OETCS/WP3/D3.7.0

openETCS

ITEA2 Project
Call 6 11025
2012 – 2015

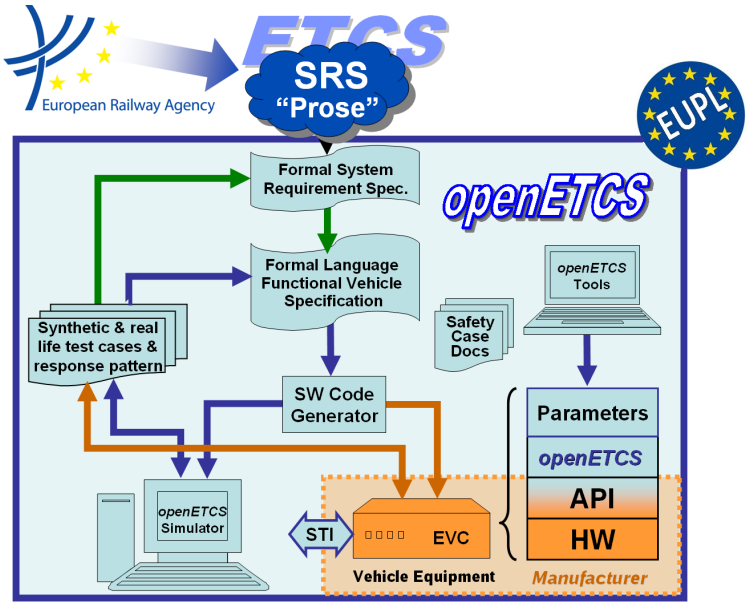
Work Package 3: “Modeling”

openETCS System Architecture requirements specification

Hardware Component requirements specification

Christian Giraud, Fausto Cochetti

November 2015



Funded by:

Federal Ministry of Education and Research

Région de Bruxelles-Capitale

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

This page is intentionally left blank

Work Package 3: “Modeling”

OETCS/WP3/D3.7.0
November 2015

openETCS System Architecture requirements specification

Hardware Component requirements specification

Document approbation

Lead author:	Technical assessor:	Quality assessor:	Project lead:
location / date	location / date	location / date	location / date
signature Fausto Cochetti (Alstom)	signature Christian Giraud (Alstom)	signature Marc Behrens (DLR)	signature Jacques Pore (Alstom)

Christian Giraud, Fausto Cochetti
Alstom

Architecture and Design Specification

Prepared for openETCS@ITEA2 Project

Abstract: The aim of this document is to introduce a *physical system perspective* matching with the Open ETCS program.

The openETCS modeling work-package does not include the development of a specific target architecture to be integrated with the train on-board system neither does it contemplate the model of such an architecture. The functional model designed in the Scade environment does refer to an application programming interface (API) to reference system resources and interface management. This approach based on an industrial proven specification [1][2][3] provides a level of abstraction from proprietary architectures. The API gives a clear insight on the platform constraints, such as dynamic aspects and timing management, principles of separation of functions and event interrupt concepts, when relevant for the application.

This document describes a generic hardware decomposition aimed to support the designer with a system perspective on the train-borne subsystem applied for the ERTMS/ETCS functionality.

The design paradigms for a suitable physical on-board system architecture remain open and need to be integrated with specific design choices. The principles refer to a proven industrial platform and to the openETCS on-board functional model. The concepts are described at a sufficiently high level so that they are easily customizable for a specific as well as for a generic not relating to any specific proprietary system design.

The hardware abstraction in the software system is obtained by referencing to a generic API. The requirements of this API are described in a separate document delivered by Alstom [4].

The functional scope of the openETCS OBU model is documented in D3.5.x, where x denotes the iteration.

Modification History

Version	Sections	Modification / Description	Author	Date
0.1	all	Initial release of the document	Fausto Cochetti	30.11.2015
0.1.l	appendix	technical compendium		30.11.2015

Table of Contents

List of Figures

1 Purpose of the document

This document is managed as a deliverable of the modeling work package with denomination D3.7.x, and contains advices and recommendation for the design of a physical system architecture.

The development of the functional model is done iteratively increasing the scope in steps, the last digit of the deliverable identifier, i.e. x, denotes the release of the model to which it applies. If the functional model requires to update the system architecture a consistent version number will be applied to this document as required by the Model release version.

This document complements the indications contained in the API requirements specification and the documentation derived from this as the generic openETCS Application Programming Interface (API), available at https://github.com/openETCS/modeling/blob/master/API/description/api-description.pdf. [4]

1.1 Input Documents

The following documents provide a context for the system perspective.

- ERA Subset-026 [5], V3.3.0
- ERA TSI CCS Documents
- openETCS API documentation, available at https://github.com/openETCS/modeling/blob/master/API/description/api-description.pdf [4][2][3]

2 Introduction

Designing a sub system integrable with the train borne system is a complex task. The designer faces a large variety of serious challenges and design complexities.

Before the functions are actually implemented, a system architect will have to select an appropriate hardware-software concept out of the large number of available boards, controllers, network and bus constraints. He will as well include robustness criteria against environmental influences.

Memories, operating systems, drivers, generic and application software segregation as well as selection criteria for sensors and actuators need to be correctly assessed.

The target architecture has to meet a large variety of requirements. Criteria of timing, Bus bandwidth, processor and peripheral performance, memory size, safety principles and possible processing or data transfer bottlenecks. Environmental conditions, timing constraints, robustness against specific interferences shall constantly be tracked.

Power requirement as well as allocation of availability, maintainability figures to enumerate only the most relevant items accompany all the design phases.

On top of this a specific vital architecture has to be selected and the required integrity level has to be granted. The relative safety constraints have to be assured and maybe exported.

Selecting the components matching these is a critical phase. Over-dimensioning the architecture may impact on cost factors relevant for the market access of the system. Under-dimensioning the architecture design could result in not achieving performance constraints, thus compromising system quality and suitability. Early architectural choices have a dominant impact on the success of the new system.

The system architects will commit to efficient design choices according to the target project margins and all this within the frame of a defined project delivery time schedule.

Due to the fact that the design verification phase, requiring to have completed all the integration steps, may be very late in the release process a high precision during the system architecture design is mandatory.

Therefore highly experienced System Designer are considered as the key factor for a reliable achievement of expected design result.

A primary goal of the openETCS ITEA2 project is to provide a formal specification and a model of an ETCS onboard functionality according to the specification defined in Subset-026 [5] by the European Railway Agency (ERA).

The Model-Based Development process is an approach that allows engineers to specify the behavior of a system and to simulate and execute it in a very early development stage.

Once a model-based development process has been established, engineers should be able to apply new technologies and tools to enhance and shorten product development cycles, e.g. by introducing generation of Model Validation test cases and target Code directly from the model. This enables to improve the V based development process to save development time and effort while preserving or improving the dependability of the developed systems.

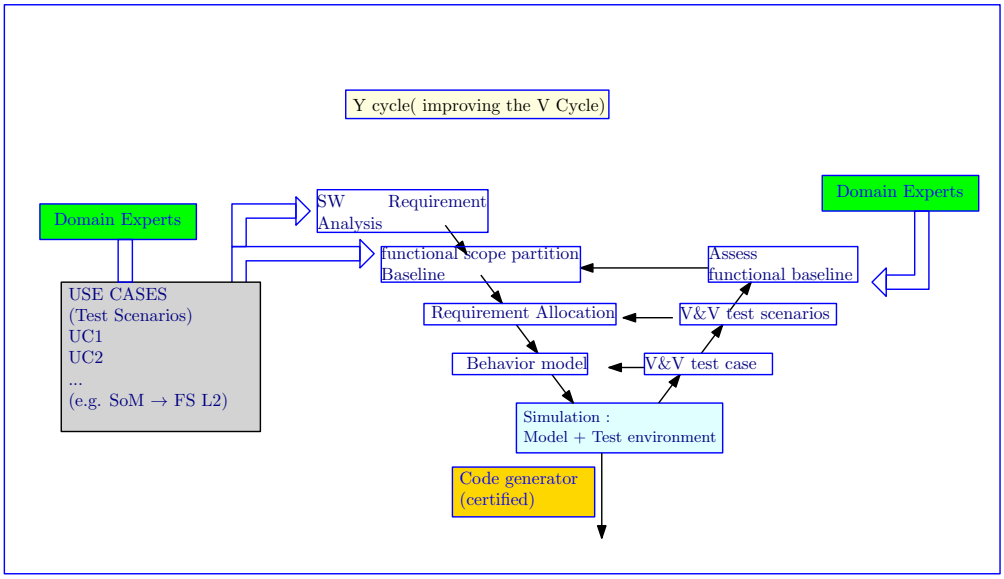


Figure 1. SRS modeling cycle

The methodology makes it easier to understand requirements and increases the correctness of the requirements, the correctness of the design and the code with respect to the requirements. An integration of system-level and design-level modeling tools allows a virtually integrated V-process that is sharpened up to a Y-based process with the required steps at the bottom of the former V being considerably automated (see figure1)

Nevertheless when specifying the overall software architecture, the designer should be aware of the implications of software design decisions on the target end system.

2.1 Safety Integrity and Functional Safety according CENELEC

The Railway Industry currently relies on the international standard group of coordinated standards: EN 50126 Railway applications The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) the EN 50129 Railway applications Safety related electronic systems for signalling and the EN 50128 Railway applications - Communications, signalling and processing systems Software for railway control and protection systems to provide a rational and consistent approach for the development of safety-related systems.

This group of standards owes much of its direction and contents to the IEC 61508 standard that is a generic safety standard for electrical/electronic/programmable electronics safety-related systems.

Both of these IEC and EN standards share the same philosophy in the sense that they:

- consider all relevant product and software safety life-cycle phases, from an initial concept phase to maintenance and decommissioning when these systems are used to perform safety functions;
- intend to shape a safety awareness;
- have been conceived with a rapidly developing technology in mind;
- provide methods and rules for defining safety requirements necessary to achieve defined functional safety.
- use Safety Integrity Levels (SIL) for specifying the target level of safety integrity for the safety functions to be implemented.
- adopt a statistical risk-based approach for the determination of the SIL requirements;
- distinguish between safe and unsafe failure modes and requires precautions against undetected failures.

According the Cenelec norms the product is subject to a certification process. The definition of the equipment under control (EUC) depends on the scope of the certification. It can be, for example the complete ERTMS/ETCS subsystem or a module of it.

The term safety-related is used to describe systems that are required to perform a specific function to ensure that risks are kept at an acceptable level. Such functions are, by definition, safety functions. Two types of requirements are necessary to achieve functional safety:

- Safety function requirements (what the function does),
- Safety integrity requirements (the required likelihood of a safety function being performed satisfactorily).

The safety function requirements are derived from a risk analysis phase, in the scope of EN 50126, where significant risks for equipment and any associated control system in its intended environment have to be identified. This analysis determines whether functional safety is necessary to ensure adequate protection from unacceptable risks. Functional safety is therefore a method of dealing with risks to eliminate them or reduce them to an acceptable level. EN 50128 specifies four levels of safety performance for a safety function. These are called Software Safety Integrity Levels (SwSIL).

2.2 Reference to the openETCS functional Model

The openETCS OBU partial model has been developed according to the specification given in ERA Subset-026 [5], Version 3.3.0. The software release is publicly available on a repository at

https://github.com/openETCS/modeling/tree/v0.3-D3.6.3

3 Design principles

3.1 Use of Scade modeling tool

While the use of the modeling tool provides a strong support for the coherent application of an uniform coding standard, it introduces some limits and constraint on the use of operators and data structures. This issues should be addressed at an early stage of design in order to provide best performing result from the code generator.

A specificity of the European Vital Computer (EVC) system is the coexistence of quite linear complexity as e.g. the one found when managing level and mode transition, together with high demanding algorithms like the ones found when approaching the braking curves in target supervision. The first type of function can be directly derived from requirements and can be coded without any additional interpretation or design criteria. This applies in the same way to other safety related systems like e.g. Interlocking applications, where functional (safety) rules are assessed and fixed and can be applied according to the specific configuration. The second type of functions need to be accurately analyzed in order to provide best performing results and this implies among others to provide suitable operators and data structures that allow fastest results lowest memory allocation and simplest rules.

3.2 System Time and Time Stamps

Basically a Scade model is based on the cycle number as its main timing source. If the model will not be clocked with an equidistant period, a different time source (actual system time) is needed as an input. The following rules shall apply for the model in such case:

- The model must not rely on a fixed period between the model clocks. Timing functions based on clock cycle counting are not feasible.
- In a first phase the model is allowed to use the actual system time for those calculations if this is not deductible form event time stamps.
- The Basic Runtime System will keep the actual system time applied to the model as an input unchanged within the same Basic Runtime System cycle.
- The actual system time will be strictly monotonic increasing between two subsequent Basic Runtime System cycles
- The interval between two subsequent Basic Runtime System cycles should not be critical for the application. Measures done on the partial Scade model indicate that a cycle of 1 ms could be achieved.
- Practically, the actual system time could be used in the model for the implementation of time intervals.

3.3 Time stamp on the physical System

In addition to the "cyclic execution" principle, the application must take into account that the various peripheral boards/sub-systems in charge of the input/output treatment are not synchronized and that their performances are not identical, nor constant in time;

- e.g. time delay due to filtering and smoothing of analog data such as MMU sensors, train digital inputs, ...
- e.g. variable propagation delay on bus systems depending on performance (chained treatments) and arbitration criteria.

Therefore,

- The input data present at the interface of the Onboard ETCS/EVC will be provided with a variable delay to the application (fluctuation of delay duration, jitter);
- The application output data are provided to the EVC/Onboard ETCS interface with a variable delay (non deterministic delay, jitter).

For each input and for each output data, a *T_min* (minimum delay of EVC input/output treatment) and a *T_max* (maximum delay of EVC input/output treatment) will be defined as shown on the diagrams in figure 2 :

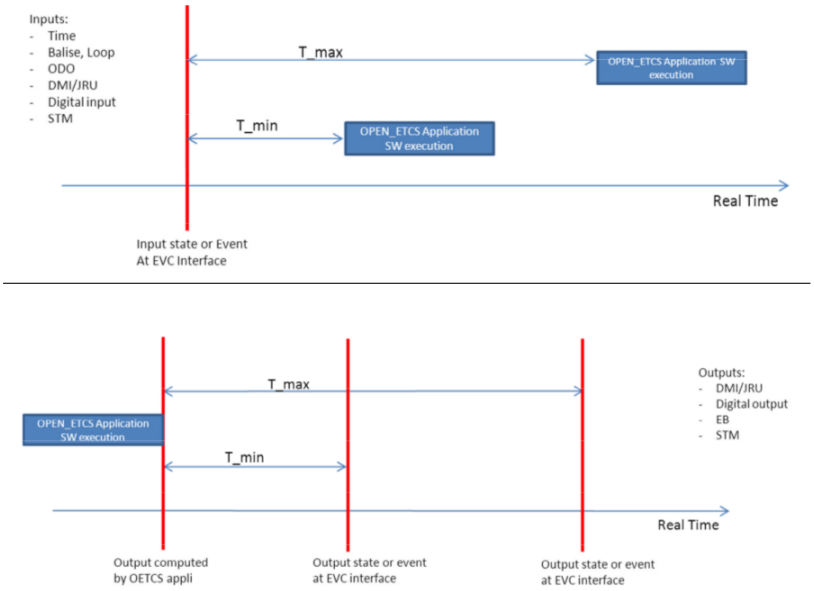


Figure 2. ***Managing delays with time stamps***

Notice that when the input data are time-stamped at the moment they are produced (by the source), the receiving applications may apply a correction in order to manage the delay of transmission of the data; assuming that the clocks of the various calculators are synchronized (e.g. the accuracy of the clock synchronization within the Model EVC is 1ms).

4 OB subsystem context

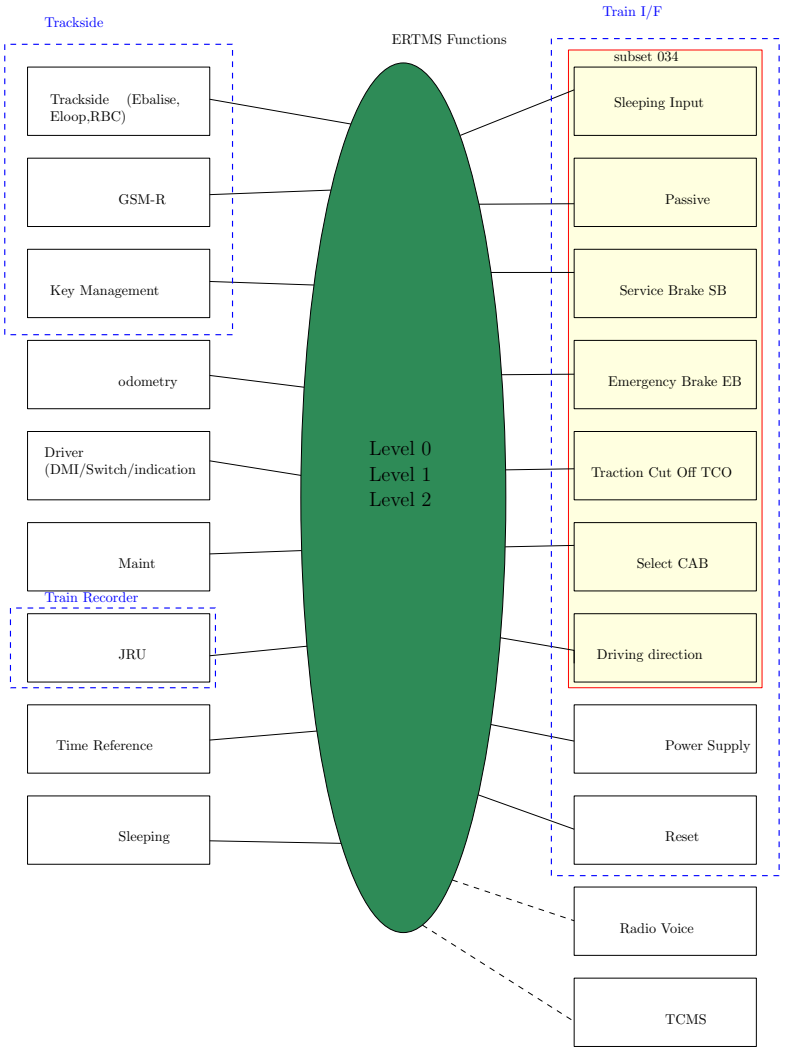


Figure 3. OB context diagram

5 Elements of the ETCS OB Architecture

The ERTMS/ETCS OB sub-system comprises at least following equipments:

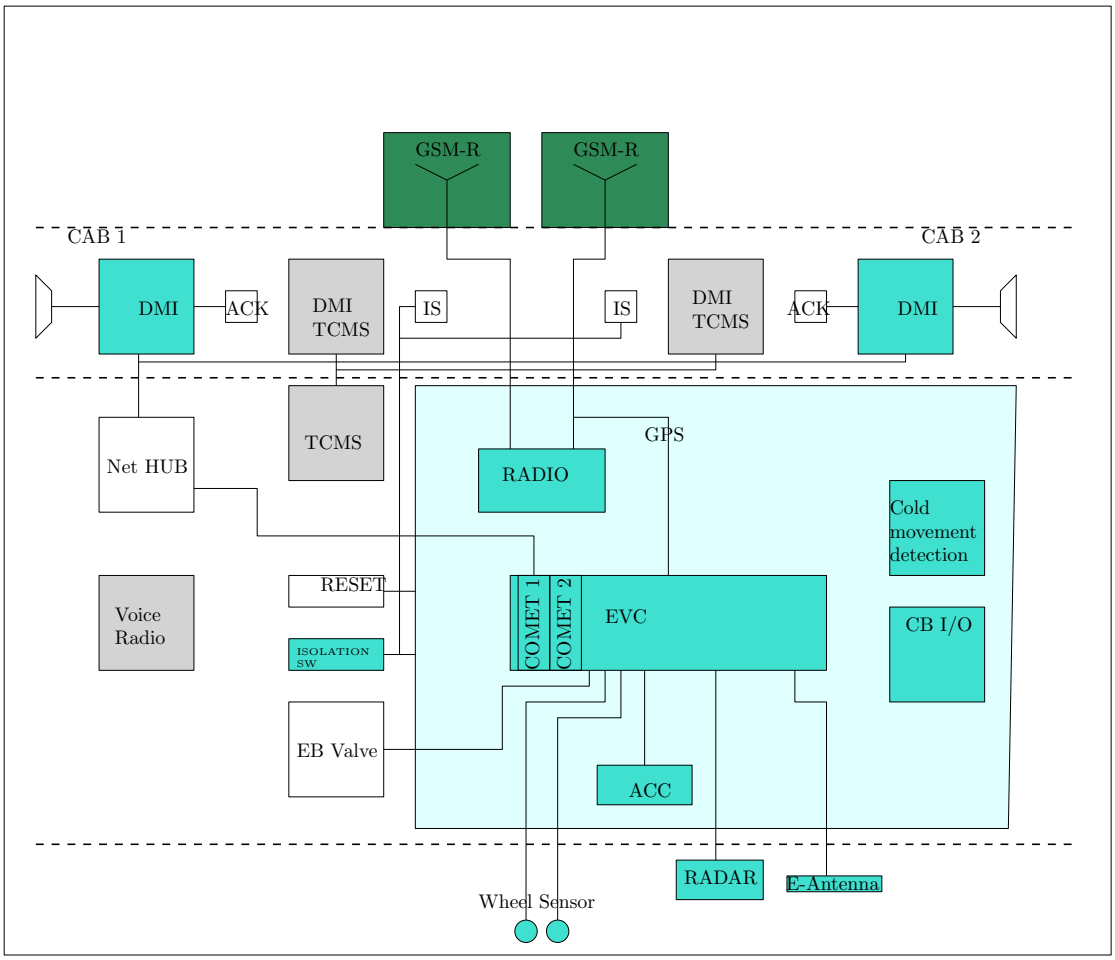


Figure 4. OB architecture Overview

The ETCS cubicle is realized as one mechanical assembly with a standard housing IP20 and will include :

- the EVC (SIL4 equipment);
- an internal recorder memory for Juridical Data and diagnostic data;
- one GSM-R Data Radio module for GSM-R communication with trackside
- relaying devices (contactors, circuit-breakers, relays);
- fan modules for the cooling function ;
- one accelerometer for the speed and distance measurement;
- a Cold Movement Detector to detect train movements when ETCS is powered off.

The EVC includes among others the Movement Measuring Unit (MMU) providing the odometry function. The odometry implies a complex vital function non mapped on specific Unisig requirements.

Other components installed elsewhere are listed below:

- The Isolation switch aims to isolate the ETCS OB sub-system from the train braking and traction interfaces when ETCS OB is excluded; associated bypass and isolation circuitry is installed to perform the isolation function. An Isolation indication must be visible to the driver in case of ETCS OB isolation.
- Reset button A periodical reset of the ETCS OB sub-system is necessary in order to ensure its required level of availability and safety.
- One ETCS driver-machine interface (DMI) (including display, loudspeaker and associated acknowledgement button) is installed in each cab. Other components (EVC, odometry, radio, ...) can be unique for both cabs or can be installed separately for each cab"
- The Eurobalise/Euroloop antenna is installed for Eurobalise and Euroloop reception.
- A radar device is installed for the speed and distance measurement.
- Wheel Sensors are installed for the speed and distance measurement.
- Redundant antennas are installed for GSM-R data communication and GPS signal acquisition.

A redundant network (bus, switch) is used for the communication between EVC, DMI(s), Train Recording Unit (TRU), and the other OB subsystems.

Train Recorder Unit The Train Recorder Unit shall record information from the ETCS OB (diagnostic and juridical data) and from the train interfaces. The GPS signal (from a GSM-R/GPS combined antenna) will be received for the data time stamping. The TRU will also compute and record the train speed. The Train Control Management System (TCMS) will exchange inputs/outputs information with the On-board (OB) ETCS OB subsystem

The TCMS will manage the TCMS DMIs and the data communication at train level (between the different vehicles in a multiple configuration). As a remark the TCMS and voice radio systems are not part of the ERTMS/ETCS system.

Braking electro-valves, train devices The braking and traction devices commanded by the ETC OB (emergency brakes, service brake and traction cut-off) are part of the Rolling Stock subsystem.

References

[1] Jan Welvaarts and Baseliyos Jacob. *Requirements on openETCS API*, 11.05.2014 edition, May 2014. https://github.com/openETCS/requirements/blob/master/D2.7-Technical_Appendix/2014-05-13-Munich-Meeting/Bullit%20point%20openETCS%20requirements_20140511.pdf.

[2] Alstom Transport. *Appendix application layer*, v1.2 edition, 2014. https://github.com/openETCS/requirements/blob/master/D2.7-Technical_Appendix/OETCS_API%20Requirements_appendix_application_layer_v1.2.pdf.

[3] Alstom Transport. *Appendix Functional Data Dictionary*, v1.1 edition, 2014. https://github.com/openETCS/requirements/blob/master/D2.7-Technical_Appendix/OETCS_API%20Requirements_appendix_functional_data_dictionary_v1.1.pdf.

[4] Nicolas Boverie. *API Requirements for OpenETCS*. Alstom Transport, v1.4 edition, September 2014. https://github.com/openETCS/requirements/blob/master/D2.7-Technical_Appendix/OETCS_API%20Requirements_v1.4.pdf.

[5] ERA. *System Requirements Specification, SUBSET-026*, v3.3.0 edition, March 2012.

[6] Philip R. Bevington D. Keith Robinson. *Data Reduction and Error Analysis for Physical Sciences*, 3rd edition, 2003.

Acronyms

API	application programming interface
DMI	driver-machine interface
EVC	European Vital Computer
MMU	Movement Measuring Unit
OB	on-board
TCMS	Train Control Management System
TRU	Train Recording Unit