STRATEGIC EUROPEAN DEPLOYMENT PLAN FOR THE EUROPEAN-WIDE IMPLEMENTATION OF THE TECHNICAL SPECIFICATION FOR INTEROPERABILITY TELEMATIC APPLICATIONS FOR FREIGHT (TAF TSI)

PROJECT No: 2005-EU-93008-S

Deliverable 2 - Definition of the functional and performance requirements and of the associated data necessary to deliver the TAF system

Appendix E – Common Interface

Table of Contents

1 Implementation Approach (ref 4.2.14)

In relation to the Common Interface, the Telematics Application for Freight Services Sub System (TAF TSI) documents the essential requirements for Telematics Applications (referring to 2.7.1 and 2.7.2 of Annex III to Directive 2001/16/EC):

> *The essential requirements for Telematic Applications guarantee a minimum quality of service for passengers and carriers of goods, particularly in terms of technical compatibility. Steps must be taken to ensure:*
>
> - *that the databases, software and data communication protocols are developed in a manner allowing maximum data interchange between different applications and operators, excluding confidential commercial data;*
> - *easy access to the information for users.*
>
> *The methods of use, management, updating and maintenance of these databases, software and data communication protocols must guarantee the efficiency of these systems and the quality of the service.*

Consequently, chapter 4.2.14.7 of the TAF TSI document that the Common Interface is mandatory for each actor in order to join the TAF TSI rail interoperability community and must have the following capabilities :

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

Working Group 5 of the Strategic European Deployment Plan project have considered, during Working Group meetings between September 2005 and February 2006, how best to achieve the requirements of the TAF TSI Regulation above and have determined that the best implementation approach should be by Public Tendering (using this document to describe the functionality required) for the delivery of the following :

1. The development of a Reference Implementation of the Common Interface;
2. The development of a Private and Common Metadata system to define, control and manage the TAF TSI messages and data flow through the Common Interface;
3. The distribution and management of the Common Metadata system;
4. Full technical documentation supporting the Reference Implementation of the Common Interface, allowing other organizations to build their own Common Interface at their own cost;
5. Distribution of the Reference Implementation of the Common Interface to all actors wishing to use it to fulfil their interface requirements of the TAF TSI;
6. The provision of an implementation support service for the internal integration of each appropriate actor's system with the Common Interface.
7. The provision of a test mechanism which will enable an installation to be validated against a reference set of messages. The tests shall be capable of being sent to

a remote validation authority. The connection of each implementation of the TAF-TSI Common Interface is dependent on successful validation.

Key Principles:

Documentation
The documentation for the Common Interface shall at least include:

- Technical requirements specifications
- Design documentation
- List of used off-the-shelf products and open source products
- List of used IT standards
- Programming guidelines
- Source code
- Installation documentation
- Firewall requirements
- User manual
- Test plan
- Test specifications
- Test reports
- Interface specifications
- Release notes
- Training documents

Each document shall be equipped with contents, scope and traceability matrix (if applicable). Where automated techniques (e.g. case tools) can be used for producing documentation, this is encouraged.

Design
The design shall focus on clarity, maintainability, modularity and reusability.

1.1 Architecture of the Common Interface

1.1.1 Architecture required by the TSI (ref 4.2.14)

Chapter 4.2.14 of TAF TSI "Networking & Communication" introduces the architecture required for the Common Interface and Message exchange :

"General Architecture (ref 4.2.14.1)

This subsystem will see, over time, the growth and interaction of a large and complex Telematic rail interoperability community with hundreds of participating actors (RUs, IMs, ...), which will compete and/or co-operate in serving the market's needs. The Network & Communication infrastructure supporting such rail interoperability community will be based on a common **Information Exchange Architecture**, known and adopted by all participating actors.

The proposed **Information Exchange Architecture:**

- is designed to reconcile heterogeneous information models by semantically transforming the data that is exchanged between the systems and by reconciling the business process and application-level protocol differences;

- has minimum impact on the existing IT architectures implemented by every actor (each may select its own Databases and applications to link to the common interface ;

- safeguards IT investments made already.

The nature of the Information Exchange Architectures indicates a Peer-to-Peer asynchronous type of interaction between all actors, while it guarantees the overall integrity and consistency of the rail interoperability community by providing a set of centralised metadata and database services.

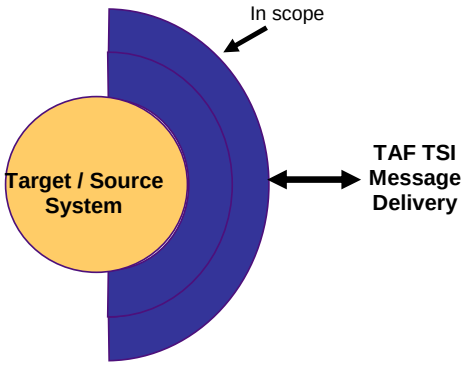
For version control, all trading partners shall continue to support current and previous versions until there is agreement to withdraw the previous version.

A Peer-to-Peer interaction model allows the best cost distribution between the different actors, based on actual usage and it will present, in general, less scalability problems."

1.1.2 Common interface Implementation Architecture (ref 4.2.14.7)

The context of the Common Interface is that it will provide the functionality between the TAF TSI messages and all target systems and human input exchanging data by TAF TSI messages. Taking the requirements of the TSI, the Common Interface implementation architecture will therefore be structured as follows :

Scope of Common Interface



The purpose of structuring the Common Interface components in the implementation architecture above is to ensure that the Common Interface can meet the technical objectives in section 1.4. By achieving these, the Common Interface will deliver early TAF TSI benefits by utilising existing systems, IT platforms and communication processes whilst allowing the scalability required to implement the longer-term TAF TSI vision.

The Common Interface architecture allows the translation of internal data to and from TAF TSI messages and the management and transmission of the messages according to the requirements of the TSI between any actor, independent of the internal systems in use by that actor.

Messages are sent and received through an open Message Queue (MQ), which is the interface between the Translation & Validation layer of the Common Interface and the Security & Transport layer of the Common Interface. The Security and Transport layer manages the delivery and receipt of messages to and from the public network side of the Message Queue. The Translation and Validation layer and API layers manage the receipt of data from and delivery of data to the systems in use on the internal side of the Message Queue.

The Common metadata is used by the Common Interface for all activities that are standardised by the TSI and the Private metadata is used by the Common Interface for local activities related primarily to the API interfaces with internal systems and local operation of the Common Interface itself. The implementation

of the Common Interface must be such that any modification of public and private metadata can be made dynamically and mustn't have any effect on operation. Metadata changes should be applied according to their validity period.

1.2 Meeting the TSI Objective (ref 4.2.14.1 & 4.2.14.6)

The objective of this Functional Requirements Document is to specify the Common Interface in sufficient detail for implementation, taking as its staring point, the high level specifications described in the TAF-TSI concerning the messaging and data model.

This document is part of the SEDP project and an SEDP deliverable.

This document describes how, using the principles of **Semantic Integration**, the freight Railways and Infrastructure Managers of the European Rail Industry will exchange data and implement the Telematics Application for Freight Regulation through an Information Exchange Architecture.

In this context, ***Semantics*** the study of meaning, is used for understanding, implementing and managing the complex TAF TSI message exchanges. Semantics = Data + Behavior. This includes data quality assurance through the implementation of data quality checking in the Common Interface.

Integration is used in this context to describe creating networks of interrelated IT applications to provide benefit to the Rail Industry. The TAF TSI Information Exchange Architecture for messaging require the co-ordination of data, messages and responses from applications across Europe via multiple implementations of its Common Interface.

The TAF TSI Semantic Integration framework is designed to focus on delivering high-returns quickly, by utilising existing applications as data sources. In this project, the adoption of a Semantic Integration framework is intended to ensure that the TAF TSI project costs don't overwhelm project benefits, particularly in the first couple of years.

An important secondary purpose of this Functional Requirement Specification is to utilise the Semantic Integration framework to deliver significant cost and timescale reductions implementation of TAF TSI.

.

To deliver value, the TAF TSI Semantic Integration framework for network applications is therefore required to:

- Be compatible with existing application systems;
- Use selected existing integration technologies;
- Support emerging integration and system technologies;
- Be capable of integration across organisational boundaries.

By its very nature integration is about the interactions between applications - therefore this Common Interface Functional Requirements Specification focuses on the semantics of the message and collaboration between systems.

A direct benefit of focusing on the interaction between systems is the ability to scale because the message is the fundamental unit of integration. The TAF TSI messages are collections of data, organised in a specific way, and grouped to

provide context. To effectively use existing messages in the TAF TSI integration framework, existing sources have been identified to avoid the expense of creating new ones wherever possible. Only those elements of existing messages that describe a reliable, consistent set of semantic properties have been incorporated. Message translation within the Common Interface will be used to reconcile differences between applications and TAF TSI messages.

The Common Interface and its Metadata management system will have the ability to add non-TAF TSI messages at any point. A small number of additional message elements have been proposed (mostly concerning intermodal units) during the Working Group stage of the project.

API Adaptors (ref 4.2.14.1 & 7)

Application Programming Interface Adaptors for linking the Common Interface to the applications/components in use by actors are required as part of the implementation process. These will provide the host application with a well defined interface and manage the technical communications between the application and the Common Interface. The proposed implementation support service will provide a centre of excellence to assist actors to achieve this cost-effectively and quickly.

Translation and Validation (ref 4.2.14.1, 6 & 7)

A standardized approach to the creation and reading of messages and the validation of those messages is a core component to having a workable implementation across many peer-to-peer actors. Common quality criteria can then be applied to the messages and data.

Metadata System (ref 4.2.14.2 & 6)

Metadata is data whose purpose is to describe other data: its definitions, structure and relationships. For the purposes of the TAF TSI project, the Information Exchange Architecture needs different types metadata for three separate purposes:

- Concrete, prescriptive metadata for the interface metadata syntax and representation - consistent and compatible for each application.
- Descriptive metadata that concretely describes the content of the actual exchanged messages so that translations can be defined within the Common Interface.
- Abstract metadata to understanding what the data means to each application and for describing the relationships between data elements.

The Metadata required for operation of the Common Interface will be applied in two parts, Private Metadata (primarily defining the local conditions of the particular implementation of the Common Interface and the APIs which will interface with the internal systems) and Common Metadata (primarily defining all the messages and their versions plus delivery addressing).

The TAF TSI metadata defines the syntax of how the data is represented, how it is structured, the order of the elements, constraints, required quality conditions and any business rules.

Security and Transport (ref 4.2.14.2, 4, 5 & 7)

A standardised approach to the security and transportation of messages will deliver end-to-end loss-less delivery across whatever networks are available, notably the public internet.

XML(ref 4.2.14.1)

XML is used to describe the metadata and messages because XML is the current, open, standardised syntax in widespread use used to parse a document into named elements. XML also provides a standardised format for structure metadata and constraints metadata, called an XML Schema (.xsd).

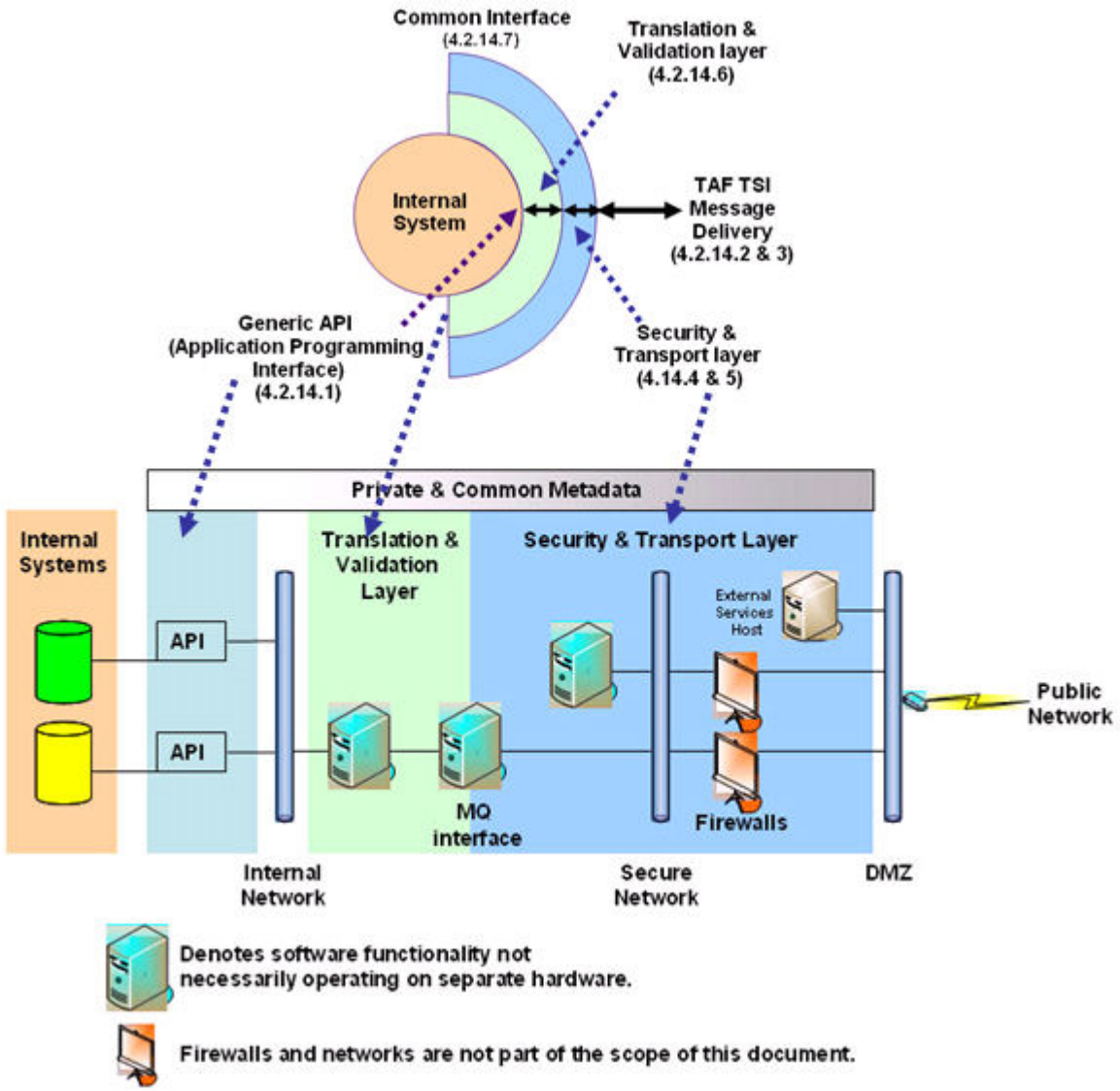
XML Schemas (.xsd) describe data precisely enough information for computers to parse any message into labeled component elements. TAF TSI XML Schemas are appended to this document. The metadata expressed in the TAF TSI XML Schemas describe what each message looks like, how it is structured, which parts are optional, which are required, and value constraints. However, for semantic integration to be successful in delivering value using existing applications, the TAF TSI XML Schemas also describe data found in messages from selected existing systems, CEN agreements and UIC leaflets. Most rail messages exchanged today are not in XML, they are in formats such as Electronic Data Interchange (EDI), flat text files or proprietary formats. The TAF TSI XML Schemas are rich enough to describe most non-XML data resources. The TAF TSI project has focused on reconciling the differences between the messages from existing applications by describing how data elements are related in maps so that it will be possible to deploy real-world integration systems that will translate the messages as they flow between existing non-compatible applications.

1.3 Functional objectives (ref 4.2.14.1)

The Functional objectives of the TAF TSI **Information Exchange Architecture** are **:**

- ..reconcile heterogeneous information models by semantically transforming the data that is exchanged between the applications and by reconciling the business process and application-level protocol differences;
- ..minimise impact on the existing IT architectures implemented already by every actor;
- ..safeguard future IT investments committed already.
- ..avoid bottlenecks, single points of failure and database access security concerns through a secure messaging-based Peer to Peer Common Interface architecture.
- ..be compatible with generally accepted Rail Industry firewall policies.
- ..be open to the inclusion and removal of Actors at short notice.
- ..be open to the creation of new message formats and versions at short notice.
- ..represent tenderable, non-proprietary, open solutions.

1.4 Detailed Structure (ref 4.2.14.1-7)



The diagram above shows the detailed structure of the Common Interface with the constituent parts. The network(s) and Firewall(s) are shown for completeness but are not part of this Functional Requirement Specification. Additionally, the Common Interface must be operable on a PC for low volume installations and be scalable to separate hardware, if required, for higher volume installations.

1.5 Description of the technical and operational environment

1.5.1 IT Platforms (ref 4.2.14.1)

The Reference Implementation of the Common Interface must be capable of reliable operation on open compliant IT platforms (for instance POSIX).

1.5.2 Programming Languages (ref 4.2.14.1)

The Reference Implementation of the Common Interface must provide Application Programming Interface (API) functionality that supports at least the following rail industry-standard programming languages:

- c
- c++ (a platform-independent version only)
- Java

1.6 References

TAF TSI : Interoperability of the trans-European conventional rail system Draft Technical Specification for Interoperability "Telematic Applications for Freight Services" Sub-System. 01/16-ST02 part 2 version EN07 23/11/2004 Directive 2001/16

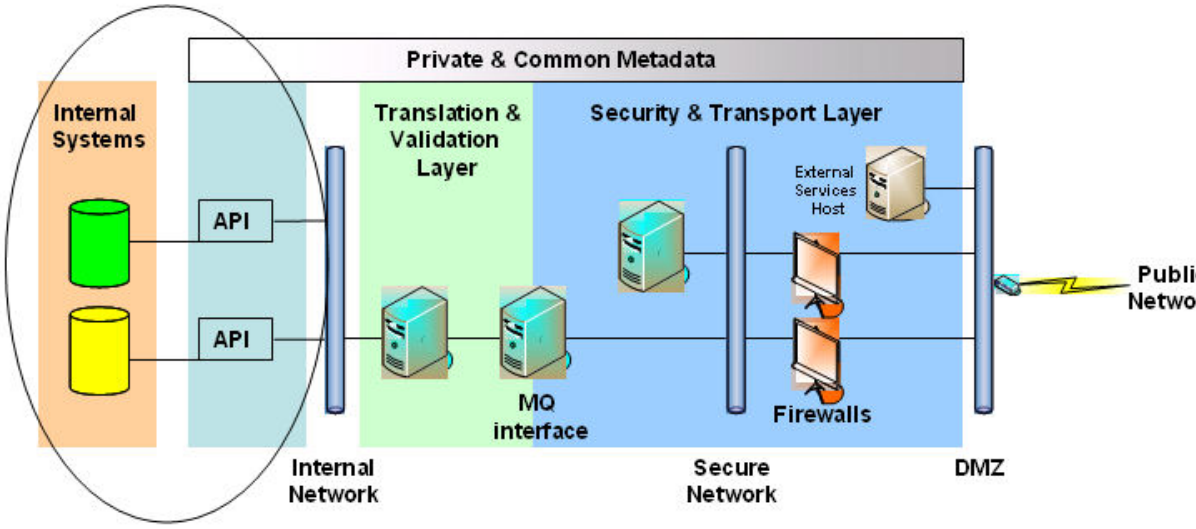
1.7 Definitions and Acronyms

See Glossary.

2 Functional Requirements

2.1 Logical Model – Generic API (ref 4.2.14.1 & 7)

Objective 2.1.1 : To make it possible to connect the ~~CI~~Common Interface to existing and future systems.
Objective 2.1.2 : To have a Platform independent API
Objective 2.1.3 : To provide support for rail industry-standard programming languages
Objective 2.1.4 : To report the success or raise the exception with reasons in case of failure conditions.



The generic APIs of the Common Interface will allow individual actors in the TAF TSI process, mainly RUs and IMs, to connect internal existing or new systems to the Common Interface. No other process for connecting internal systems to the Common Interface is envisaged.

The APIs must support the platforms detailed in 1.5.1. and languages in 1.5.2.and present messages. It shall not address the functionality of the Transport Layer with calls such as Encryption, network protocols, etc.

The functionality that must be supported in the generic APIs is as follows :

The APIs must provide all necessary open-standard functionality for sending and receiving messages to internal RU/IM systems.

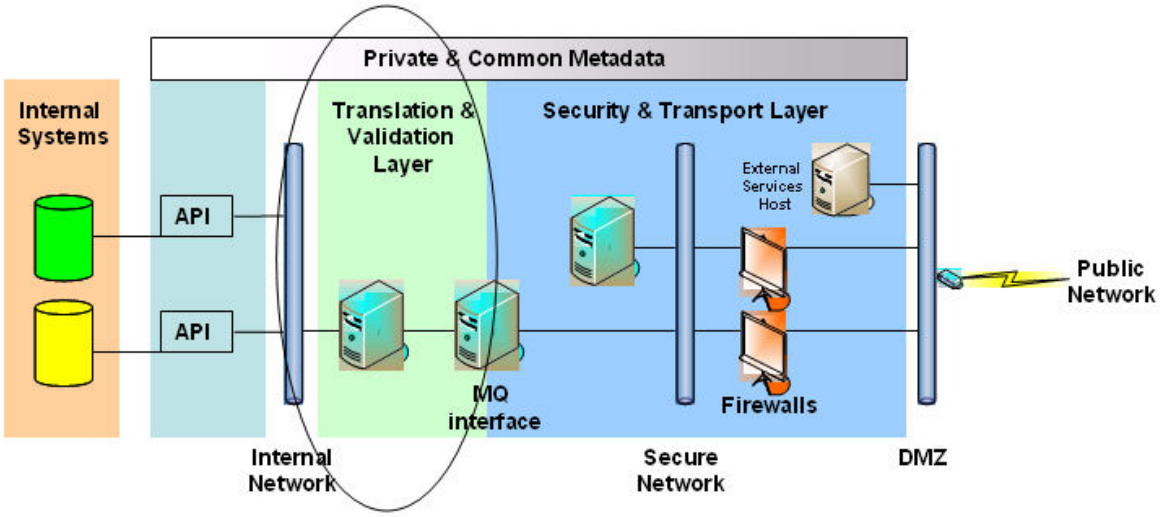
In order for the API to be successfully integrated into the target application, the following must be included:

- Description of how messages are sent and received
- Description of services provided by the interface
- Services for the following functions for C, C++ and Java of the application interface covering
    - Opening and closing the session
    - Sending and receiving messages
    - Request and Response

    - Message Destination
    - Name and Value Elements
    - Error Handling
- Object and Class References
    - Base Classes, Help Classes and Exception Classes
- Installation
    - Defining Services, Policies and Policy Handlers
- LDAP (for authenticating infrequent access)
    - Security
    - Problem Determination
    - Monitoring

2.2 Logical Model – Translation & Validation Layer (ref 4.2.14.1, 6 & 7)

Objective 2.2.1 : To translate from API data to XML-formatted TAF TSI messages and vice versa
Objective 2.2.2 : To apply multiple levels of syntactic and/or semantic validation of API data against rules and reference data
Objective 2.2.3 : To report on data quality, service quality, manageability and volumes.



The Translation and Validation layer of the Common Interface receives data from and sends data to the API layer on the one side and receives from and presents TAF TSI messages to the Security and Transport layer of the Common Interface utilising the Common Interface Metadata for its translation and validation rules.

The Translation and Validation layer must be able to handle each of the TAF TSI data elements, TAF TSI messages and Metadata shown in the Common interface XSD.

The Translation and Validation layer must be able to provide the following functionality :

1. Receive all the TAF TSI messages shown in the Common Interface XSD from the Security and Transport layer

2. Present the relevant, translated and validated data elements from these TAF TSI messages to the APIs in use at the particular implementation of the Common Interface (as recorded in the Private Metadata).

3. Create all the TAF TSI messages from the data elements passed to it by the APIs in use (recorded in the Private Metadata), ensuring that each data element is conformant to TAF TSI metadata definitions, both in format and data quality ('annotation' and 'facets' shown in the Common Interface XSD) according to the version of TAF TSI messages in use at the destination Common Interface (registered in the Common Metadata) and to pass

completed messages to the open Message Queue interface between the Translation & Validation layer and the Security & Transport layer.

4. To reduce implementation effort and encourage early adoption, the translation & validation layer must provide the following modules to translate existing railway messages to their nearest TAF TSI equivalent. The modules are to be selected on implementation and loaded into the Private Metadata :

Existing Message	**TAF TSI TSI message(s)**
HERMES Application 30 A	Wagon Interchange Notice Wagon ETI Wagon Interchange sub-notice
HERMES Application 30 B	Wagon Received at Interchange
HERMES Application 38-A	(enquiry & reply message to WIMO)
HERMES Application 39	Wagon Exception (possible distance data to WIMO)
HERMES messages 41	Wagon Departure Notice
HERMES messages 42	Wagon Arrival Notice
ORFEUS messages CTD, UTD,	Wagon Order
IFCSUM 97B	Wagon Order
IFTMIN 97A	Wagon Order
RCA XML (CTD)	Wagon Order
ISR IFTSTA & WSM 01	Wagon Departure Notice
ISR IFTSTA & WSM 02	Wagon Yard Arrival
ISR IFTSTA & WSM 03	Wagon Yard Departure
ISR IFTSTA & WSM 04	Wagon Interchange Notice / Sub (Border crossing)
ISR IFTSTA & WSM 05	Wagon Arrival Notice
ISR IFTSTA & WSM 06	Wagon Exception
UIC 407-1 2001	Train Running Forecast
UIC 407-1 2002	Train Running Information Train Running Interruption
UIC 407-1 2090 Europtirails	Path Request Path Details
UIC 407-1 2003	Train Delay Performance
UIC 407-1 Generic 2004 /2201 & Europtirails 2004	Train Composition
UIC 407-1 2701	Train Running Interruption

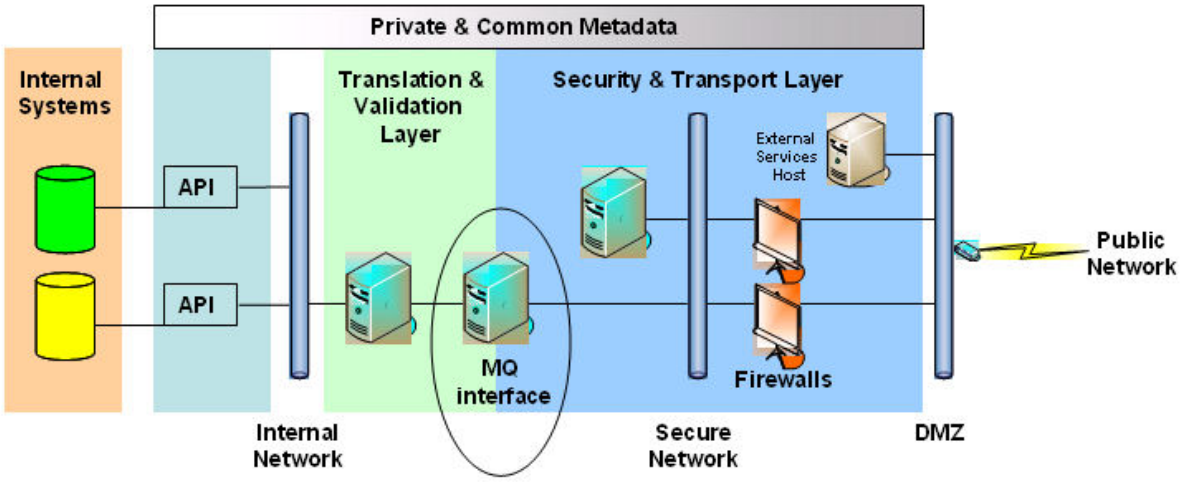
It is expected that translation from existing messages as per the above table will be fully implemented by Jan 2009, noting that not all TAF TSI mandatory data may be supplied from the existing messages shown above. It is further expected that implementation of all remaining mandatory elements of TAF TSI messages which are themselves derived from the existing messages shown above, will be achieved by Jan 2011.

The public metadata will hold the TAF-TSI XML Schema shown in the Common Interface XSD, allowing internal systems to process correctly formatted TAF TSI messages into and out of the Queues without Translation.

5. The translation & validation layer must handle the following validation :
    - Compliance checking of received messages from the network and logging and sending of error reports to both parties when compliance checks have failed. The software should allow configuration at installation.
    - Compliance checking of incoming messages from the internal systems and logging and sending of error reports to the internal party only when compliance checks have failed. The software should allow configuration at installation.
    - Management of message rejection at the Translation and Validation layer should make the cause of the rejection clear.
    - Management of message rejection at the remote Common Interface, including identification of the cause of rejection.
    - Entire rejection of a non-compliant message.

2.3 Logical Model – Interface between Translation & Validation Layer and Security and Transport Layer (ref 4.2.14.1, 2 & 7)

Objective 2.3.1 : Provide a clear message queue interface between the T&V and S&T layers for inbound and outbound messages in order to provide a platform for end to end delivery between an application and the trading partner.

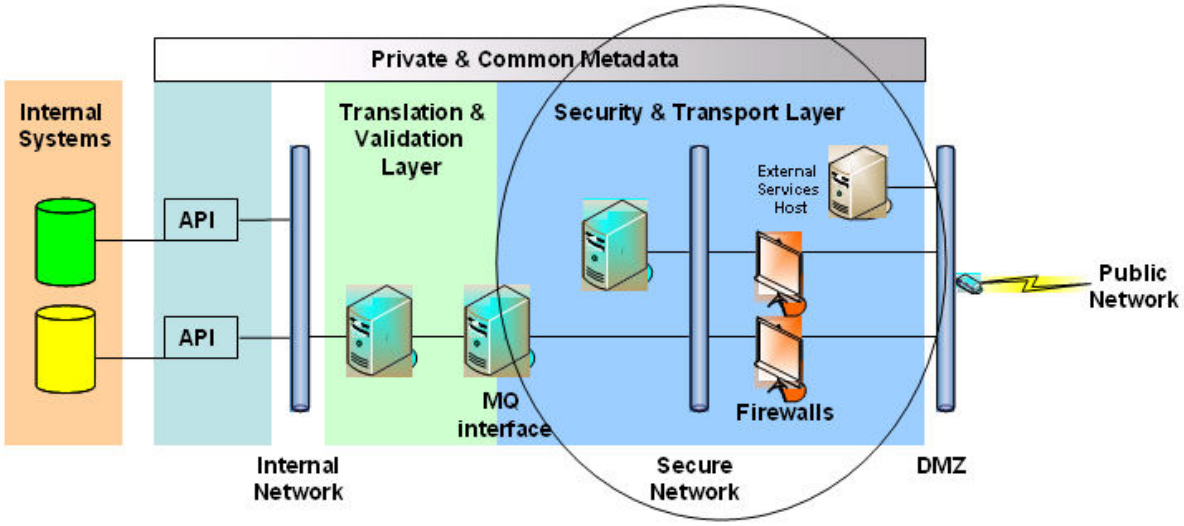


This objective will be met by the implementation of an Open Message Queue as shown. By using an open Message Queuing interface between the Translation & Validation layer and the Security & transport layer, it is possible for the following functional requirements to be met :

- Only correctly formatted messages are placed on the queue.
- The solution is scalable from a PC to a larger system.
- Inbound messages will be directed to a standard-name inbound public queue which includes company ID.
- Outbound messages will be directed to a standard-name outbound queue which includes the company ID of the recipient.
- All public queue names will be recorded in the Common Metadata.
- All private queue names will be recorded in the Private Metadata.
- Messages received into the inbound queue will then be directed (according to local implementation of the security and transport layer) for example to a queue per application as defined in the private metadata.
- Properties of queues will be configurable as part of the installation process.
- Appropriate security capability will be proposed during development and/or tendering phases.

2.4 Logical Model – Security and Transport Layer (ref 4.2.14.2, 4, 5 & 7)

Objective 2.4.1 : To provide appropriate security against specific issues
Objective 2.4.2 : To provide a transaction mechanism that gives the authenticated sender of a TAF TSI-formatted message on an outbound queue the guarantee that the message is received in the inbound queue of the destination Common Interface.
Objective 2.4.3 : To ensure that messages are sent to the correct destination Common Interface.
Objective 2.4.4 : To handle the delivery of messages through IP networks.
Objective 2.4.5 : To implement asynchronous message exchange.
Objective 2.4.6 : Message delivery from one Common Interface to another Common Interface must be achieved via any available IP network.



The Common Interface must ensure security addressing the following specific issues :

- privacy & confidentiality
- authentication
- integrity
- non-repudiation
- denial of service attack or flooding of queues

Inbound functionality: The Security and Transport layer of the Common Interface receives TAF TSI messages from other Common Interface implementations elsewhere in the Rail Industry either by the TAF TSI message being placed on the public queue if it is received from a regular trading partner whose connection details are permanently stored in the local firewall, or via an external services host if the message is received from an irregular trading partner whose network connection details are not permanently stored in the local firewall. The Security and Transport layer also dynamically manages the inbound queue to ensure that each queue entry is processed onto the correct receiving queue within 5 seconds, for further processing by the Translation & Validation layer and the API layer.

Outbound functionality: The Security and Transport layer sends TAF TSI messages to other Common Interface implementations elsewhere in the Rail Industry by transferring the TAF TSI messages presented to it from the Translation and Validation layer of the Common Interface from the outbound message queue within 5 seconds, using the Common Interface Metadata.

External Services Gateway must authenticate casual users using the commonly accepted appropriate security processes. The external services gateway must deny service to non-authenticated users.

The following error handling and reporting is required :

- Queue size
- Request for opening a non-existent Queue
- Message lifetime per queue

2.4.1 Data Compression (ref 4.2.14.1)

Since the files with TAF messages may have large size, it is appropriate to perform data compressing. For XML messages, this procedure ensures significant decrease of the data volumes. Therefore the Security & Transport layer of the Common Interface~~l~~ will perform compression of the outgoing and decompression of the incoming TAF XML messages. An open industry standard mechanism has to be used for compression/decompression.

Encryption or data compression in Message Queuing is available at 'channel exit time'. Selection of messages to apply encryption or data compression to should be agreed as part of implementation.

2.5 Required processing for Wagon & Intermodal Unit Operating Database Instances (WIMO) (ref 4.2.12.2)

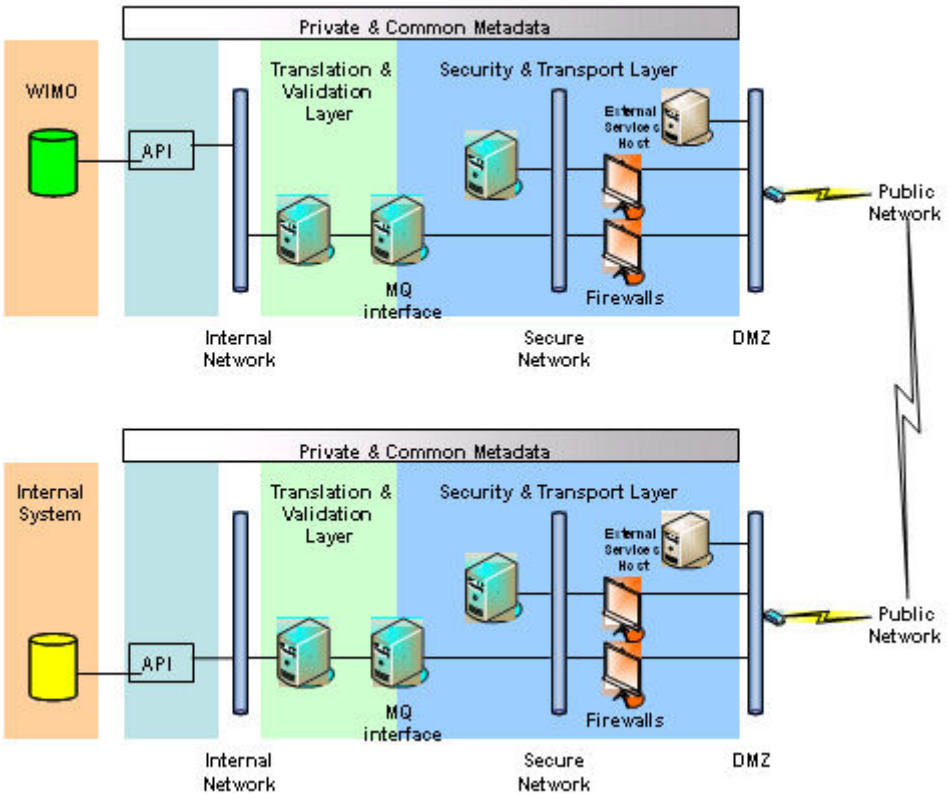
Near-real-time updates for the data required by TAF TSI to be sent to WIMO instances will be transferred to the relevant WIMO instance via the ~~CI~~Common Interface, so that each actor obliged to place data in the WIMO calls a 'WIMO update API' in the API layer of the ~~CI~~Common Interface just after the relevant internal-system transaction has been processed internally.

~~The WIMO API will identify the relevant WIMO for the particular wagon from the public metadata by comparing the wagon number with the relevant wagon number mask in the metadata.~~

~~For example :~~

~~Wagon Number 012345676890~~
~~Matching WIMO mask in metadata : **34*********~~
~~The mask address for the WIMO applicable to wagon numbers with that mask will then be used by the WIMO API to place the message on the outbound queue – in this example addressed for the VR (Finland) WIMO~~
~~In this example, all wagons registered in Finland (code 10) are stored in the VR WIMO.~~

~~Distributed WIMOs would also be able to continue to exist in parallel to the mandatory posting of data to a central WIMO.~~



WIMO enquiries/reporting

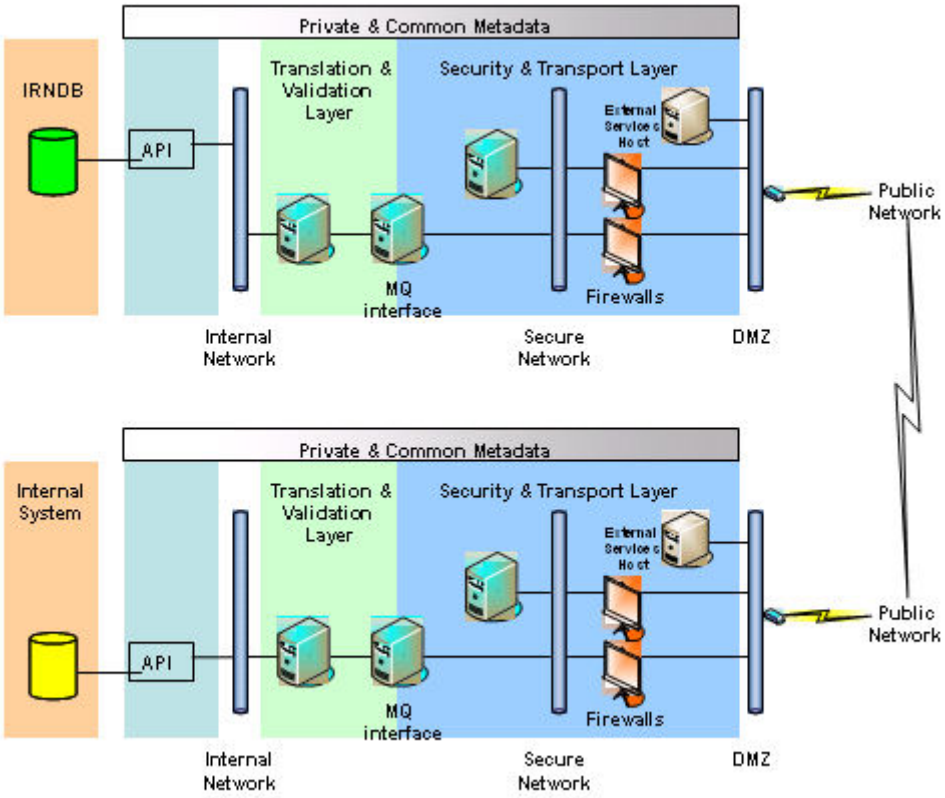
2.6 Reference Files and Databases (ref 4.2.12.1)

Access to the TAF TSI Reference files will be controlled and managed through the Common Interface. Some of the data in the TAF TSI reference files will be part of the public metadata (see the FRS documents for Reference Files) and each will have its own Human-Machine Interface for administration purposes. These Human-Machine Interfaces will be separate to the Common Interface.

2.7 Infrastructure Restriction Notice Data (IRNDB) (ref 4.2.3.1)

IRNDB updates requested by RUs will be transferred to RUs from the requested IM's IRNDB via the ~~CI~~Common Interface, so that each actor obliged to read data from an IRNDB calls an 'IRNDB access API' in the API layer of the ~~CI~~Common Interface to obtain the relevant data from each of all of the IMs involved in the proposed route.

Distributed IRNDBs may exist in parallel to any multi-IM IRNDB.



2.8 Metadata (ref 4.2.14.2 & 6)

The Metadata required for operation of the Common Interface is in two parts, Private Metadata (defining the local conditions of the particular implementation of the Common Interface) and Common Metadata (defining all the public information available to all actors and centrally managed and distributed).

2.8.1 Private Metadata (ref 4.2.14.2 & 6)

- Specific Implementation Metadata about APIs in use,
- Translation tables from internal system data to and from TAF TSI Messages
  - one-to-one mapping
  - code-to-code mapping
  - units translation (accuracy, rounding)
- Translation of message header information
- Audit and logging settings
- Administrative rights to amend the metadata (User & system profiles, access rights, authentication details, security)
- Local Transport protocol & system information linking the APIs and Transport & Validation layer (refer to documentation regarding systems referred to in section 2.2)
- Single or multiple instances of Private Metadata may be required.

2.8.2 Common Metadata (ref 4.2.14.2 & 6)

API Metadata (ref 4.2.14.1 & 7)

- Generic information describing the implementation of APIs
- API audit and logging requirements

Translation & Validation Metadata (ref 4.2.14.1, 6 & 7)

- TAF TSI Message Metadata (XSD)
- Data quality (validation)
- Registered Common Interface installations (addresses and Information about message versions in use at each destination Common Interface)

Security & Transport layer Metadata (ref 4.2.14.2, 4, 5 & 7)

- Partner definitions including partner queue names, TCP/IP definitions for each partner, network(s) to be used for each partner, service times for each partner.
- Open Message Queue definitions, names, restart/recovery, persistency, time outs, dead letter queue, maximum size, heartbeat checking
- Channel definitions queue manager to queue manager
- Logging definitions, locations, size, cyclic parameters etc
- Security definitions
- Message definitions (these override defaults above on a per message basis) expiry, triggering, priorities

- Error handling
- Naming standards
- LDAP usage, queue names
- Performance of ~~CI~~Common Interface throughput, response times
- Technical implementation, scalability
- Support details
- Security information (Authentication and security data for messages, Reference file, WIMO and IRN database access information, encryption keys)

2.8.3 Queue Naming (ref 4.2.14.1, 2 & 7)

Public Queue names should describe their principle in the common metadata. The Common Interface should not require the internal application to know the public queue names – the application should receive a message - or request that a message is sent - to/from the actors, not the public queues.

Private queues may be named as required for the operation of each specific implementation of the Common Interface. Private Queue names will be stored in the Private Metadata.

Public Queue names will be named according to the following convention :

Queue Name	**Content**
Characters 1234	Company Code (numeric)
Character 5	Queue type – 0, production, 1-5 test
Character 6	Direction – O, Outbound, I, Inbound

2.8.4 Metadata Management & Distribution (ref 4.2.14.2 & 6 and 4.4.2)

The common metadata required for operation of the Common Interface must be managed by a central body responsible to the actors using the Common Interface. The common metadata must be implemented using a secure, replicated metadata repository capable of secure, automatic distribution in near real-time to the Common Interface metadata instances in use by the actors. Change management, including the use of the metadata for non-TAF TSI messages, is the responsibility of the metadata management governance.

Private metadata may be managed by each individual actor implementing the Common Interface.

2.9 Human-~~Machine~~Computer-Interface (ref 4.2.14.1)

The following administrative functionality is required:

The following administrative functionality is required for each layer in the Common Interface

- Start/stop/reset
- Parameterised process activity logging (entry, translation/validation & exit of the Translation & Validation layer)
- Monitoring of activity in real-time
- Automatic Queue Recovery
- Version management of the Metadata

This functionality must be available via a standard internet browser.

3 Performance and Data Quality

3.1 Sizing and performance (ref 4.4.1)

Capacity
A single instance of a Common Interface should be capable of communicating with up to 10000 other Common Interface instances.
A single instance of a Common Interface should be capable of communicating simultaneously with up to 30 “existing applications”.

Performance
A Common Interface instance should be capable of sending/receiving:

- nominal stress: a sustained rate of up to 30 TSI TAF messages/database accesses (in a random mix) per second;
- peak stress: a 1 minute peak of up to 50 TSI TAF messages/database accesses (in a random mix) per second.
- Take into account the wagon reporting to WIMO

The delay of any message passing through the chain Common Interface – internet – Common Interface should be less than 2000 ms, and 90% should be within 500 ms (assuming an infinitely fast internet and a nominal stress).
The expected average transaction volume to WIMO is 2.9m per day.

Minimum unplanned unAvailability, MTBF, MTTR
A Common Interface instance should be capable of running continuously. Availability of a Common Interface should be designed to deliver at least 99.9% measured on a monthly basis (maximum total outage 525 minutes/year).
Maximum number of outages per year is 50 (MTBF=1 week)
Automatic recovery of software-related errors shall take place within 30 minutes (MTTR).

Minimum planned unAvailability, MTBF, MTTR
Automated update the public metadata within 10 seconds.

Internet resource utilisation
Under nominal stress, the communication between two Common Interfaces should not cause more than 600 kb/s (thousand bits per second) load per direction on the internet. (30 messages/s x 1000 B x 8b/B x 2 overhead factor = 480kb/s). Strongly recommend compression.

Computer resource utilisation
A Common Interface communicating with 5 existing applications and 10 other Common Interfaces under nominal stress, should not cause a load on the cpu or any other critical resource of the computer the Common Interface is running on greater than 50%. The vendor should indicate the requirements on the computer, such as cpu speed and memory.

3.2 Data Quality (ref 4.4.1)

3.2.1 Prerequisite

Chapter 4.4.1 of the Telematics Application for Freight Services Sub System (TAF TSI) documents the essential requirements for Data Quality. This is a prerequisite for effective data exchange and comprises the following elements:

- Completeness
- Accuracy
- Consistency
- Timeliness

The sender of each message will be responsible the for the correctness of the data sent and must verify that it is in compliance with the guidelines stipulated for that message. This means that the data must not only be complete and conform to the metadata requirements (syntax-level), but must also be accurate, timely and consistent for the receiving application to effectively import the message. This requires two distinct levels of validation, as described below:

3.2.2 Level 1 Compliance Checking

As the TAF TSI messages are defined using WC3 XSDs according to Recommendation 28, the schema contain all metadata needed for strict Level 1 compliance checking. This syntactical-level check validates the interchange, or part of it, for compliance with the schema. This checking normally happens at the translation and validation layer, before the data is treated by the API. It includes validation for field lengths, data types, codification (where enumerations exists), presence or absence of required data, valid payload entries where defined and the order of data transmitted. The schema validation is more robust and provides a higher level of compliance checking than traditional EDI.

The XSD metadata provides a perfect solution to meet the needs for Completeness and some of the Accuracy requirements as defined above.

As a minimum, Level 1 Compliance Checking should be implemented in the early phases of the TAF TSI message exchange. This should be part of the Common Interface Translation and Validation Layer.

3.2.3 Level 2 Application Validation

According to the TAF TSI the originator of the message must ensure a data quality assurance check using their own resources. Data quality assurance includes comparison of data from reference file databases provided as part of the TSI plus, where applicable, logic checks to assure the timeliness and continuity of data and messages.

Data must be of high quality if they are fit for their intended uses, which means they

- Are Error free: accessible, accurate, timely, complete, consistent with other sources, etc., and
- Possess desired features: relevant, comprehensive, proper level of detail, easy-to-read, easy-to-interpret, etc.

For example, while the Schema can validate that a CompanyIdent contains 4 integers, it cannot assess the validity of that code against a common reference file in the translation and validation layer. It is therefore up to the sender to assure that the information is valid

in his own application before generating the message. The receiver must also perform the same validity check before the data is imported into his system. Additionally, Level 2 Application Validation should also provide consistency and timeliness checks according to the requirements defined by the target application.

This level of validation is a function of the internal systems as it presupposes that the necessary reference data are in place and applied consistently in the senders' systems.

3.2.4 TAF Acknowledgments (ref 4.2.14.7 & 4.4.1)

The TAF TSI states in the Common Interface section that "based on the results of authenticity verification of incoming messages, a minimum level of message acknowledgement can be implemented." Message acknowledgement can be positive or negative. XML acknowledgment messages have already been implemented within the industry, particularly for use with the ENEE application. These messages have the ability to communicate either syntax or specific application error back to the sender. The messages also have the capability to inform the sender whether the message have been accepted or rejected by the application.

Messages that require acknowledgement are those which update and delete database entries (excluding event reporting) and those which have application or technical errors.

3.2.5 TAF Acknowledgement Message (ref 4.2.14.7 & 4.4.1)

Based on existing acknowledgements, a prototype message has been developed to comply with the recommendation as stated above.

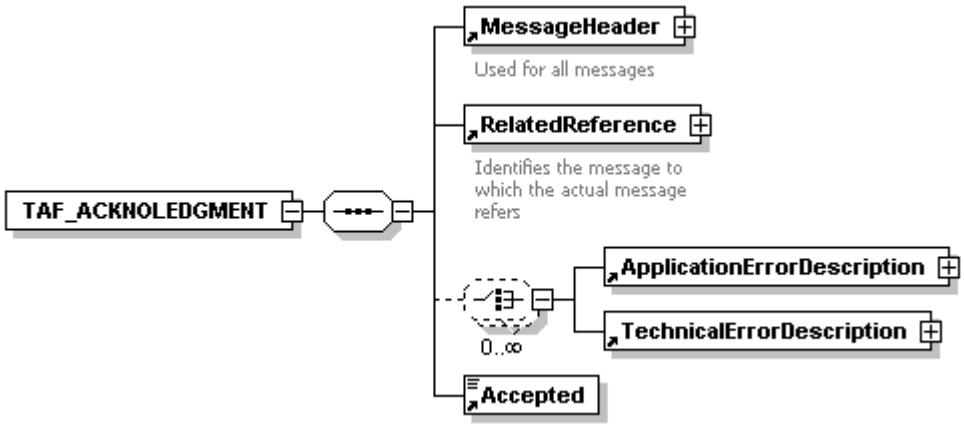


Diagram 3.1 – TAF_Acknowledgment

This message serves the dual purpose of providing reporting both Level 1 and Level 2 errors, a link back to the original message being acknowledged and an acceptance/reject flag.

3.2.6 Level 1 – Compliance Checking (ref 4.2.14.7 & 4.4.1)

This action (acknowledgement or rejection) indicates the result of a syntactical check of the complete received XML document based on the schema and is acknowledged with the TechnicalErrorDescription as seen below:

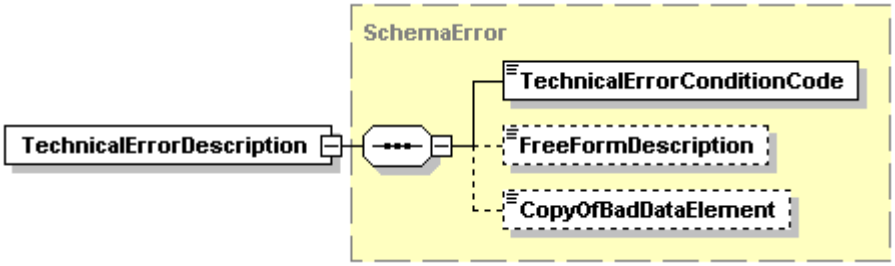


Diagram 3.2 – Technical Error Description

This element contains

- TechnicalConditionCode that needs to be defined by the user community and is based on criteria established by the metadata defined in the relevant XSD.
- Free-form description of the technical error, if needed
- Copy of the bad data element.

3.2.7 Level 2 – Application Validation (ref 4.2.14.7 & 4.4.1)

This action (acknowledgement or rejection) indicates the result of an application validation of the complete received XML document based on the schema and is acknowledged with the ApplicationErrorDescription as seen below:

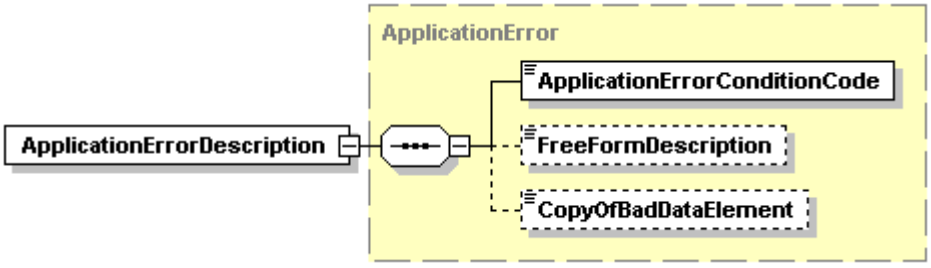


Diagram 3.3 – Application Error Condition Description

This element contains

- ApplicationErrorConditionCode that needs to be defined by the user community and is based on criteria established by the application.
- Free-form description of the application error, if needed
- Copy of the bad data element.

4 Change Management

All change management will be in accordance with Chapter 7.3 of the TAF – TSI ("Management of Change"). The following is an extract from paragraph 7.3.5:
Change management procedures should be designed to ensure that the costs and benefits of change are properly analyzed and that changes are implemented in a controlled way. This requires the defined change management process and associated tools to ensure that changes are recorded and applied to the specifications in a cost-effective manner. Whatever the specific details of such a process might eventually be, the latter should be broadly mapped on a structured approach as follows:
Any user may propose a modification to the reference file. Basically, these change requests (CR) may be due to enhancements or corrections of the system. Enhancements are meant to add value to the system and corrections are to repair errors. Serious errors are to receive priority and be fixed immediately. Minor errors (i.e. documentation, screens, etc.) are to be processed as CRs.
CRs are logged in a dedicated change management register. Each CR should contain the proposed change, the reason, the urgency / importance and the originator. The CAS approves or rejects CRs based on a change management procedure as shown in the diagram below. Approved CRs will be included in the CR-plan with a description of the type of the change.

There are four types of CRs:

- **Functional**: to add a new function to the application or to repair a small error in a system function
- **Technical**: to improve the performance or to connect another device or system
- **Organisational**: to modify a procedure or to repair a fault in a rule
- **Documentary**: improve or update documentation, hand books, help screens

System updates (functional and technical) will be tested first in a test environment before they can be switched over to the production system. A system change in the test system must be validated by the CAS before it can be put into operation. The users are to be informed via the defined communication channels on planned changes (including the scheduled date).

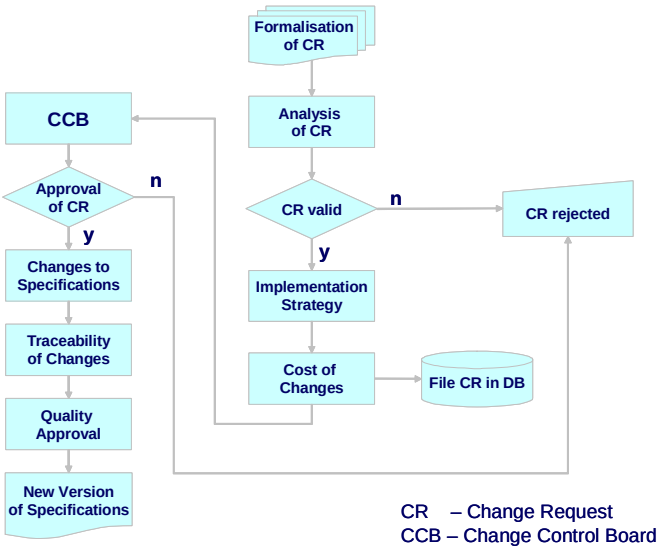


Diagram 4.1 Change Management Procedures