <Header version="3.2"
        technicalId="bd5137b9-113e-4b7d-ad2b-324cf2d8af50"
        workflowId="9a21ed26-1d72-4def-9710-a8d549113008"
        sentAt="2018-04-05T11:12:53Z"
        from="UK"
        to="IE"/>
      <Body businessCaseId="ICDL Request Example">
        <IssuedCard driverCardNumber="C6235B5410196509"
          drivingLicenceNumber="CREIG312249P991R"
          statusCode="OK"/>
      </Body>
    </IssuedCardDrivingLicence_Response>
  </s:Body>
</s:Envelope>

```

Figure 32 - ICDL SOAP Response Message Example

7.4. Check Card Status (CCS)

The CIC and CCS services return the same data in the response (if a tachograph card is found). However, a CCS search will return the details for just the card that has been requested, whereas a CIC search will return the details of all the tachograph cards which have been issued to drivers whose name matches the CIC search keys; and so the requesting MS must determine which of the found tachograph cards is relevant.

A CCS search is to be preferred if the tachograph card number is known.

The CCS request contains the card number, and the response details the data contained in the tachograph register: holder details, card details and driving licence details.

7.4.1. *CCS Version Mappings*

7.4.1.1. *v1.5 to v3.0*

The requesting Member State (MS1) sends a v1.5 MS2TCN_CheckCardStatus_Req message to the Hub. This is converted (as necessary) to a v3.0 CheckCardStatus_Request message and forwarded to the responding Member State (MS2).

MS2TCN_CheckCardStatus_Req	CheckCardStatus_Request	Notes
n/a	businessCaseId	GUID.NewGuid()
n/a	requestPurpose	"Other"
n/a	requestSource	"Other"
SearchCriteria/cardNumber	SearchCriteria/cardNumber	
SearchCriteria/IssuingMemberStateCode	Header/to	Convert UNECE to ISO

MS2 replies with a v3.0 CheckCardStatus_Response message that is converted at the Hub to a v1.5 TCN2MS_CheckCardStatus_Res message and forwarded to MS1.

CheckCardStatus_Response	TCN2MS_CheckCardStatus_Res	Notes
businessCaseId	n/a	Ignored / lost
SearchCriteria/cardNumber	SearchCriteria/CardNumber	
Header/from	SearchCriteria/IssuingMemberStateCode	Convert ISO to UNECE
SearchCriteria/statusCode	SearchCriteria/SearchStatusCode	
SearchCriteria/statusMessage	SearchCriteria/SearchStatusMessage	
MSContactInfo/*	n/a	Ignored / lost
CardHolderDetails/familyName	DriverDetails/Surname	
CardHolderDetails/firstName	DriverDetails/FirstName	
CardHolderDetails/dateOfBirth	DriverDetails/BirthDate	
CardHolderDetails/placeOfBirth	DriverDetails/PlaceOfBirth	
CardDetails/cardNumber	n/a	See SearchCriteria/CardNumber
CardDetails/cardStatus	CardDetails/CardStatus	
CardDetails/cardIssuingAuthority	CardDetails/CIA	
CardDetails/cardStartOfValidityDate	CardDetails/StartOfValidityDate	
CardDetails/cardExpiryDate	CardDetails/ExpiryDate	
CardDetails/cardStatusModifiedDate	CardDetails/StatusModifiedAt	
CardDetails/validForDriving	CardDetails/CardValidity	
CardDetails/temporaryCard	n/a	Ignored / lost
n/a	CardDetails/AdditioanlCardStatus	Optional – empty
n/a	CardDetails/AdditionalStatusModifiedAt	Optional – empty
DrivingLicenceDetails/drivingLicenceNumber	DrivingLicenseDetails/DLNumber	

CheckCardStatus_Response	TCN2MS_CheckCardStatus_Res	Notes
DrivingLicenceDetails/drivingLicenceIssuingCountry	DrivingLicenseDetails/DLIssuingNation	Convert ISO to UNECE
DrivingLicenceDetails/drivingLicenceStatus	DrivingLicenseDetails/DLStatus	
DrivingLicenceDetails/drivingLicenceIssuingDate	DrivingLicenseDetails/DLIssueDate	
DrivingLicenceDetails/drivingLicenceExpiryDate	n/a	Ignored / lost
WorkshopDetails/workshopName	WorkshopDetails/WorkshopName	
WorkshopDetails/workshopAddress	WorkshopDetails/WorkshopAddress	
CardHolderDetails/familyName	WorkshopDetails/Surname	
CardHolderDetails/firstName	WorkshopDetails/FirstName	
CardHolderDetails/dateOfBirth	WorkshopDetails/BirthDate	

7.4.1.2. v3.0 to v1.5

The requesting Member State (MS1) sends a v3.0 CheckCardStatus_Request message to the Hub. This is converted (as necessary) to a v1.5 TCN2MS_CheckCardStatus_Req message and forwarded to the responding Member State (MS2).

CheckCardStatus_Request	TCN2MS_CheckCardStatus_Req	Notes
businessCaseId	n/a	Ignored / lost
requestPurpose	n/a	Ignored / lost
requestSource	n/a	Ignored / lost
SearchCriteria/cardNumber	SearchCriteria/cardNumber	

MS2 replies with a v1.5 MS2TCN_CheckCardStatus_Res message that is converted at the Hub to a v3.0 CheckCardStatus_Response message and forwarded to MS1.

MS2TCN_CheckCardStatus_Res	CheckCardStatus_Response	Notes
n/a	businessCaseId	Taken from the CheckCardStatus_Request
SearchCriteria/CardNumber	SearchCriteria/cardNumber	
SearchCriteria/SearchStatusCode	SearchCriteria/statusCode	
SearchCriteria/SearchStatusMessage	SearchCriteria/statusMessage	
n/a	CardDetails/cardNumber	Taken from the CheckCardStatus_Request
CardDetails/CIA	CardDetails/cardIssuingAuthority	
CardDetails/CardStatus	CardDetails/cardStatus	
CardDetails/StatusModifiedAt	CardDetails/cardStatusModifiedDate	
CardDetails/AdditioanlCardStatus	n/a	Ignored / lost
CardDetails/AdditionalStatusModifiedAt	n/a	Ignored / lost
CardDetails/StartOfValidityDate	CardDetails/cardStartOfValidityDate	
CardDetails/ExpiryDate	CardDetails/cardExpiryDate	
CardDetails/CardValidity	CardDetails/validForDriving	
n/a	CardDetails/temporaryCard	Optional – empty
DriverDetails/Surname	CardHolderDetails/familyName	
DriverDetails/FirstName	CardHolderDetails/firstName	
DriverDetails/BirthDate	CardHolderDetails/dateOfBirth	
DriverDetails/PlaceOfBirth	CardHolderDetails/placeOfBirth	
WorkshopDetails/WorkshopName	WorkshopDetails/workshopName	
WorkshopDetails/WorkshopAddress	WorkshopDetails/workshopAddress	
WorkshopDetails/Surname	CardHolderDetails/familyName	

MS2TCN_CheckCardStatus_Res	CheckCardStatus_Response	Notes
WorkshopDetails/FirstName	CardHolderDetails/firstName	
WorkshopDetails/BirthDate	CardHolderDetails/dateOfBirth	
DrivingLicenseDetails/DLNumber	DrivingLicenceDetails/drivingLicenceNumber	
DrivingLicenseDetails/DLIssuingNation	DrivingLicenceDetails/drivingLicenceIssuingCountry	Convert UNECE to ISO
DrivingLicenseDetails/DLStatus	DrivingLicenceDetails/drivingLicenceStatus	
DrivingLicenseDetails/DLIssueDate	DrivingLicenceDetails/drivingLicenceIssuingDate	
n/a	DrivingLicenceDetails/drivingLicenceExpiryDate	

7.4.2. *CCS Request Message Sample*

```

<?xml version="1.0" encoding="UTF-8"?>
<CheckCardStatus_Request xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="1d061fe6-5dd1-4325-bed3-3f6f388a538e"
    workflowId="782dec67-f3ac-48f9-9fc6-9d0cc0123f9b"
    sentAt="2016-01-01T00:00:00Z"
    from="IE"
    to="UK"/>
  <Body businessCaseld="CCS Request Example" requestPurpose="Control" requestSource="RSI">
    <SearchCriteria cardNumber="C6235B5410196509"/>
  </Body>
</CheckCardStatus_Request>

```

Figure 33 - CCS HTTP Request Message Example

```

Content-Type: text/xml;charset=UTF-8
SOAPAction: "Call_CCS_Workflow"

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:CheckCardStatus_Request>
      <ns:Header version="3.2"
        technicalId="19bef15a-c2ca-4358-97a0-9e0e7ddb87e7"
        workflowId="7c6e5959-e0da-4aa2-8dbb-0b6570884336"
        sentAt="2018-04-05T13:12:47Z"
        from="IE"
        to="UK"/>
      <ns:Body businessCaseld="CCS Request Example" requestPurpose="Control" requestSource="RSI">
        <ns:SearchCriteria cardNumber="C6235B5410196509"/>
      </ns:Body>
    </ns:CheckCardStatus_Request>
  </soapenv:Body>
</soapenv:Envelope>

```

Figure 34 - CCS SOAP Request Message Example

7.4.3. *CCS Response Message Sample*

```

<?xml version="1.0" encoding="UTF-8"?>
<CheckCardStatus_Response xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="37b0929f-eb16-4347-8291-b42285ef5d77"
    workflowId="782dec67-f3ac-48f9-9fc6-9d0cc0123f9b"
    sentAt="2016-01-01T00:00:09Z"
    from="UK"
    to="IE"/>
  <Body businessCaselId="CCS Request Example">
    <SearchCriteria cardNumber="C6235B5410196509"
      statusCode="Found">
      <SearchResults>
        <CardHolderDetails familyName="Creighton-Ward"
          firstName="Penelope"
          dateOfBirth="1939-12-24"
          placeOfBirth="London"/>
        <CardDetails cardNumber="C6235B5410196509"
          cardStatus="Replaced"
          cardIssuingAuthority="VOSA, PO Box 343, Swansea SA1 2YS"
          cardStartOfValidityDate="2015-09-30"
          cardExpiryDate="2020-09-29"
          cardStatusModifiedDate="2015-09-02T13:42:09Z"
          validForDriving="true"
          temporaryCard="false"/>
        <DrivingLicenceDetails drivingLicenceNumber="CREIG312249P99IR"
          drivingLicenceIssuingCountry="UK"
          drivingLicenceStatus="Valid"
          drivingLicenceIssuingDate="1986-09-30"
          drivingLicenceExpiryDate="2019-09-29"/>
      </SearchResults>
    </SearchCriteria>
  </Body>
</CheckCardStatus_Response>

```

Figure 35 - CCS HTTP Response Message Example

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <CheckCardStatus_Response xmlns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
      <Header version="3.2"
        technicalId="50552e73-6d90-4266-be7b-a9110c9e31a3"
        workflowId="7c6e5959-e0da-4aa2-8dbb-0b6570884336"
        sentAt="2018-04-05T11:12:49Z"
        from="UK"
        to="IE"/>
      <Body businessCaseId="CCS Request Example">
        <SearchCriteria cardNumber="C6235B5410196509" statusCode="Found">
          <SearchResults>
            <CardHolderDetails familyName="Creighton-Ward"
              firstName="Penelope"
              dateOfBirth="1965-09-30"
              placeOfBirth="London"/>
            <CardDetails cardNumber="C6235B5410196509"
              cardStatus="HandedOver"
              cardIssuingAuthority="VOSA, PO Box 343, Swansea SA1 2YS"
              cardStartOfValidityDate="2015-09-30"
              cardExpiryDate="2020-09-29"
              cardStatusModifiedDate="2015-09-29T00:00:00Z"
              validForDriving="true"/>
            <DrivingLicenceDetails drivingLicenceNumber="CREIG312249P99IR"
              drivingLicenceIssuingCountry="UK"
              drivingLicenceStatus="Valid"
              drivingLicenceIssuingDate="1982-09-30"/>
          </SearchResults>
        </SearchCriteria>
      </Body>
    </CheckCardStatus_Response>
  </s:Body>
</s:Envelope>

```

Figure 36 - CCS SOAP Response Message Example

7.5. Modify Card Status (MCS)

Any modification of the status of a card (such as stolen, lost, defective, suspended or exchange) must be sent to the MS that issued the card.

The MCS workflow is split into two, disconnected message flows. Firstly an MCS request is sent from the notifying MS to the issuing MS and the latter acknowledges receipt of the MCS request by responding with an MCS acknowledgement message.

The issuing MS then follows any national procedures to update (or reject) the status of the tachograph card and when this is complete the changed (or unchanged) status of the tachograph card is returned to the notifying MS via an MCS response message. The notifying MS acknowledges receipt of the MCS response with an MCS acknowledgement message.

The *workflowId* (and *businessCaseId*) will be the same on all four messages.

7.5.1. Automatic Register Updates

A two-part disconnected message exchange has been introduced as the majority of MS do not wish to automatically apply an update to their card register. Nonetheless, if MS do automatically apply status updates to their register the two part message exchange must be adhered to:

- An MCS request is received and an acknowledgment created and sent.
- The automatic register update is applied.
- An MCS response is created and sent and the corresponding acknowledgement received and processed.

7.5.2. MCS Message Flow

The diagram below describes the MCS workflow. If no response is received from MS2 within the 10 calendar days then the Hub will create a Timeout response and send it to MS1.

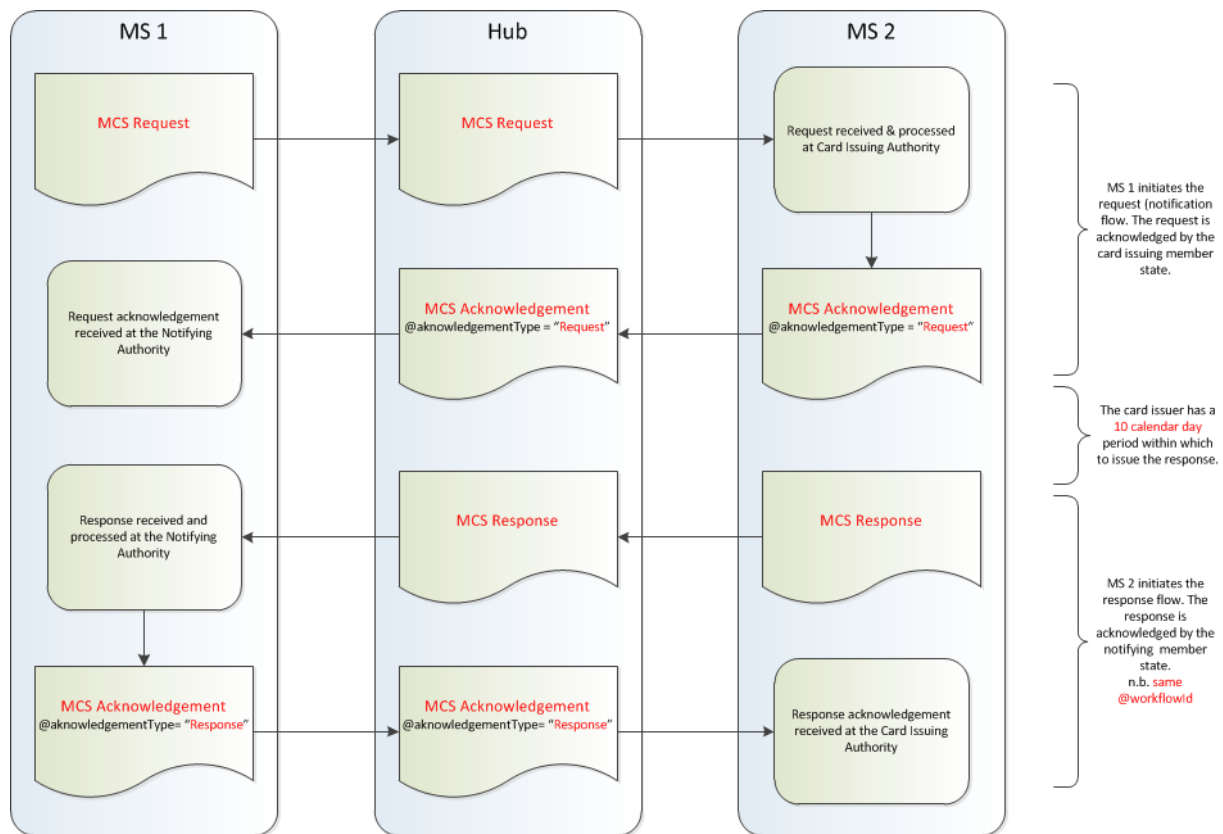


Figure 37 - MCS Message Flow

7.5.3. Synchronous and Asynchronous MCS Messages

As described in 7.5.2, the exchange of MCS messages occurs in two disconnected stages. If a MS chooses to use synchronous message exchange with the Hub then the two stages will be:

- If the notifying member state wants to use synchronous connections the MCS request (notification) and associated acknowledgement will be one synchronous exchange.
- If the responding member state wants to use synchronous connections the MCS response and associated acknowledgement will be on synchronous exchange.

There is no synchronous relationship between the MCS request and the MCS response.

7.5.4. MCS Conversion Mappings

MCS conversion is more complex than the other message types as the workflow itself has changed, not just the format and content of the messages. The Hub must not only map the message content between versions but it must also create and absorb v3.0 MCS acknowledgements and responses as necessary.

7.5.4.1. v1.5 to v3.0

The requesting Member State (MS1) sends a v1.5 MS2TCN_ModCardStatus_Req message to the Hub. This is converted to a v3.0 ModifyCardStatus_Request message and forwarded to the responding Member State (MS2).

As MS2 is using the v3.0 workflow it will first generate a ModifyCardStatus_Acknowledgement (@acknowledgementType="Request") within the 10 second timeout period. The Hub must use this acknowledgement to generate the v1.5 TCN2MS_ModCardStatus_Res as MS2 has up to 7 calendar days to create the actual response ModifyCardStatus_Response message, by which time the original MS2TCN_ModCardStatus_Req at MS1 will have timed out.

MS2 will, at some point within 7 calendar days, create a ModifyCardStatus_Response message and send it to the Hub. MS1 is no longer expecting a TCN2MS_ModCardStatus_Res message as the Hub created and forwarded this message when the ModifyCardStatus_Acknowledgement was received from MS2. Therefore, the Hub must absorb the ModifyCardStatus_Response received from MS2 and create and send a ModifyCardStatus_Acknowledgement (@acknowledgementType="Response") to MS2.

The following diagram shows the message flow described above.

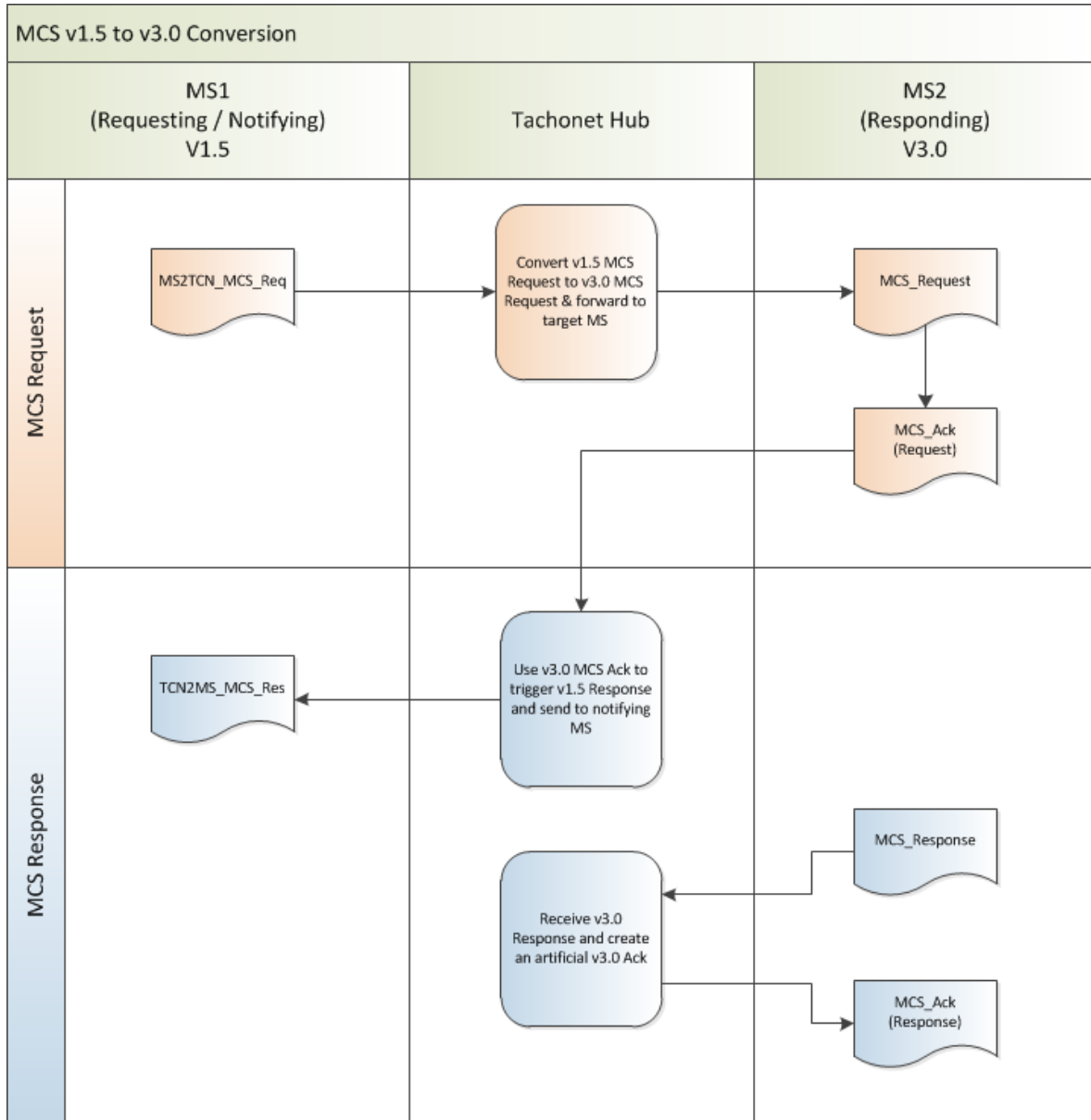


Figure 38 - MCS Message Flow 1.5 to 3.0 Version Mapping

The original v1.5 MS2TCN_ModCardStatus_Req from MS1 is mapped to the v3.0 ModifyCardStatus_Request as follows:

MS2TCN_ModCardStatus_Req	ModifyCardStatus_Request	Notes
n/a	businessCaseId	GUID.NewGuid()
n/a	requestPurpose	"Other"
n/a	requestSource	"Other"
CardDetails/CardNumber	CardDetails/driverCardNumber	
CardDetails/NewCardStatus	CardDetails/newDriverCardStatus	
CardDetails/Reason	n/a	Ignored / lost
CardDetails/ReasonCode	n/a	Ignored / lost
CardDetails/StatusModifiedAt	CardDetails/driverCardStatusModifiedDate	
CardDetails/IssuingMemberStateCode	Header/to	
DeclaredBy/Origin	DeclaredBy/organizationName	Use for both name & address
DeclaredBy/Origin	DeclaredBy/organizationAddress	Use for both name & address
DeclaredBy/Surname	DeclaredBy/contactFamilyName	
DeclaredBy/FirstName	DeclaredBy/contactFirstName	
DeclaredBy/Phone	DeclaredBy/contactPhone	
DeclaredBy/Fax	n/a	Ignored / lost
DeclaredBy/Email	DeclaredBy/contactEmail	

MS2 responds with a v3.0 ModifyCardStatus_Acknowledgement and this is mapped to a v1.5 TCN2MS_ModCardStatus_Res message and forwarded to MS1:

ModifyCardStatus_Acknowledgement	TCN2MS_ModCardStatus_Res	Notes
businessCaseld	n/a	Ignored / lost
acknowledgementType	n/a	Ignored / lost
CardDetails/driverCardNumber	CardDetails/CardNumber	
n/a	CardDetails/NewCardStatus	Taken from the MS2TCN_ModCardStatus_Req
n/a	CardDetails/Reason	Taken from the MS2TCN_ModCardStatus_Req
n/a	CardDetails/ReasonCode	Taken from the MS2TCN_ModCardStatus_Req
n/a	CardDetails/StatusModifiedAt	Taken from the MS2TCN_ModCardStatus_Req
CardDetails/statusCode	CardDetails/ModStatusCode	Taken from the MS2TCN_ModCardStatus_Req (OK, Timeout, NotAvailable)
CardDetails/statusMessage	CardDetails/ModStatusMessage	Taken from the MS2TCN_ModCardStatus_Req
n/a	DeclaredBy/Origin	Taken from the MS2TCN_ModCardStatus_Req
n/a	DeclaredBy/Surname	Taken from the MS2TCN_ModCardStatus_Req
n/a	DeclaredBy/FirstName	Taken from the MS2TCN_ModCardStatus_Req
n/a	DeclaredBy/Phone	Taken from the MS2TCN_ModCardStatus_Req
n/a	DeclaredBy/Fax	Taken from the MS2TCN_ModCardStatus_Req
n/a	DeclaredBy/Email	Taken from the MS2TCN_ModCardStatus_Req
n/a	MSContactInfo/*	Optional – empty

7.5.4.2. v3.0 to v1.5

The requesting Member State (MS1) sends a v3.0 ModifyCardStatus_Request message to the Hub. This is converted to a v1.5 TCN2MS_ModCardStatus_Req message and forwarded to the responding Member State (MS2).

As MS2 is using the v1.5 workflow it will generate only a MS2TCN_ModCardStatus_Res message and from this message the Hub must create the v3.0 ModifyCardStatus_Acknowledgement (@acknowledgementType="Request") and v3.0 ModifyCardStatus_Response messages. The acknowledgement will be sent to MS1 followed immediately by the response message.

MS1 will complete the workflow by sending a ModifyCardStatus_Acknowledgement (@acknowledgementType="Response")

The following diagram shows the message flow described above.

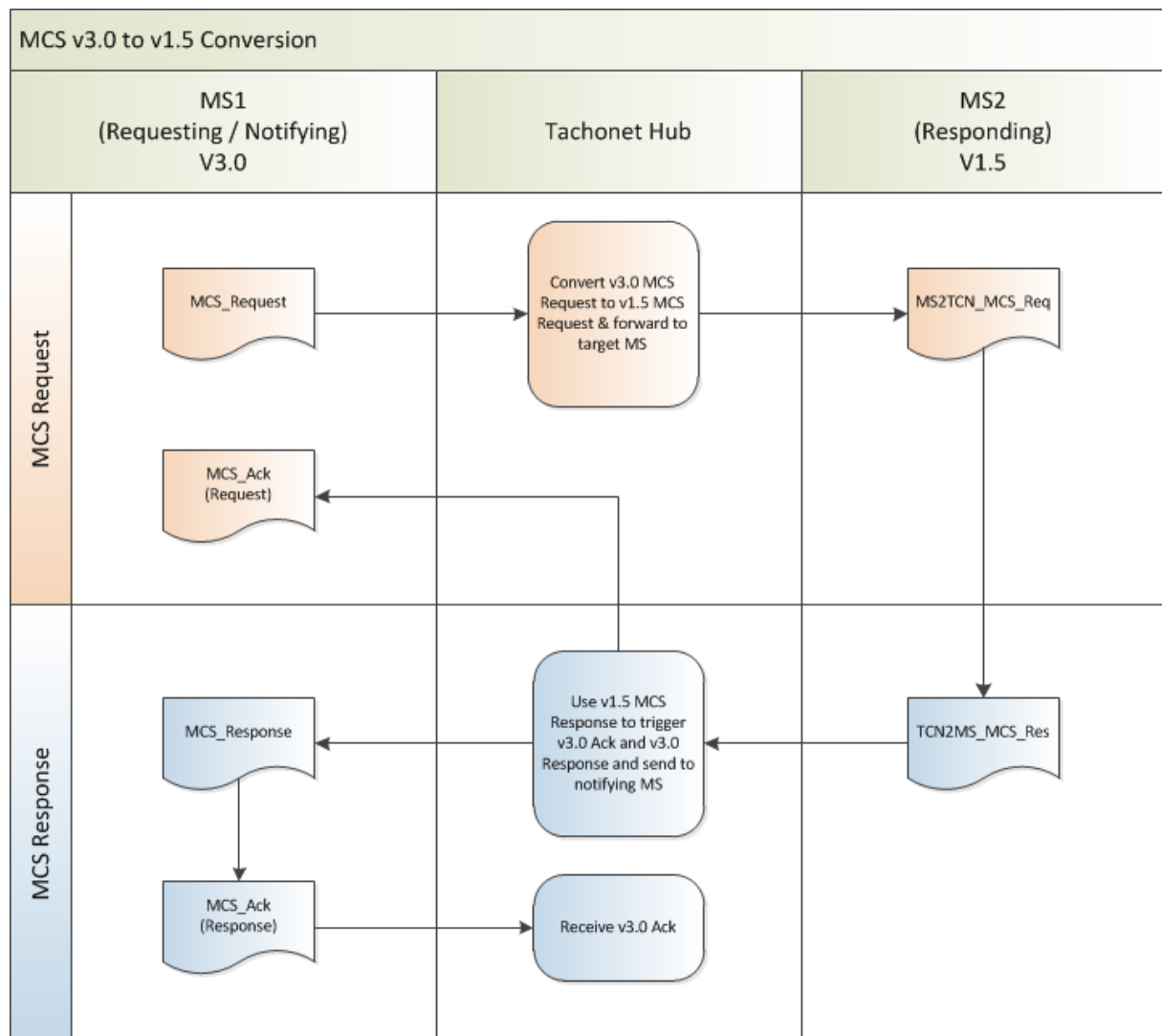


Figure 39 - MCS Message Flow 3.0 to 1.5 Version Mapping

The original v3.0 ModifyCardStatus_Request from MS1 is mapped to the v1.5 TCN2MS_ModCardStatus_Req as follows:

ModifyCardStatus_Request	TCN2MS_ModCardStatus_Req	Notes
businessCaselId	n/a	Ignored / lost
requestPurpose	n/a	Ignored / lost
requestSource	n/a	Ignored / lost
CardDetails/driverCardNumber	CardDetails/CardNumber	
CardDetails/newDriverCardStatus	CardDetails/NewCardStatus	
n/a	CardDetails/Reason	optional – empty
n/a	CardDetails/ReasonCode	optional – empty
driverCardStatusModifiedDate	CardDetails/StatusModifiedAt	
DeclaredBy/organizationName	DeclaredBy/Origin	
DeclaredBy/organizationAddress	n/a	Ignored / lost
DeclaredBy/contactFamilyName	DeclaredBy/Surname	
DeclaredBy/contactFirstName	DeclaredBy/FirstName	
DeclaredBy/contactPhone	DeclaredBy/Phone	
n/a	DeclaredBy/Fax	optional – empty
DeclaredBy/contactEmail	DeclaredBy/Email	
n/a	DeclaredBy/MemberStateCode	Header/From

MS2 responds with a v1.5 MS2TCN_ModCardStstus_Res and this is used to trigger a v3.0 ModifyCardStatus_Acknowledgement (Request) message forwarded to MS1:

MS2TCN_ModCardStatus_Res	ModifyCardStatus_Ack	Notes
n/a	businessCaseId	Taken from the ModifyCardStatus_Request
n/a	acknowledgementType	"Request"
n/a	CardDetails/DriverCardNumber	Taken from the ModifyCardStatus_Request
n/a	CardDetails/statusCode	"OK"
n/a	CardDetails/statusMessage	Optional – empty

The same v1.5 MS2TCN_ModCardStstus_Res is then used to map to the v3.0 ModifyCardStatus_Response message forwarded to MS1:

MS2TCN_ModCardStatus_Res	ModifyCardStatus_Response	Notes
n/a	businessCaseId	Taken from the ModifyCardStatus_Request
CardDetails/CardNumber	CardDetails/driverCardNumber	
CardDetails/NewCardStatus	CardDetails/driverCardStatusCode	
CardDetails/ModStatusCode	CardDetails/statusCode	OK = Updated CardNumberNotFound = NotFound Timeout = Timeout else NotUpdated
CardDetails/ModStatusMessage	CardDetails/statusMessage	
CardDetails/Reason	n/a	Ignored / lost
CardDetails/ReasonCode	n/a	Ignored / lost
CardDetails/StatusModifiedAt	n/a	Ignored / lost
DeclaredBy/Origin	n/a	Ignored / lost
DeclaredBy/Surname	n/a	Ignored / lost
DeclaredBy/FirstName	n/a	Ignored / lost
DeclaredBy/Phone	n/a	Ignored / lost
DeclaredBy/Fax	n/a	Ignored / lost
DeclaredBy/Email	n/a	Ignored / lost
DeclaredBy/MemberStateCode	n/a	Ignored / lost

7.5.5. *MCS Request Message Sample*

```

<?xml version="1.0" encoding="UTF-8"?>
<ModifyCardStatus_Request xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="5bf6cf87-62a2-4a15-b881-69386d429da8"
    workflowId="8e8d9829-9874-4316-b4ef-630de4e6969f"
    sentAt="2016-01-01T00:00:00Z"
    from="IE"
    to="UK"/>
  <Body businessCaselId="MCS Request Example" requestPurpose="Exchange" requestSource="CIA">
    <CardDetails driverCardNumber="C6235B5410196509"
      newDriverCardStatus="Malfunctioning"
      driverCardStatusModifiedDate="2015-12-31T17:05:00Z">
      <DeclaredBy organizationName="Road Safety Authority"
        organizationAddress="Clonfert House, Bride Street, Loughrea, Co. Galway"
        contactEmail="digitaltacho@rsa.ie"
        contactPhone="091 872600/ 1890 40 60 40"/>
    </CardDetails>
  </Body>
</ModifyCardStatus_Request>

```

Figure 40 - MCS HTTP Request Message Example

```

Content-Type: text/xml;charset=UTF-8
SOAPAction: "Call_MCS_Req_Workflow"

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:ModifyCardStatus_Request>
      <ns:Header version="3.2"
        technicalId="05936385-a8e1-4e2a-a7c5-41dc683da457"
        workflowId="d0e544f2-3fc6-4453-b5e4-712412ece25f"
        sentAt="2018-04-05T15:13:31Z"
        from="IE"
        to="UK"/>
      <ns:Body businessCaselId="MCS Request OK" requestPurpose="Exchange" requestSource="CIA">
        <ns:CardDetails driverCardNumber="C6235B5410196509"
          newDriverCardStatus="Malfunctioning"
          driverCardStatusModifiedDate="2015-12-31T17:05:00Z">
          <ns:DeclaredBy organizationName="Road Safety Authority"
            organizationAddress="Clonfert House, Bride Street, Loughrea, Co. Galway"
            contactEmail="digitaltacho@rsa.ie"
            contactPhone="091 872600/ 1890 40 60 40"/>
        </ns:CardDetails>
      </ns:Body>
    </ns:ModifyCardStatus_Request>
  </soapenv:Body>
</soapenv:Envelope>

```

Figure 41 - MCS SOAP Request Message Example

7.5.6. MCS Response Message Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<ModifyCardStatus_Response xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="9a118c2e-483d-4c09-a488-5a1764daa5d3"
    workflowId="8e8d9829-9874-4316-b4ef-630de4e6969f"
    sentAt="2016-01-03T15:05:54Z"
    from="UK"
    to="IE"/>
  <Body businessCaselId="MCS Request Example">
    <CardDetails driverCardNumber="C6235B5410196509"
      driverCardStatusCode="Malfunctioning"
      statusCodeUpdated="Yes"/>
  </Body>
</ModifyCardStatus_Response>
```

Figure 42 – MCS HTTP Response Message Example

```
Content-Type: text/xml;charset=UTF-8
SOAPAction: "Call_MCS_Res_Workflow"

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:ModifyCardStatus_Response>
      <ns:Header version="3.2"
        technicalId="01918b15-b3f6-4745-9ab4-a9d3a3a74e08"
        workflowId="d0e544f2-3fc6-4453-b5e4-712412ece25f"
        sentAt="2018-04-05T15:55:36Z"
        from="UK"
        to="IE"/>
      <ns:Body businessCaselId="MCS Request OK">
        <ns:CardDetails driverCardNumber="DC235B5410196509"
          driverCardStatusCode="InExchange"
          statusCode="Updated"/>
      </ns:Body>
    </ns:ModifyCardStatus_Response>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure 43 - MCS SOAP Response Message Example

7.5.7. MCS Request Acknowledgement Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<ModifyCardStatus_Acknowledgement xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="a396f8a8-9fc1-4cf6-97eb-1cf53185c840"
    workflowId="8e8d9829-9874-4316-b4ef-630de4e6969f"
    sentAt="2016-01-01T00:00:09Z"
    from="UK"
    to="IE"/>
  <Body businessCaselId="MCS Request Example" acknowledgementType="Request">
    <CardDetails driverCardNumber="C6235B5410196509" statusCode="OK"/>
  </Body>
</ModifyCardStatus_Acknowledgement>
```

Figure 44 - MCS HTTP Request Acknowledgement Message Example

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ModifyCardStatus_Acknowledgement xmlns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
      <Header version="3.2"
        technicalId="35c2d2ee-b101-44ce-a097-765bcc0711e1"
        workflowId="d0e544f2-3fc6-4453-b5e4-712412ece25f"
        sentAt="2018-04-05T13:13:50Z"
        from="UK"
        to="IE"/>
      <Body businessCaselId="MCS Request OK" acknowledgementType="Request">
        <CardDetails driverCardNumber="C6235B5410196509" statusCode="OK"/>
      </Body>
    </ModifyCardStatus_Acknowledgement>
  </s:Body>
</s:Envelope>

```

Figure 45 - MCS SOAP Request Acknowledgement Message Example

7.5.8. MCS Response Acknowledgement Sample

```

<?xml version="1.0" encoding="UTF-8"?>
<ModifyCardStatus_Acknowledgement xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2">
  <Header version="3.2"
    technicalId="83c9cb5f-cab5-4a72-b700-e3374b0d2e0d"
    workflowId="8e8d9829-9874-4316-b4ef-630de4e6969f"
    sentAt="2016-01-03T15:06:07Z"
    from="IE"
    to="UK"/>
  <Body businessCaselId="MCS Request Example " acknowledgementType="Response">
    <CardDetails driverCardNumber=" C6235B5410196509" statusCode="OK"/>
  </Body>
</ModifyCardStatus_Acknowledgement>

```

Figure 46 – MCS HTTP Response Acknowledgement Message Example

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ModifyCardStatus_Acknowledgement xmlns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
      <Header version="3.2"
        technicalId="35c2d2ee-b101-44ce-a097-765bcc0711e1"
        workflowId="d0e544f2-3fc6-4453-b5e4-712412ece25f"
        sentAt="2018-04-05T15:55:41Z "
        from="IE"
        to="UK"/>
      <Body businessCaselId="MCS Request OK" acknowledgementType="Response">
        <CardDetails driverCardNumber="C6235B5410196509" statusCode="OK"/>
      </Body>
    </ModifyCardStatus_Acknowledgement>
  </s:Body>
</s:Envelope>

```

Figure 47 – MCS SOAP Response Acknowledgement Message Example

7.6. Error Notification (EN)

When the Hub or a MS receives a message that cannot be processed (because of incorrect data or format), or when other technical issues prevent a message from being processed, the sending MS is notified of the error via an EN message.

7.6.1. Error workflowId

The *workflowId* of the message that caused the error should be used in the ErrorNotification message. However, it may not be possible to determine the *workflowId* from the incoming message – it may be corrupted for example. In this case a new GUID should be created and assigned to the *workflowId* attribute of the ErrorNotification message.

If the message is not readable as XML (it's is not correctly formatted) then an attempt should be made to extract the workflowId from the original message by treating it as a string and using a regular expression (workflowId="(.*?)"). Only if this fails should a new workflowId be created for the ErrorNotification.

The following diagrams show the Error_Notification message for failures at different points in the workflow:

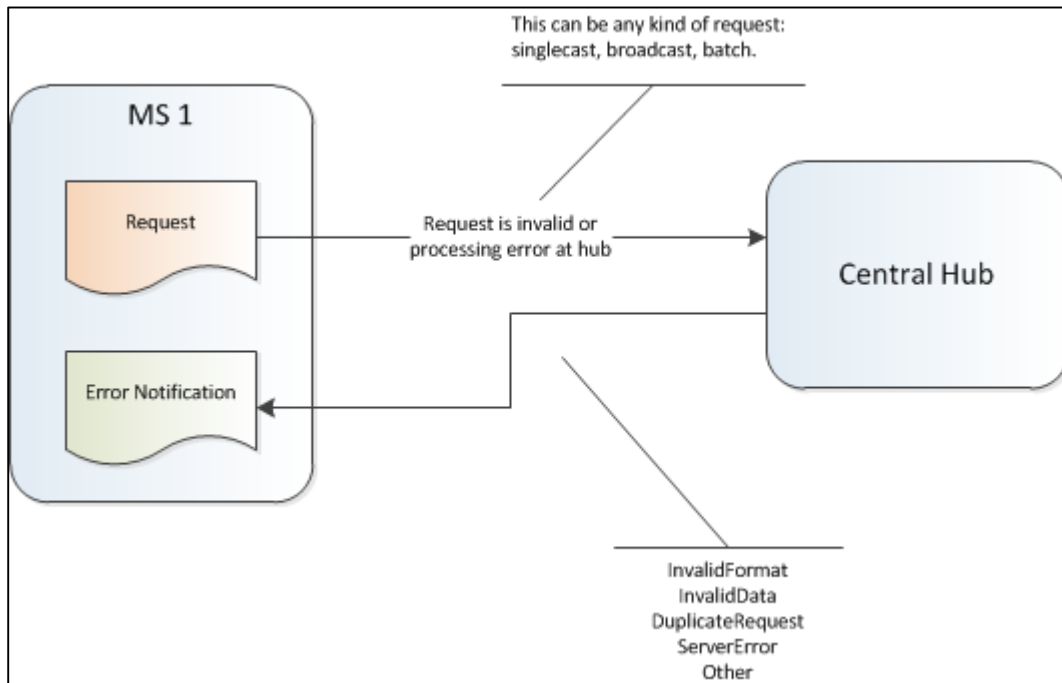


Figure 48 - Error in Request

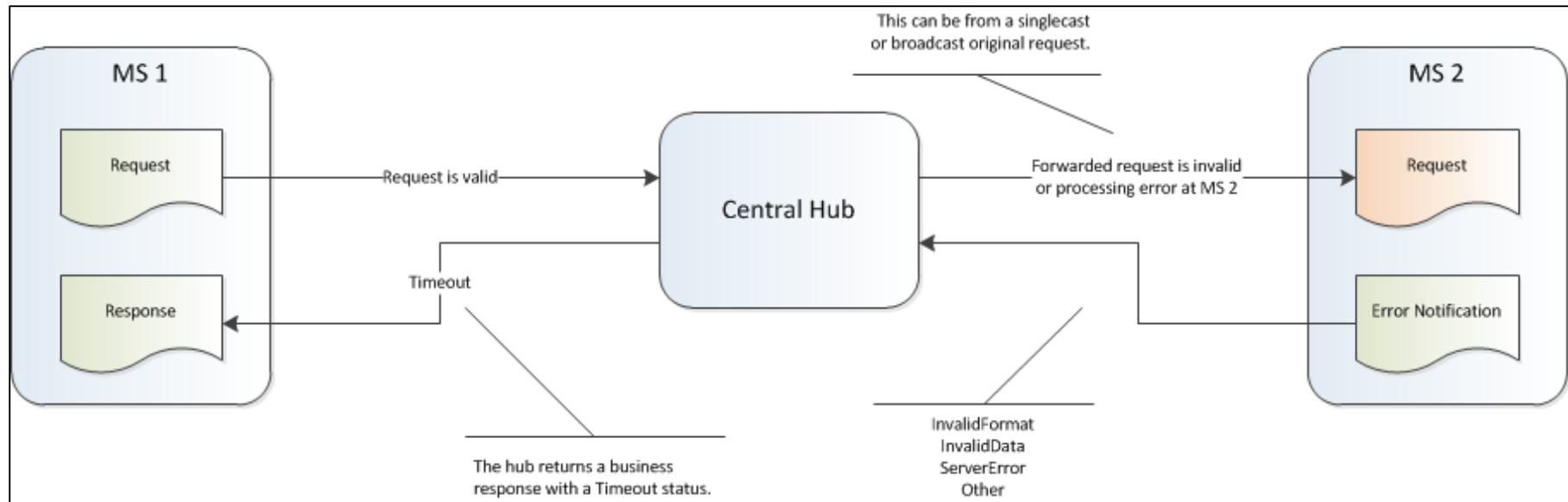


Figure 49 - Error in Forwarded Request

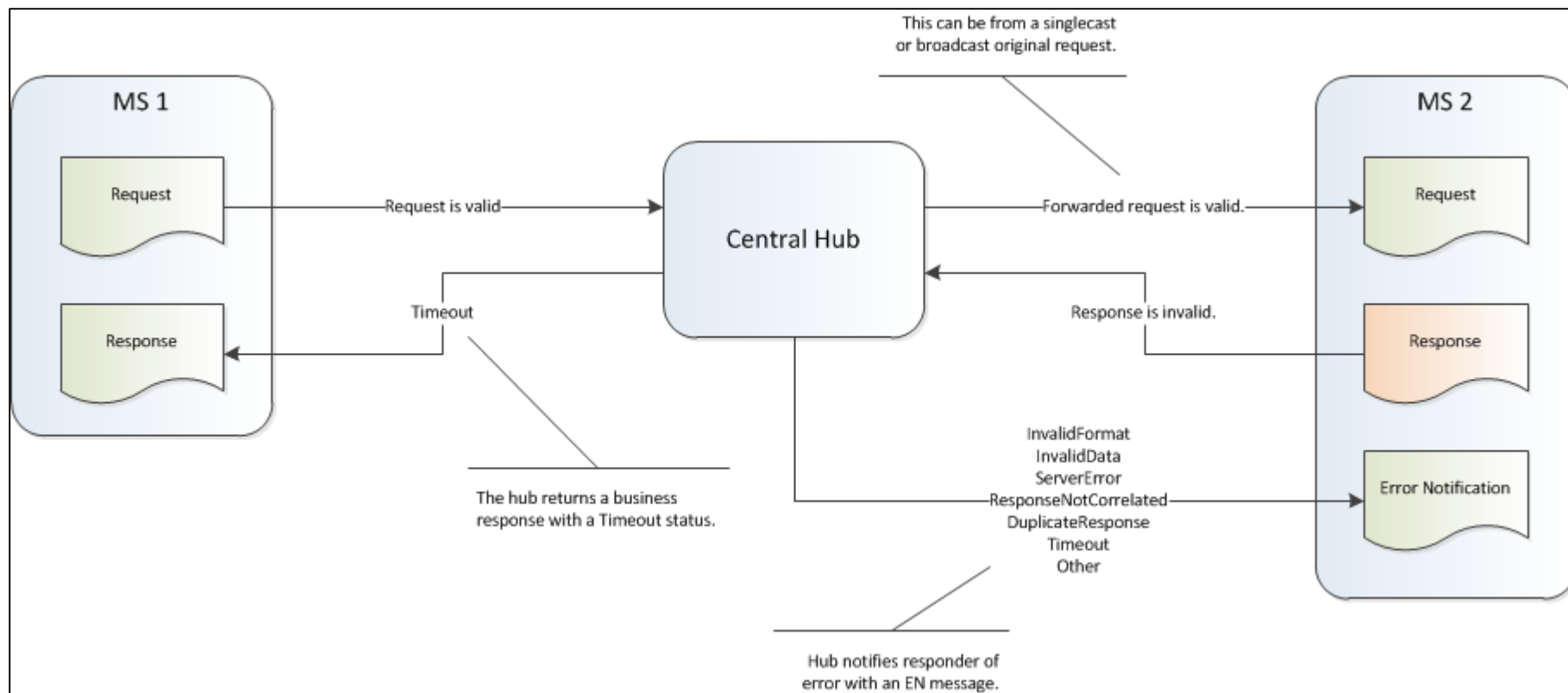


Figure 50 - Error in Response

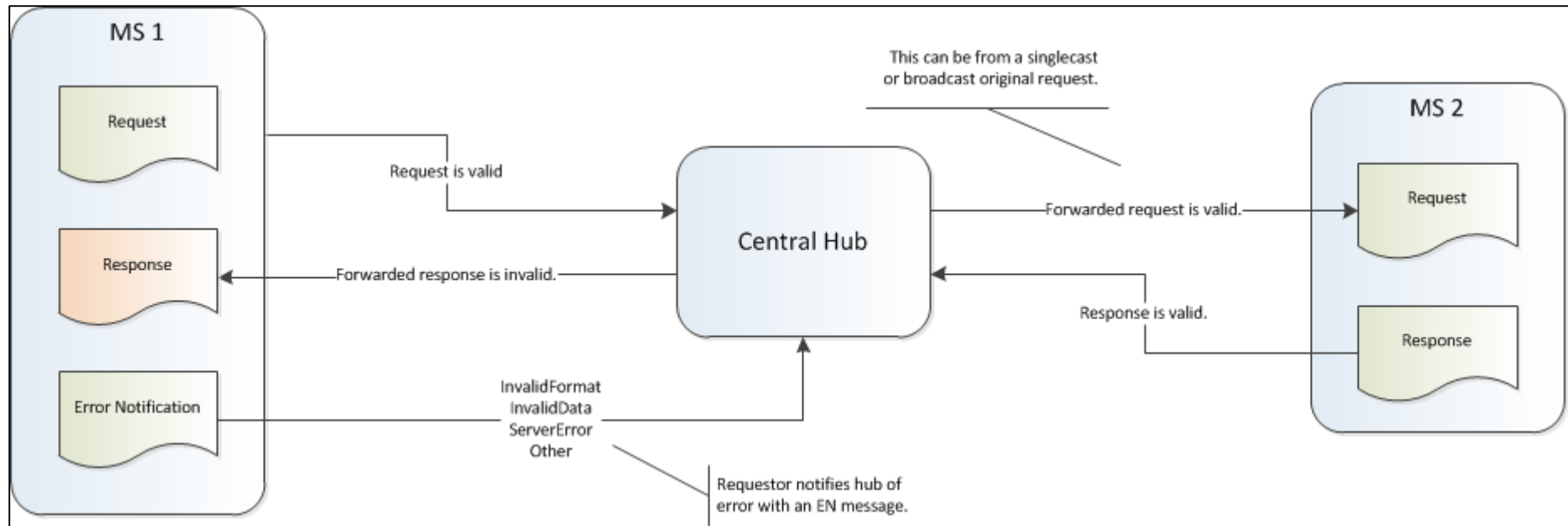


Figure 51 - Error in Forwarded Response

7.7. EN Message Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<ErrorNotification xmlns="https://webgate.ec.testa.eu/move-Hub/tachonet/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header version="3.2"
    technicalId="5a4b12de-272a-41ed-a27c-37622d9a2528"
    workflowId="782dec67-f3ac-48f9-9fc6-9d0cc0123f9b"
    sentAt="2016-01-01T00:00:00Z"
    from="IE"
    to="EU"/>
  <Body businessCaselId="CIC Request Example"
    statusCode="InvalidFormat"
    statusMessage="XML not well-formed."/>
  <OriginalMessage>
    &lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;&lt;CheckCardStatus_Request
    xmlns=&quot;https://webgate.ec.testa.eu/move-Hub/tachonet/3.2&quot;&gt; &lt;Header
    version=&quot;3.0&quot; technicalId=&quot;1d061fe6-5dd1-4325-bed3-3f6f388a538e&quot;
    workflowId=&quot;782dec67-f3ac-48f9-9fc6-9d0cc0123f9b&quot; sentAt=&quot;2016-01-
    01T00:00:00Z&quot; timeoutValue=&quot;2016-01-01T00:00:20Z&quot; from=&quot;IE&quot;
    to=&quot;UK&quot; &gt;&lt;Body businessCaselId=&quot;CIC Request Example&quot;
    requestPurpose=&quot;Control&quot; requestSource=&quot;RSI&quot;&gt;&lt;SearchCriteria
    cardNumber=&quot;C6235B5410196509&quot;/&gt;&lt;/Body&gt;
  </OriginalMessage>
</ErrorNotification>
```

Figure 52 - Error Notification HTTP Message Example

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <s:Fault>
      <faultcode>s:Typed Exception Error Code</faultcode>
      <faultstring xml:lang="fr-BE">ErrorNotification typed exception</faultstring>
      <detail>
        <ErrorNotification xmlns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2">
          <Header version="3.2"
            technicalId="90fe08e1-85bf-499a-a513-49e519c5de86"
            workflowId="b2a5a6fd-216c-4866-9ea3-454cac01030c"
            sentAt="2018-04-05T15:12:00Z"
            from="EU"
            to="IE"/>
          <Body businessCaselId="CCS Request Example"
            statusCode="InvalidFormat"
            statusMessage="The element 'Body' in namespace 'https://webgate.ec.testa.eu/move-
            hub/tachonet/3.2' has incomplete content. List of possible elements expected:
            'https://webgate.ec.testa.eu/move-hub/tachonet/3.2:SearchCriteria'.">
            <OriginalMessage><![CDATA[<ns:CheckCardStatus_Request
            xmlns:ns="https://webgate.ec.testa.eu/move-hub/tachonet/3.2"><ns:Header version="3.2"
            technicalId="ae019b65-b749-444f-a401-cfa65a418b61" workflowId="b2a5a6fd-216c-4866-9ea3-
            454cac01030c" sentAt="2018-04-05T17:11:59Z" from="IE" to="UK"/><ns:Body
            businessCaselId="CCS Request Example" requestPurpose="Control" requestSource="RSI"> <!--
            <ns:SearchCriteria cardNumber="C6235B5410196509"/>-->
          </ns:Body>
          </ns:CheckCardStatus_Request>]]</OriginalMessage>
        </Body>
      </ErrorNotification>
    </detail>
  </s:Fault>
</s:Body>
</s:Envelope>
```

Figure 53 - Error Notification SOAP Message Example