Report

EU Train Driver Licence

Technical Requirements Specification

16522R/TD

Version:1b Draft

D
Y
Y
REPORT.DOT v3.0 01-Feb 2001

Document control

Superseded documents

Nil

Version history

Version	Date	Comments
1a	4/09/2006	First draft
1b	22/09/2006	Comments by MH, CD and FFM

Changes since last version

Outstanding issues and omissions

Issue control

Owner and approver: M.Haynes

Signature: Date:

Distribution:

File reference(s)

Contents

1. Introduction

1.1 Background

The aim of this study is to investigate systems for implementing the proposed EU Directive on the certification of train crews operating locomotives and trains on the Community's rail network.

The aim of this document is to outline the technical design for systems to implement this Directive.

In this document, the system will generally be considered as a whole. Where that is inappropriate, or confusing, the discussion will be split into:

- The licence register system, and
- The certificate register system

Where the licence is the document that shows that a person has passed initial checks to show they are fit to drive (issued by a state's rail safety body, the "competent authority"), and the certificate shows that they have undergone training and are capable of driving certain traction types and/or over certain routes (issued by their employer or a training provider). Further details of the proposals can be found in:

- The proposed Directive itself (COM(2004) 142 final of 3 March 2004)
- The functional specification for this project

1.2 System construction

Although this document details technical parameters to be taken into consideration for the systems, it does not deal with how the physical system is realised. The actual database, package, programming language and hardware (or whatever may be used in construction) are left to the individual authority to decide upon, as this does not form a part of the Directive. As such, much technical detail is left to the implementer: This allows existing system code, where it fits the required model, to be re-used, or adapted for the model. It also allows for (e.g.) an existing personnel system to be extended to deal with these functions. In the survey work for the study, a number of railway undertakings made it clear that they would expect to feed the driver data from their existing personnel system.

Some of the data items are left to the practices of the individual Member State: for instance, in system terms, it would be advantageous to have a single European set of route and traction codes: this, however, is clearly unworkable because it would involve large amounts of change to working practices in many States. Indeed, such codes are not even specified at country level, leaving countries free to continue using their existing arrangements, which may or may not include codified routes and/or traction types.

The only required items are the ability to access certain data fields (how and where they are stored and linked is not mandated), and the ability to make enquiries of the data, as

well as adding and updating it.

1.3 TAF common interface TSI

This document has been structured to allow the inclusion, if required, of the intercommunication between systems into the methodology described in the above specification. Nothing in this document is intended to contravene that specification, nor should it be taken to do so.

1.4 Protection of personal data

Directive 95/46/EC applies to the data held in this system, and nothing in this specification is intended to contravene the content of that directive.

2. Technical design options

2.1 Interactions between systems

This section looks at three possible alternative technical options for interactions between the various components of the system, then selects the most appropriate for various types of company.

The terms ‘loosely coupled’ and ‘tightly coupled’ (use in sections 2.2, 2.3, and 2.4 below) refer to the way that the individual states (or employers) systems intercommunicate with each other. The next two sections will explain these terms in some detail.

2.1.1 Loosely coupled

Loose coupling is where systems pass messages to and from each other, typically using a form of middleware queuing system, such that the target system deals with the request as and when it can. Thus, the target system may not even be available at the time of the request: this is simply queued for later action.

Loose coupling of systems is effective where systems run by different authorities, with different availability rules, wish to intercommunicate. It allows for buffering of messages due to system overloads, and system outages. While it does guarantee a response, it does not guarantee the response time – indeed, due to system downtime, the response could be passed back many hours later (by which time the issuing system may be down). Thus, the separate systems should exist without detailed knowledge of (or even specific methods to handle) system downtime.

2.1.2 Tightly coupled

Tight coupling is where systems directly query each other with the expectation of an immediate reply. This gives a guaranteed response time, but at the cost of additional system complexity in dealing with failure modes (e.g. communications failure mid transaction, target system not available, etc.)

Tightly coupled systems require a great deal of effort to design, code and test effectively. This effort is typically only worthwhile when systems have a great deal of intercommunication, and the response must be available rapidly. This may even involve shared database tables, to which interested parties have access. This can allow a remote user direct access to the data, cutting out any interaction by the target system’s code at all (except perhaps for enforcing access permissions).

2.2 Intercommunication Option 1 – Loosely coupled systems connected manually

2.2.1 Physical access by system owner alone

With this option, the interactions between systems are purely manual in nature. When

queries are required to be made between the various parties, these are done by letter, email or phone. Using this scenario, the link between systems is wholly administered by systems staff.

This option means that the individual systems need to have no similarity between themselves at all (save that required by the Directive itself), because the systems do not physically interact with each other, and the parties dealing with the query can arrange language and code translation between themselves as appropriate. In start-up costs, it is cheap, but running costs (physically providing resources to answer queries) are greater than purely automated systems.

This level of interaction is the simplest (and most cost effective) to specify and implement. It also allows for conditions not catered for in the specifications, because it is assumed that the human system actors will use their own initiative to resolve such issues. However, it would have serious issues should the number of queries ever exceed a relatively small volume.

The database design, implementation, platform and languages of the systems can be totally different. There is also no need for complex enquiry processes to be written. This option suits well where a system already exists to perform one (or other) function, and there is no case for replacing it

2.2.2 Physical access via the internet for trusted users

An alternative to the above is to allow trusted users to access the system directly via the internet. By using a suitable logon and password, the user could access/update a suitable subset of the data. This removes one link of the chain (where physical intercommunication is required between the enquirer and the relevant staff of the target system), thus speeding the process. It still separates the systems in the view of the end user (i.e. they have several systems to use not just one logical view of the data)

2.3 Interconnection Option 2 – Loosely coupled systems connected using MQ series

With this option, systems intercommunicate with each other via a series of MQ Series queues, issuing requests, and, at some later stage, receiving a response. This can be sub-divided into 2 other options:

a) Interconnection via a ‘hub’

b) Interconnection by directly accessing the queues of the required target system

This option is really just a more technically advanced version of the option detailed in section 2.2, and shares all the advantages of it, except that an enquiry process must be made available. This could, therefore, apply to an existing system where a modern railway wishes to provide an electronic external interface to its data. The big advantage of this option to the end user is that the data appears in on logical interface, owned by his undertaking, so they only have to go to one place to access all the required data (even though, physically, this is not the case).

2.3.1 Intercommunication via a 'hub'

Using this solution, the sender of the message passes that message to a central message switching hub, with the receiving system as the virtual address (using some form of universally-recognised coding scheme). The hub then performs validation (is that party permitted to make requests of this type to that party?) and, if valid, forwards this to the relevant authority, as identified by its lookup tables of virtual to physical address.

The advantages of this approach over a direct addressing scheme are:

- Trusted third party performs validation and security checking
- Physical address of partners hidden from each other
- Centralised control and monitoring of traffic levels
- No requirement for all users to be directly connected

2.3.2 Intercommunication by direct access

Using this solution, the sender of a message must obtain the physical address of the recipient, and pass the message directly to them. This requires each party to have a table of the physical address of all possible recipients, which must be kept up to date.

The recipient must then perform validation to ensure that the query is valid from this sender, then reply to the message. This requires a set of security checking logic to be present in all parties' back ends, and it must all work in the same way.

With this option, everyone is aware of the physical address of everyone else, and they all have to be interconnected. There is also no way for a central authority to monitor traffic levels. This is a technically simpler option that option 2.3.1, but having less central control.

2.4 Interconnection Option 3 – Tightly coupled systems with common database design

With this option, all parties will need to share a common database structure, and implement security on their databases, which can then be accessed directly by other parties. This mandates a common coding structure for all tables. This solution is impractical should different existing systems already exist, or should any two participants not agree on coding issues. Whilst it is an excellent technical solution, it is very unlikely to be capable of being made workable in the real-world environment.

2.5 Choice of intercommunication option

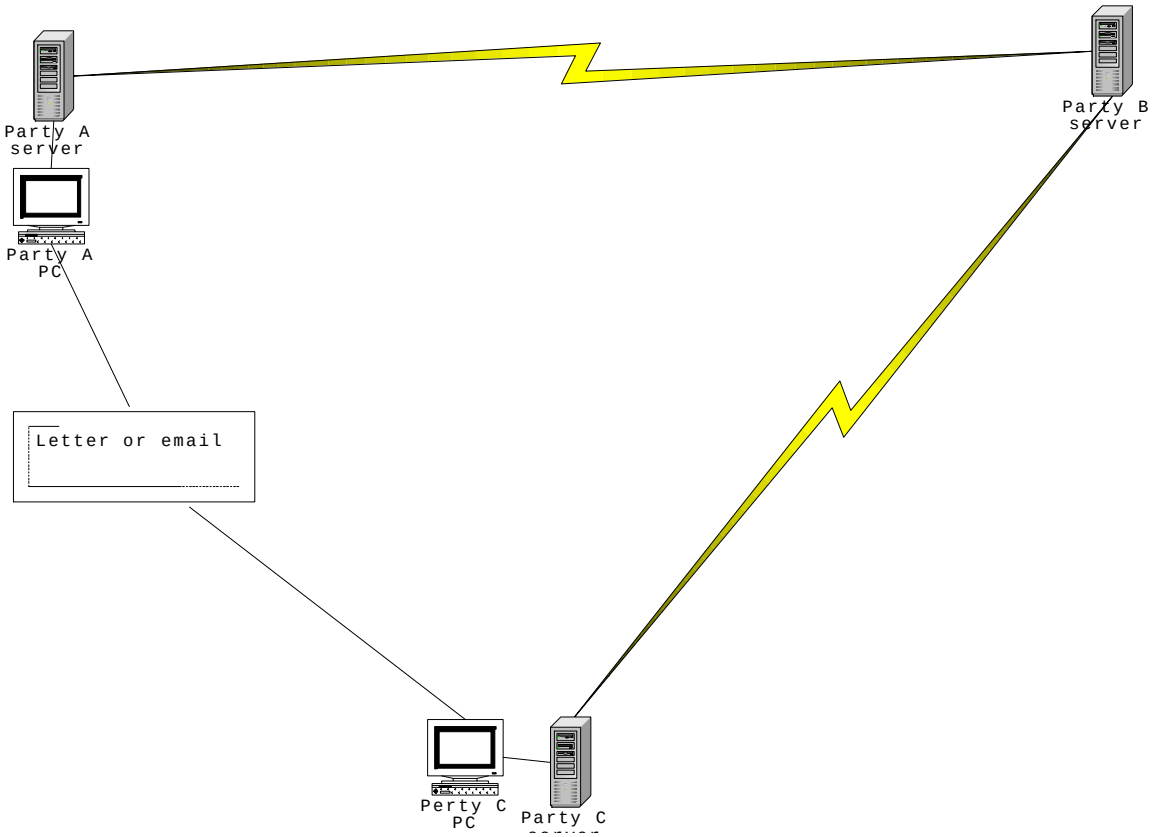
Due to the anticipated low volume of inter-system queries and the varied nature of the systems that may underpin the directive, the first option (detailed in section 2.2) is considered the most viable and cost effective option.

For undertakings where large shifts of drivers between States or employers are anticipated, the second option (detailed in section 2.3.2, using direct communication)

would probably be better.

Intercommunication should be electronic. The most cost effective (and simplest to implement for current system owners) answer to this is email, while wholesale adoption of the MQ series solution would also fulfill this need, albeit at increased initial cost. An advantage of the latter method is security: Email is a very insecure method of data transmission, whereas the latest version of MQ series supports SSL-level encryption of the message data, thus protecting the sensitive data over the public internet.

This situation is summarised in the diagram below, where parties A and B have direct links to each other, as do B and C, but A and C communicate via letter or email.



3. Register design

The details presented here suggest a design for the tables used by the registers. As long as loosely-coupled systems are used, the actual design used could vary substantially from these, so long as the required data items remain accessible via an interconnection method. This approach allows for the use of any solution for the system, from a packaged one (where table formats may be prescribed or even unknown to the user), through to custom-written code.

3.1 Licence system

Table	Table status	Data item	Suggested format	Column status	Notes
Licence	Required				
		Family name	Char(30)	R	
		Maiden name (if married)	Char(30)	O	
		Other names	Char(60)	R	
		Sex	Char	R	
		Title	Char(10)	R	
		Date of birth	Date	R	
		Town of birth	Char(30)	R	
		Country of birth		R	1
		Driver's record reference number		R	16
		Drivers employer	Char(100)	O	20
		Drivers reference number (employers version)		O	15
		Licence status	Char	R	14
		Duplicate		O	8

		number			
		Original status		O	8
		Employer code	Integer(4)	R	6
		Expiry date	date	R	
		Photograph		O	2
		Signature		O	3
		Street address	Char(90)	R	
		Town	Char(30)	R	
		Postal Code	Char(10)	O	22
		Country		R	1
		Validation done by third party	Char(1)	R	21
Licence change audit trail	Required				
		Driver reference number		R	
		Date of change	date	R	
		Field code		R	10
		Original value		R	
		New value		R	
Employers	Optional				
		UIC code	Integer(4)		
		Name	Char(50)		
		Address	Char(200)		
		Country			1

		Drivers contact person name	Char(50)		
		Drivers contact address	Char (100)		
		Drivers contact country			1
		Drivers contact phone	Integer(15)		7
		Drivers contact email	Char (100)		
Electronic message log	Required				
		Message text			
		Message source			17
		Organization code	Integer(4)		24
		Timestamp			18
		Message type	Char(1)	New insert, Amend, Delete	
		Message accepted ?	Char(1)	Y/N	

3.2 Certificate system

Table	Table status	Data item	Suggested format	Column status	Notes
Certificate	Required				
		Family name	Char(30)	O	9
		Maiden name	Char(30)	O	9

		(if married)			
		Other names	Char(60)	O	9
		Title	Char(10)	O	9
		Date of birth	Date	O	9
		Town of birth	Char(30)	O	9
		Country of birth		O	1,9
		Expiry date	Date	R	
		Driver licence record reference number		R	16
		Drivers licence issuing authority	Char(100)	R	19
		Employers driver reference number		O	15
		Status	Char	R	14
		Last medical date	Date	R	
		Driver type	Char	R	13
		Permitted rolling stock type(s)		R	4
		Permitted route(s)		R	5
		Language skills		R	11
		Other information	Char(100)		12
Certificate change audit	Required				

trail					
		Driver reference number		R	
		Date of change	Date	R	
		Field code		R	10
		Original value		R	
		New value		R	
Administrations	Optional				
		UIC code	Integer(4)		
		Name of administration	Char (100)		
		Country code	Integer(2)		
		MQ Server hostname	Char(100)		
		Enquiry qname	Char(100)		23
Electronic message log	Required				
		Message text			
		Message source			17
		Organization code	Integer(4)		24
		Timestamp			18
		Message type	Char(1)	New insert, Amend, Delete	
		Message accepted ?	Char(1)	Y/N	

Inbound sources	Required				25
		Hostname	Char(100)		26
		UIC code	Integer(4)		

Notes:

1) Country code should be either char(30) or char(2). The latter is the EC country code.

2) Photograph is optional, passport sized photo, 100k max (.BMP or .JPG format)

3) Scanned version of signature optional, 100k max (.BMP format, .JPG not acceptable due to possible loss of clarity)

4) Rolling stock coding and/or description is country specific, as is number of occurrences. Other data may also be attached (Date of passing, date qualification expires etc.)

5) Route coding and/or description is country specific, as is number of occurrences. Other data may also be attached (Date of passing, date qualification expires etc.)

6) Employer code is the UIC standard code, and is optional (a driver may not be employed presently) It may also be the employers name is plain text

7) Phone numbers to be stored in full international format, with the country code but without leading zeros/plus etc.: e.g. 4420883101744

8) In the event of a duplicate licence being issued, these fields allow for the recording of this fact. It is assumed that the original licence will be duplicate number 0, subsequent reissues starting from1 etc. Original status can be from:

Status	Meaning
L	Lost
S	Stolen
D	Destroyed
R	L/S/D then recovered

9) These fields are technically optional as they can be obtained from

the licence. It may, however, be simpler or more effective to have local copies of this data, but this must allow for the fact that the data on the certificate and licence can become out of step. In this case, the master system (licence) is taken as the correct source of data, no matter how the data is sourced in the certificate system. This may appear illogical: the employer may well access this information by reference to his own personnel system which is known to be correct, but for licensing purposes the latest version of that record is deemed to be correct. This anomaly means that employers must strive to keep the licensing authority fully informed of changes to personal information about the drivers it employs.

10) Audit trail may take other forms, this is the logically simplest one. Another alternative would be simply to store the new and old records in the audit log. All that the directive requires is that an audit trail exists and may be used to determine the actions which have taken place.

11) Language skills should be coded as defined in section 8 of Annex VII. This will include:

    i. Language

    ii. Competence on a scale of 1 to 5.

12) Other information is a free format field, used to record other information or free-format restrictions or conditions of use of the certificate where such conditions exist.

13) Driver type is shunter or main-line (i.e. hauling passenger or freight trains)

14) Status is one of:

    i. Suspended

    ii. Original (i.e. as originally granted)

    iii. Updated (i.e. one or more fields have been updated since original issue)

    iv. Withdrawn

15) The drivers' reference number with his employer may optionally be on the licence, therefore it is recorded here if required. It is may also be recorded on the certificate record, or referenced via the personnel system (in the case where an employee number, not just driver number, is used)

16) The drivers record reference number is the number printed on the licence, which allows simple and accurate cross reference back to the data in the system. It is included in both registers to enable quick and accurate location of the relevant data. It could have a number of different names or meanings: Licence number being the

most obvious, or database reference number. No meaning beyond this is implied or required by the system.

17) Source of the message: hostname

18) Local time the message was received

19) Name of the authority who issued the drivers licence. With their reference number, provides a direct key into the data of that authority, which simplifies the process of updating it

20) Name of employer, which, with their employee reference number, provides a direct key into the data of that company

21) Where licence data has been amended by electronic message from a trusted third party, this flag records the fact

22) The postal coding systems used by countries vary, and some Member States (Ireland, for example) do not have them. This field records that data if available.

23) The queue name to which enquiries for that authority should be posted. Due to the possibility of change between production and test environments, queue names should never be hard coded in programs, but always read from database reference tables.

24) The UIC code of the originator, as extracted from the message. As part of the validation, this should have been checked against the hostname (Note 17) and shown to be a good host from that administration.

25) Table required for safety authorities only, to validate the source of inbound requests. All valid hostname/UIC code pairs should be present in this table: Should a message arrive from any other, it should be logged as a possible intrusion attempt, and no other processing done upon it.

26) IP address may also be used where hostname lookup or resolution is an issue.

4. Security

4.1 General

Security of this data is a very important issue. The cards issued confer the right to drive railway trains within Europe. In some Member States a car driving licence also doubles (informally) as an identity card, a status likely to filter down to these documents. Therefore, the security of the data and the cards issued is something which must be taken seriously.

4.2 Cards

The issued cards must contain security features to:

1. Validate their authenticity, and
2. Check that data has not been fraudulently amended

These features could include:

1. Check digits on numbers
2. Holograms on the card itself to detect tampering
3. Smart card technology to maintain an on-card electronic version (with suitable cryptography) which can be validated to the displayed information in a similar fashion to ‘chip and pin’ credit cards
4. Pre-laminated card technology so that it cannot be de-constructed and rebuilt later

As well as other security measures as determined by the relevant authorities or suitable international bodies.

4.3 General system security

The data held by these systems is (at least) as private and confidential as data held in a personnel system, and the measures used to protect it should reflect that. EC Directive 95/46/EC applies to the data, which confers protection to the subjects of the data.The consultants expect that this security will include, but not be limited to:

1) Passwords on the application
2) Logging of access
3) Passwords for access to the physical data, however stored

4.4 Licence system

The data and its storage must be secure, both in the register proper and if transmitted via any means, especially where access via MQ Series queries over the internet is permitted.

4.5 Certificate system

The certificate system's security is paramount – this document confers the right to the holder to physically drive a train. Therefore, data security should be a prime concern to implementers of this system. The issue should be viewed in the same light as that for the company pay and personnel system – the data is at least as private. The physical database tables should be password protected for all access, whatever the users right to other parts of the system. This helps secure the system from access by a user who has knowledge of the database system itself.

4.6 Intercommunication

Intercommunication and the protection of data over the Internet is a complex subject. This section does not intend to cover all the issues, or provide all the answers. Individual implementers of the system will need to consider the security of their system, the threats they perceive as possibly happening, and how they can counteract these. IBM and others have much useful security advice about protecting an MQ series implementation to be used over the Internet.

This section will look at two main issues:

1. Validation of identity of a connecting party
2. Security of data (from snooping or tampering) over the internet

4.6.1 Identity of connecting party

The connecting party (the physical machine, not the person or code operating it) should be validated for being an allowed connectee. This is because unauthorised connectors could:

- Attempt to gain data they have no right to
- Attempt to de-stabilise the system (e.g. denial of service attack)

Therefore, the connector must be validated somehow. The channel security exit is the place to do this, and it is recommended that this exit is used by all parties who expose their MQ series server to the Internet. Sample exits are provided by IBM, which can then be taken an modified suitably to suit the individual requirements. It is suggested that users use this exit to:

- Only allow authorised users to connect (validate by IP address or similar) – to prevent unauthorised users from connecting at all
- Only allow those users to put/get from selected queues – to stop

unauthorised access to your system

As a further level of security, the hostname (or IP address) should also be validated against the originating organization's UIC code, i.e. show that the originator is not attempting to bypass security by impersonating another organization. This is done by having a database table with the permissible hostname (or IP address) and UIC code combinations. A check is made, and, for any records that do not have a match on this table, they are logged for investigation as a possible intrusion attempt, and no further processing should be carried out on them.

As a final check, the destination field should be checked to verify that it refers correctly to this organization. This is a final syntax check of the message to ensure that the receiving party is the intended recipient of the message.

4.6.2 Security of data

Even without connecting to an MQ series server, an unauthorised person could gain access to the data by packet sniffing. This is a process whereby a device is inserted into the communications network which simply sniffs (i.e. reads) packets from the network. This type of user is generally undetectable, because they cause no side effects. The most effective way to deal with this type of attack is to render the packets worthless to such an intruder, by implementing packet-level encryption.

In previous releases of MQ series, such encryption had to be done by the channel exits, however, it is now possible to use SSL (Secure Sockets Layer) to do this. This is the suggested method, using as long a key as is possible.

4.6.3 Interconnection methods

Interconnection between two parties can be done in two ways:

1. A link between two MQ series servers, to allow queues to exist remotely, and

2. Client access to the remote queue manager

The former is the more secure option because the number of connectees is then relatively small, allowing validation by host name (or IP address) within the channel exit to be done simply and quickly. However, due to the potential number of interconnections between authorities, this method should only be used for the likely pairs of administrations, i.e. between the operators and safety authority of the same, or neighbouring countries.

For pairings which are less common (or not seen as even possible at system design time), it is suggested that the second option is considered. While security is more of an issue with this solution, it remains very flexible and cost effective for very low volume enquiries.

Queue names should meet the standards laid down in the common interface TSI. The suggested names are:

Queue name	System	Function
XXXXXI_LICENCE_QUERY_INPUT	Licence issuing	To receive queries

		about licences for drivers
XXXXXI_LICENCE_QUERY_REPLY	Railway undertaking	To receive replies to queries about licences for drivers

Where XXXXX is the company and system type prefix as defined in the TSI. This is made up as follows:

Character positions	Content
1234	Company code (numeric)
5	Queue type: 0= production, 1-5= test

5. Technical guidelines for the processes

The following (high level) processes are evident:

1. Create new licence
2. Update licence details
3. Revoke/Withdraw licence
4. Create new certificate
5. Update certificate details
6. Revoke/Withdraw certificate

These processes are basic Information Technology processes: create/update/delete records in a database. As such, the technical details will depend on the platform used for the database and also selected for the client system to run on.

If the option of system-based intercommunication between parties is used, then the following processes should be added:

1. Request for licence details
2. Review licence details request and send response
3. Request for certificate details
4. Review certificate details request and send response

Otherwise the following processes will be required:

1. Get licence details
2. Get certificate details

6. System interface methods

6.1 Principles

The system shall use XML-encoded messages for all data transfer, because XML is the common, open, standardised syntax in widespread use for this role. XML tags shall be encoded in the Latin character set (ISO 8859). Message formats are defined in section 8. Transfer shall be accomplished by the use of MQ series, which is the preferred method of transfer due to its inherently more secure and reliable transfer. Encryption as required in the functional and other specifications is provided by the use of SSL in the MQ series transport, as detailed in section 4.6. All these functions can be carried out by the common interface layer of the TSI.

Most messages can be directly applied, but the some (e.g. licence detail enquiries) require manual intervention to check the reason for the enquiry. This will require several separate, but logically interconnected processes in order to accomplish the final task. It also requires a method of physically alerting the person responsible for validating requests that messages are available for them to validate, as well as a logical queuing method to allow for several messages to arrive before being processed.

6.2 Process

The process for intercommunication is, therefore:

1. Format and send query message
2. Receive query, validate sender host. If not valid, log and ignore
3. Validate sender name in the query body
4. If message does not require manual authorisation, go to step 8
5. Display reason to user, accept (go to step 6) or reject (go to step 10). *Note: this step will require a break in the process, probably by queuing the message for review and emailing a message to the reviewer to get them to go and review the request.*
6. Obtain data and format reply
7. Go to step 11
8. Apply data
9. Go to step 11
10. Format rejection message
11. Send reply message

12. Receive response

7. Safety Implications

This section considers the safety aspects of the system itself, not the data that it contains. Thus, validation of that data (especially the certificate), is the responsibility of the employer/trainer (as it is now): This system is simply a method of recording this data, and does not change existing practice with regard to safety.

The following safety-related issues arise from this system:

1. Licence shows a driver as being medically and educationally fit to drive
2. Certificate lists traction and routes that a driver can drive

8. Data transfer details

Data transfer is to be undertaken in one of two ways:

1. By email or letter, and
2. By communication using XML-encoded messages over MQ series

This section will consider the latter option, and give formats for the messages to be used:

8.1 XML definitions

At this stage in the design, outlines only of the message are proposed. These are presented as XML schema definitions (XSD's), for use in the common interface TSI framework.

8.2 Group definitions

The driver group will always be keyed by the record reference number of the competent authority. This is not to confer any special status on that authority, rather it is a unified way of representing the driver which can be understood by both parties.

```
<xs:element name=”driver” type=”driver reference”>

      <xs:complextype>

      <xs:sequence>

            <xs:element name=”driver record reference number”
type=”xs:string”/>

            <xs:element name=”family name” type=”xs:string”/>

            <xs:element name=”dob” type=”xs:date”/>

     </xs:sequence>

     </xs:complextype>

</xs:element>
```

Message origin and destination are two identical groups, of type organization:

```
<xs:element name=”message recipient” type=”organization”>

      <xs:complextype>

      <xs:sequence>
```

```
            <xs:element name="type">
            <xs:simpletype>
                  <xs:restriction base="xs:string">
                  <xs:enumeration value="E"/><!—Employer -->
                  <xs:enumeration value="C"/><!—CA -->
                  <xs:enumeration value="O"/><!—Other -->
                  </xs:restriction>
            </xs:simpletype>
            </xs:element>
            <xs:element name="code">
            <xs:simpletype>
                  <xs:restriction base="xs:integer">
                  <xs:length value="4"/>
                  </xs:restriction>
            </xs:simpletype>
            </xs:element>
      </xs:sequence>
      </xs:complextype>
</xs:element>
```

The common content of all messages will, therefore, be:

```
<xs:element name="message header" type="header">
      <xs:complextype>
      <xs:sequence>
                  <xs:element name="driver" type="driver
            reference"/>
                  <xs:element name="origin" type="organization"/>
                  <xs:element name="destination"
```

```
            type="organization"/>

                <xs:element name="message type" type="xs:string"/>

                <!—- Valid message type codes defined later in
            this document -->

        </xs:sequence>

        </xs:complextype>

</xs:element>
```

The following table summarises the messages and their permitted origin and destinations:

Message	Origin	Destination
Start of employment	Employer	Competent Authority
End of employment	Employer	Competent Authority
Renewal (no changes)	Employer	Competent Authority
Change of name	Employer	Competent Authority
Change of address	Employer	Competent Authority
Licence enquiry	Employer or Competent Authority	Competent Authority

The following sections define the messages themselves.

8.3 Start of employment

Message type = SE

No other data is mandated, as the employer is identified in the message origin field. Optionally, the date of start of employment can be added, but is assumed to be the current date:

```
<xs:element name="hdr" type="header">

</xs:element>

<xs:element name="start date" type="xs:date" maxoccurs="1"
minoccurs="0">

</xs:element>
```

The response is a message of type SE:

```
<xs:element name="hdr" type="header">
</xs:element>
<xs:element name="reason" type="xs:string">
</xs:element>
```

Where reason is either OK or the reason for the rejection in plain text.

8.4 End of employment

Message type = EE

```
<xs:element name="hdr" type="header">
</xs:element>
```

The response is a message of type EE:

```
<xs:element name="hdr" type="header">
</xs:element>
<xs:element name="reason" type="xs:string">
</xs:element>
```

Where reason is either OK or the reason for the rejection in plain text.

8.5 Renewal

Message type = RN

```
<xs:element name="hdr" type="header">
</xs:element>
```

The response is a message of type RN:

```
<xs:element name="hdr" type="header">
</xs:element>
<xs:element name="reason" type="xs:string">
</xs:element>
```

Where reason is either OK or the reason for the rejection in plain text.

8.6 Change of name

Message type = CN

The message comprises the parts of the name which have changed, from:

```
<xs:element name="hdr" type="header">

</xs:element>

<xs:element name="family name" type="xs:string">

</xs:element>

<xs:element name="maiden name" type="xs:string" maxoccurs="1"
minoccurs="0">

</xs:element>

<xs:element name="other names" type="xs:string">

</xs:element>

<xs:element name="title" type="xs:string">

</xs:element>
```

The response is a message of type CN:

```
<xs:element name="hdr" type="header">

</xs:element>

<xs:element name="reason" type="xs:string">

</xs:element>
```

Where reason is either OK or the reason for the rejection in plain text.

8.7 Change of address

Message type = CA

The message comprises the parts of the address which have changed:

```
<xs:element name="hdr" type="header">

</xs:element>

<xs:element name="street address" type="xs:string" maxoccurs="1"
minoccurs="0">
```

```
<xs:element name="town" type="xs:string" maxoccurs="1"
minoccurs="0">

<xs:element name="postal code" type="xs:string" maxoccurs="1"
minoccurs="0">

<xs:element name="country" type="xs:string" maxoccurs="1"
minoccurs="0">
```

The response is a message of type CA:

```
<xs:element name="hdr" type="header">

</xs:element>

<xs:element name="reason" type="xs:string">

</xs:element>
```

Where reason is either OK or the reason for the rejection in plain text.

8.8 Request licence details

Message type = RD.

```
<xs:element name="hdr" type="header">

</xs:element>

<xs:element name="request reason" type="xs:string">

<!-- Plain text, the reason for the request, to be reviewed by the
recipient -->

<xs:element name="enquirer id">

     <xs:complextype>

     <xs:sequence>

          <xs:element name="name" type="xs:string"/>

          <xs:element name="phone" type="xs:string"/>

          <xs:element name="email" type="xs:string"/>

     </xs:sequence>

     </xs:complextype>

</xs:element>
```

The response is a message of type RD, with either:

```
<xs:element name=”hdr” type=”header”>

</xs:element>

<xs:element name=”reject reason” type=”xs:string”>

</xs:element>
```

Or:

```
<xs:element name=”hdr” type=”header”>

</xs:element>

<xs:element name=”family name” type=”xs:string”/>

<xs:element name=”maiden name” type=”xs:string” maxoccurs=”1”
minoccurs=”0”/>

<xs:element name=”other names” type=”xs:string”/>

<xs:element name=”sex” type=”xs:string”/>

<xs:element name=”title” type=”xs:string”/>

<xs:element name=”date of birth” type=”xs:date”/>

<xs:element name=”town of birth” type=”xs:date”/>

<xs:element name=”country of birth” type=”xs:string”/>

<xs:element name=”drivers employer” type=”xs:string”/>

<xs:element name=”licence status” type=”xs:string”/>

<xs:element name=”duplicate number” type=”xs:integer”/>

<xs:element name=”original status” maxoccurs=”1” minoccurs=”0”>

    <xs:simpletype>

          <xs:restriction base=”xs:string”>

          <xs:enumeration value=”L”/>

          <xs:enumeration value=”S”/>

          <xs:enumeration value=”D”/>

          <xs:emuneration value=”R”/>

          </xs:restriction>

     </xs:simpletype>
```

```
</xs:element>
<xs:element name=”employer code”>
      <xs:simpletype>
      <xs:restriction base=”xs:integer”>
      <xs:length value=”4”/>
      </xs:restriction>
      </xs:simpletype>
</xs:element>
<xs:element name=”expiry date” type=”xs:date”/>
<xs:element name=”street address” type=”xs:string”/>
<xs:element name=”town” type=”xs:string”/>
<xs:element name=”postal code” type=”xs:string”/>
<xs:element name=”country” type=”xs:string/>
<xs:element name=”external validation” type=”xs:string”>
     <xs:simpletype>
           <xs:restriction base=”xs:string”>
           <xs:enumeration value=”Y”/>
           <xs:enumeration value=”N”/>
           </xs:restriction>
      </xs:simpletype>
</xs:element>
```

Notes:

- In the case of optional fields, if that field is not present on the database, then the relevant group will be omitted. This also applies if the data in that field is null.
- It is assumed that all fields will be returned in displayable format, i.e. character not binary data
- Dates will be in the format: YYYYMMDD
- Date and time will be in the format: YYYYMMDDHHMM (optionally SS as well)

9. Suggested production system management

The directive does not specify target availability for these systems, so they could be run as working-hours, weekday only systems. There is, of course, no reason why they cannot be run as true 24x7 systems. However, due to different timezones in parts of Europe, any system which requires to interconnect with another should be aware that the target system may not be available. Therefore, 24x7 operation would be preferable for systems where interconnection is done by electronic means.

Production systems would require resilient hardware, frequent backup and stringent recovery procedures. This is because the data is critical to the running of the railways.

The premises which are used to house the equipment should be secure, to remove the possibility of unauthorized physical access to the machine. The room in which the equipment is located should itself be secure, with access only by authorized (and preferably access to it logged) personnel.

Authorities may wish to outsource the management of this hardware to a suitably qualified company.

10. Validation and quality control

Data validation is important for a system like this, where the system is maintaining a computerised record of paper documents. This should be done by comparison of the data against the original source documents, which have to be supplied as part of the process.

In the case where data is supplied electronically, and validated by a third party (e.g. where an employer supplies updates to an employees licence) then the validation can only be done by that third party. Such updates should be flagged on the database so that it is clear that the data has been validated by another authority.

11. Sizing and performance

The following sizing figures are based on:

- A national body with 20,000 licences, using 50k photographs and signature files, and
- A train operator with 2,000 employees (assume 10 traction types of 30 characters each, 20 route codes of 30 characters each)
- 2,000 new/amended licences per year
- 200 new/amended certificates per year

and simply calculate the space required for the data elements, not allowing for:

- Indexing
- Free space
- Other, database-dependant features

11.1 Licence register

Table name	Rows	Approx size
Licence	20000	2000Mb
Licence audit trail	2000 p.a. (assume retain for 40 years)	8Mb
Employers	500	0.2Mb
Electronic message log	2000	8Mb

11.2 Certificate register

Table name	Rows	Approx size
Certificate	2000	2.5Mb
Certificate audit trail	200 p.a.	0.8Mb
Administrations	100	0.1Mb
Electronic message log	2000	8Mb

11.3 Performance

As the data movement is low (once the initial load of data is done), performance is not seen to be an issue.

Appendix A - Format of certificate

Driver Licence no. **EC1234567890**

Issued: dd mm yyyy **Expires: dd mm yyyy**

Issued by: International North & South Railway SE (Code: 9999)

Issued to: 12345678(30 characters)4567890
12345678(30 characters)4567890

Address: 12345678(30 characters)4567890
12345678(30 characters)4567890
12345678(30 characters)4567890
12345678(30 characters)4567890

Date of birth dd mm yyyyy

Signature

Employee reference no. 1234567890

Driving Category: B

Authorised traction types:

12345678(55 characters)4567890123456789012345678901234567890123456789012345
12345678(55 characters)4567890123456789012345678901234567890123456789012345
12345678(55 characters)4567890123456789012345678901234567890123456789012345

Authorised infrastructure:

12345678(55 characters)4567890123456789012345678901234567890123456789012345
12345678(55 characters)4567890123456789012345678901234567890123456789012345
12345678(55 characters)4567890123456789012345678901234567890123456789012345
12345678(55 characters)4567890123456789012345678901234567890123456789012345

Additional information

12345678(40 characters)45678901234567890

Languages	Skill level

Issued by: <name of authorised person> Signature:

Note: The 'Issued by' fields are optional, the certificate does not lose its validity if they are not completed.

Official Stamp