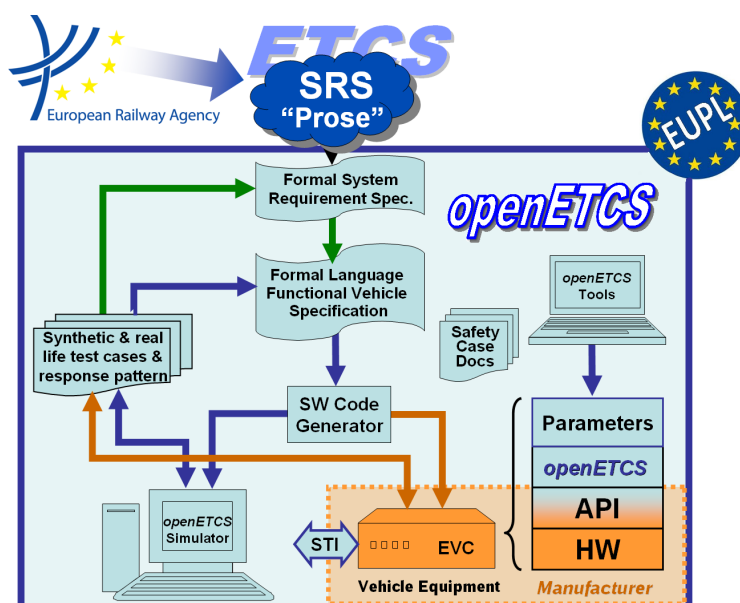OETCS/WP3/D3.5.1.2

openETCS

Work-Package 3: "Modeling"

openETCS System Architecture and Design Specification

Second Iteration: Scope of openETCS ITEA2 Functions

Baseliyos Jacob, Bernd Hekele, Marc Behrens, David Mentre, Jos Holtzer, Jan Welvaarts, Vincent Nuhaan and Jacob Gärtner

October 2014

This page is intentionally left blank

Work-Package 3: "Modeling"   **OETCS/WP3/D3.5.1.2**
October 2014

openETCS System Architecture and Design Specification

Second Iteration: Scope of openETCS ITEA2 Functions

Document approbation

Lead author:	Technical assessor:	Quality assessor:	Project lead:
location / date	location / date	location / date	location / date
signature	signature	signature	signature
Baseliyos Jacob	[assessor name]	Izaskun de la Torre	Klaus-Rüdiger Hase
(DB Netz)	([affiliation])	(SQS)	(DB Netz)

Baseliyos Jacob, Bernd Hekele

DB Netz AG
Völckerstrasse 5
D-80959 München Freimann, Germany

Marc Behrens

DLR

David Mentre

Mitsubishi Electric R&D Centre Europe

Jos Holtzer, Jan Welvaarts, Vincent Nuhaan

NS

Jacob Gärtner

LEA Engineering

Architecture and Functional Specification

Prepared for   openETCS@ITEA2 Project

Abstract: This document gives an introduction to the architecture of openETCS. The functional scope is tailored to cover the functionality required for the openETCS demonstration as a target of the ITEA2 project: the Utrecht Amsterdam use-case. It has to be read as an add-on to the models in SysML, Scade and to additional reading referenced from the document.

Modification History

Version	Section	Modification / Description	Author
0.1	Document	Initial document providing the structure	Baseliyos Jacob
0.2	Document	Workshop Results included and some pretty-printing	Bernd Hekele

Table of Contents

Figures and Tables

Figures

Tables

1 Introduction

1.0.1 Motivation

The openETCS work package 3 (WP3) aims to provide – amongst others - the software architecture for the openETCS kernel in order to eventually build the software itself. WP3 partner has put great effort in the openETCS software design, thus far without making definite choices on the software architecture itself respective of functional breakdown and data structures of the openETCS kernel. Since the project planning foresees in the production of a reference software to be used as a demonstrator by June 2014, it is of paramount importance that a design freeze of the openETCS kernel architecture be finalized shortly but no later than November 2014.

In compliance with the agreements made during the last WP 3 meeting at the 10.09.2014 in Brussels, DB has taken the initiative to design the aforesaid architecture including of functional breakdown and data structures in order to safeguard a timely delivery of these products. Furthermore, DB has ensured that these developments are focused on including end user requirements so as to develop a design in conformity with the needs and requirements of the operators. Specialists of DB and NS have cooperated together with other partners in WP3 to produce this document.

As referred to above, the architecture description has to be finalized in the month of November 2014. This version of the document is a draft version, demonstrating the general directions and philosophy of the architectural design, the functional breakdown of the software and the data structures. The design is focused on maximum efficiency in order to maximize on RAMS performance of the end product.

This document, named second iteration, is a draft document and will be developed until a complete architecture.

Since this is a work in progress, any remarks referring to the improvement of the document, including reporting errors, are more than welcome. Any additional work done thus far on the subject by other WP3 partners will be incorporated in this document as long as it is aligned with and consistent or complementary to the fundamental viewpoints advocated in this document after a review in respect to the openETCS process. At the same time, any contributions to the integration of which will demand discussion or changes of the fundamentals as proposed in this document, will be discarded with . Only in this way the ITEA2 project is able to meet its objectives as mentioned above. There will be two workshops in which there is due time and opportunities to fine-tune this document and its contents. Any comments will be addressed there.

It is urgent to definitely finalize the architecture on a short notice and therefore this document will rather prescribe than describe the openETCS architecture, functional decomposition of the system and the data structures within the limits as stated above. The document is divided in two parts, i.e.:

• A description of the general architecture of the openETCS OnBoard Kernel (software) including data structures prepared by NS. . . .
• A description of the functional decomposition of the openETCS OBU (software) in alignment with the general architecture prepared by DB. . . . .

Furthermore, this document describes the preconditions on which said descriptions are based on, the status and planning of upcoming activities and the main objectives of DB and NS as the End User. Wherever necessary, reference will be made to documents that underline the agreements that have been made during the openETCS architecture design process and the activities and meetings of WP3.

1.0.2 Objectives

The prime objective of WP3 is to produce a rapid prototype for the openETCS reference system that can function as a demonstrator in collaboration with WP 4 and WP 5 for the openETCS approach and will be used as such in the final phase of the project. That phase is the first half of 2015. This objective is defined as . . .

High level Objectives of this work:
«any further general statements on the ITEA2 objectives, like. . . »
• Work on a model bases approach and process for effective collaborative work within an international ETCS developer team as stated above, the project needs a definite architecture design by the end of 2014. This document targets:
• Defining the general design and conditions of the openETCS architecture, functional breakdown and data structures;
• Providing the guidelines for discussion during the workshops that are planned in October and November 2014 that will result in the final and decisive version of this document;
• Being the 'platform' for finalization i.e. whatever be the products or results of the workshops shall be integrated in this document. Apart from these general objectives, the document means to provide for the materials that will enable WP3 partners to improve the efficiency of the Work Package activities:
• The comprehensive architecture design shall enable splitting the work load according to the building blocks defined by the architecture and allocate strictly compartmented work parcels or activities to WP3 partners.
• Doing so will enable WP3 to avoid any double work
• Compartmenting the work load according to the functional building blocks as defined by the architecture will enable efficient planning of activities, be it individually or the integrated WP3 planning for the coming period, aiming at a just in time delivery of all results and products;
• Each partner that is responsible for one of the work parcels shall abide by the requirements in terms of quality and timeliness as defined by this document and prior documents and agreements made within the ITEA2 project.

1.0.3 Roles, responsibilities and tasks

In this section, the roles and responsibilities of the WP3 partners are confirmed, especially where they divert from what has been agreed upon at the start of WP3:
• First of all, in the last WP 3 meeting in Brussels on 10.09.2014 DB proposed to take over the lead of the architecture design and functional breakdown. At the subsequent weekly scrum

meeting on 12.09.2014, it was agreed upon by all participants that DB will take over the lead
(see Appendix . . .);
•**Planning:** Alstom as WP 3 leader will remain to be responsible for the planning and the
allocation of the defined tasks to the different partners
•**Roles:** Alstom will also coordinate the work and safeguard that the defined results will be
delivered according to the quality requirements that are agreed within the ITEA2 project and the
schedule and the milestones that will be agreed upon during the coming workshops;
•All WP3 partners will deliver the results or products according to planning as will be agreed
upon during the said workshops.

In the interest of a swift production of the critical documentation of which this version is a draft,
specific tasks will be defined in terms of concrete results to be delivered, the timeframes in which
these results must be produced and the partner who shall be responsible for that specific result
and the planning. This is to safeguard the timely delivery. The process will be described in the
next sections.

1.0.4 Process

- Alstom as WP 3 leader will be responsible for planning

- Time and quality aspects should be respected

- openETCS tools and methodology must be respected

Most of the operational requirements to WP3 in the last phases of the ITEA2 project have been
described in the former paragraphs. This section will describe the process which has to lead to
the final result: the reference software to be used in the demonstrator next year, more specifically
the final description of the openETCS architecture including the data structures and its functional
decomposition. The process will run as follows:
• DB will supervise the development of the first 'firm' draft of the specified products, 'firm'
meaning that changes can only be made within the framework of these products and not to the
fundamentals of these products as described in this document;
• DB will supervise the preparation of the two workshops that are proposed by Alstom and aim
at defining the final and definite architecture, data structures and functional decomposition. It
will make proposals for a planning of the critical tasks that remain to be done;
• Alstom will lead the two workshops following the preparations and the instructions of DB.
Since all participants are intrinsically involved in the development work and tend to immerse
themselves in technical discussions, for productivity purposes it is proposed to make use of
a (non-technical) moderator that will be made responsible for coordinating the meeting, the
discussions and the team efforts according ot the agenda.
• Also for productivity reasons, introductory presentations will be restricted to the contents and
setup of this document since all prior efforts have to be merged with this document and not the
other way around. Following a general introduction into the work that has been so far, the other
contributions will be scrutinized on their consistency with this document and any useful sections
will be merged with this document.
• During the workshops, there will be ample room reserved for enhancing this document, using
other documents pertaining to the same field of work that have been delivered by other partners.
Only material that is aligned with the general philosophy and structures proposed by this docu-
ment, will be integrated;

• In case conflicting views emerge over the benefits and value of certain contributions, at the very moment that parties conclude that they have conflicting views, these will be listed in an inventory for later discussion. The moderator shall note any such conflicts on the said inventory. Conflicting views will be treated at the end of each workshop whenever there is sufficient time or will be treated in a separate meeting that will be chaired by DB as coordinator of the ITEA2 project.

• The workshop shall be attended by a secretary provided for by DB who is responsible for making the workshop minutes. Within a week after each workshop these minutes shall be distributed among the partners that have cooperated in the workshop and be reviewed by those.

• The main objective of the workshops shall be the finalization of this document. In order to reach the specified result, the remaining tasks shall be identified and split into separate tasks or work parcels. Every task or work parcel will be allotted to one single responsible partner. Responsibility relates to the timely delivery of the defined result and according to quality requirements;

• Alstom, as WP3 leader, will be responsible for the planning, allocation of tasks or work parcels to partners and will ensure timely delivery of results;

• In case there will be tasks or work packages that cannot be finalized during the workshops or will be identified during the workshops and do not fit in the actual planning, these will be allotted in such a way that deadlines are perfectly clear and acknowledged by the party that is responsible for the results, fit within the general requirements of the project and are agreed upon in writing and executed by the responsible partners according to agreement;

• DB as partner that has integral responsibility for both the ITEA2 project and responsible as well for the architecture etc. , is entitled to interfere take over the role as leader / coordinator in case the workshops prove to be insufficiently productive;

• All output will be such, that it can be integrated in this document. It is the responsibility of DB to integrate the results and to deliver the final and definite version of this document.

• The document concept will follow the openETCS process and tools (LaTex and Git-hub).

1.0.5 Assumption and Preconditions

• All future contributions shall be fully aligned an compliant the finalized and approved document

• Alls documents produced by the partners are requested to be compliant and merge to this document; other contributions will be discarded

The workshops are all about working as swift, as efficient and as productive as possible and make full use of the potential made available for these workshops by the partners. It is expected that the partners in the workshops will have the express intention to:

• Contribute to the workshops with the intention to finalize the openETCS architecture;

• Provide resources according to the agreements made prior to the Workshops;

• Focus primarily on getting concrete results regardless of methodological issues that might arise. Where necessary or opportune, classical project management methodology will be applied;

• Provide full transparency with respect to experience, knowledge base and information touching the subjects to be treated in the workshops;

• Document on paper or electronically all output of the workshops and integrate these with the underlying document;

• Restrict discussions only to topics that have an immediate impact on the content or the quality of the end product: the improved version of this document.

1.1 Functions ERTMS/ETCS

The ERTMS / ETCS system was developed with a view to interoperability of trains on the different European rail networks. It is divided into "tracks" - and "board" finishes and shall establish a mutual message operation, by beacons or through a "radio" - The transmission system (in this case a mobile telephone network GSM-R) is performed. It defines several operating levels, and the system must also interfaces with the existing monitoring systems of the trains (using STM) have. The ERTMS / ETCS system provides the transport operator (the track) the choice of conditions concerning the use and operation. The train must therefore may go with different operating conditions on routes. Thus has the onboard equipment but must be implemented, to the interoperability of the train to ensure on the other networks. These functions must therefore correspond to one standard: the SRS (version 3.3.0).

application functions, which have two different species of origin: defined in the SRS: here one finds in particular the speed monitoring- and transfer functions; these functions must be implemented in full accordance with the SRS; they can in indeed be on any network on which the train is used; these functions are described below in Section 5;

Moreover, there are functions to adapt to the train: so, for example, the processing a "separation distance" in the airborne equipment trigger: This is dependent on the distribution of functions between the Control monitoring equipment (which the ERTMS / ETCS), and the other CCS Systems.

1.2 openETCS Architecture: Iterations and History

The openETCS Architecture and Design is implemented in iterations [**?**]. The current step (second iteration) is based on a step to implement the kernel functions of the ETCS system [1]. For a better understanding of the scope the Iteration is described in the following.

1.2.1 First Iteration Functional Scope: The Minimum OBU Kernel Function

The openETCS first iteration architecture and the design of the openETCS OBU software as mainly specified in [2] UNISIG Subset_026 version_3.3.0.

The appropriate functionality has been divided into a list of functions of different complexity (see the WP3 function list [3]).

All these functions are object of the openETCS project and have to be analysed from their requirements and subsequently modelled and implemented. With limited manpower, a reasonable selection and order of these functions is required for the practical work that allows the distribution of the workload, more openETCS participants to join and leads to an executable—limited—kernel function as soon as possible.

While the first version of this document focuses on the first version of the limited kernel function, it is intended to grow in parallel to the growing openETCS software.

The first objective of the first iteration was

- "Make the train run as soon as possible, with a very minimum functionality, and in the form of a rapid prototype."

This does not contradict the openETCS goal to conform to EN50128.

- After a phase of prototyping, the openETCS software shall be implemented in compliance to EN50128 for SIL4 systems.

1.2.2 How to find the functions of the First Iteration in the Architecture

The functions will be merged with the new architecture. Wherever a function has already been in the scope it will be marked as "first iteration".

1.3 Glossary and Abbreviations

ERTMS/ETCS Glossary of Terms and Abbreviations SUBSET-023, ISSUE: 3.1.0, 12.05.2014,

API Application Programming Interface
EVC European Vital Computer
BTM Balise Transmission Module
SRS System Requirements Specification

SRS-Subset 26
QA-Plan: D1.3.1
Process: D2.3
Methods: D2.4
API: D2.7

2 The openETCS Architecture

3 SRS Architecture
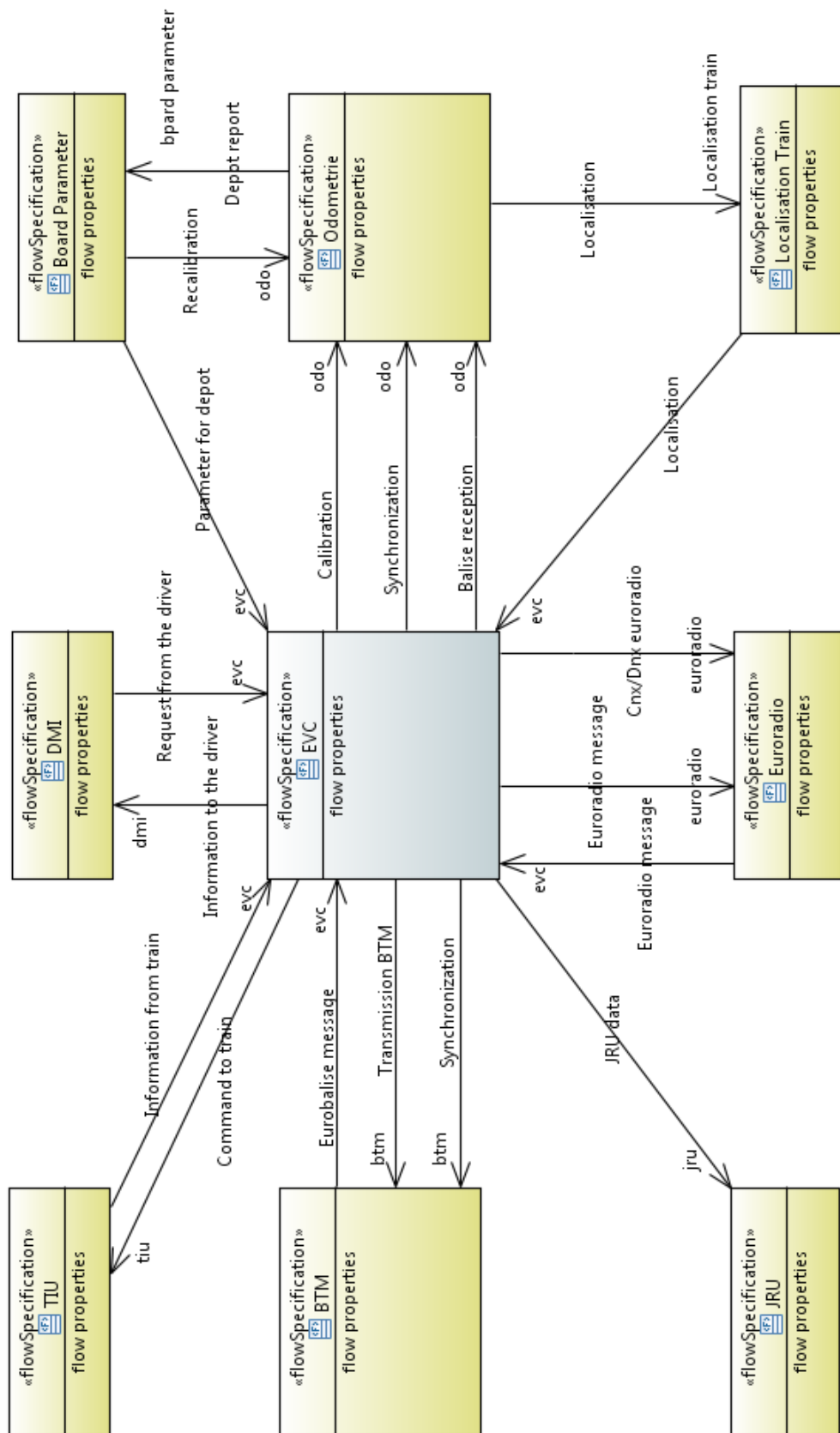
Figure 1. SRS architecture

4   Functional Breakdown
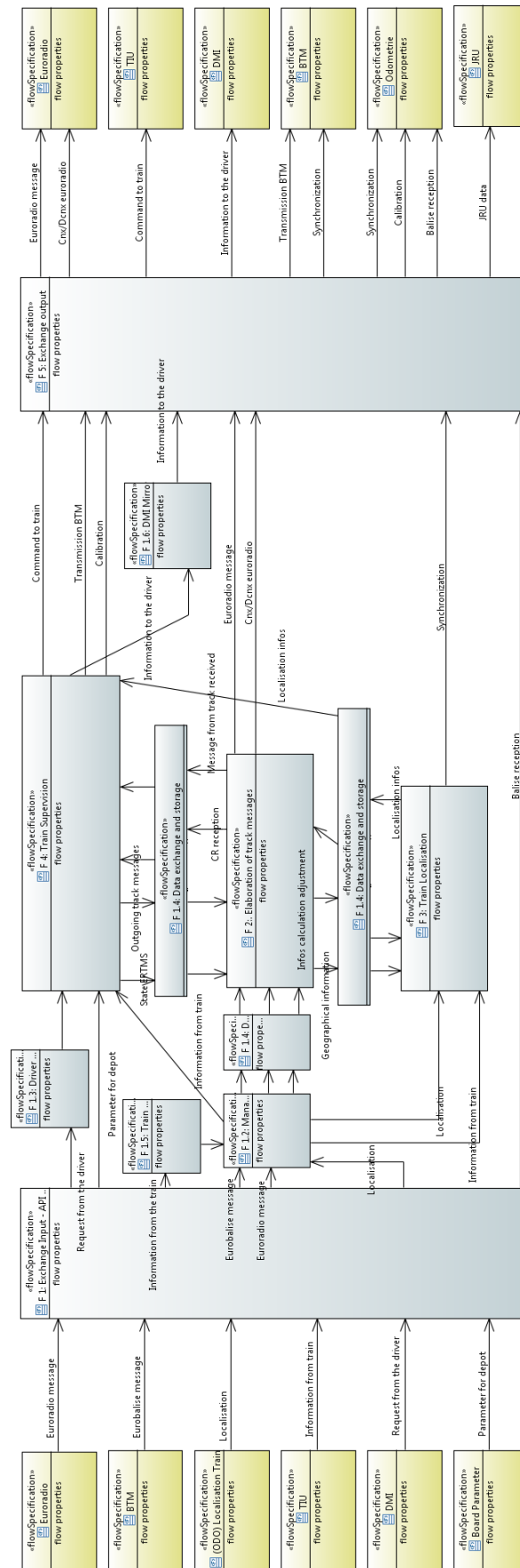
Figure 2. SRS Breakdown

5   Description of the SRS Functions

5.1   Reference abstract hardware architecture

5.1.1   Why a reference abstract hardware architecture?

For proper understanding of openETCS API and of constraints imposed on both sides of the API, we need to define a *reference abstract hardware architecture*. This hardware architecture is "abstract" is the sense that the actual vendor specific hardware architecture might be totally different of the abstract architecture described in this chapter. For example, several units might be grouped together on the same processor.

However the actual vendor specific architecture shall fulfil all the requirements and constraints of this reference abstract hardware architecture and shall not request additional constraints.

5.1.2   Definition of the reference abstract hardware architecture

The reference abstract hardware architecture is shown in figure 3.

The reference abstract hardware architecture is made of a bus on which are connected *units*:

- EVC (European Vital Computer);

- TIU (Train Interface Unit);

- ODO (Odometry);

- DMI (Driver Machine Interface);

- STM (Specific Transmission Module, up to 8 units);

- BTM (Balise Transmission Module);

- LTM (Loop Transmission Module);

- EURORADIO;

- JRU (Juridical Recording Unit);
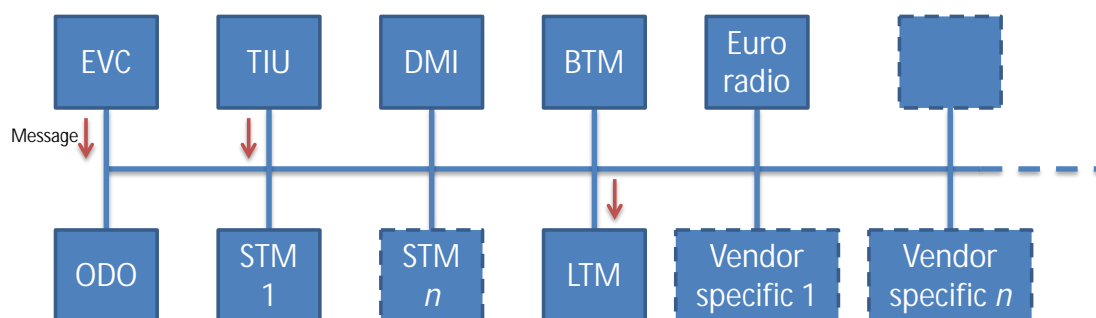
Figure 3. Reference abstract hardware architecture

- Zero or more Vendor specific unit.

A given instance of openETCS might not have all of above units. FIXME: Define a set of mandatory units? FIXME

Those units shall working concurrently. They shall exchange information with other units through asynchronous message passing.

5.1.3 Reference abstract software architecture

5.1.4 Overall architecture
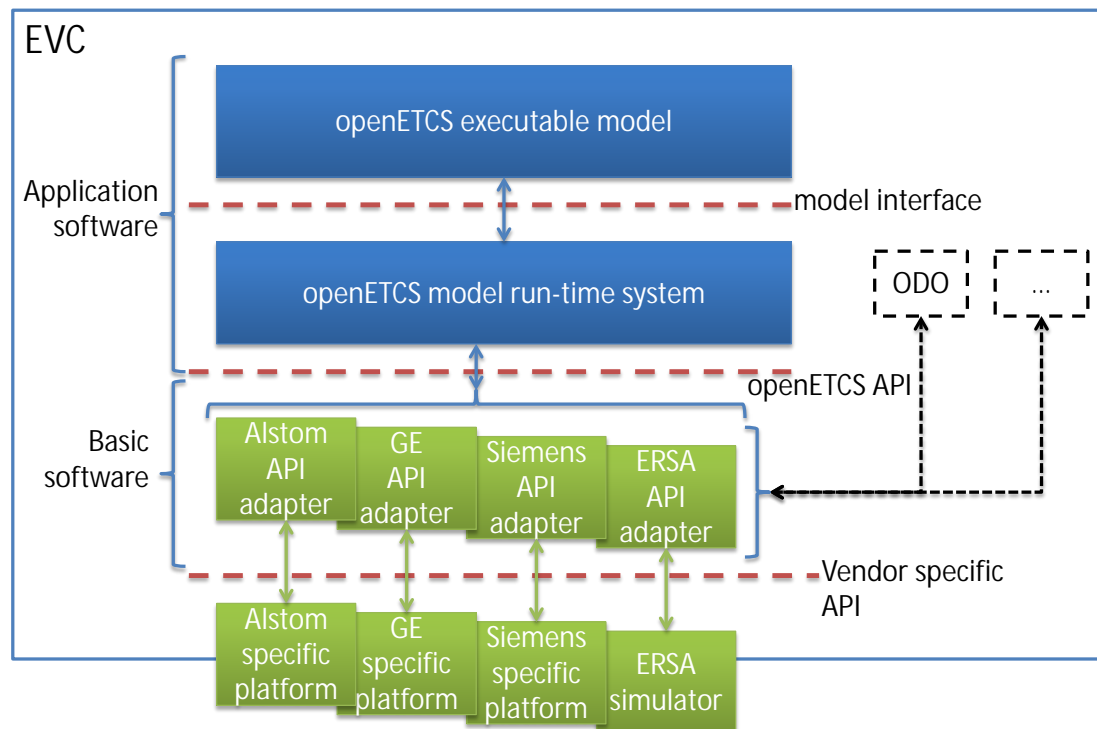
Figure 4. Reference abstract software architecture

The *reference abstract software architecture* is shown in figure 4. This architecture is made of following elements:

- *openETCS executable model* produced by the [4]. It shall contain the program implementing core ETCS functions;

- *openETCS model run-time system* shall help the execution of the openETCS executable model by providing additional functions like encode/decode messages, proper execution of the model through appropriate scheduling, re-order or prioritize messages, etc. This block shall be described in another openETCS document. FIXME: ref? FIXME

- *Vendor specific API adapter* shall make the link between the Vendor specific platform and the openETCS model run-time system. It can buffer message parts, encode/decode messages, route messages to other EVC components, etc.

- All above three elements shall be included in the EVC;

- *Vendor specific platform* shall be all other elements of the system, bus and other units, as shown in figure 3.

We have thus three interfaces:

- *model interface* is the interface between openETCS executable model and openETCS model run-time system. It shall be described in another openETCS document FIXME: ref?;

- *openETCS API* is the interface between openETCS model run-time system and Vendor specific API adapter. It is described in this document;

- *Vendor specific API* is the interface between Vendor specific API adapter and Vendor specific platform. This interface is not publicly described.

The two blocks openETCS executable model and openETCS model run-time system are making the *Application software* part. This Application software might be either openETCS reference software or vendor specific software.

The Vendor specific API adapter is making the *Basic software* part.

5.1.5 Information exchange between blocks

At this level of description, we do not explain how the various blocks of above architecture are calling themselves. We only assume they are exchanging *messages* in an asynchronous way. A message is a set of information corresponding to an event of a particular unit, e.g. a balise received from the BTM. The possible kind of messages are described in chapter **??**.

How the exchange of messages in implemented in actual software, e.g. function call, storage of data in a shared buffer, ..., is described in chapter **??**.

5.1.6 Architectural variations

Please note that the reference abstract hardware and software architectures do not forbid architectural variations. For example, the Odometry function could be put within the EVC (see ODO on figure 4) instead of a separate hardware unit (as it was shown on figure 3). Such Odometry function would be part of the Application software. But communication between this Odometry function within EVC and the openETCS model run-time system shall be done through the openETCS API and shall follow its conventions.

5.1.7 openETCS Model Runtime System

The openETCS model runtime system also provides:

- Input Functions From other Units
  In this entity messages from other connected units are received.

- Output Functions to other Units
  The entity writes messages to other connected units.

- Conversation Functions for Messages (Bitwalker)
  The conversion function are triggered by Input and Ouput Functions. The main task is to

convert input messages from an bit-packed format into logical ETCS messages (the ETCS language) and Output messages from Logical into a bit-packed format. The logical format of the messages is defined for all used types in the openETCS data dictonary.

Variable size elements in the Messages are converted to fixed length arrays with an used elements indicator.

Optional elements are indicated with an valid flag. The conversion routines are responsible for checking the data received is valid. If faults are detected the information is passed to the openETCS executable model for further reaction.

- Model Cycle
  The executable model is called in cycles. In the cycle

  – First the received input messages are decoded

  – The input data is passed to the executable model in a predefined order. **(Details for the interface to be defined)**.

  – Output is encoded according to the SRS and passed to the buffers to the units.

As another example, part of Vendor specific platform could be on EVC and thus the Vendor specific API would be within the EVC.

5.1.8   F1: openETCS API (Input)
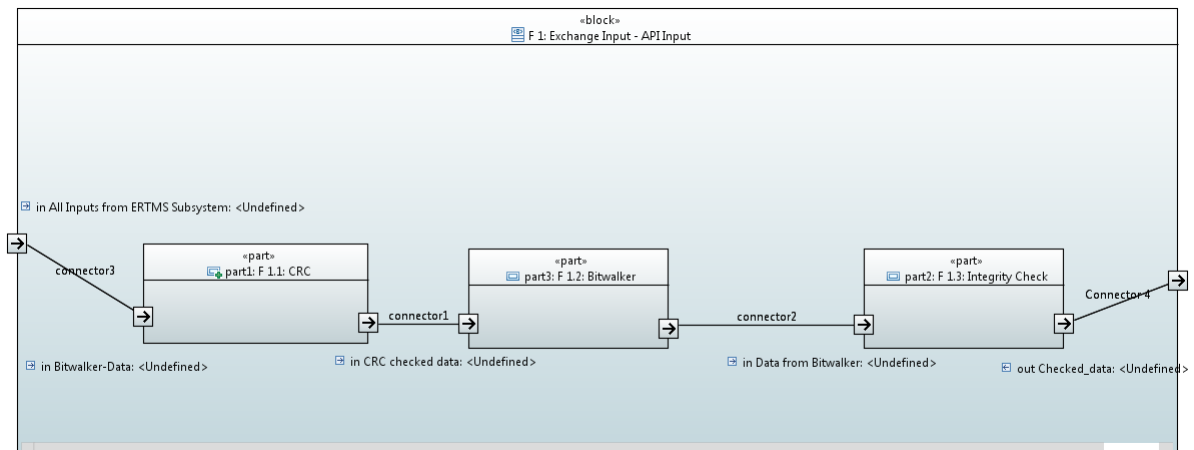
Figure 5. Exchange input - API input

Interfaces aligned with the Alstom API to be added
The exchange input needs to be seperated in Basic SW
and
Application SW
since the functions CRC and Bitwalker belong to a basic function. The integrity Check belongs
to Application function.

See Figure 5

Inputs:

- TIU

- DMI

- BTM

- EURORADIO

- ODOMETRY

- JRU

- Parameterization board ?

- Displacement measurement?

Outputs:

- The decoded and checked messages from other units are are output of F1. The information is
  processed by procedure call provided by the executable model.

- Details of the granualarity of those interfaces are to be clarified.

5.2   F1.2: Manage Input
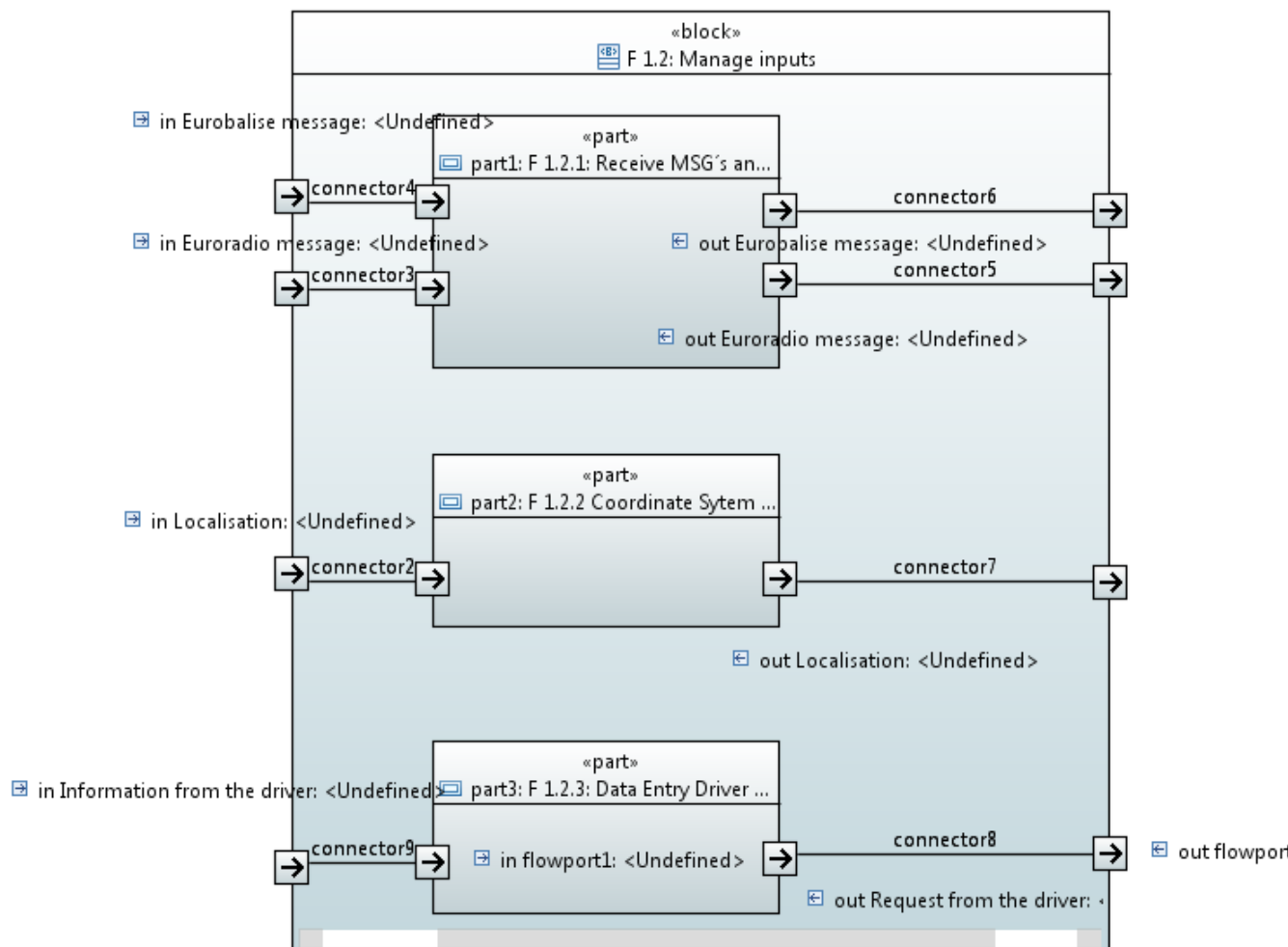
Figure 6. Manage input

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: ???

5.3 F1.4: Data structure
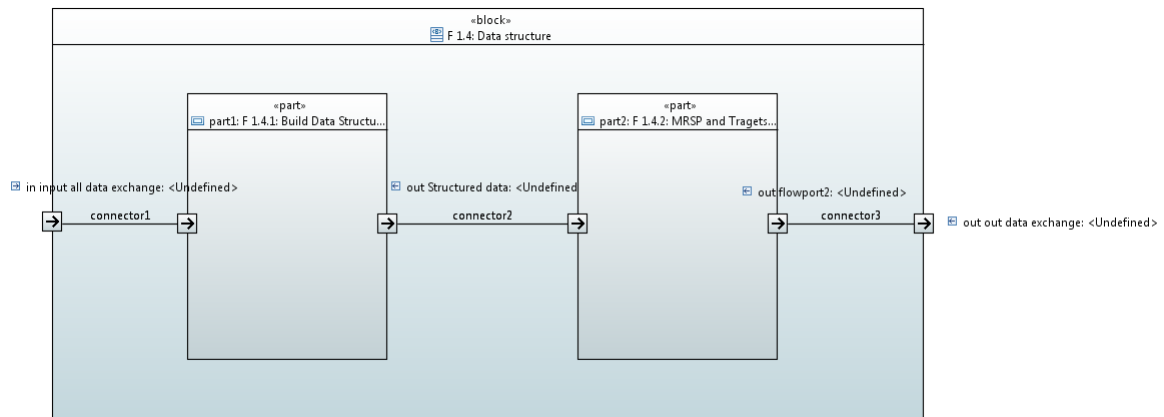
Figure 7. data structure

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: ???

5.4 F2 Elaboration track messages

See Figure 2 - Block F 2

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module provides the encoding and decoding of track messages that Management of the port and power off RBC, and the management of reference Beacons, which are based on the track messages.

5.4.1 F21: Receive Eurobalise Messages

SRS: § 3.4, § 3.6, § 3.16, § 3.17, § 4.8

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module provides the summary of telepowering information and the review of coherence the content of telepowering message. This module separates the eurobalise message as follows:

- The Balise telegram heads for checking the consistency of the different Telegrams, the management of the duplicated beacons and the acquisition of meaning.

- Then the information that the balise group, their reading direction and are Wegmessungsposition where the first balise group was read, in Baliseninformationen for monitoring the function "Manage the eurobalises" summarized (see the term Identifzierung and meaning in this function), which the Continuation of reading via the information Referenzbalise with a Balisenposition in Kernel-reference document approved.

- The ETCS ETCS version and packages:

The first review is the version ETCS ETCS track with the version Zugbord (Version 1.x Class1 =) to compare; these must be compatible (that means 1.y) to continue processing (a Balise version of the type is 0.y as incompatible considered, but they must not generate a negative reception report).

The second review is in compliance with the ETCS grammar in the resulting Packets with a packet 255, which terminates the beacon information of each of the group.

The reception of a packet 245 (standard package) in a eurobalise) includes the Rejection of all other packets received and generated a the same mistake as Grammatical errors ETCS, unless the OBU is in Level 2.

Only the packets 44, 65, 66 and 136 may be located several times in a same eurobalise message
For the 136 package:
• a single packet 136 per Balise telegram,
• the different packages 136 in a same message are identical.

The read packets are filtered as a function:

- from the direction of the transition to the balise group (the packet is the variable QDIR oriented), the bi-default state (level, level of announced and ERTMS / ETCS mode).

The tables of filtering per level and ERTMS / ETCS mode are in the description of the Function "received EUR radio messages" contain.

These packets are transmitted in four different groups:

- Geographical information (packet 79 for the "train locations")

- Information CNX / DCNX (packets 42, 131) for the "the CNX and DCNX Euro Radio Management "

- Information Balise (package 136: Referenzbalise for in-fill information) for the function "Manage eurobalises"

- Track-mail received (other packages) for the "monitor train".

Any error of Referenzbalise, the version or the grammar ETCS is in CR (radio channel) Receiving reported.

5.4.2   F22: Receive Euroradio Messages

SRS § 3.4, § 3.6, § 3.16, § 4.8

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module provides the summary of the Euro radio information and the Verify the consistency of the contents of the Euro radio message. The first guaranteed by this module check is to monitor the radio link. This only applies to a normal communication channel (see § 8.3.1 interface gSM-R).

The review is broken down as follows:

- Basic principle: the RBC is providing its messages with a time stamp based on the Time marking the bi-standard the equipment (if the timestamp of the messages on position is undefined, the message is accepted, but only during the initialization of the Communication session EVC - RBC).

Verification of the sequence:

- the ETCS OBU train equipment rejects any message that is "older" than the last received message.

- the ETCS OBU train equipment is up to any message that is "younger" than the last received message.

5.4.3 F23: Manage Eurobalise Messages

SRS § 3.4, § 3.6, § 3.16

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module manages the Balise reference document of the ETCS On-board Unit, is said to know that the information supplied by the track following terms have:

- either the balise which provides the information,

- or delivered in the radio message Referenzbalise,

- or delivered in the infill balise message Referenzbalise (package 136).

A balise group contains 1-8 eurobalises. The balise group is referenced by an identification NIDLRBG (NIDC: identification + NIDBG region: identification beacon). Each eurobalise has an internal number from 1 to 8, which describes the position of the beacon relaltive in the group.

A consisting of only one beacon balise group called "simple beacon", as any other balise group managed; However, it produces features that described in Function module F23 and in the function module F25.

5.4.4 F24: Manage Cnx and Dncx Euroradio

SRS § 3.5, § 3.15.1, § 5.15

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module ensures the production ending a Euroradio- Communication. For the ETCS OBU equipment it is possible to Euro Radio communication session initiate:

- for a "start of mission": train data

- after one obtained from the track command: info CNX / DCNX. The command, an RBC to contact, the identity of the RBC and its telephone number (Packets 42 or 131).

5.4.5   F25: Send Euroradio Messages

§ 3.4, § 3.16

Inputs:
"will be complete"

Outputs:
"will be complete"

Description:

This function module ensures the completeness (location and time stamp) and the Transmission of radio messages euro on the basis of information "radio session message", "Outgoing track message", "message acknowledgment" (for a radio message 136) and "CR Radio channel reception "(for a radio message 136 along with a package 4).

A radio message is sent only if between the ETCS OBU equipment and the RBC opened a euro radio session.

The radio messages, with the exception of the message confirmation (radio message 146) and the Session messages (radio message 154, 155, 156, 159) contain a "position report" package (O or 1 packet).

The structure of this "position report" package based on the following information:

- Info for locating the data LRBG, position, velocity, position error, moving direction, by default state for the data concerning the level and the ERTMS / ETCS mode.

The package 1 "special position report" is used for one or two LRBG whose Crossing direction is not known ((LRBG type simple balise group).

The package 0 "position report" is used when the direction of the LRBG is known (group not simple beacons, or group simple beacons, whose Balisenrichtung over by the track was positioned a link or package 135) (see function F23)).

See the functional breakdown of the function in F2 in the next paragraph (Figure 3)

5.4.6   F2 breakdown
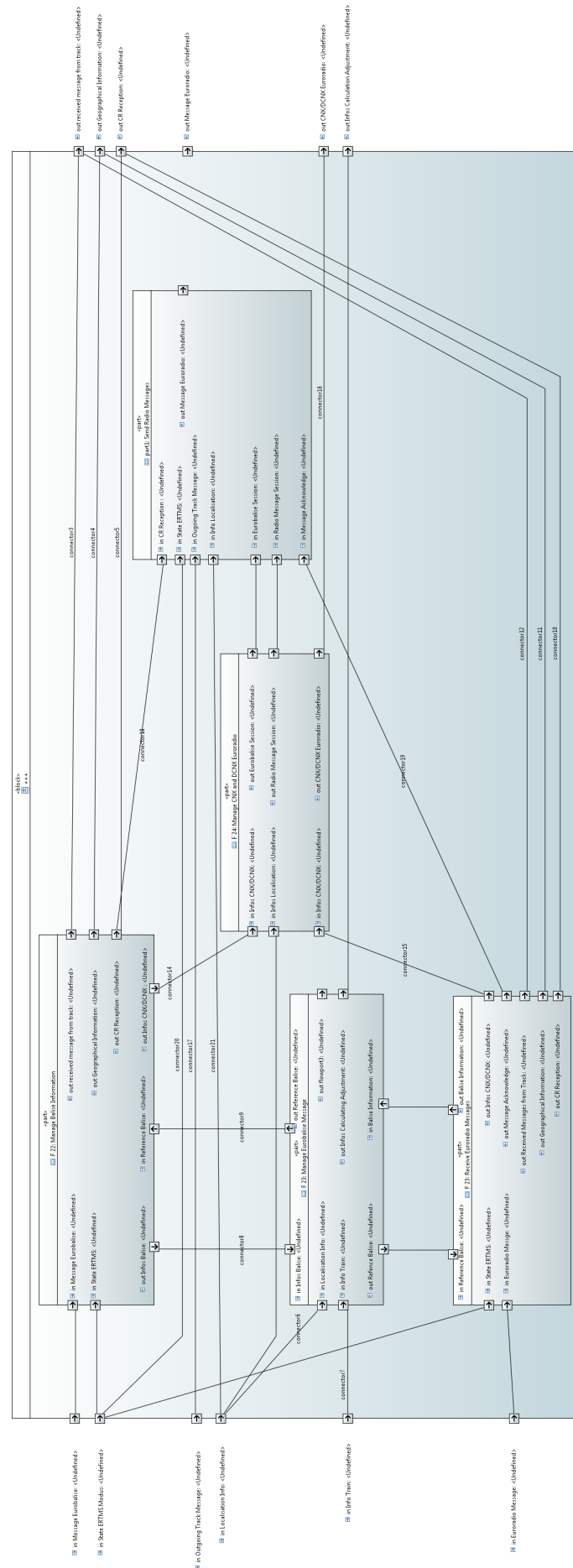
Figure 8. Elaboration track messages breakdown

5.5 F3 Train loction

See Figure 2 - Block F 3

SRS § 3.6.4, § 3.6.5, 3.6.6

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module provides the location of the train in a kernel-reference document (Locating information), based on the following reference documents:

- path measurement calculation reference documents the function "distance measurement": train location,

- Balise the "track messages draw": Information computational comparison.

5.6 F4 Train Supervising in ERTMS Modus

See Figure 2 - Block F 4

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module ensures the monitoring of the train by working out:

- The level and mode of control,

- Monitoring the speed, position and the movement of the train,

- The driver interfaces,

The data, which is based on the monitoring.

5.6.1   F41: Manage Level ERTMS/ETCS
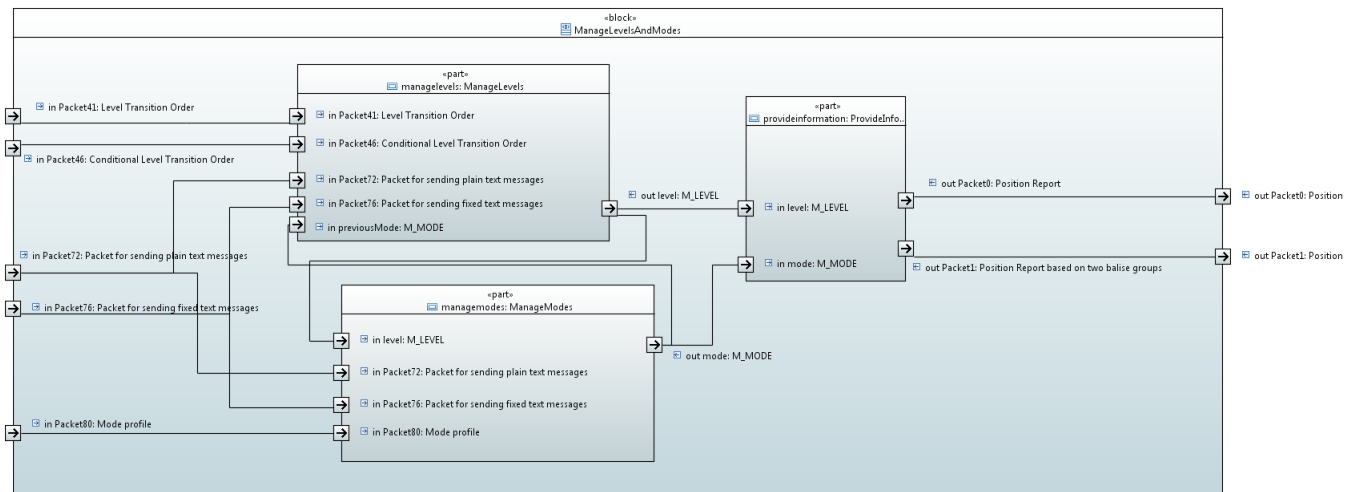
Figure 9. Elaboration track messages breakdown

SRS§ 5.1, § 3.6.5

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function is only applicable for the Eurobalise or Euro radio messages announced transitions. The aspect of "use" of the transitions is in Chapter 5 (5.10] described.

This function module manages the transitions of the level ERTMS / ETCS and the associated driver acknowledgments and releases associated with these transitions related actions. The function depends on the current level (managed by this function) and the following inputs:

- "Get track message": notice the transition, the immediate transition command;

- «Info locating": position of the train;

- "Request by the driver": acknowledgment of the transition;

- «Info train": information exchange of the active driver's cab.

5.6.2 F42: Manage Modus ERTMS/ETCS

SRS§ 4, § 3.6.5, § 3.15.4, § 5.5, § 5.6, § 5.7, § 5.9, § 5.11, § 5.13

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: description This function module manages the transitions of the ERTMS / ETCS mode and the associated acknowledgments by the driver, and triggers associated with these transitions associated actions.

The function depends on the following inputs:

- "Get track message": Adopted train data, SH mode approved indication signal (VMAIN = 0), Information «danger for SH" Information "stop if in SR";

- Info locating": train position and indication information;

- "Mapper": available allocation to in SR mode, FS, OS or RV drive requirement of "train trip" (triggering the train-emergency);

-"Request driver»: acknowledgment of the driver proposed modes SH and NL, acknowledgment at the end of TR mode, Preventing the TR mode;

- Info train": TIU Inputs (off, active cab, signal leading / led Cab, field monitoring system, condition of equipment;

- "ETCS Mode": current mode and current level;

- "CR reception": a reported on a balise error or loss of the radio link and related response;

- "Train protection»: in SH mode Balise prohibited, banned in SR mode Balise and the status of theETCS OBU equipment forming equipment (TIU, DMI, BTM, Euro radio, board-parameterization, position measurement): any error in the equipment (not shown "material flow").

5.6.3 F43: Train Speed supervision

SRS§ 3.7, § 3.8, § 3.10, § 3.11, § 3.12, § 3.13, § 5.7, § 5.8, § 5.9

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: description This function module monitors the train speed in comparison to a Track assignment / a Movementauthority.

You must track the assignment using the information obtained track work out, ie the Completeness of the assignment using the information obtained from the track information concerning MA, check profiles and parameters.

A complete assignment is based on the following information:

- The radio messages 2 (Approval SR), 3 (MA), 33 (MA reference it) and the packages 12 (MA Level 1), 15 (MA Level 2), 80 (mode profiles), 138 (Reversing area information), 139 (Reversing supervision information) give information on the approval of Movement on the track assignment;

- The radio messages 9 (requirement of shortening of MA), 15 (conditional urgent Stopping), 18 (refusal of urgent arrest) and 16 packets (repositioning), 70 (Route Suitability) give the "extensions" of the movement permit information concerning the track assignment, that is, that they complement the original track assignment, without replacing them;

- The radio message 16 (unconditional urgent stop) changes the target point, so the train to travel (emergency) passes (equivalent to the train which his Has exceeded track assignment);

- The packages 21 (Gradientprofil), 27 (velocity profile), 51 (axle load), 65 (temporary restriction) and 66 (refusal temporary restriction) enter the profile information on the track assignment;

- The package 57 (MA requirements) are the parameters concerning the track assignment; for the technical operation modes, the assignment only to provide information to technical mode are limited;

- The assignment can also on the position of the train (stop) at the end of a mission, a missed beacon or a radio or loss in accordance with the current technical Be reduced mode (SRS §4.10).

- The package 71 (adhesion factor) called the coefficients of the delay tables Change train (train data service braking and emergency braking): 70% or 100%. It should be mentioned that the change of these coefficients by the driver on the ETCS onboard unit is not approved. To ensure interoperability must the application but the "track adhesion" function on the driver manage (see the national value of QNVDRIVERADHES).

- For mode FS and OS mapping is complete when the gradient profiles and SSP to End of the MA are known; Otherwise MA is denied.

5.6.4 F44: Train Movement supervision

SRS§ 3.14

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module supervises train movements, ie it ensures protection against the involuntary movements of the train:

- Protection against the elopement of the train,

- Protection against rolling back of the train,

- and monitoring the maintenance of the train.

This module triggers an unwanted movement of the train from the service brake.

Protection against the elopement of the train
The protection against the elopement of the train to avoid the train in one direction moves that do not match the current position of the field monitoring of the active driver's cab matches. This protection is applicable only in SH mode, FS, SR, OS, UN, PT and RV. If the removal of unwanted motion exceeds the national value DNVROLL, The brake is activated. When the train goes back again to stop, the brake is by consent of the Train driver released.

Protection against rolling back of the train

The protection against rolling back of the train to avoid the train in the his Direction opposite direction, i.e., his current assignment, moved. This protection is applicable only in FS mode, SR, OS, PT and RV. If the removal of unwanted motion the national value DNVPOTRP for the Mode PT and PT and et DNVROLL exceeds for the other modes, the brake is activated. When the train goes back again to stop, the brake is by consent of the Train driver released.

Monitoring the stop of the train

The monitoring of the stop of the train to avoid the train moves, if he in SB mode is stationary. If the removal of unwanted motion exceeds the national value DNVROLL, The brake is activated. When the train goes back again to stop, the brake is by consent of the Train driver released.

5.6.5   F45: Train position supervision

SRS § 3.6.5, 4.4.8, § 4.4.11

Inputs:

"will be complete"

Outputs:

"will be complete"

Description: This function module monitors the trainposition, namely:

- protection against overshooting the unapproved mode in SH or SR beacons,

- and the report "report" the pull position (on the conditions of the train position) to the RBC.

Protection of unauthorized beacons

The protection against overshooting of non-approved mode in SH or SR beacons based on the list of approved packages ETCS balise the 49 and 63rd This means in each affected mode (SH or SR):

- if a list was supplied, it contains the list of approved beacons: the Driving over a balise not included in this list causes the transition to mode TR the ETCS Onboard equipment;

- if an empty list is supplied, this means that no beacon is approved: the Driving over any Balise causes the transition to mode TR ETCS On-board equipment;

- if no list is supplied, each beacon can be run over.

position report
At each stop the train in FS mode, OS and SR is systematically a position report on a radio message 136 produced.

Depending on the requirements of the package 58 ETCS another position report is generated ("Position report parameters"), i.e. .:

- a temporary cyclical position report,

- a moderate distance zykischer position report,

- an immediate position report,

- a position report at each crossing a balise group,

- a position report at specific locations, ie if the max. Head of the train (max safe front end) or the min. End of the train (min safe rear end) a specific Has crossed position.

- Otherwise, without a request by the track, the position report is limited to each Driving over a balise group.

5.6.6   F46: Data storage

SRS (§ 4.3), § 3.18

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module guarantees the storage of various data in the "Train data"; these are then used for the entire function modules of the kernel available (it doesnt show the flow, they are distributed to the total functions).

This train data are broken down as follows:

- Parameterization train maintenance; These are the characteristics of the train and the track (Train data, fixed data and otherwise national standard data),

- Additional data; these are the data of the mission of the train.

The additional data from the driver at the end of the process "start of mission» adopted (see function "with the driver dialoguing) and § 6.3 procedure "Start Of Mission). This validates the additional data and allows the validation of the beginning a mission.

See § 4.3 DATA for details of the data.
The train data are sent over a track auszusendende message to the RBC (Radio message 129, Parcel 11). The data obtained from the track national values (Package 3) will be stored and considered when the track region (NIDC the Balise of locating information) in one of the received national values specified region.

5.6.7   F47: Dialog with the driver

SRS § 5.4, § 3.12.3

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module ensures the dialogue with the driver at a start a mission and for all on-board advertisements (messages, symbols, dynamic data) and the track Text messages.

The method "start of mission" consists in starting or restarting (end of mission the train) for a new mission.

See Chapter 5 for the method "start of mission."

- The track text message is a resulting free track text message (packet 72) with display and Display end conditions that are dependent:
• Level of ERTMS / ETCS,
• mode of ERTMS / ETCS,
• from a distance,
• from one time interval,
• of an acknowledgment by the driver.
The condition can be single or multiple (combination of multiple conditions). In the case of the

expectation of an acknowledgment by the driver at the end of other Combined display conditions, and if these are met, the managed Eurocab- Board equipment service braking (lack of acknowledgement).

A resulting encoded text message (packet 76) generates an error message because the text codes are not defined.

The track text messages and the driver displays are in messages summarized, which are intended for the DMI: dynamic message (coming from the KV Information) symbols and train-text messages.

The train-text messages are managed by internal text codes that the DMI an ad in chosen by the driver language enable (choice at the beginning of the "start of mission "or during the mission).

5.6.8   F48: Manage brake controll

SRS § 3.14.1

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module ensures the release of the brake control (output "Brake control").

The allowances are as follows:

- The technical TR and SF mode (input "State ETCS") produce a emergency braking,

- The lack of acknowledgment of the level, the mode or text message (input "Lack of acknowledgment") causes a braking operation,

- The protection of the train by the movement monitoring (input "train protection") causes a Service braking

- FS a reaction to a Bali mustard Ehlers or a wireless loss (input "CR Receiving "causes a braking operation,

- The direction indicated by the speed monitoring overspeed FS or FU (Input "overspeed") causes an operating or emergency braking.

Each service brake application (except for overspeed) is an emergency braking replaced when the service brake:

- Does not exist (train data),

- Not after the brake reaction time (train data) after reading the service brake (input "Info train") is not working.

The solution braking is managed by the requesting functionalities (at a Change the mode or the rectification of the fault).

5.6.9 F49: Manage Train Control

SRS § 3.12.1, FFFIS TIU

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: This function module ensures the development of the using of the train control:

- to Control: service brake or emergency brake

- the resulting track message "tracks conditions" (Package 68)

- the resulting track message "track condition big metal masses" (Package 67)

- the resulting track message "linking", which allows a calibration range to define (Package 5).

The drafting of "track conditions" and the "calibration range" is national in § 4.4 And functions described in § 4.5. The "track conditions" (Package 68) activate the corresponding train control systems and give them the driver to. The package 68 provides the start and end of the "tracks conditions ".

The "big metal masses track condition" (Package 67) puts the transmission in BTM state OFF. This causes the disruption of the broadcast antenna of the ETCS onboard equipment. The package 67 provides the beginning and end of the track condition "big metal masses." At the end of this condition in the transmission mode is "not modulated", so the the Default is to read the eurobalises.

There is a manual calibration of the measurement system exists, it is, however, directly from the Linear Position Sensing subsystem through the acquisition of the Wegmessungsdaten Bord-parametrierungsmoduls managed. However, the bi-standard onboard equipment is capable of automatic calibration ranges of To capture Bordwegmessungssystems: A calibration range is announced between two Eurobalisegruppen Calibration area having the following properties:
• Distance between the Euro balise 'groups greater than 1000m,
• Linking error between the Euro balise 'groups smaller than 1 m,
• Gradientprofil between the eurobalises, the closest to a zero-gradient are,
• Travel between the beacons with a least possible number of curves.
Therefore, these properties are the responsibility of implementing this track Areas, but the first two conditions are those of the ETCS onboard equipment allow to detect a calibration area.

5.6.10   F4*: Building Most Restrictive Speed Profile and List of Targets i.r.t. the Train Front End

The Statric Speed Profile (SSP) and speed restrictions in the "location based data" FIXME: missing inputs to MRSP need to be added like Mode related speed restrictions have to be combined to a most restrictive speed profile (MRSP) and list of targets for braking monitoring given as a distance from the LRBG. This is the task of the current function. FIXME

5.6.10.1   Building of the MRSP

The current function will combine the SSP and the list of speed restrictions (coming from TRS's, most restrictive braking distances, axle load profiles and level crossings FIXME: missing inputs to MRSP need to be added like Mode related speed restrictions) into a most restrictive speed profile (MRSP) and a list of targets subsequently used for braking curve monitoring. Inputs: SSP and list of speed restrictions (including a mark which SR is not yet included in the MRSP and "list of targets". FIXME: missing inputs to MRSP need to be added like Mode related speed restrictions Outputs: MRSP and "list of targets for braking curve monitoring", given as distances (one value without tolerance) from the front end of the train (taking into account the train length for speed increases if necessary). FIXME   FIXME

The train specific static speed profile and the speed restrictions are used for:

- braking curve monitoring; according to the most restrictive targets

- ceiling speed monitoring: according to the most restrictive speed at a certain location

- release speed monitoring: according to the most restrictive speed at a certain location

- indication monitoring: according to the most restrictive speed at a certain location

- displaying the planning area to the driver; most restrictive speed profile in the area ahead.

For displaying to the driver (planning area) and guarding the actual ceiling- and release-speed, the maximum speed at any place shall be calculated (MRSP). For braking curve monitoring the actual most restrictive target is sufficient. The latter will not change till it's passed or revoked. Therefore a list of targets can be derived which has to guarded one by one, by braking curve monitoring. In the modes FS (full supervision), OS (on sight) an LS (limited supervision) the MRSP can be derived from the SSP and speed restriction. In shunting and staff responsible (SH and SR) the mode profile can be given in the message commanding the transition to the state.

In all cases the applicable most restrictive speed has to be respected. The restrictions can come from:

- Static Speed Profile (SSP)

- Axle load Speed Profile (ASP)

- Temporary Speed Restrictions (TSR)

- Maximum Train Speed

- Signalling related speed restriction (only level 1)

- Mode related Speed Restriction.

- STM Max speed

- STM System speed

- Level Crossing speed restriction (LX SR)

- Override function related Speed Restriction

- Speed restriction to ensure a given permitted braking distance

5.6.10.2   Usage of the MRSP

The most restrictive speed profile is giving the lowest speed of the resulting SSP, and the list of speed restrictions. When constructing the MRSP the distance to be taken into account that for speed decreases, is the minimum safe distance from the train front end (therefore the maximum "correction distance") and for speed increases the maximum safe distance from the train front end (therefore the minimum "correction distance"). The MRSP is to be used for supervising the actual speed and for displaying at the DMI (planning area).

5.6.10.3   Updating the MRSP

- The MRSP has to be calculated every time a new SSP is received and when a new Last Relevant Balise Group (LRBG) is detected.

- New speed restrictions can be included in the existing MRSP, without recalculating the complete MRSP. Therefore it's necessary to mark in the "list of speed restrictions" which speed restrictions have not yet been taken into account in the MRSP.

- After every change of the train position (every cycle) the MRSP shall be updated according to the maximum and minimum distance driven (the maximum is relevant for speed decreases, the minimum is relevant for speed increases). This could lead to changing the order of points where the speed increases and points where the speed decreases.

- the conditions for application of the single speed restriction has to be stored with the speed restriction and to be recalculated at least if one of the conditions change

An example is the case with two Temporary Speed Restrictions (TSRs) at a short distance:
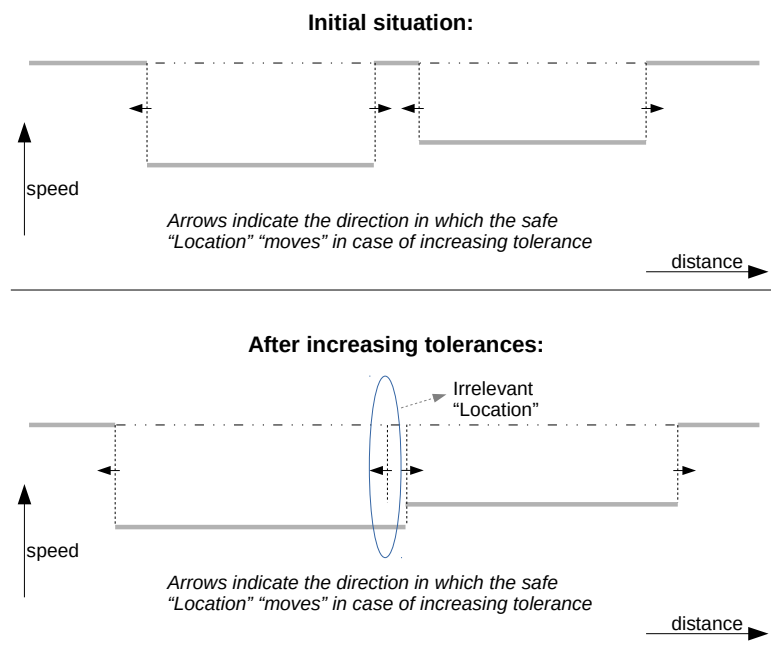
Figure 10. Example for overlapping TSRs due to increasing tolerances.

The example of figure 10 shows the effect of overlapping TSRs due to increasing tolerances. One point (distance from the train front end) shall be removed from the MRSP. FIXME: further explanation on removing points from MRSP could be good here   FIXME

For braking curve monitoring (BCM) only speed decreases are relevant, and only those which do not lead to a higher "braking curve speed" than further targets with a lower target speed (see figure 11).

5.6.10.4 Building the Most Restrictive Targets for BCM

For braking curve monitoring the most restrictive target shall be known at every moment. In general a speed decrease in the MRSP is a potential target. But not every speed decrease is a relevant target because they can be above the braking curve to another target (see figure xxx).

Figure 11 shows examples of the selection of the "list of most restrictive targets" which have to be handled by "Braking Curve Monitoring" one by one. In the above example the "green" target will not be "most restrictive" and will thus not be relevant in the "list of most restrictive braking curve targets" unless the red target being more restrictive is revoked.
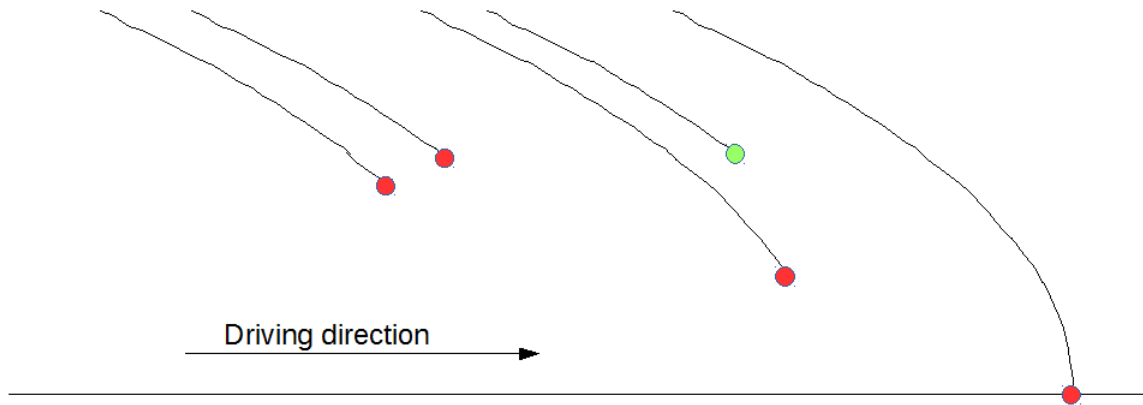
Figure 11. Example for "list of most restrictive targets" handled by braking curve monitoring.

When analysing the MRSP from the furthest location towards the train front end, all relevant speed decreases (most restrictive targets for braking curve monitoring) are found. These locations will be stored (including the speed level) and have to be guarded one by one for braking curve monitoring.

5.6.10.5   Updating the Targets for BCM

The list of most restrictive targets shall be recalculated every time the MRSP has changed.

The data structure of the MRSP and list of target is different from the structure for general location based data because all locations are given as a distance from the LRBG only. So the data structure will not be based on the type "Location".

5.6.11   F4 breakdown
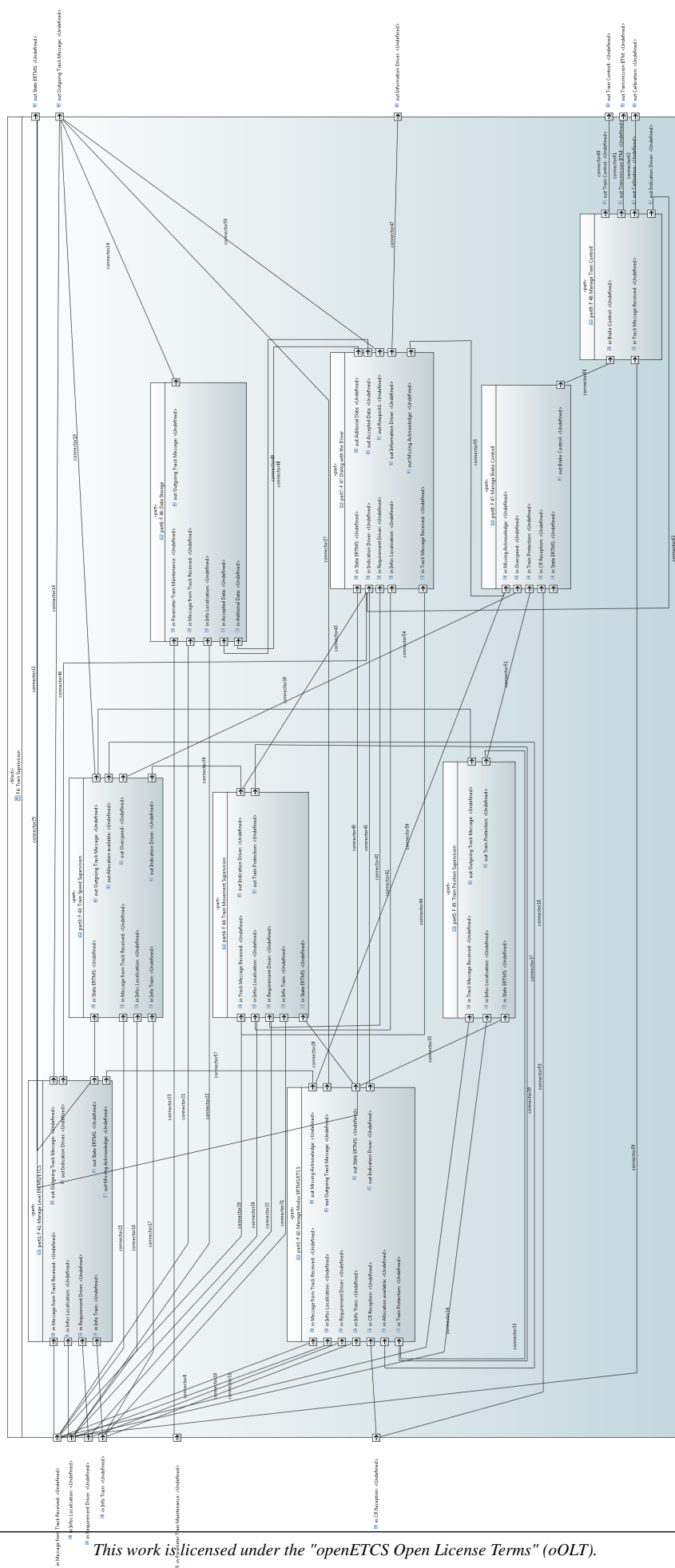
Figure 12. Train Supervising in ERTMS Modus

5.7   F5 Exchange output

See Figure 2 - Block F 5

Inputs:
"will be complete"

Outputs:
"will be complete"

Description: description This function module manages the exchange at the output of the ETCS
On board equipment with the following modules:

- TIU,
- DMI,
- BTM,
- EURORADIO,
- SAM,
- Displacement measurement.

It complements their functions at the level of exchange protocols and the "safety layer" for
the entire module with the exception of Wegmessungsmoduls (that manages its own Exchange
protocols). Note: The manner in which these functions are implemented, is free; the details of
which are specified in the software specification documents.

6 Data Flow Description

6.1 List of Data FlowsID

Version	Section	Modification / Description	Author
1.0	Missing Acknowledge	Kernel xx/Kernel xx	This information indicates the absence of the Acknowledgment of the driver to a confirmatory text message on the Level Transition, the transition mode or a track Train-text message
2.0	Acknowledge	Kernel xx/Kernel xx	Radio message 146 transmission Confirmation - Confirmation message depending on the variable § 8.6.7
3.0	Action Driver	Kernel xx/Kernel xx	Election / confirmation / detection Driver via DMI
4.0	Display Driver	Kernel xx/Kernel xx	For the driver specific DMI display
5.0	Reference Balise	Kernel xx/Kernel xx	information; in the Flow "information Balise" delivered Balise confirmed as a reference beacon.
6.0	Allocation available	Kernel xx/Kernel xx	Information message that an allocation for a given technical mode is available.
7.0	Calibration	Kernel xx/Kernel xx	Specifying a calibration range of the path measurement.
8.0	Cnx/Dcnx Euroradio	Kernel xx/Kernel xx	Command on the connection and disconnection of a or of a RBC, the identity and Phone RBC contains.

Number	Flow	Source/Sink	Description
9.0	Brake Controll	Kernel xx/Kernel xx	Control of the service and the emergency braking.
10.0	Train Controll (brake)	Kernel xx/Kernel xx	Functional output.
11.0	CR Reception (Radio Channel)	Kernel xx/Kernel xx	CR (radio channel) on receipt of a balise or Radio message: Checking version ETCS, Grammar ETCS, testing reference balise (Position), monitoring radio link
12.0	Accepted Data	Kernel xx/Kernel xx	Information valid confirmation of the train data and Consequently, at the very beginning of the mission.
13.0	Additional Data	Kernel xx/Kernel xx	Data relating to the mission of the train: Identity of the driver, selected level, identity and mandatory. of the RBC, if Level 2, no. Mission
14.0	JRU Data	Kernel xx/Kernel xx	Recorded legal data.
15.0	Input Train	Kernel xx/Kernel xx	From the train coming TOR digital inputs and Digital outputs
16.0	State ERTMS	Kernel xx/Kernel	Current Level and operating mode ERTMS, announced ERTMS Level and Current Level and operating mode ERTMS, announced ERTMS Level
17.0	Indication Driver	Kernel xx/Kernel xx	each specific for the driver Specification. This includes the dynamic information speed monitoring the different messages and symbols, the Exchange during the "start of mission"

Number	Flow	Source/Sink	Description
18.0	Info Train	Kernel xx/Kernel xx	From Train comming functional information TIU
19.0	Message Eurobalise	Kernel xx/Kernel xx	Information, which the ID Balise group (country ID + ID Balise), their reading direction and their Contains path measurement for review (see Flow "reference balise")
20.0	Information cnx/dcnx	Kernel xx/Kernel xx	Information identifying the RBC ID (Identification of land + the identification of the RBC), RBC his phone number and the authorization
21.0	Information Driver	Kernel xx/Kernel xx	Specific for the driver Function DMI information. They contain the different specifications for the Driver, which are generated from the different Function modules.
22.0	Geographical information	Kernel xx/Kernel xx	Absolute current position of the train, which the add the flow "location information"
23.0	Information Localisation	Kernel xx/Kernel xx	Locating the train: position and speed of the train in the different reference documents, including the absolute geographical Position.
24.0	Message Eurobalise	Kernel xx/Kernel xx	Flow, summarizing the message / messages eurobalise. BTM to F1: the entire telegrams F1 to F2: Message (the safety layer and the the safety layer and the transfer of Telegrams in a message by F1 F1 guaranteed)

Number	Flow	Source/Sink	Description
25.0	Infos Calculation Adjustment	Kernel xx/Kernel xx	Information, which contains the ID Balise group (country ID + ID Balise), their path measurement and her for the computational comparison of the on-board Contains reference document expected position
26.0	Localisation Train (ODO)	Kernel xx/Kernel xx	path measurement in comparison to a Balise (+ safety margin), Speed (+ Safety distance), direction of travel and capture of stopping
27.0	Euroradio Message (train to track)	Kernel xx/Kernel xx	Transmitted Euro radio message (the safety layer is guaranteed by F5)
28.0	Euroradio Message (track to train)	Kernel xx/Kernel xx	Received Euro radio message (the safety layer is guaranteed by F1)
29.0	Radio Message Session	Kernel xx/Kernel xx	Message / parcel Euro Radio for creating and Termination of the radio connection
30.0	Outgoing Track Message	Kernel xx/Kernel xx	Before Inform creation and timestamp (except the message flow Radio link, message confirmation and CR Reception) sent to the track to function message

Number	Flow	Source/Sink	Description
31.0	Message from Track Received	Kernel xx/Kernel xx	Via radio or beacon message received function (except the Riverside current Information, cnx / dcnx information, Balise information)
32.0	Parameter Train maintenance	Kernel xx/Kernel xx	Fixed data, train data, domestic values
33.0	Train Protection	Kernel xx/Kernel xx	Information, protection against rolling Twitching, monitoring of stopping, the Transition to an unauthorized Balise in SH or SR reports
34.0	Balise Reception	Kernel xx/Kernel xx	Information of receipt of a reference Balise NPIG = 0 from the associated type for the computational comparison of the displacement measurement
35.0	Requirement Driver	Kernel xx/Kernel xx	By driver outbound (see flow "Action Driver") or from DMI DMI produced function information
36.0	Euroradio Session	Kernel xx/Kernel xx	State of the Euroradio connection
37.0	Signal Eurobalise	Kernel xx/Kernel xx	Airgap Eurbalise -> BTM
38.0	Signal Euroradio	Kernel xx/Kernel xx	Airgab RBC ->Euroradio Board
39.0	Output Train	Kernel xx/Kernel xx	Train digital outuputs
40.0	Overspeed	Kernel xx/Kernel xx	Information message overspeed of the train relative to the speed curves
41.0	Synchronization	Kernel xx/Kernel xx	Synchronization (xw, V, T) between the kernel, the BTM and the path measurement

Number	Flow	Source/Sink	Description
42.0	Transmission BTM	Kernel xx/Kernel xx	Control BTM - antenna

Number	Flow	Source/Sink	Description

7 Functional and data structure architecture ETCS on-board

Within the following chapter Description of Functional and data structure architecture ETCS on-board the datastrucutre is explained and detailed on location based data. FIXME: To be   FIXME
considered and merged to the overall architecture view!!

7.1 Data Structures

7.1.1 Introduction

The data flows and data structures used in the application are the interface between the functional blocks. A clear definition of the data will allow sharing the workload between different groups without causing too much communication overhead. The data structures are part of the **Data Dictionary**.

Data can be distinguished in data storage which is remembered as a system state: "data stored on-board" see SUBSET-026-A.3.4, and data temporarily stored.

Data structures shall be designed for: FIXME

FIXME: API Synchronization Inputs (to be synchronised with the API specification):

- structures for messages

- data from other inputs: odometer, DMI, TIU, STM's
                                                                                                                              FIXME

FIXME: API Synchronization Outputs (to be synchronised with the API specification):

- structures for output messages

- data to other outputs: DMI, TIU/BIU, STM's, JRU (not including technical communication, like distributing the system time)

7.1.2 Data structures to store received messages

Messages are received either by Euroradio or Eurobalise (Euroloop is not yet taken into account). The Euroradio will provide a message in one block of data, from a Eurobalise Telegrams (the part of a message sent by one balise) are received, which have to be put together into a message.

Eurobalise Messages contain a header to give the following information: FIXME: Needs syn-   FIXME
chronization according to SUBSET-026-8.4.2.1

- valid version of ERTMS/ETCS system

- `NID_BG` identifier of the BG

- `NID_C` identifier of the country

- reference to which information is given (identifier of the LRBG), in case of a radio message.

- information for constructing the message out of the Telegrams and for checking message consistency (balise numbering, message counter, duplicate information etc.)

Euroradio Messages contain a header to give the following information: FIXME: Needs synchronization according to SUBSET-026-8.4.4.6.1

- `NID_MESSAGE` - identifying the message number

- `L_MESSAGE` - Message length including everything from field 1 to padding inclusive

- `T_TRAIN` - Time Stamp from RBC

- `M_ACK` - Inidicating of the message must be acknowledged

- `NID_LRBG` - identifying the

- further variables as required by `NID_MESSAGE`

The message header is used to check the consistency, to check if the message is valid for the current direction, define the reference for location based data and construct the message out of Telegrams.

Further messages contain "packets" as defined in SUBSET-026 chapter 7.

Radio-messages are received as a bit stream in a minimised data format. To allow efficient processing the data has to stored using standard data-types (fixed point) in structures for individual packets. The chain of components between receiving the message from the antenna to storing packets using standard types is: FIXME: needs a design decision - consistent with high level architecture? radio - EVC basic software – bit walker – filtering. Functions are assigned as follows to the different modules and functional blocks:

- CRC checking shall be done in the radio. The result of the CRC check shall be made available for the application software. FIXME: Exact definition of CRC error needs to be added here (because a CRC error is to handled by the application as a consistency error, i.e. "linking reaction").

- Translating the bit stream into variables of a standard (fixed point) data-type and store the data in a memory accessible by the application software. The way the data stream has to be put into variables and stored depends on the version of the track-side ETCS installation. (see `M_VERSION` and SUBSET-026-6) Indexes ("pointers") shall be set to packets contained in the message, i.e. an array containing the indexes of the first packet of a certain type. The array shall have one element per possible packet. Each packet shall have an extra field to refer to the next packet of the same type in the message (only possible for a few packet types). This shall be done by the "bit walker".

- Detecting the version of the track-side ETCS system is to be handled in the application.

- Check on illegal values in the packets. (if detected, set a flag for "consistency error").

- Couple the location reference to the different packets (those can be different for different packets in case infill information is included in the message, i.e. all packets after packet 136 contain infill information (have to be reference to the LRBG using linking information).

- Translate the identity of the location reference (a BG) from `NID_BG` to the index and array where the concerning information is stored.

- Check if the packets have to be taken into account or have to be stored in a transition buffer according to the current mode and level (filtering) and mark the packets if they shall be used.

To facilitate the above described functions, a data structure is necessary capable of storing at least 2 complete messages plus extra variables to store the result of CRC checking and the check on illegal values. Further for each packet variables are needed to store the reference position for the information in the packet (`NID_BG` and index plus array where the BG information is stored). In addition an array is needed to store "pointers" to the start of each packet in the message.
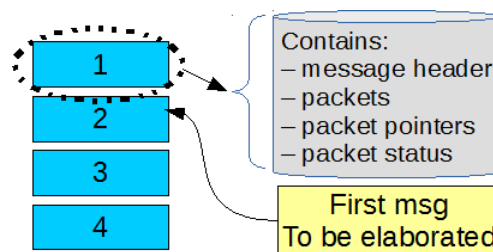
Figure 13. Data structure to store not yet elaborated messages.

Data structure to store not yet elaborated messages as found in figure 14 are further described in 17 and 17. A special value for the "pointer" is defined for the case no radio messages are available. Further pointers to the individual packets are required for efficient elaboration later on.

The different functions can work on the same data structure, i.e. the structure is filled once and is than used up to the point where the packets are elaborated.

Eurobalise (BTM) -messages are received as a bit stream in a minimised data format per balise telegram. To allow efficient processing the data has to be stored using standard data-types (fixed point) in structures for individual packets. The chain of components between receiving the message from the antenna to storing packets using standard types is: BTM - EVC basic software – bit walker – filtering.   FIXME

FIXME: needs a design decision - chain of components consistent with high level architecture? Functions are assigned as follows to the different modules and functional blocks: FIXME: FIXME CRC checking to be better described

- CRC checking shall be done in the BTM per Telegram. The result of the CRC check shall be made available for the application software (because a CRC error is to handled by the application as a consistency error, i.e. "linking reaction").

- Translating the bit stream into variables of a standard (fixed point) data-type and store the data in a memory accessible by the application software . If there is already a Telegram from the same balise stored and depending on the message counter of the telegram header, the new

data shall overwrite that Telegram. The way the data has to be stored depends on the version of the track-side ETCS installation.

- "Pointers" shall be set to packets contained in the message, i.e. an array containing the indexes of the first packet of a certain type. The array shall have one element per possible packet. Each packet shall have an extra field to refer to the next packet of the same type in the message (only possible for a few packet types). This shall be done by the "bit walker".

- Detect the version of the track-side ETCS system, to be done in the application.

- Check on illegal values in the packets. (if detected, set a flag for "consistency error").

- Couple the location reference to the different packets (those can be different for different packets in case infill information is included in the message, i.e. all packets after packet 136 contain infill information.

- Compose a message out of the Telegrams if a balise of another BG is found, if all balises of a BG are found, if the maximum distance (12m) since the last detected balise has been passed.

- Gather the BG information needed to build the "reference (coordinate) system".

- Translate the identity of the location reference (a BG) from `NID_BG` to the index and array where the concerning information is stored, see figure 15.

- Check if the packets have to be taken into account or have to be stored in a transition buffer according to the current mode and level (filtering) and mark the packets if they shall be used.

FIXME: what is the rationale behind only storing 15 telegrams?

To facilitate the above described functions, a data structure is necessary capable of storing at least 15 telegrams plus extra variables to store the result of CRC checking and the check on illegal values. Indexes ("pointers") indicating where the information of a completed message can be found shall be set. BG information (location, the identity of the BG, etc.) has to be stored in a separate structure for further evaluation. Further for each packet variables are needed to store the reference position for the information in the packet (`NID_BG` and index plus array where the BG information is stored). In addition an array is needed to store "pointers" to the start of each packet in the message.

A completed message can during the consistency check and the check on linking, be kept in the same structure.

In the example shown in figure 14:

- A message is just made available consisting of 5 telegrams stored at position 16, 1,..,4. If no message is available the pointers **"last telegram of delivered BG message"** and **"First telegram of delivered BG message"** message have a special value. After elaboration of the message and retrieving the packets (done in the same cycle) the pointers will be reset to this special value

- Since receiving the last Telegram of the delivered message, 3 more (5, 6, 7) Telegrams were detected.

- **"Oldest received Telegram"**: the oldest Telegram stored by the "bit walker" which has not been used in a completed message. If no Telegram is waiting for the completion of a BG-message the "pointer" shall have a special value.
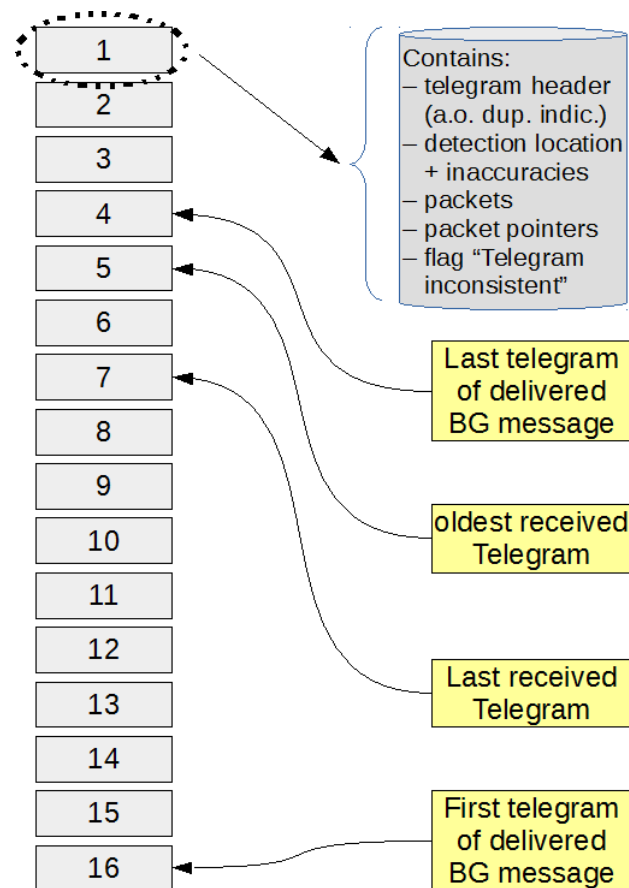
Figure 14. "Telegram structure": Structure in which Telegrams shall be stored by the "bit walker".

- **"Last received Telegram"**: the last Telegram stored by the "bit walker". If no Telegram is waiting for the completion of a BG-message the "pointer" shall have a special value.

For packages which are not yet to be taken into account a data structure has to be designed to store the packages for later use inside the **"transition buffer"**. The stored packages shall be elaborated if the mode and level for which they are stored is reached (generic condition or shall those be stored with the packet). The minimum number of packages which could have to be remembered, i.e. the minimum storage capacity to be reserved, is to be determined. FIXME: determine minimum packages to be remembered within the transition buffer   FIXME

7.1.3 Data structures to store input data from other sources

Other input data:

- TIU: Isolation, brake status (commanded, pressure, inhibited,...) FIXME: to be completed   FIXME

- DMI: data-entry, acknowledgements, etc. FIXME: to be completed   FIXME

- STM's: out of scope for the current project.

TIU The data coming from the TIU is wrapped up in SUBSET-034 and needs amendments for the specific use of the OBU inside a locomotive. A data structure for storing all possible information shall be designed. FIXME: complete design of TIU related data   FIXME

DMI The data coming from the DMI, consists of inputs required based on information sent to the DMI in the context of a procedure.

7.1.4 Data structures for "data stored on-board"

This section is independent to the fact that what happens if the OBU is switched off FIXME: data stored on board needs to be completed

1. Location based data (a.o. announced BG's, SSP, speed restrictions, location based level transitions, text messages,..)

2. Actual MRSP and list of targets as a distance from the trains front end. (taking into account the train length for speed increases if and where necessary).

3. Train Position

4. Reference (Coordinate) system

5. announced BG's

6. Direct orders

7. Intermediate analysis results (linking error, number of missed BG's, intervention curve passed,......)

8. BG lists for shunting and staff responsible

9. Train data

10. Data describing national rules (national and default values)

11. track data: version, level priority FIXME: to differentiate the difference between track data and location based data

12. Driver ID

13. Procedure status information (technical procedures and driver interactions)

14. Parameters tuning procedures (MA request parameters, Train position report parameters,....)

15. System mode and level

16. System status (including time to last self test)

Below the types of data listed above are further described.

7.1.4.1 Location based data

In general "Location based data" is data giving a specific type of information related to a location on the track. Requirements concerning storing:

- The reserved space shall be sufficient to store the maximum number of elements.

- The elements shall be stored in the order they are passed on the track.

- The location shall be available as a distance from the LRBG.

The minimum number of elements for which space shall be available is given in chapter 7, "Management of location based data". For all types of location based data the same basis structure can be used, except for the location based information which can only contain one element (level transition order (LTO), reversing area, platform conditions, current limitation). The structure is described in chapter 7, "Management of location based data":
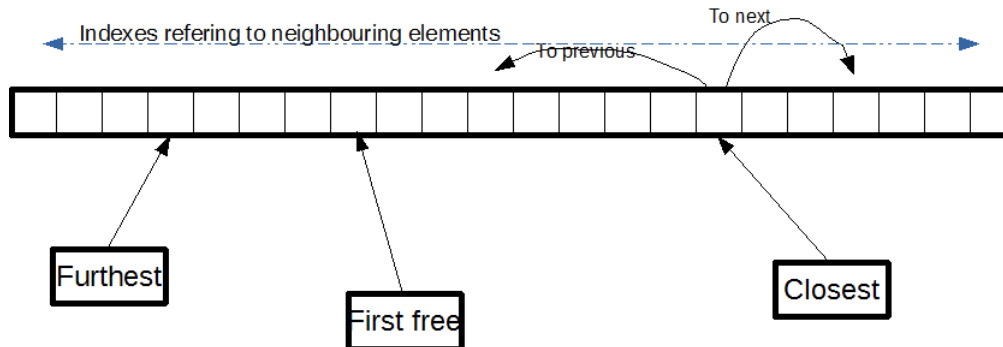
Figure 15. Data structure of "location based data"
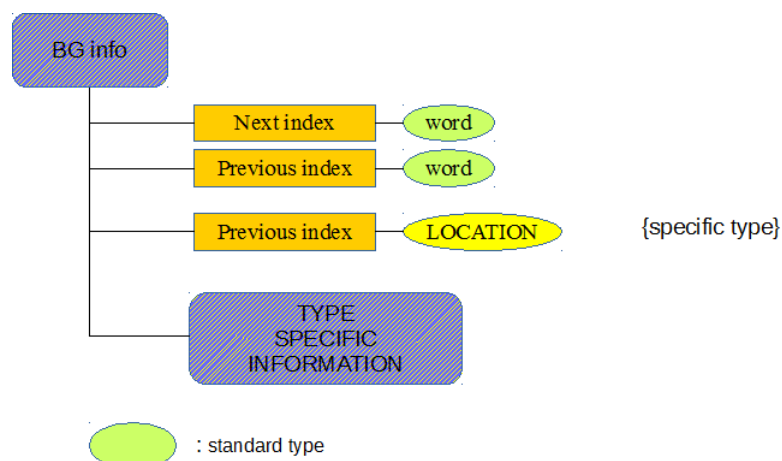
Figure 16. Single Element of data structure of "location based data" including backwards linking.

Actual MRSP and list of targets As described in chapter 7 the MRSP and list of most restrictive targets is not stored as location based data, but as a set of distances with related speed level from the train front end. The rationale behind this is the ability to recalculate the MRSP when a restriction is revised. At figure 17 the structure of one element of the MRSP or list of targets is shown graphically.
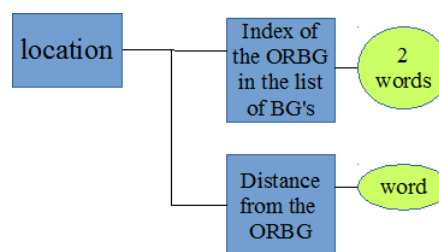
Figure 17. MRSP element as dependant from distance to target and further conditions   FIXME

FIXME: Mode is also a condition, better list all conditions or write "Further conditions" FIXME: FIXME explain BCM

The elements shall be stored in the order in which they are passed. At each cycle the structure shall be update according to the distance driven. As for speed increases the maximum distance

and for speed decreases the minimum distance is valid, the order of the elements in the MRSP may change during driving from the LRBG.

7.1.4.2 Train Position

The train position shall be available for the train's safe front end, the safe rear end in case train integrity is used and the active antenna as a minimum, estimated and maximum distance from the nominal position of the LRBG:

-	Minimum	Estimated	Maximum
front	X	X	X
rear	O	O	O
antenna	X	X	X

X - mandatory

O - optional in case train integrity is used

Table 1. Availability of safe distances

To be able to retrieve this data the following data is stored:

- The measured distance (raw data from the odometer) stored as a passing location (of the active antenna) for the LRBG (to be stored in the structure of the "list of passed BG's").

- The passing direction of the LRBG (nominal or reverse): for reporting to the RBC. (to be stored in the structure of the "list of passed BG's").

- The antenna (A or B side in case a second antenna in installed) which reads the LRBG (to be stored in the structure of the "list of passed BG's")

- The distance between the antenna A and front end of cab A (cab A)

- The inaccuracy in the distance between the antenna A and cab A (installation inaccuracy only)

- The distance between the antenna A and cab B (installation inaccuracy only)

- The inaccuracy in the distance between the antenna A and cab B

- The distance between the antenna B and cab A

- The inaccuracy in the distance between the antenna B and cab A (installation inaccuracy only)

- The distance between the antenna B and cab B (installation inaccuracy only)

- The inaccuracy in the distance between the antenna B and cab B

Together with the inputs "actual train position" and the "train orientation", all combinations given in the table above can be generated.

7.1.4.3 Reference (Coordinate) system

FIXME: for the correction distance a design decision is needed and additional justification since this is not part of the SRS - relates to the complete subsubsection "Reference (Coordinate) System) The data stored shall allow to calculate the distance between the train and a track-side element whose location is given as a distance from the nominal position of any BG. Therefore data shall be available concerning the relation between the distance from the nominal position of the LRBG and the distance from the nominal position of any other BG (further called ORBG), see figure 17. This distance is further called the "correction distance" and is BG specific. Therefore is shall be stored in the data structure for detected and announced BG's. This "correction distance" can be derived from linking (exact) or from odometer data (with a tolerance). Further the possible distance between the position where a BG is detected and it's nominal position is to be taken into account.
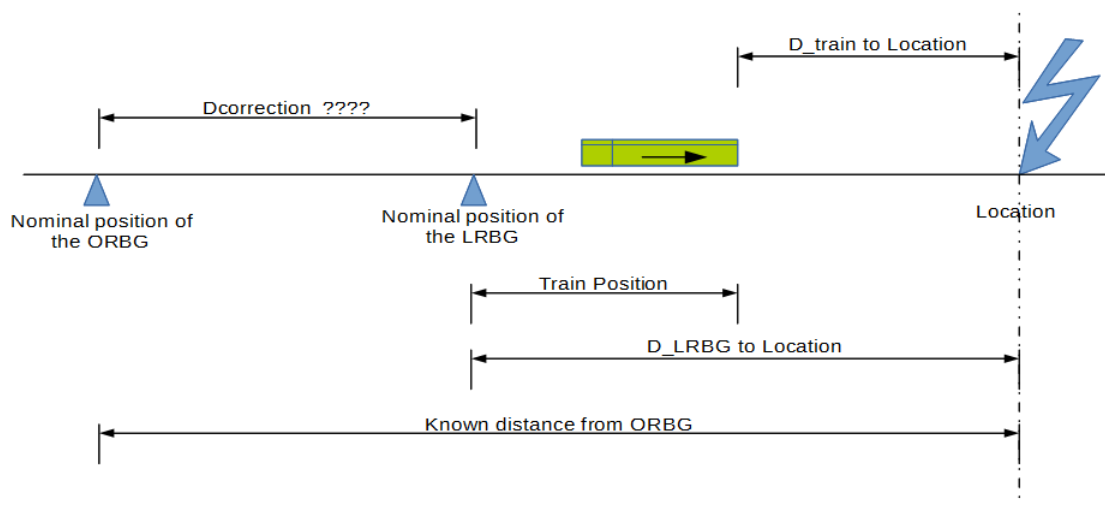
Figure 18. Calculation of the distance from the train to an a track-side element ("Location")   FIXME

FIXME: for this principle a justification and design decision is needed

Therefore for each BG already passed, in reference to which the location of any track-side element is given a minimum and maximum correction distance shall be stored in order to be able to calculate the minimum and maximum distance between the train front end and the track-side element:

Distance between train and location = Distance from ORBG – Train Position (I.r.t. LRBG) - the stored correction distance (all including tolerances).

As the Location of a track-side element can also be given in reference to a BG in advance of the train (infill information), the same correction distance shall be available for announced BG's. This is only with one difference. For announced BG the exact linking distance is available (if not the infill information can and must not be taken into account). The correction distance exact (no minimum and maximum), and equal to the linking distance.

This leads to the need for two data structures, one storing the correction distances for already passed BG's, and one storing the announced (linking) distance from the nominal position of the LRBG to the announced BG's. The latter structure also contains the data needed for checking the linking consistency, i.e. the expectation window. A graphical representation of the data structure is given below: FIXME
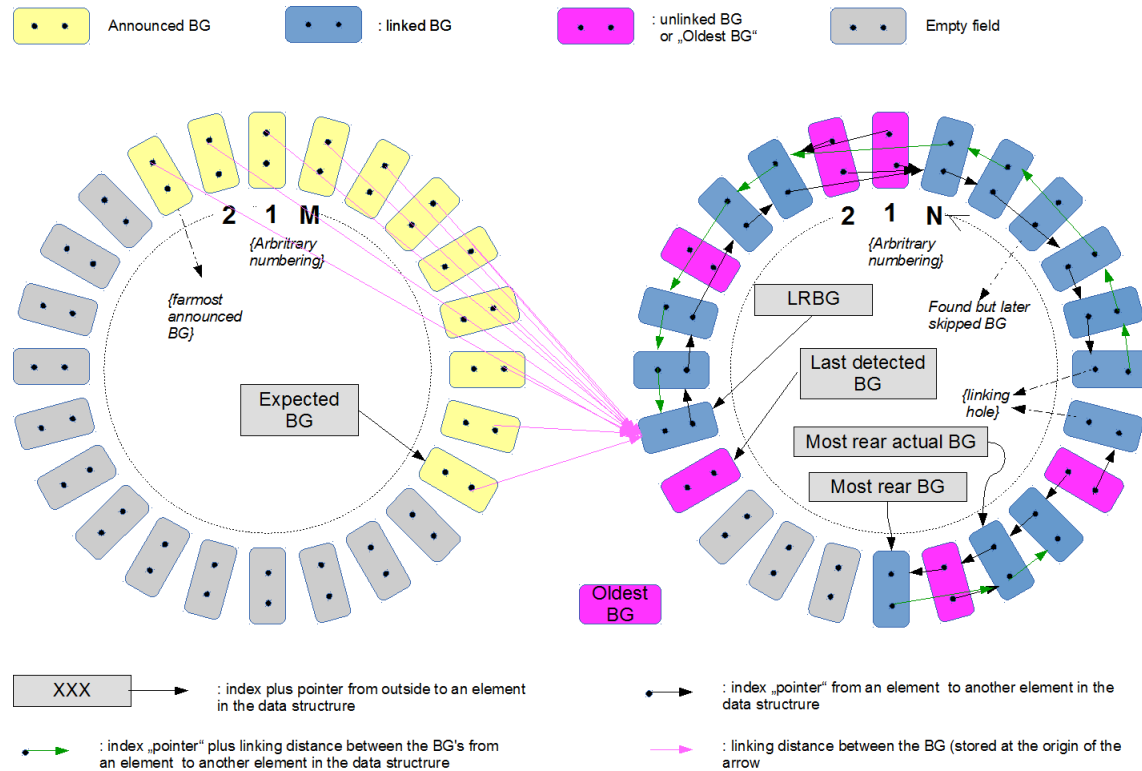
Figure 19. Data structures for announced, detected linked and detected unlinked BG's

FIXME: for this principle a justification and design decision is needed

1. LRBG: "last relevant BG" = the last detected linked BG

2. Last detected BG: can be different from the LRBG if the last detected BG is marked as "unlinked"

3. Most rear actual BG: most rear BG to which one or more "Locations" refer as being their ORBG (original reference BG). All BG's more in rear may be deleted from the list.

4. Most rear BG: the most rear BG whose information is still stored in the "list of detected BG's"

5. "Oldest BG": BG to which all "Locations" with an ORBG in rear of the "most rear BG" refer. This is used if the "List of detected BG's" is full. Between the "oldest BG" and the "most rear BG" no linking distance is available. Therefore the accuracy for the concerned locations will decrease. linking hole: Number of linked BG's for which no linking information is available (see figure xx). skipped BG: Linked BG which was detected when no BG's were announced, but which is not included in linking information received after wards (see figure xx) "pointer": index of the element in the data structure where is referred to.

7.1.4.4 Direct orders

Direct orders can be given for several actions, e.g. command the service or emergency brake, "stop if in staff responsible" etc. Direct orders which have immediate effect at the first evaluation like direct level and mode transitions (have immediate effect in the functional block "filtering" because they shall be taken into account at filtering further information in the message) can be rejected. Direct orders shall be stored in the internal data structure to allow the correct execution

of the order by the concerning functional block. For each possible order a field shall be reserved, i.e. an array with a field for every possible order shall be build.

This data structure is needed to avoid getting event driven software, as the orders are stored and read by the a functional block which is commanded cyclically. Direct orders to be stored (as a flag or variable):

- command service brake

- "stop in staff responsible"

- SR authorisation

- Exit of trip mode

- Inhibition of revoking TRS's

FIXME

FIXME: List needs to be completed by all orders including a special value "now" according to SUBSET-026-7   FIXME

FIXME: To what extent shall mode changes following from input data be taken into account before filtering?

Intermediate analysis results (linking error, nb of missed BG's, intervention curve passed,......) Just like direct orders, some analysis results shall lead to a certain specified action. For example, having a consistency error shall lead to the linking reaction (given from track-side or the default value which is command service brake). Just like direct orders those intermediate results are stored as an order for other functional blocks, in stead of immediately (event driven) calling the specific function. Intermediate results to be stored (as a flag or variable):

- From BCM: EB intervention, SB intervention, acoustical warning, Vsoll, Vwarning,...

- From linking consistency checking: type of linking reaction, number of missed BG's

- From message consistency checking: type of linking reaction …... FIXME: needs to be completed   FIXME

7.1.4.5   BG lists for shunting and staff responsible

Together with the order to go to shunting or to staff responsible a list of BG's which may be passed can be given. Both sets need to be storable in parallel. The data structure shall consist of an array storing the identities of the BG's which may be passed. As the passing order is not always predictable it doesn't make sense to reorder them. BG's may thus be stored in the same order as they were received.

7.1.4.6   Data describing the train

To allow the usage of the system on all different train types, data describing the specific train shall be stored. The source of the data can be the TIU, the data entry unit or default values. Data to be stored includes:

- Train category(-ies)

- Train length

- Traction / brake parameters

- Maximum train speed

- Loading gauge

- Axle load category

- Traction system(s) accepted by the engine

- Train fitted with airtight system

- List of National Systems available on-board

- Number of axles

Each parameter shall have a flag to indicate if this parameter can be edited by the driver. Each parameter shall have a flag to indicate if this parameter can be displayed on demand by the driver.

7.1.4.7 Data describing national ruling

Maximum speeds, braking curve margins, roll away distance, timing for certain conditions, procedural information etc. may differ from country to country. Therefore a list of national values is defined according to SUBSET-026-A3.2. These national values shall be stored on board, to be available for all functional blocks.

For those national values, defaults can be defined for the case no national values have been received from track-side. A similar structure shall be provided to remember those default values.

7.1.4.8 Track Data

The version of the track-side ETCS system and the priority of levels supported from track-side shall be available for the functional blocks. Data structures to store this information in the same format as it is received from track-side shall be provided.

7.1.4.9 Driver Data

Driver data is used when displaying data to the driver (language) and for juridical recording (driver ID). A data structure shall be defined to store all possibly available driver data, see also chapter 7.1.4.6

7.1.4.10 Procedure status information

In the SRS (especially SUBSET-026 chapter 5) a number of procedures between track-side, train and driver are described, e.g.

- data-entry

- reporting train position

- SoM, EoM

- override

- Train data management....

- text message acknowledgements

- establishing and ending a connection to the RBC

- RBC-RBC hand over

- transition to shunting

- transition to staff responsible

- …...   FIXME

- FIXME: complete procedure

For each of these procedures the status shall be remembered, in some cases including the time since specific events. Therefore for each procedure a global data structure shall be designed to remember the status of each procedure.

7.1.4.11 Parameters for procedures

As the requirements concerning procedures may differ, parameters are defined to tune the procedures. (timing requirements, etc.). Data structures to store this information in the same format as it is received from track-side shall be provided.

7.1.4.12 System mode and level

The system mode and level shall be available to all functional blocks at any time. Therefore variables to store the current state and level shall be defined.

7.1.4.13 System status

Data structures are necessary to store the current status of the subsystems (Radio, BTM, LTM, STM, JRU, odometer, TIU, brakes).

7.1.4.14 Data structures to store messages to be sent to the RBC

The information needed which as to be sent to the RBC is provided by "train supervision", "mode/level management", diverse procedures etc. To sent them to the RBC the information has to be build in a specific format (specified in ss026, chapter 7 and 8). This structure has to be synchronised with the API specification

7.1.4.15 Data structures to sent to subsystems (DMI, TIU/BIU, JRU, STM)

Information to be sent to the DMI and TIU/BIU shall be build in data structures as described in the API specification. The JRU and STM are out of scope for the current iteration.

7.2 Management of location based data

A lot of information given from ETCS track side to on-board is "Location based", i.e. the information is valid for a certain location given as a distance from a reference BG. The current function will manage the elaboration of packets containing "Location based data" into the (already available) sorted internal data structures which shall enable efficient further elaboration of the data (e.g. braking curve monitoring, constituting the planning area, etc.).

Description: Elaboration of the packets containing "Location based data" into the (already available) sorted internal data structures.

Inputs:

- Packets containing "Location based data" (see below)

Affected data in the "data stored on-board":

- Internal data structures storing "internal data structures"

- Status of received packets (change to "elaborated", thus may be forgotten)

Outputs: -

"Location based" data is categorized for the purpose of defining data structures:

- **movement authority (MA) list of sections**, message 37, packet 12 (level 1), message 3, packet 15 (level 2), 16 (repositioning, i.e. extending the current section), message 33 (??), packet 70 (route suitability), message 9 (request to shorten MA), minimum number of elements to be stored: 6

- **list of announced BG's linking information:** packet 5 minimum number of elements to be stored: 30

- **adhesion factor:** packet 71; only one element

- the **"gradient profile"** (in: pkt 21) minimum number of elements to be stored: 50

- **Speed profiles:** packet 27 (SSP) the worst case can be determined at reception Packet 13 minimum number of elements to be stored: 50

- **Speed restrictions** and non-continuous speed profiles: packet 51 (axle load profile), packet 52 (permitted braking distance), packets 65/66 (TSR), packet 88 (level crossing, incl. stop condition to be reset at standstill). minimum number of elements to be stored: TSR: 30, axle load: 30, permitted braking distance: 5 , level crossing: 10. Reversing area's: packets 138, 139 minimum number of elements to be stored: 1 Mode dependent speeds: message 2 and packet 80 minimum number of elements to be stored: 6

- **Level transitions:** packet 41 minimum number of elements to be stored: (see ss26, 5.10.1.6): 1

- **RBC Transitions:** packet 131

- **Radio infill area entry or exit:** packet 133 Loop announcement: packet 134

- **Conditional emergency stop:** message 15

- **DMI information:** packets 72,76 (text messages), packet 79 (geographical position information), message 34 (track ahead free request) minimum number of elements to be stored: fixed text: 5, free text: 5, geographical position:

- **Track conditions** (to be passed to the TIU and displayed at the DMI): packet 39 (traction system), packet 40 (current limitation), packet 68 (diverse track conditions), packet 69 (platform conditions). Pkt 139 minimum number of elements to be stored: 20, + 1 for change power supply + 1 for platform conditions, + 1 for current limitation

- **Route suitability:** minimum number of elements to be stored: 3

- **Big Metal Mass:** Technical information (to be used for BG-filtering): packet 67 (ignore BG integrity) minimum number of elements to be stored: 5

- **Virtual balise covers:** minimum number of elements to be stored: 10

- **list of position report locations.** In: pkt. 58 minimum number of elements to be stored: 15

All above information contains "Locations". These "Locations" can all be managed in the same manner. A location is defined using the following data structure:

For each location the "original reference" (ORBG) and the distance from this ORBG are remembered. In the data structure for passed BG's a correction distance is stored such that:
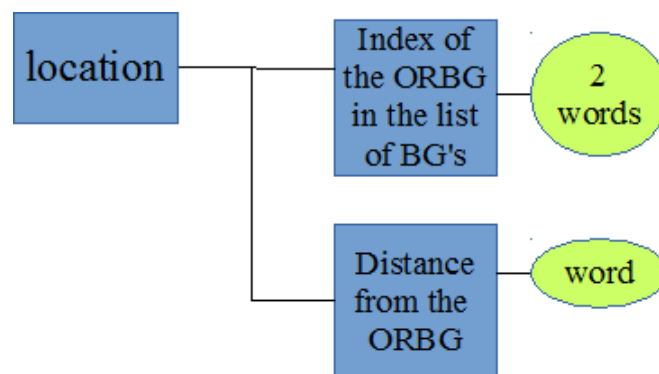
Figure 20. SRS architecture

$$d_{between\ train\ and\ location} = d_{from\ ORBG} - P_{Train\ to\ LRBG} - d_{correction\ stored} \qquad \text{FIXME}$$

FIXME: formula needs to be documented with a picture or refined description of the variables

where

d - represents a distance

P - represents the train position

$the stored correction distance (all including tolerances).$

A location is thus always stored as a distance from the nominal location of the ORBG (thus not having tolerances). Location based data is given from track-side as a chain of incremental distances. This shall be converted during the elaboration and storage.

Typically location based data has to be available in the order it is passed. This requires ordering and reordering if an element The following situations have to be taken into account:

- some types of location based data are not sent in the order they are passed, i.e. an element can be received which has to be included halfway the list.

- some types of location based data can be withdrawn from the list.

- Some types of location based data can be updated

- Information stored for a location can change, e.g. if the axle load changes, then the speed restrictions stored for this item shall be changed.

- Shifting of data shall be avoided.
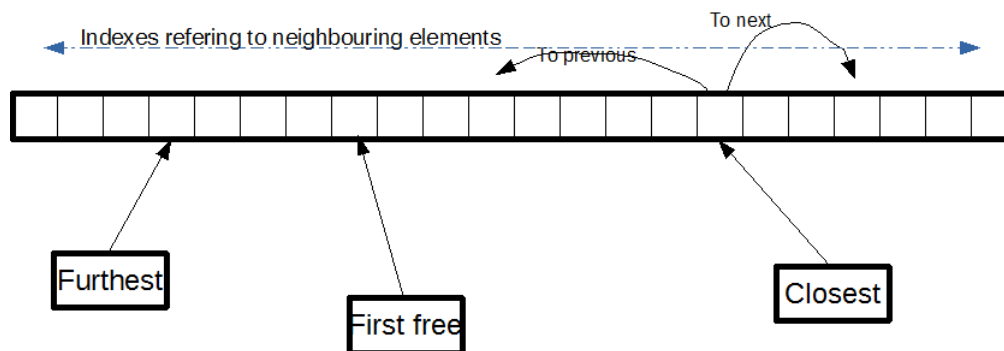
Figure 21. Indexing of internal data structure

General data structure for location based data of a specific type: "LOCATION BASED DATA STRUCTURE"

closest: The element related to the location which is the closest to the train front end

furthest: The element related to the location which is the closest to the train front end

first free: The element where the next received "location based data element of the specific type

Each element shall contain the index of the element where the preceding location is stored, the index of the element where the next further location is stored, the location (see type description above) and fields for information depending of the type of location based data.

Generic structure for an element in a data structure for location based data

General functions to be performed on a "LOCATION BASED DATA STRUCTURE" when new information is received are:

- Replace all data from a certain location onwards with the new received information.
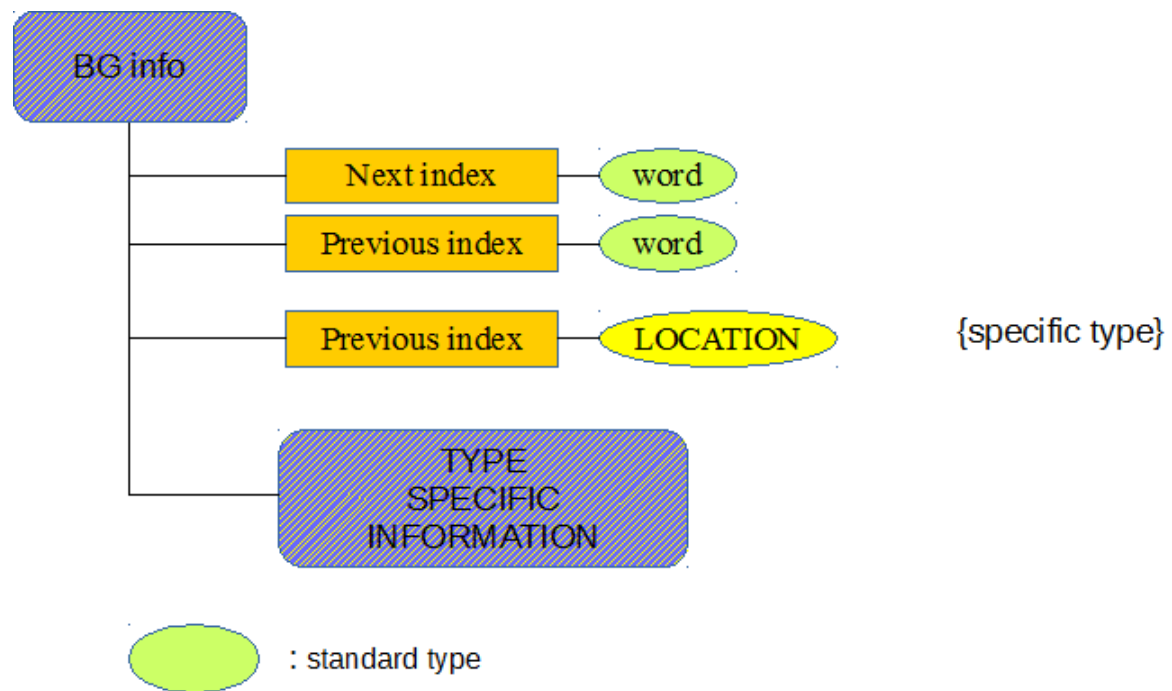
Figure 22. Location based data structure

- Insert one element in the structure at the right location, i.e. between the last preceding and the first next location of the same type.

- Delete one element of the structure, i.e. restore the order and free the memory where the information was stored.

- Update the information for a specific (already stored) element containing location based data, e.g. update the speed of "axle load dependent speed restrictions" in case the axle load changes.

ORDER IN WHICH PACKETS CONTAINING LOCATION BASED DATA SHALL BE ELABORATED

There are a few requirements determining the order in which different types of data shall be analyzed:

- MA's may only be accepted if a static speed profile and gradient information is available. Therefore the latter ones shall be elaborated before the MA information is elaborated.

- xxxx

Scope of the function The current function includes all elaboration of packets containing location based data into the internal data structure for location based data, further elaboration is not included. MRSP and the list of targets for braking curve monitoring are also location based, but they are a further elaboration of the received data. The building of the MRSP and list of braking curve targets is therefore not handled in the current function but in "Build MRSP + list of targets at the LRBG" (see figure 23).

11   Build structures for location based data: green structures

1) Select applicable SSP info and update the SSP structure
   (in: pkt 27, train characteristics)

2) Calculate speed restrictions from various sources and add them to the data
   structure for speed restrictions. (in pkt: 51,axle load profile, pkt 52,perm.
   Braking dist., pkt65/66: TSR, pkt 88: level crossing)

3) MA-management, in msg 3, 9, 33, 37 and pkt 12, 15, 16, 70, 90,
   including calculation of the SvL and LoA/EoA

4) Mode specific profiles: In: pkt 80

5) Fill the data structure for location based level transitions (in: pkt 41)

6) Update the "list of announced BG's" (in: pkt 5)

7) Update the "gradient profile" (in: pkt 21)

8) Update adhesion data (in: pkt 71)

9) Update location dependent test messages in: pkt 72, txt messages,
   pkt 76 is not used anymore (rejected in version 1 as well as version 1).

10) Updata geografical position information, i.e. list of locations where the track km
    changes. In: pkt 79, ,

11) Update list of track ahead free requests, in pkt 34

12) Update track conditions (a.o. "big metal mass"). In pkt 39, 40, 67, 68,138

13) Update list of position report locations. In: pkt. 58

Arbitrary order

Packets/ Msg's

MRSP (at LRBG)

Braking curve targets

mode

12

Build MRSP At LRBG + List of targets

SSP, LoA,SvL

Speed Restrictions,

MA structure

Mode profiles

Level transitions

Announced BG's

Gradients

Adhesion

Txt messages

km indicators

T.A.F. requests

Track conditions

Position rep. locations

10

Deleting passed locations

Train Position

As the reference location is given as an index for the list of detected BG's, all locations can
be stored as received from track-side and read in reference to the LRBG

Figure 23. Build Structure of Location Based Data

8 current partly openETCS Architecture - first iteration

Needs to be integrated into the overall architecture .....
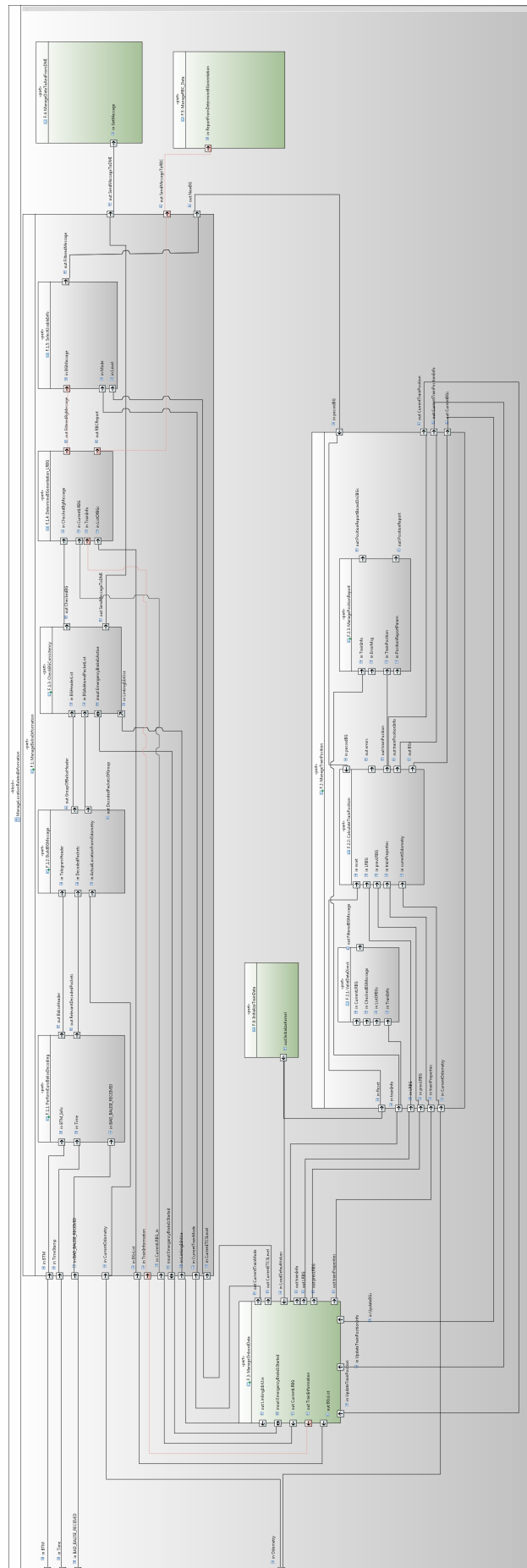
Figure 24. partly openETCS architecture - first iteration

9 merge the first iteration architecture with the overall architecture

Needs to be integrated into the overall architecture .....
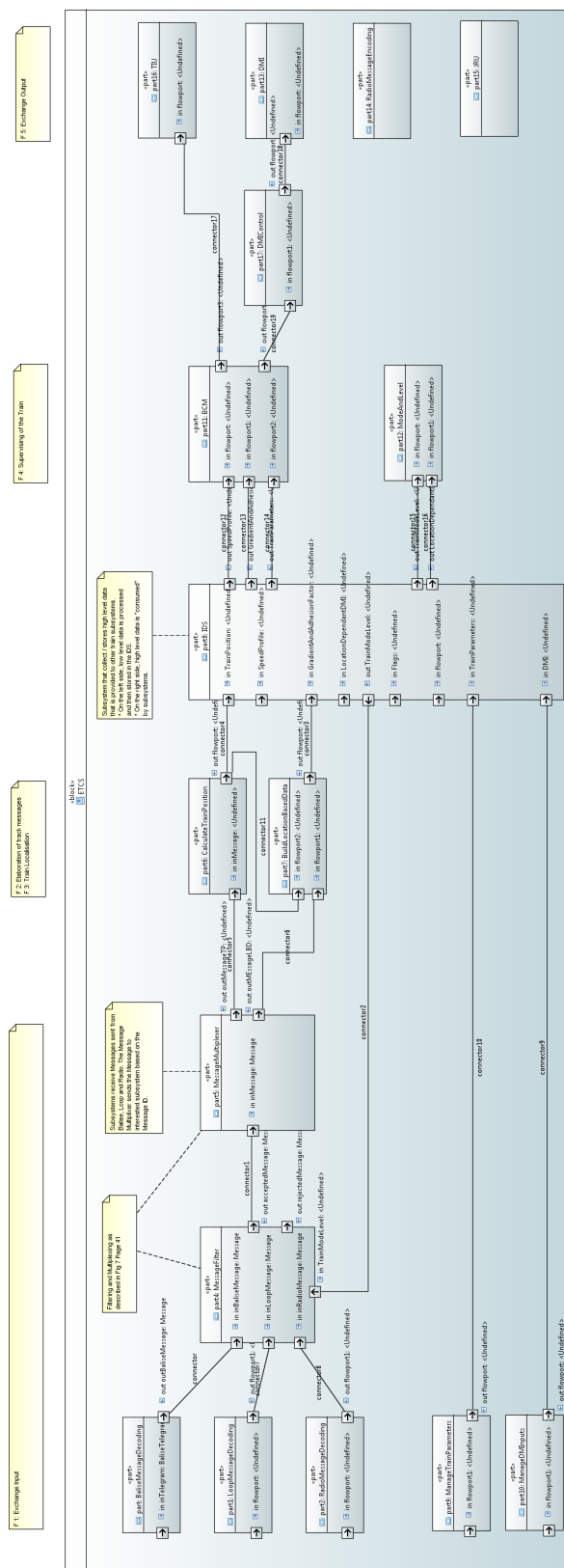
Figure 25. concept to merge

10 Alstom High Level Approach
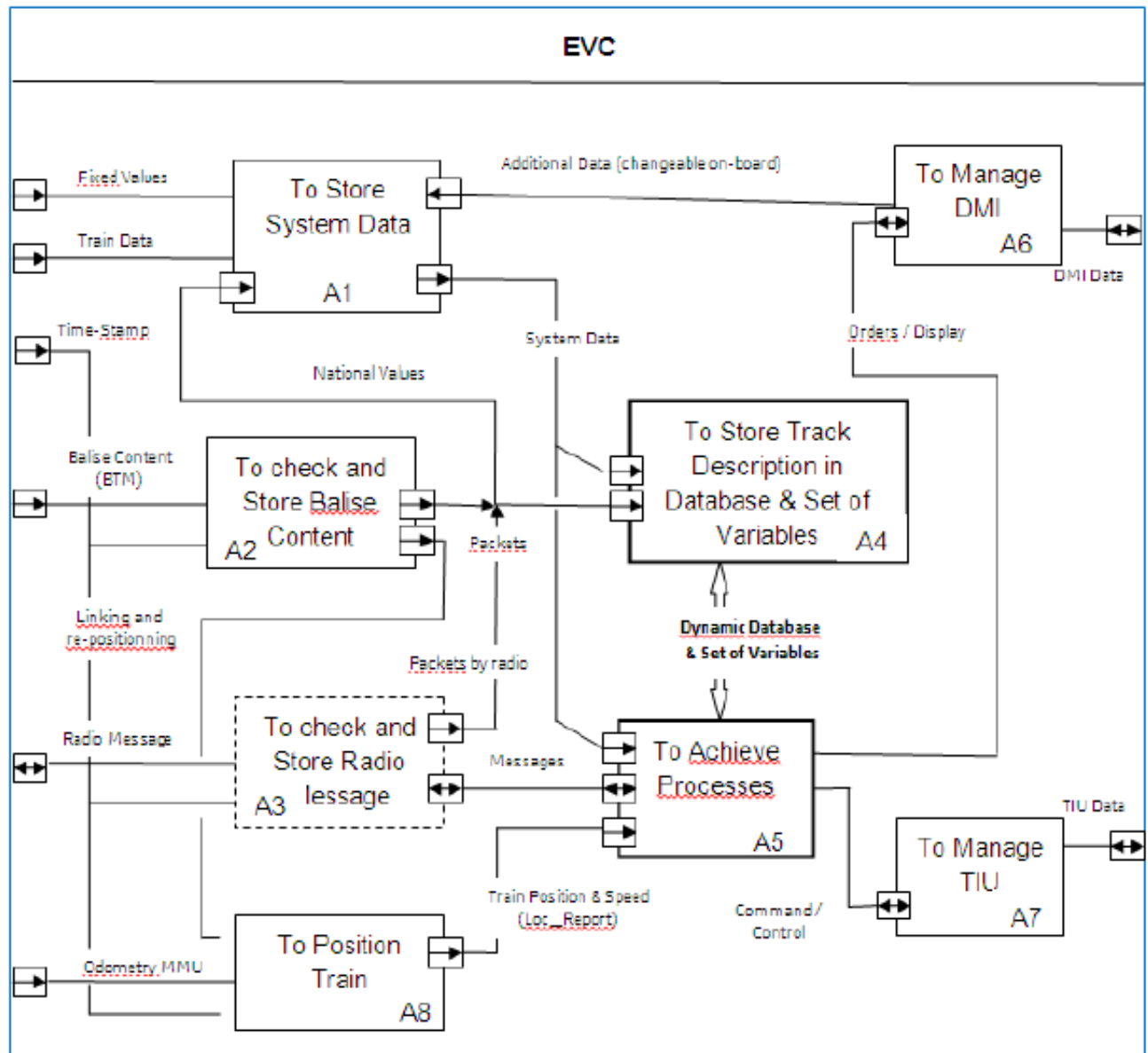
Figure 26. Alstom SRS Architecture Approach

11 Scope of Utrecht - Amsterdam

This chapter should give an overview about the necessary functions, message, ... for the open proof of concept on the Utrecht - Amsterdam line.

Below an overview of the input and output information for the system is given. The information is described in the following sources: Trackside: subset 26 chapter 7 and 8. Chapter 6 (baseline 2) has not yet been analysed, however the expectation is that this chapter will not contain more information (only diffferent format)

- TIU: annex of the API specification
- DMI: annex of the API specification
- JRU: not used
- odometer: the API specification

From the available input data the part which is necessary for the demonstration on Amsterdam-Utrecht has been selected. Data only needed for functions which are not realised in this first implementation are marked in grey. Those functions can be added later-on without affecting the core functionality realised in the demonstrator for Amsterdam-Utrecht.

Based on the replacement strategy for the data (a.o. ss26, 3.7.3) data structures will be defined to store the received data and to update the data if new information is received from the same type (sometimes overwrite, sometimes extend, or a combination).

For the output data it is defined which functional blocks shall be the source of the data, and which internal data shall be available to enable the determination of the required outputs.

Most of the input information shall be stored in a way that it can be remembered, updated and maintained (e.g. when driving). Other inputs are always used in the current cycle, have their effect and can be deleted.

11.1 Inputs from track-side

nb	Packet Name	Action process and/or storage	Replacement strategy of existing information / remarks	Resulting data structure	
0	Virtual Balise Cover marker	Store in "17": Marker given in the packet to indicate the BG can be ignored if the marker is stored on-board (see packet 6). no on board data	Not used on the infrastructure		
2	System Version order	Store in "11", Track data			
3	National Values	Store in "10", national/default data			
5	Linking	Store in "1", location based data: list of announced BG's			
6	Virtual Balise Cover order	Store the identity of the balise cover in "11" track data (new packet 6 in addition or replacing?)	Not used on the infrastructure		
12	Level 1 Movement Authority	Store in "1", location based data: MA			
13	Staff Responsible distance information from loop	Store in "1" location based data: distance in SR, Store in "1", location based data: MA	No loops		
15	Level 2/3 Movement Authority	Store in "1", location based data: MA			
16	Repositioning Information	Reset of the length of the current section of an MA What is the effect on the other location based data? Used to update the MA			
21	Gradient Profile	Store in "1", location based data: gradient profile			
27	International Static Speed Profile	Store in "1", location based data: static speed profile			
39	Track Condition Change of traction system	Store in "1", location based data: track conditions	No track conditions		
40	Track Condition Change of allowed current consumption	Store in "1", location based data: track conditions	No track conditions		
41	Level Transition Order	If no distance is given but a special value, i.e. an immediate level transition order, then the level available on-board with the highest given priority shall be stored in "6": the list of direct orders If a distance is given then store in "1", location based data: level transition order. Further for transitions to level 1, level2/3 of SN a "flag" has to be set to indicate to the function "filtering" that a (and which) level transition is announced.			
42	Session Management	Store in "13", procedure management: "establish connection" or			
		"terminate connection"			
44	Data used by applications outside the ERTMS/ETCS system.	Store as a complete packet for forwarding to the addressed STM.	No STM management		
45	Session Management	Store in "13", procedure management: ?????			
46	Conditional Level Transition Order	Like immediate order given by pkt 41, unless the current level is given in the priority list. In that case no action and no storage.			
49	List of balises for SH Area	Packet given together with the acceptance of shunting. Store in "8": List of BG's which may be passed in SH or SR (only one of the modes is active at a time)			
51	Axle load Speed Profile	Store in "1", location based data: speed restrictions			
52	Permitted Braking Distance Information	Evaluate the maximum speed based on the permitted braking distance and data stored on-board. Store in "1", location based data: speed restrictions			
57	Movement Authority Request Parameters	Store in "14" parameters for procedures: MA request			
58	Position Report Parameters	Store in "14" parameters for procedures: position report			
63	List of Balises in SR Authority	Packet given together with the acceptance of staff responsible. Store in "8": List of BG's which may be passed in SH or SR (only one of the modes is active at a time)			
64	Inhibition of revocable TSRs from balises in L2/3	Store in "17": filtering data			
65	Temporary Speed Restriction	Store in "1", location based data: speed restrictions			
66	Temporary Speed Restriction Revocation	Update "1", location based data: speed restrictions			
67	Track Condition Big Metal Masses	Store in "1", location based data: big metal mass	Technical information		
68	Track Condition	Store in "1", location based data: track conditions	No track conditions implemented		
69	Track Condition Station Platforms	Store in "1", location based data: track conditions	No track conditions implemented		
70	Route Suitability Data	Store in "1", location based data: route suitability data			
71	Adhesion Factor	Store in "1", location based data: adhesion factor			
72	Packet for sending plain text messages	Store in "1", location based data: DMI data			

76	Packet for sending fixed text messages	Store in "1", location based data: DMI data			
79	Geographical Position Information	Store in "1", location based data: DMI data			
80	Mode profile	Store in "1", location based data: mode dependent speeds	Replaced at the reception of a new MA (also if no Mode Profile is included), but only beyond the infill reference location.		
88	Level crossing information	Store in "1", location based data: speed restrictions (including a speed restriction to stop if necessary.	No level crossings at Amsterdam-Utrecht		
90	Track Ahead Free up to level 2/3 transition location	Leads to an MA request. To be used in MA management ("1")			
131	RBC transition order	Store in "1", location based data: RBC transitions			
132	Danger for Shunting information	?????????			
133	Radio infill area information	Store in "1", location based data: Radio infill area	No radio infill foreseen		
134	EOLM Packet	Gives the distance to the start of a loop plus further relevant information. I this information relevant for the LTM, or is the LTM passing any loop information and shall the check been done by the application software. In the latter case it is location based data: Store in "1", location based data: Loop announcement	No loops		
135	Stop Shunting on desk opening	Store in "6" direct and conditional orders			
136	Infill location reference	To be used in calculating the reference for location based data in the (which?) part of the packet (or telegram?) following this packet 136. As the reference is in advance of the LRBG, the concerning locations are stored in reference to the LRBG (only if the linking distance is provided) no data stored			
137	Stop if in Staff Responsible	Store in "6" direct and conditional orders			
138	Reversing area information	Store in "1", location based data: reversing area's			
139	Reversing supervision information	Store in "1", location based data: reversing area's What's the difference between D_reverse and L_reversearea?			
140	Train running number from RBC	Store in "9", train data			
141	Default Gradient for Temporary Speed Restriction	Store in "11", track data			
143	Session Management with neighbouring Radio Infill Unit	Store in "13": procedure status??	No RIU		
145	Inhibition of balise group message consistency reaction	Information to be used during consistency checking. This requires an early analysis of the message			
180	LSSMA display toggle order	?	No limited supervision		
181	Generic LS function marker	?	No limited supervision		
254	Default balise, loop or RIU information	Packet used in switchable BG's when track side information is not available. Shall be taken into account when storing information from the message.			
Msg Id		For all messages If M_ACK=1 (acknowledgement from the system (?) required), this information shall be stored in "13": procedure status information			
2	SR Authorisation	Store in "6" direct and conditional orders			
3	Movement Authority	Contains the MA: see packet 15			
6	Recognition of exit from TRIP mode	- no action (acknowledgement see above)			
8	Acknowledgement of Train Data	??			
9	Request to Shorten MA	Update "1", location based data: MA's			
15	Conditional Emergency Stop	Store in "1", location based data: conditional emergency stop			
16	Unconditional Emergency Stop	Store in "6": direct and conditional orders.			
18	Revocation of Emergency Stop	Update "1" or "6"/"13", In case of an unconditional em. Stop, revocation during braking is a kind of procedure "braking due to unc. em. Stop number xxxx", thus stored in "13"			
24	General message	-			
27	SH refused	stored in "13": procedure status information			
28	SH authorised	stored in "13": procedure status information			
33	MA with shifted reference location	Update "1", location based data: MA's			
34	Track ahead free request	Store in "1", location based data: DMI information			
37	Infill MA	RIU only..........	No RIU		
40	Train Rejected	stored in "13": procedure status information			
32	RBC/RIU system version	Store in "11", Track data			
38	Initiation of a communication session	stored in "13": procedure status information			
39	Acknowledgement of termination of a communication session	stored in "13": procedure status information			
41	Train accepted	stored in "13": procedure status information			
43	SoM position report confirmed by RBC	stored in "13": procedure status information			
45	Assignment of coordinate system	Update "3" reference system, i.e. store the direction in which a BG was passed (train orientation at the time of the passage) in the data structure for "passed BG's")			

11.2   Output to track-side

nb	Packet Name	Action to build the message from the originating data structure / remark	Originating data structure	
0	Position Report	Functional block: xxxxxxxxxx	"3" train position and reference	
		Store the actual distance plus tolerances from the on-board LRBG, the identity of the LRBG and the passing direction in packet 0	system ("List of passed BG's")	
1	Position Report based on two balise groups	As above, but add the identity of the previous BG if no passing direction is available	"3" train position and reference system ("List of passed BG's")	
3	Onboard telephone numbers	Functional block: xxxxxxxxxx	"9" train data	
4	Error Reporting	Functional block: xxxxxxxxxx	"18" error status	
5	Train running number	Functional block: xxxxxxxxxx	"9" train data	
9	Level 2/3 transition information	?????		
11	Validated train data	Functional block: xxxxxxxxxx	"9"train data	
44	Data used by applications outside the ERTMS/ETCS system.	From STM management, STM management not implemented		

Mes. Id.	Message Name	Action to build the message from the originating data structure	Originating data structure	
129	Validated Train Data	See above (packets)		
130	Request for Shunting	Function "procedure management" will generate the request	"13" procedure management	
132	MA Request	Function "MA management" will generate the request	"13" procedure management	
136	Train Position Report	See above"7" intermediate analysis results"	"7" intermediate analysis results"	
137	Request to shorten MA is granted		"7" intermediate analysis results"	
138	Request to shorten MA is rejected	"7" intermediate analysis results"	"7" intermediate analysis results"	
146	Acknowledgement	Acknowledgement of a received message, identified by the time stamp in the message. Functional Block: "Radio message consistency" because this function determines if the message is ready for evaluation. Information then stored in "7" intermediate analysis results"	"7" intermediate analysis results"	
147	Acknowledgement of Emergency Stop			
149	Track Ahead Free Granted			
150	End of Mission			
153	Radio infill request	No Radio infill		
154	No compatible version supported			
155	Initiation of a communication session			
156	Termination of a communication session			
157	SoM Position Report			
158	Text message acknowledged by driver			
159	Session Established			

11.3   Inputs from TIU derived from the API

- Status of EB, SB, traction cut off; (active, not active, unknown, failed)

- Flag indicating that the TIU is commanding SB, EB or traction cut off (interface between TIU and EVC, not TIU and train)

- ETCS system "Isolated"

- Train controlled from other unit: "Sleeping"

- Tilting on/off, relevant in case of selection of a train specific SSP: "train data"

- Direction controller (forward, "0", "backward")

- Selected cabin: (cab A, none, cab B, failure)

- Train integrity (state: yes/no, in the API a variable for length and/or time is missing!!!!!)

- Vigilance control (state?, external device active?)

What's meant with filtered values

11.4   Brake model parameters, API packet 3 xxx (to be filled)

Further (in the API) defined inputs from the TIU, but not needed for Amsterdam-Utrecht demonstration:
Text messages from the train to the ETCS DMI: API packet 1
Test and failure detection: API packet 6
STM specific behavior: API packet 7
Specific from MVB: Alstom application specific, packet 8
Diagnostic information: packet 12
Inhibition level: packet 13.

11.5   Outputs to TIU (derived from the API)

Brake and traction commands (EB, SB and TCO): packet 0
- EB command
- SB command
- TCO command
Vigilance disable order (order to disable train-side (?) vigilance control)

Further (in the API) defined outputs to the TIU, but not needed for Amsterdam-Utrecht demonstration
Track conditions packets 1, 5,
Odometric data, packet 2,
mode and level information, packet 3
Train type, i.e. the type of brake R, P or G, packet 4
Location reference, packet 6
Test and vigilance commands: packet 7
STM state: packet 8
diverse information (train length, train number, level inhibition ????????, data change due to change of running direction????)

What's specified in general for vigilance??

Conclusion: for Amsterdam-Utrecht only the emergency brake, service brake and traction cut of commands are needed as outputs to the TIU.

11.6   Inputs from DMI (derived from the API)

Acknowledgement from the driver on information (free text) sent to the DMI earlier: packet 50. message identifier corresponds to packet 9, packet 68 "EVCmessage"

Acknowledgement from the driver on information (fixed text) sent to the DMI earlier: packet 77. message identifier corresponds to packet 76 and 78 "Track message"

Operation of a button (for buttons displayed, see packet 5) by the driver: packet 52

Data entered by the driver: packet 53 and packet 54 (confirmation). As far as used for level selection at start of mission. A data-entry procedure is not foreseen in the first demonstration.

DMI controlled window exit request: packet 67, needed for the demonstration???

Question box reply: packet 73, only used for track ahead free request (in the current specification)

Question: How is the confirmation on data given with packet 3 returned to the EVC?

Further (in the API) defined inputs to the DMI, but not needed for (first implementation of the) Amsterdam-Utrecht demonstration
STM error on DMI ????: packet 48.
Text message deleted (due to full stack): packet 56
Connection confirmation: packet 57, not needed in a demonstrator (technical information only)
DMI state: packet 58, used for connection mgt, not needed for the demonstration
data for the JRU: packet 59: not needed for the demonstration
Report on-going acknowledgements: packet 70, shouldn't be necessary (what is done with this information in the EVC?)
Report on deletion due to full stack: packet 79, not needed for the demonstration
iBox state: packet 85: used in case of a remote connection to the DMI, not needed for the demonstration
ATP test result, packet 87: no test used for the demonstration
Results for additional Data-Entry (for STM's): packet 180, not implemented in the demonstration (STM and data-entry)

11.7   Outputs to DMI (derived from the API)

Minimum functionality for the demonstration:
- speed dial including braking curve ("hook")
- planning area: most restrictive speed profile ahead
- text messages (including acknowledgements)
- Level selection at start-up

- acknowledgement of level transitions

Message to delete elements on the DMI: packet 2
It's not clear if this only applies to text messages and acknowledgement requests or to all data?
Further for specific data (a.o. text messages) special packets are sent to delete the data from the screen.

Confirmation request: packet 3
Sending text or data fields plus an identifier of the confirmation asked (for data entry?) How is responded to the EVC after receiving this packet?

Text message with or without confirmation or acknowledgement request: packet 9 + packet 68 in case of required acknowledgement One line of free text

Fixed text message to be displayed (identifier from track to train packet 76): packet 76
Fixed text message to be deleted: packet 78

Displaying and hiding buttons: packet 5
list of buttons is not included in the API (available in the standardised screen layout?, TS50459)

Sound activation: packet 7
list of sounds is not included in the API (available in the standardised screen layout?, TS50459)

Speed, braking curve and target: packet 8, packet 43 and packet 71.
New information shall overwrite the existing information.

Display an icon (not, constant, flashing): packet 11
list of icons is not included in the API (available in the standardised screen layout?, TS50459)
Data for viewing after a driver request: packet 15
list of data is not included in the API

Planning area speed restrictions: packet 19, a translation between speed and width of the indication is needed.

Planning area gradients: packet 26

Planning area displaying on/off: packet 31, in certain modes the planning area shall not be shown at all (also not an empty planning area).

Geographical position information: packet 33

DMI controlled window request: packet 66, needed for the demonstration???

Question box request: packet 72, only used for track ahead free request (in the current specification)

11.8   Further (in the API) defined outputs to the DMI, but not needed for (first implementation of the) Amsterdam-Utrecht demonstration

- Screen control packet 4,
- Screen request: packet 6, is used for displaying windows, but the demonstration can work with a standard screen/window - EVC connection request: packet 16, for the demonstration PC's with addressable screens can be used. - Local time: packet 17, for the demonstration the displays local time can be used - Planning area icons: packet 18, for the demonstration elaborating track data is not needed, thus displaying not possible - data transmission start/stop: packet 20, not needed in case of communication between PC's - Set speed info: packet 23, no cruise control foreseen for the demonstration - Data entry related information: packet 27, data freeze: does this mean that the data is confirmed, final and can not be changed by the - - - driver anymore?, if so what's the use? The EVC could simply ignore changes. nice to have, but not essential for the demonstration - Driver language transmission: packet 30, only needed if translations are made in the EVC. However it seems more efficient to translate "fixed texts" in the DMI instead of in the EVC. According to the evaluation of DMI packet76, the translation is done in the DMI. - Tunnels stopping area distance: packet 32 - STM specific test request: packet 37, STM not supported in the demonstration. - STM accessibility packet 47, STM not supported in the demonstration. - Acknowledgement retention: packet 69, For the demonstration (and further applications) this function shouldn't be necessary - Manage ATP test: packet 82: test not implemented for the demonstration. - Request, confirmation, view and echo additional Data-Entry (for STM's): packet 179/181/183/185, not implemented in the demonstration (STM and data-entry)

11.9   Input from the odometer (according to the API specification):

Distance
minimum distance driven since start up
nominal distance driven since start-up
maximum distance driven since start-up

Speed
Safe nominal speed
Raw nominal speed
Raw minimum speed
Raw maximum speed

Motion state: motion/no motion

Motion direction: cabA/cabB

Appendix: References

[1] openETCS Product Owner. *WP3 Product Backlog, Selection of First Iteration.* `https://github.com/openETCS/SRS-Analysis/labels/backlog%20item`.

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

[3] openETCS WP3-team. *openETCS Working Result: List of ETCS Functions*, November 2013. `https://github.com/openETCS/SRS-Analysis/blob/master/System%20Analysis/List_Functions.xlsx`.

[4] openETCS. *openETCS SCADE model*, 2014. `https://github.com/openETCS/modeling/tree/master/model/Scade/System`.