Ref. Ares(2023)8798999 - 21/12/2023

ERJU FP2 WP27 Digital Register Specification, Development, and Implementation

D27.2 – Specification of Digital Register Implementation(s) required in R2DATO

Due date of deliverable: 30/11/2023

Actual submission date: 10/11/2023

Leader/Responsible of this Deliverable: Danny Kreyenberg, DB Netz AG

Reviewed: Y

Document status		
Revision	Date	Description
01	23/10/2023	First issue for review within WP27 and stakeholders (WP13, WP44, WP45)
02	10/11/2023	Updated version post review and first version of D27.2 for release submitted for TMT Review

Project funded from the European Union's Horizon Europe research and innovation programme		
Dissemination Level		
PU	Public	X
SEN	Sensitiv – limited under the conditions of the Grant Agreement	

Start date: 01/12/2022 Duration: 42 months

ACKNOWLEDGEMENTS

This project has received funding from the Europe's Rail Joint Undertaking (ERJU) under the Grant Agreement no. 101102001. The JU receives support from the European Union's Horizon Europe research and innovation programme and the Europe's Rail JU members other than the Union.

REPORT CONTRIBUTORS

Name	Company	Details of Contribution
Harish Narayanan	Nextrail	Author
Frederik Düpmeier	Nextrail	Author
Benedikt Wenzel	Nextrail	Author
Rodion Altshuler	DB	Author
Franziska Kempf	SBB	Reviewer
Anders Ekberg	Chalmers/TRV	Reviewer
Said El-Fassi	SNCF	Reviewer
Adusumalli Vijaya Bhaskar	GTS-D	Reviewer
Christan Sadowski	DB	Reviewer
Thomas Renner	DB	Reviewer
Albert Ledermann	SBB	Reviewer
Bažant Lubor	AZD	Reviewer
Konstantinos Emmanouil	DB	Reviewer
Bertrand Badot	Alstom	Reviewer
Simon Chadwick	Siemens	Reviewer
Frank Skowron	DB	Reviewer
Gelavizh Ahmadi	SMO	Reviewer
Peter Tummeltshammer	GTS-D	Reviewer
Gregorio Velasco	ADIF	Reviewer

Disclaimer

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author's view – the Joint Undertaking is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

EXECUTIVE SUMMARY

The system Digital Register (DR) was introduced to realise a centralised static reliable infrastructure data management system to facilitate other systems that are part of ERJU R2DATO like Automatic Train Operation Transactor (AT), Traffic Management System (TMS), Plan Execution (PE), Moving Block System (MBS), Absolute Safe Train Positioning (ASTP), Perception On-Board (PER-OB), Automatic Processing Module On-Board (APM-OB), and Automatic Driving Module (ADM) to access and function with reliable static infrastructure data.

This deliverable D27.2 “Specification of Digital Register Implementation(s) required in R2DATO” is part of WP27 in FA2 R2DATO. It focuses on specifying the different Functional Interfaces along with their implementation specifications for different required implementations as they are part of task 27.2, 27.3, and 27.4 of WP27. The basis of the system DR has been defined in deliverable D27.1 – “Set of requirements on the Digital Register in R2DATO”. This deliverable is based on requirements defined in deliverable D27.1. Ergo, we recommend reading D27.1 before you read D27.2 to improve the understanding of D27.2.

A semiformal method of definition of requirements using scenarios and descriptions is used in this document for defining the interface messages and their sequences. The current state of requirements (see chapter 4 - Functional Interface Specification, and chapter 5 - Implementation specification) in this document are considered plausible and can be shared with relevant stakeholders as the status quo of WP27.

The current state (the first version) of this document focuses on Functional Interface Specifications between DR and the Trackside Consuming Systems (MBS, PE, TMS, AT) with an emphasis on the implementation specification for the interface DR-MBS for use in release 2 of WP44/45 to realise a Moving Block Demonstrator. The interface specification with the On-Board systems will be included in this document in a subsequent release.

The document does not yet define the requirements for all the identified logical internal and external interfaces of DR. An iterative approach, corresponding to the WPs which need data from DR, would be put into play to update the requirements to be defined in this deliverable. In parallel, another deliverable D27.3 is planned within the framework for WP27 to provide a detailed set of requirements for DR during the month 36 (i.e., July 2025) of the project timeline. The state of this deliverable will also be updated correspondingly when needed.

ABBREVIATIONS AND ACRONYMS

ADM	Automatic Driving Module
AGG	Aggregator
APM	Automatic Processing Module
ASTP	Absolute Safe Train Positioning
AT	ATO Transactor
ATO	Automatic Train Operation
ATP	Automatic Train Protection
COMP	Compiler
DM	Data Manager
DR	Digital Register
ENG	Engineering
ERJU	Europe's Rail Joint Undertaking
EUG	ERTMS Users Group
FA	Flagship Area
FP	Flagship Project
GoA	Grade of Automation
IM	Infrastructure Manager
INFRA	Infrastructure
IP	Innovation Pillar
MBS	Moving Block System
MDM	Maintenance and Diagnostic Management
OB	On-Board
OC	Object Controller
OPC-UA	Open Platform Communications – Unified Architecture (protocol by OPC found.)
PE	Plan Execution
PER	Perception
PREP	Preparation
PUB	Publish
R2DATO	Rail to Digital and Automated Train Operation
RCA	Reference CCS Architecture
RU	Railway Undertaking
SP	System Pillar
SS	Subset
TCCS	Transversal CCS
TMS	Traffic Management System
TP	Train Protection
TS	Trackside
URA	Usage Restriction Area
VAL	Validator
WP	Work Package

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The deliverable D27.2 “Specification of Digital Register Implementation(s) required in R2DATO” in the framework of WP27 of the ERJU R2DATO project constitutes of the specifications to enable the implementation/demonstration activities as part of the work package WP tasks 27.2, 27.3, and 27.4. This document uses the initial set of requirements defined in D27.1 as a foundation to specify the functional interface requirements required for the different implementation activities.

The deliverable primarily includes the logical architecture and functional interface specifications for the interfaces between the Digital Register and Consuming Systems. In addition, due to ongoing demonstrators in ERJU IP FA2, an implementation specification for the WP44/45 to realise a moving block demonstrator is also defined.

The document starts with the high-level logical architecture definition. The subsequent chapters provide data representations and functional interface specifications for trackside systems. Respective specifications for On-Board Systems will follow in a subsequent release of this document. The data representation, as well as the functional interface specifications, are defined generic and can be used for any implementation specification.

The deliverable currently does not cover implementation/function interface specifications for all the required interfaces defined in this deliverable. The corresponding specifications for the same will be updated iteratively according to the needs of the different demonstrators.

1.1 DEFINITIONS

This chapter only defines the new terms encountered in this document. For all other data definitions and object relations, please refer to chapter 1.1 in [2].

Term	Definition
Operational Plan	The Operational Plan is the result of the planning process performed by TMS. It describes either a planned Operational Movement, Operational Restriction, or Operational Warning Measure through a temporal sequence of Operational Events to be implemented by ATO Execution and/or Plan Execution in the Area of Control. For more details refer to SCI-OP concept

2 LOGICAL ARCHITECTURE OF DIGITAL REGISTER

The following high-level logical architecture is derived from the system boundary/environment definitions in chapter 4.2 in [2].

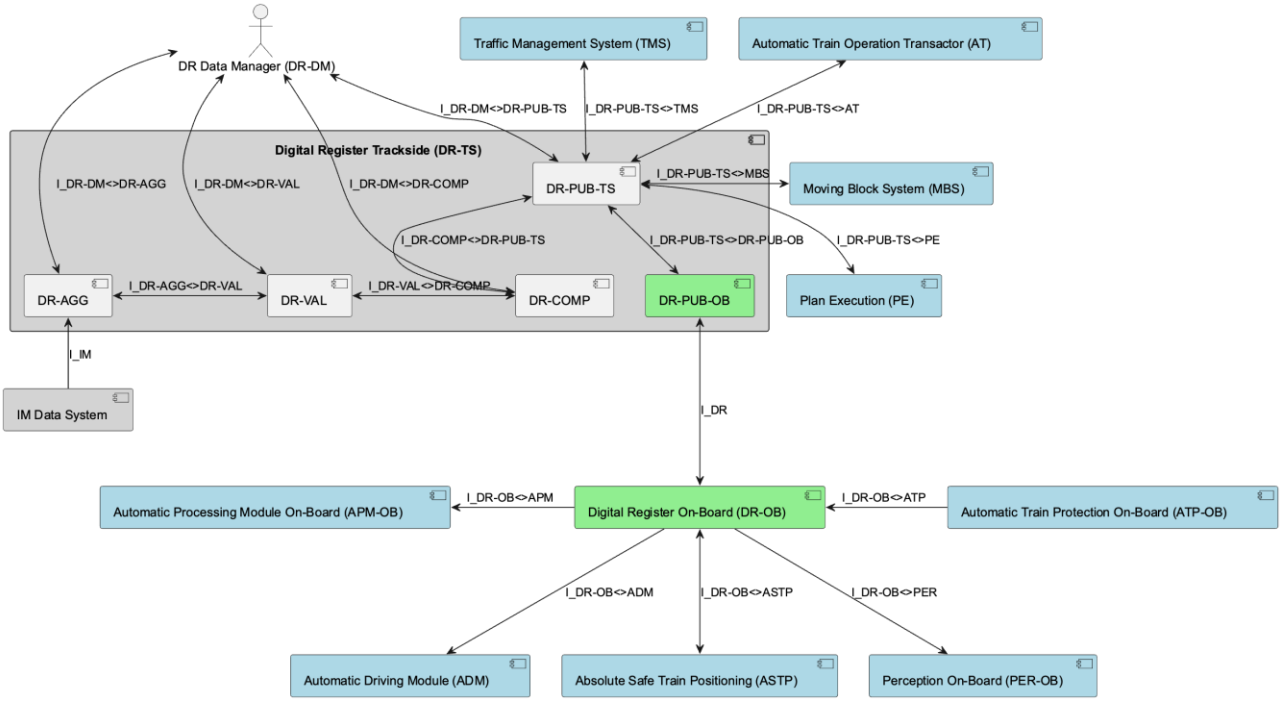

Figure 1: Logical Architecture for Digital Register in WP27

According to DR's functional responsibilities for data aggregation, validation, compilation, and publication, DR-TS is further divided into logical components (shown in the upper grey box) along the process flow of data sets within DR. The new logical components implicate new logical interfaces between the different components of DR-TS. These corresponding components and interfaces are described in the next chapters.

With regards to the Domain Data publishing components, they are further divided into two, one for trackside (DR-PUB-TS) and one for on-board (DR-PUB-OB) publication. Consuming Systems that receive their data from DR-PUB-TS are referred to as Trackside Consuming Systems, and Consuming Systems that receive their data from DR-OB as On-board Consuming Systems. It is to be noted that DR-PUB-OB is a Trackside Consumer of DR-PUB-TS. The Domain Data is then transacted to DR-OB from DR-PUB-OB via the airgap interface I_DR.

2.1 LOGICAL COMPONENT DEFINITION

2.1.1 Digital Register Aggregator (DR-AGG)

The system component DR-AGG of DR-TS imports Engineering Input Data from the IM Data System with the help of DR Data Manager (a Human actor) and aggregates the data into a common set of Engineering Data. The import process in DR-AGG is triggered using a 'Import Trigger' by the DR Data Manager. After completing the aggregation process successfully, DR-AGG sends the Engineering Data to DR-VAL for validation.

The following function defined in chapter 6.1.2 in [2] shall be realised as a part of this component.

1. SysF2 Import and aggregate Engineering Data

2.1.2 Digital Register Validator (DR-VAL)

DR-VAL validates the aggregated Engineering Data based on a set of engineering and validation rules. All issues found during the validation phase are reported to the DR Data Manager. After completing the validation process successfully, DR-VAL sends the Engineering Data to DR-COMP for compilation into Domain Data.

The following function defined in chapter 6.1.2 in [2] shall be realised as a part of this component.

1. SysF3 Validate Engineering Data

2.1.3 Digital Register Compiler (DR-COMP)

DR-COMP compiles the validated Engineering Data into a standardised Domain Data format (see System Pillar CCS/TMS Data model) that meets the requirements of each of the Consuming Systems. After completing the compilation process successfully, DR-COMP sends Compile state to DR Data Manager, and the Domain Data to DR-PUB-TS for publishing it to the Consuming Systems in a safe and harmonised process.

The following function defined in chapter 6.1.2 in [2] shall be realised as a part of this component.

1. SysF5 Compile Domain Data

2.1.4 Digital Register Publisher Trackside (DR-PUB-TS)

DR-PUB-TS publishes the Domain Data to the trackside Consuming Systems (incl. DR-PUB-OB). The publish process in DR-PUB-TS is triggered using a ‘Publish Trigger’ by the DR Data Manager. This component also ensures a synchronised activation (which is to be ensured by DR Data Manager) of the Domain Data in the Trackside Consuming Systems.

The following functions defined in chapter 6.1.2 in [2] shall be realised as a part of this component.

1. SysF8 Maintain Domain Data
2. SysF9 Transact Domain Data
3. SysF12 Transact Domain Data activation request
4. SysF13 Authorise activation of Domain Data

2.1.5 Digital Register Publisher On-Board (DR-PUB-OB)

DR-PUB-OB publishes the Domain Data, received from DR-PUB-TS, to registered instances of DR-OB on the active vehicles in the area of responsibility of DR-TS for further distribution to the on-board Consuming Systems.

The following functions defined in chapter 6.1.2 in [2] shall be realised as a part of this component.

1. SysF8 Maintain Domain Data
2. SysF9 Transact Domain Data
3. SysF15 Receive on-board Map Reference Data trigger

2.1.6 Digital Register On-Board (DR-OB)

DR-OB obtains parts of the Domain Data (viz. On-Board Domain Data) that are relevant for the on-board system from DR-PUB-OB and provides it to the Consuming Systems on the train. In the context of the on-board architecture, On-Board Domain Data is referred to as “On-Board Map Data”.

The following functions defined in chapter 6.1.2 in [2] shall be realised as a part of this component.

1. SysF14 Perform Map Data relevancy check
2. SysF16 Maintain Domain Data
3. SysF17 Validate on-board Map Data
4. SysF18 Activate Map Data
5. SysF19 Deactivate Map Data

2.2 Logical Actor Definition

For actor definitions, please refer to chapter 4.3.1 in [2].

2.3 Logical Interface Definition

The following table provides an overview of the logical interfaces in DR-TS and DR-OB.

Table 1: Logical Interfaces Overview

Id No.	Name of Interface	Interface Endpoints
1.	I_DR	DR-PUB-OB and DR-OB
2.	I_DR-AGG<>DR-VAL	DR-AGG and DR-VAL
3.	I_DR-VAL<>DR-COMP	DR-VAL and DR-COMP
4.	I_DR-COMP<>DR-PUB-TS	DR-COMP and DR-PUB-TS
5.	I_ DR-PUB-TS<>DR-PUB-OB	DR-PUB-TS and DR-PUB-OB
6.	I_DR-DM<>DR-AGG	DR-DM and DR-AGG
7.	I_DR-DM<>DR-VAL	DR-DM and DR-VAL
8.	I_DR-DM<>DR-COMP	DR-DM and DR-COMP
9.	I_DR-DM<>DR-PUB-TS	DR-DM and DR-PUB-TS
10.	I_IM	IM and DR-AGG
11.	I_DR-PUB-TS<>MBS	DR-PUB-TS and MBS
12.	I_DR-PUB-TS<>TMS	DR-PUB-TS and TMS
13.	I_DR-PUB-TS<>PE	DR-PUB-TS and PE

Id No.	Name of Interface	Interface Endpoints
14.	I_DR-PUB-TS<>AT	DR-PUB-TS and AT
15.	I_DR-OB<>ASTP	DR-OB and ASTP
16.	I_DR-OB<>ADM	DR-OB and ADM
17.	I_DR-OB<>PER	DR-OB and PER
18.	I_DR-OB<>ATP	DR-OB and ATP
19.	I_DR-OB<>APM	DR-OB and APM

3 DIGITAL REGISTER DATA MODEL FOR WP27

The data representation for WP27 Digital Register data model shall use the standardised data model defined in systems pillar within Transversal CCS Domain. Refer to System Pillar CCS/TMS Data model.

Note:

1. Even though the functional interface specification defined below describes different messages like requests, acknowledgements, commits, etc. these are currently not a part of the said data model but will be integrated within relevant interface specification deliverables in Transversal CCS.
2. The link mentioned above provides a copy of the system pillar data model v0.3 in Project Place. For the actual model in polarion refer to this link.

4 FUNCTIONAL INTERFACE SPECIFICATION

This chapter contains the functional interface specifications for the interfaces towards the external systems MBS, TMS, AT, ADM, ASTP, PER-OB, APM-OB from DR-PUB-TS and DR-OB. The requirements are specified using sequence diagrams with sufficient explanations using sequence diagram descriptions. The current Functional Interface Specification is restricted to the following interfaces:

1. I_DR-PUB-TS<>MBS
2. I_DR-PUB-TS<>TMS
3. I_DR-PUB-TS<>PE
4. I_DR-PUB-TS<>AT
5. I_DR-DM<>DR-PUB-TS

The other remaining interfaces identified in chapter 2.3 will be brought in iteratively with respect to the different demonstrator needs into this deliverable.

4.1 INTERFACE TO TRACKSIDE CONSUMING SYSTEMS

The interfaces to the Trackside Consuming Systems shall be defined as a standardised interfaces . These can be used irrespective of implementation strategies. This is because,

- The Trackside Consuming Systems might change their data needs in the future, so a generic interface prevents the necessity of performing (major) changes in DR.
- There might also be new systems that might need to consume Domain Data from DR in the future. Generic interface specifications shall allow us to add the said consumers as Trackside Consuming Systems without implying (major) changes in DR.
- Therefore, we describe the basis scenario processes of the interfaces to the Trackside Consuming Systems for all Trackside Consuming Systems together.
- The following scenario processes are based on the Processual Functional Flows defined in [2].

4.1.1 Trackside exchange scenarios

4.1.1.1 Trackside Consuming Systems acquire Domain Data on start-up

During the start-up or restart of a Trackside Consuming System (i.e., when the system does not have any version of Domain Data present or cannot know if it has the right version of Domain Data), every Trackside Consuming System shall request Domain Data from DR-PUB-TS using a Domain Data request. This scenario describes how trackside consuming systems acquire Domain Data during start-up.

If the Trackside Consuming System already has a version of Domain Data, it can include the version number in the request so that DR-PUB-TS can check and confirm if that is the latest active version. Trackside Consuming Systems like TMS that store multiple versions of Domain Data may include multiple Domain Data versions in the Domain Data request. Having multiple Domain Data versions enables TMS to have a forecast of upcoming infrastructure changes for operational planning purposes

If the version currently stored as the active version in the Trackside Consuming System equals the current active version in DR-PUB-TS, DR-PUB-TS confirms that the Trackside Consuming System has the correct version by sending a 'Domain Data version acknowledgement' message. Otherwise, DR-PUB-TS sends the current active version to the requesting Trackside Consuming System. If the Trackside Consuming System has asked for multiple versions, DR-PUB-TS sends these versions as separate 'Domain Data' messages.

Note: Alternatively, the safe version check can be implemented on consumer side, so DR-TS has less safety-related functionality. This is to be defined in accordance with best practices and WP44/45 MBD implementation results in the next document increments.

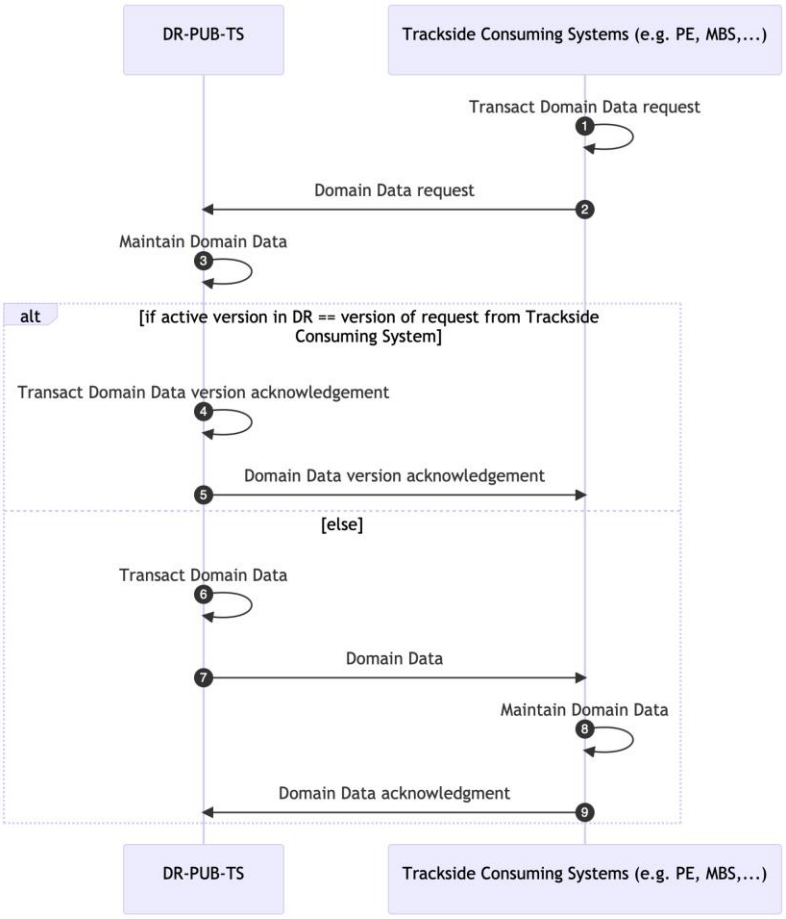

Figure 2: Scenario: Acquire new version of Domain Data during start-up

Table 2: Scenario description: Acquire new version of Domain Data during start-up

Scenario Condition	Scenario Description
ALT: If active version in DR == version of request from Trackside Consuming System	Precondition: Trackside Consuming System has a version of Domain Data present AND DR-PUB-TS has established a communication session with each Trackside Consuming System. 1. Each Trackside Consuming System sends a 'Domain Data Request' message to DR-PUB-TS.

	2. DR-PUB-TS executes the function ‘Maintain Domain Data’. 3. DR-PUB-TS executes the function ‘Transact Domain Data version acknowledgement’. 4. DR-PUB-TS sends a ‘Domain Data version acknowledgement’ message to each Trackside Consuming System. Postcondition: DR-PUB-TS has sent a ‘Domain Data version acknowledgement’ message to all Trackside Consuming Systems.
ALT: Else	Precondition: Trackside Consuming System has an older or no version of Domain Data present AND DR-PUB-TS has established a communication session with each Trackside Consuming System. 1. A Trackside Consuming System sends a ‘Domain Data Request’ message to DR-PUB-TS. 2. DR-PUB-TS executes the function ‘Maintain Domain Data’. 3. DR-PUB-TS executes the function ‘Transact Domain Data’. 4. DR-PUB-TS sends a ‘Domain Data’ message to each Trackside Consuming System. 5. Each Trackside Consuming System executes the function ‘Maintain Domain Data’. 6. Each Trackside Consuming System sends a ‘Domain Data version acknowledgement’ message to DR-PUB-TS. Postcondition: DR-PUB-TS has received a ‘Domain Data Acknowledgement’ message from all Trackside Consuming Systems.

4.1.1.2 Distribution of a new version of Domain Data to the Trackside Consuming Systems

For activation of a new Domain Data version, a usage restriction for some parts/objects of the railway infrastructure is necessary (cf. chapter 4.1.1.3). To minimise the time when a usage restriction for Domain Data activation is active, DR-PUB-TS shall send in advance the new version of Domain Data to Trackside Consuming Systems. This scenario describes how a new version of Domain Data is distributed to the Trackside Consuming Systems.

Initially, after the new Domain Data version is ready for distribution in DR-PUB-TS, DR-PUB-TS sends the new Domain Data version to TMS so that TMS can plan an activation time (the Domain Data version might have a predefined time window for Domain Data update planning). TMS sends the planned activation time to DR-PUB-TS. Based on the received activation time, DR-PUB-TS shall calculate a distribution time in advance of the activation time. By calculating the distribution time, DR has to take into consideration that some Trackside Consuming Systems might only be able to store the active version of Domain Data and one additional version (the version to be activated next). Therefore, the distribution time shall be set well in advance of the activation time so that the distribution process will be finished before the activation process begins, but also not too early, especially not before the previous Domain Data version has been activated to avoid unnecessary usage of the Consuming System’s performance levels.

When the distribution time has been reached, DR-PUB-TS sends the new Domain Data version to all Trackside Consuming Systems. The Trackside Consuming Systems store the new Domain Data version, send an acknowledgement to DR-PUB-TS confirming that they have received the new Domain Data version correctly, and wait for the activation command to activate the new Domain Data version. Meanwhile the Trackside Consuming Systems continue to use the active version of Domain Data present in the Consuming System.

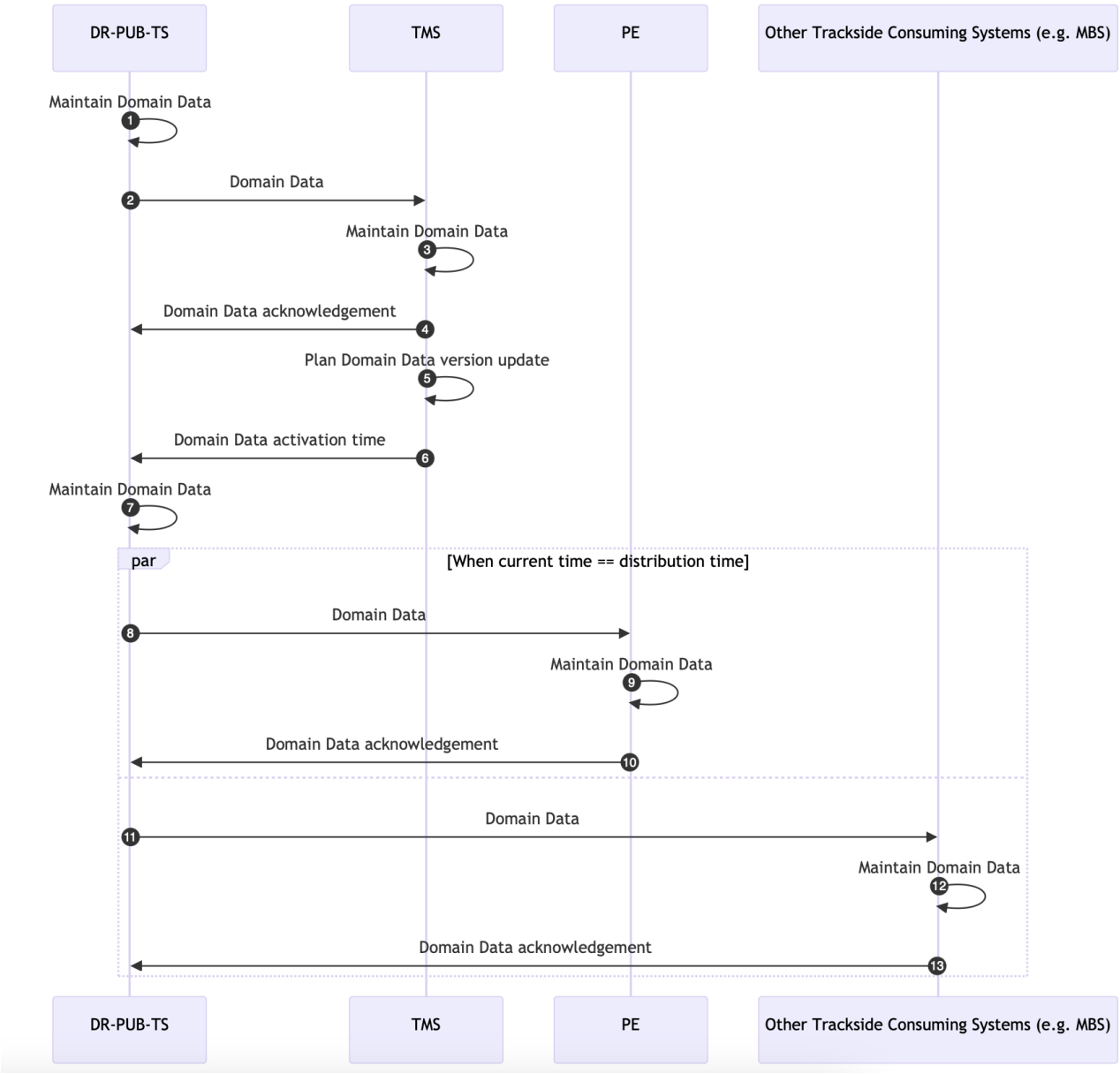

Figure 3: Scenario: Distribution of new version of Domain Data

Table 3: Scenario description: Distribution of new version of Domain Data

Scenario Condition	Scenario Description

Nominal case	Precondition: DR-PUB-TS has established a communication session with each Trackside Consuming System. 1. DR-PUB-TS executes the function 'Maintain Domain Data'. 2. DR-PUB-TS sends a 'Domain Data' message to TMS. 3. TMS executes the function 'Maintain Domain Data'. 4. TMS sends a 'Domain Data Acknowledgement' message to DR-PUB-TS. 5. TMS executes the function 'Plan Domain Data version update'. 6. TMS sends a 'Domain Data activation time' message to DR-PUB-TS. 7. DR-PUB-TS executes the function 'Maintain Domain Data'. 8. Parallel: a. DR-PUB-TS sends a 'Domain Data' message to PE. b. PE executes the function 'Maintain Domain Data'. c. PE sends a 'Domain Data acknowledgement' message to DR-PUB-TS. d. DR-PUB-TS sends a 'Domain Data' message to each Trackside Consuming System. e. Each Trackside Consuming System executes the function 'Maintain Domain Data'. f. Each Trackside Consuming System sends a 'Domain Data acknowledgement' message to DR-PUB-TS. Postcondition: DR-PUB-TS has received 'Domain Data acknowledgement' message from all Trackside Consuming Systems.

4.1.1.3 Restrict topology for Domain Data Activation

For safe railway operations, it is necessary that all involved systems use the current active version of Domain Data. Therefore, a safe process is needed which ensures that all Trackside Consuming Systems activate the new Domain Data version at the same time and that the infrastructure to be affected by changes caused by the Domain Data version update (e.g., deleted or changed elements) is not in use by railway vehicles during the activation process (e.g., occupied by a railway vehicle or part of a Movement Permission). This scenario describes the first step of the activation process that establishes a usage restriction with the help of a Usage Restriction Area (URA) in MBS that covers all elements to be changed or deleted within the Domain Data version update in order to prevent usage of these affected elements as well as the affected area.

Preconditions for this step are that

1. the DR Data Manager has authorised the activation with the Authorisation command, which sets a flag for the Domain Data version that it is authorised for activation, and that

2. DR has received acknowledgements from all Trackside Consuming Systems that they have received the version of Domain Data to be activated.

When DR-PUB-TS has met these preconditions and the activation time provided by TMS has been reached, DR-PUB-TS shall start the activation process by sending a 'Domain Data activation request' to TMS.

PE, based on the Domain Data version to be activated according to the operation plan, requests a usage restriction for these infrastructure objects to MBS. MBS, being a safety-critical system, is responsible for setting this usage restriction over the infrastructure elements that are affected by the Domain Data version update. While setting this up, MBS shall ensure that all necessary preconditions are fulfilled[1]. Correspondingly, MBS also confirms that the necessary usage restrictions are active by sending 'Domain Data usage restriction acknowledgement' to DR-PUB-TS.

On receiving this acknowledgement from MBS, DR-PUB-TS shall start the second step of the activation process (cf. chapter 4.1.1.4). When this has been successfully finished, MBS removes the usage restrictions related to the activation of the new Domain Data version and sends an acknowledgement message to PE.

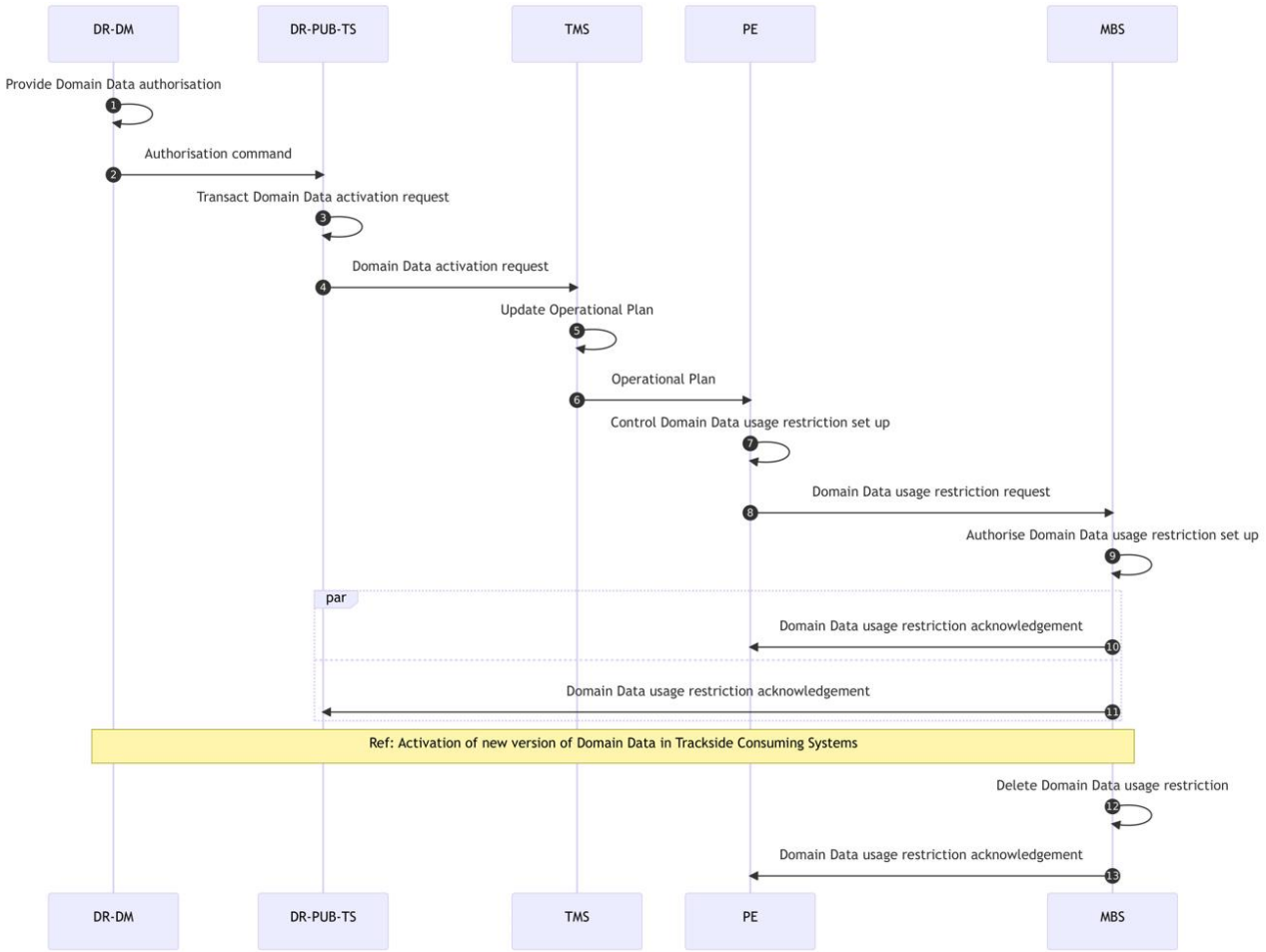

Figure 4: Scenario: Restriction of topology

[1] Conditions such as no active Movement Authority in the affected area, no train communications in the affected area, etc. are to be ensured by MBS. The conditions list is not exhaustive.

Table 4: Scenario description: Restriction of topology

Scenario Condition	Scenario Description
Nominal Case	Preconditions: DR-PUB-TS has established a communication session with each Trackside Consuming System AND DR has received 'Domain Data Acknowledgement' messages from all Trackside Consuming Systems. 1. DR Data Manager executes the function 'Provide Domain Data Authorisation'. 2. DR Data Manager sends an 'Authorisation Command' message to DR-PUB-TS. 3. DR-PUB-TS executes the function 'Transact Domain Data activation request'. 4. DR-PUB-TS sends a 'Domain Data activation request' message to TMS. 5. TMS executes the function 'Update Operational Plan'. 6. TMS sends an 'Operational Plan' message including planned Domain Data activation information to PE. 7. PE executes the function 'Control Domain Data usage restriction set up'. 8. PE sends a 'Domain Data usage restriction request' message to MBS. 9. MBS executes the function 'Authorise Domain Data usage restriction set up'. 10. Parallel: a. MBS sends a 'Domain Data usage restriction acknowledgement' message to PE. b. MBS sends a 'Domain Data usage restriction acknowledgment' message to DR-PUB-TS. 11. Ref: Scenario: 'Activation of Domain Data' 12. MBS executes the function 'Delete Domain Data usage restriction'. 13. MBS sends a 'Domain Data usage restriction acknowledgment' message to PE. Postcondition: DR-PUB-TS has received a 'Domain Data usage restriction acknowledgement' message from MBS.

4.1.1.4 Activation of new version of Domain Data in Trackside Consuming Systems

For safe railway operations, it is necessary that all involved systems use the current active version of Domain Data. This scenario process describes the second step of the activation process to activate a new Domain Data version in a safe way.

There are two preconditions for starting the actual activation process:

1. The necessary usage restriction must be active, which was confirmed by MBS with the 'Domain Data usage restriction acknowledgement' as a result of the previous scenario process described in chapter 4.1.1.3
2. The authorisation command by the DR Data Manager must have been received. The authorisation command has already been used for triggering the process described in chapter 4.1.1.3 to avoid unnecessary usage restrictions. However, DR should double-check this second precondition before starting the activation by controlling whether the authorisation command flag is set for the Domain Data version to be activated.

When all preconditions are met (usually after receiving the 'Domain Data usage restriction acknowledgement'), DR-PUB-TS sends the 'Activation command' message to the Trackside Consuming Systems. The Trackside Consuming Systems check for themselves if all preconditions for activation of a new Domain Data version are fulfilled[2]. In general, during the activation the Trackside Consuming Systems with Domain Data updates shall not perform any activities (i.e., sending requests) on the part of topology affected by a Domain Data update. When all preconditions are fulfilled, the Trackside Consuming Systems shall send an 'Activation acknowledgement' message to DR-PUB-TS. Correspondingly, they shall mark the new Domain Data version as active version and the previous version as inactive. This means the Trackside Consuming Systems shall not use the previous version anymore. However, the new active version is still marked with a flag that indicates that confirmation is missing, stating that all other Trackside Consuming Systems have also activated the new Domain Data version. Ergo, after receiving the 'Activation acknowledgement' messages from all relevant stakeholders, DR-PUB-TS sends an 'Activation commit status' message to the Trackside Consuming Systems and the DR Data Manager to indicate the successful completion of the activation process of the new Domain Data version.

[2] potentially there are necessary checks to be performed by the Trackside Consuming Systems at this step of the process, for example manual consistency checks (safe binding) for new field elements.

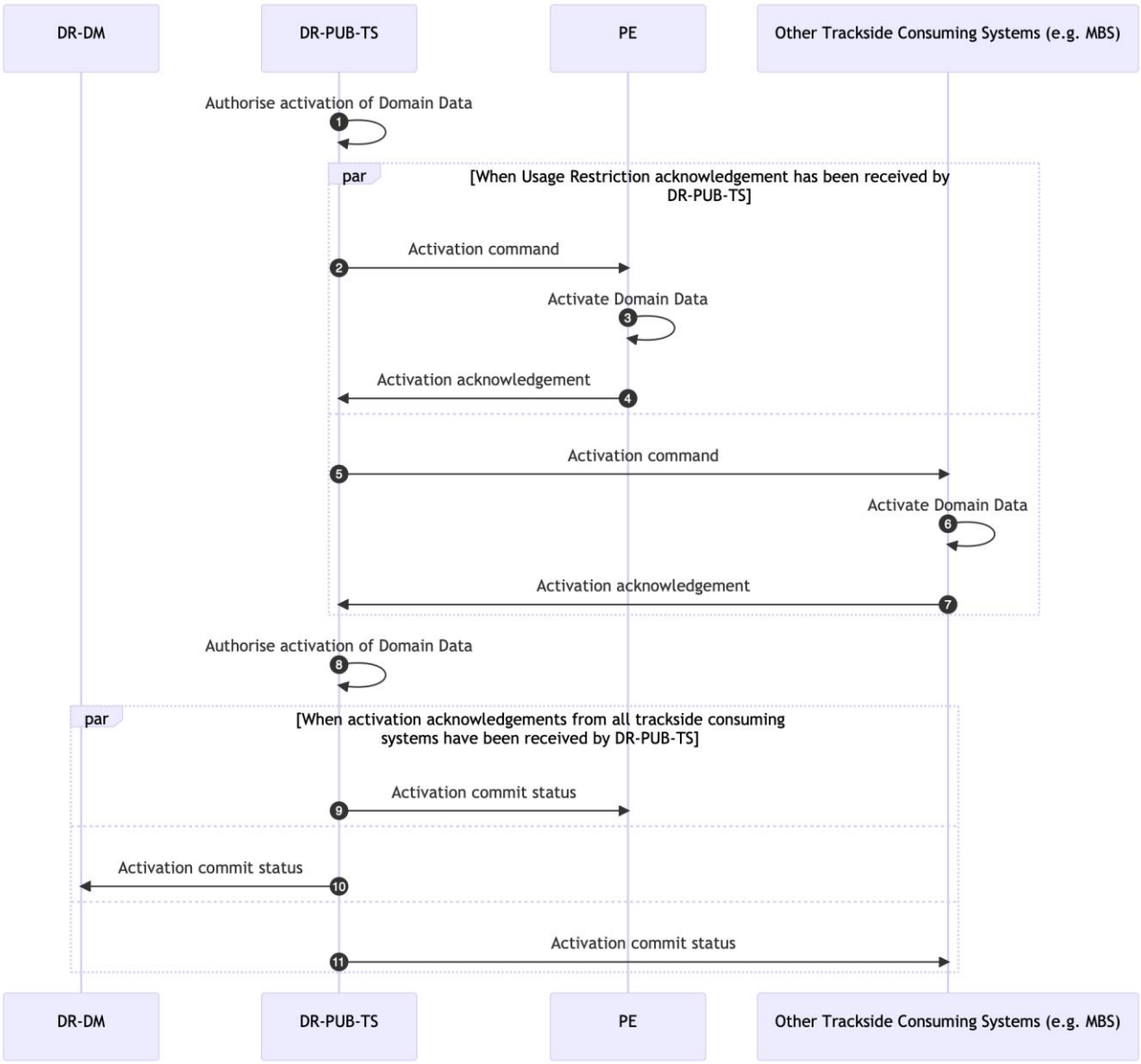

Figure 5: Scenario: Activation of new version of Domain Data

Table 5: Scenario description: Activation of new version of Domain Data

Scenario Condition	Scenario Description
Nominal Case	Precondition: DR-PUB-TS has established a communication session with each Trackside Consuming System AND The Domain Data version to be activated has the flag ‘authorisation command received’ AND DR-PUB-TS has received a ‘Domain Data usage restriction acknowledgment’ message from MBS for the Domain Data version to be activated.

1. DR-PUB-TS executes the function ‘Authorise activation of Domain Data’.
2. Parallel:
    a. DR-PUB-TS sends an ‘Activation command’ message to PE.
    b. PE executes the function ‘Activate Domain Data’.
    c. PE sends ‘Activation acknowledgement’ message to DR-PUB-TS.
    d. DR-PUB-TS sends an ‘Activation command’ message to each Trackside Consuming System.
    e. Each Trackside Consuming System executes the function ‘Activate Domain Data’.
    f. Each Trackside Consuming System sends ‘Activation acknowledgement’ message to DR-PUB-TS.
3. DR-PUB-TS executes the function ‘Authorise activation of Domain Data’.
4. Parallel:
    a. DR-PUB-TS sends an ‘Activation commit status’ message to PE.
    b. DR-PUB-TS sends an ‘Activation commit status’ message to each Trackside Consuming System.
    c. DR-PUB-TS sends an ‘Activation commit status’ message to DR Data Manager.

Postcondition: DR-PUB-TS has sent ‘Activation commit status’ messages to all Trackside Consuming Systems AND DR-PUB-TS has sent ‘Activation commit status’ message to the DR Data Manager.

4.1.2 Logical Message Definitions

Table 6: Logical Messages for interfaces to trackside consuming systems

Id No.	Message	Source	Sink	Scenario Ref.
1.	Domain Data request	Trackside Consuming Systems	DR-PUB-TS	4.1.1.1

2.	Domain Data	DR-PUB-TS	Trackside Consuming Systems	4.1.1.1, 4.1.1.2
3.	Domain Data version acknowledgement	DR-PUB-TS	Trackside Consuming Systems	4.1.1.1
4.	Domain Data acknowledgement	Trackside Consuming Systems	DR-PUB-TS	4.1.1.1, 4.1.1.2
5.	Domain Data activation time	TMS	DR-PUB-TS	4.1.1.2
6.	Authorisation command	DR-DM	DR-PUB-TS	4.1.1.3
7.	Domain Data activation request	DR-PUB-TS	TMS	4.1.1.3
8.	Operational Plan	TMS	PE	4.1.1.3
9.	Domain Data usage restriction request	PE	MBS	4.1.1.3
10.	Domain Data usage restriction acknowledgement	MBS	DR-PUB-TS and PE	4.1.1.3
11.	Activation command	DR-PUB-TS	Trackside Consuming Systems	4.1.1.4
12.	Activation acknowledgment	Trackside Consuming Systems	DR-PUB-TS	4.1.1.4
13.	Activation commit status	DR-PUB-TS	Trackside Consuming Systems and DR-DM	4.1.1.4

4.1.3 Message Structures

The body of a message is conceptually defined as following:

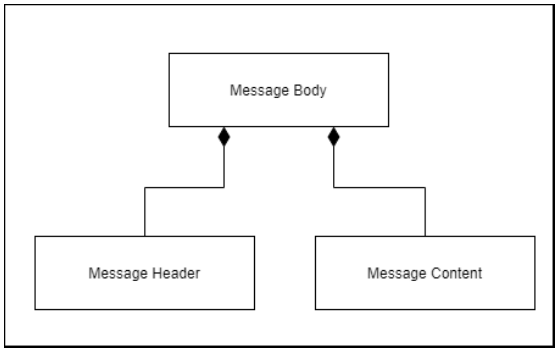

Figure 6: Message Structure

Every message consists of a header with predefined information used to identify the message. The header definition remains constant in terms of included parameters. The content of this header is defined in section 4.1.3.1. The Message Content is defined in the other subchapters of chapter 4.

4.1.3.1 Message Header

Contains required information to identify and interpret the transmitted message, i.e., Message number, Date, and Time.

Table 7: Message Header definition

Name	Type	Multiplicity	Length	Description
messageId	INT	1	8 bits	Message number, used in the header for each message, allowing the receiving systems to identify the data which follows.
timestamp	String	1	24 bits	Timestamp in ISO8601 datetime format (yyyy-MM-dd'T'HH:mm:ss.SSSZ); message timestamp generated by the ATO onboard, SS126 limitation: min: '2010-01-01T00:00:00.000Z', max: '2099-09-18T00:23:59:59.000Z'

4.1.3.2 Domain Data Request

Table 8: Domain Data Request definition

Name	Type	Multiplicity	Length	Description
header		1		Defined as per Section 4.1.3.1
consumerId	INT	1	32 bits	Unique identifier of the requesting consumer
domainDataAreaId	INT	0..1	32 bits	Unique identifier for Domain Data area e.g., to identify the required data parts to cover the complete Area of Control of the Consuming System. Alternatively, DR-PUB-TS provides pre-configured Domain Data packages that already correspond with the consumer-specific Area of Control and data needs. Therefore, the attribute is optional and must be clarified in the following development process.
domainDataVersion	INT	0..*	16 bits	Currently loaded/activated Domain Data version. In format x.y where x represents the major version and y represents the minor version. 1 byte for the major version and 1 byte for the minor one.

4.1.3.3 Domain Data

Table 9: Domain Data definition

Name	Type	Multiplicity	Length	Description
header		1		Defined as per Section 4.1.3.1
domainDataAreaId	INT	1	32 bits	See Section 4.1.3.2
domainDataVersion	INT	1	16 bits	See Section 4.1.3.2

positionInActivationQueue	INT	1	5 bits	'0' if the transmitted version is the currently active version '1' if the transmitted version is the next version to be activated '2-31' indicates how many other versions are waiting in the activation queue
Specific Domain Data Elements				Placeholder for definition of the specific objects that are part of this Domain Data Message. The element definitions shall correspond to CCS/TMS Data model.

4.1.3.4 Domain Data Version Acknowledgement

Table 10: Domain Data version acknowledgement definition

Name	Type	Multiplicity	Length	Description
header		1		Defined as per Section 4.1.3.1
domainDataAreaId	INT	1	32 bits	See Section 4.1.3.2
domainDataVersion	INT	1	16 bits	See Section 4.1.3.2

4.1.3.5 Domain Data Acknowledgement

Table 11: Domain Data acknowledgement definition

Name	Type	Multiplicity	Length	Description
header		1		Defined as per Section 4.1.3.1
domainDataAreaId	INT	1	32 bits	See Section 4.1.3.2
domainDataVersion	INT	1	16 bits	See Section 4.1.3.2
preloadingStatus	ENUM	1	1 bit	Unique status indicating the status of Domain Data preloading. 0 = loaded 1 = Unable to load

4.1.3.6 Domain Data Activation Time

Table 12: Domain Data activation time definition

Name	Type	Multiplicity	Length	Description
header		1		Defined as per Section 4.1.3.1
domainDataAreaId	INT	1	32 bits	See Section 4.1.3.2
domainDataVersion	INT	1	16 bits	See Section 4.1.3.2
activationTimestamp	String	1	24 bits	Timestamp in ISO8601 datetime format (yyyy-MM-dd'T'HH:mm:ss.SSSZ); message timestamp generated by the ATO onboard, SS126 limitation: min: '2010-01-01T00:00:00.000Z', max: '2099-09-18T00:23:59:59.000Z'

4.1.3.7 Authorisation Command

Table 13: Authorisation command definition

Name	Type	Multiplicity	Length	Description
header		1		Defined as per Section 4.1.3.1
domainDataAreaId	INT	1	32 bits	See Section 4.1.3.2
domainDataVersion	INT	1	16 bits	See Section 4.1.3.2

4.1.3.8 Domain Data Activation Request

Table 14: Domain Data activation request definition

Name	Type	Multiplicity	Length	Description
header		1		Defined as per Section 4.1.3.1
domainDataAreaId	INT	1	32 bits	See Section 4.1.3.2
domainDataVersion	INT	1	16 bits	See Section 4.1.3.2

4.1.3.9 Operational Plan

The Operational Plan to be sent between TMS and PE with Activation Time for next Domain Data version shall be based on the SCI-OP concept.

4.1.3.10 Domain Data Usage Restriction Request

The Domain Data Usage Restriction Request to be sent between PE and MBS to restrict the topology for Domain Data update shall be based on the SCI-CMD concept. Note: A reference to SCI-CMD concept is currently not available. The reference will be added in the later version of this document.

4.1.3.11 Domain Data Usage Restriction Acknowledgement

The Domain Data Usage Restriction Request to be sent between MBS – PE and MBS – DR-PUB-TS to acknowledgement the usage restriction of the topology for Domain Data update shall be based on the SCI-CMD concept.

4.1.3.12 Activation Command

Table 15: Activation command definition

Name	Type	Multiplicity	Length	Description
header		1		Defined as per Section 4.1.3.1
domainDataAreaId	INT	1	32 bits	See Section 4.1.3.2
domainDataVersion	INT	1	16 bits	See Section 4.1.3.2

4.1.3.13 Activation Acknowledgment

Table 16: Activation acknowledgement definition

Name	Type	Multiplicity	Length	Description
header		1		Defined as per Section 4.1.3.1
domainDataAreaId	INT	1	32 bits	See Section 4.1.3.2
domainDataVersion	INT	1	16 bits	See Section 4.1.3.2
activationAcknowledgement	ENUM	1	1 bit	Unique status indicating the status of activation of Domain Data. 0 = Activated 1 = Unable to activate

4.1.3.14 Activation Commit Status

Table 17: Activation commit status definition

Name	Type	Multiplicity	Length	Description
header		1		Defined as per Section 4.1.3.1
domainDataAreaId	INT	1	32 bits	See Section 4.1.3.2

domainDataVersion	INT	1	16 bits	See Section 4.1.3.2
activationCommitStatus	ENUM	1		Unique status indicating the completion of activation process. 0 = Activation Completed 1 = Activation abort.

5 IMPLEMENTATION SPECIFICATION

As required by the first implementation within WP44, this chapter provides a detailed specification of the interface I_DR-PUB-TS<>MBS from an implementation point of view. The I_DR-PUB-TS<>MBS interface uses OPC-UA methods to implement the interface. The defined specification is used for Release 2 of the WP44.2 Moving Block Demonstrator.

This implementation specification is based on SMI interface, an evolution of EULYNX, which is OPC UA based. If further developments of required quality attributes specifications and/or functional use cases are not fulfilled by OPC UA or better fulfilled by other approaches (architectures, protocols), these alternatives should be considered.

From the standpoint of interoperability, security, performance (including real-time) and data modelling capabilities OPC UA architecture and protocol is considered as option #1 for DR interfaces.

In addition, the current implementation specification does not yet completely correspond to the Functional Interface Specification defined in Chapter 4. This is due to the iterative development of the Moving Block System in Releases in WP44. The specifications will evolve as we progress in releases. Following the goals of Release 2 from WP44.2, the current implementation specification focuses on Domain Data distribution and does not cover the data aggregation, validation, compilation, authorisation, activation, and activation commit functionalities. Since the distribution is limited to one Trackside Consuming System (MBS) for this release, the consideration of consumer-specific data containers for different data needs and areas of control is neglectable.

5.1 INTERFACE SCENARIOS

5.1.1 Acquire Domain Data (pull)

The scenario “Trackside Consuming Systems acquire Domain Data on start-up”, as defined in 4.1.1.1, is partially realised for the current scope of Release 2 as follows:

The Trackside Consuming System (i.e., MBS) sends requests to the Digital Register Publisher Trackside (DR-PUB-TS) to pull required Domain Data packages. Once the consuming system completes the download and internal processing of the data, it sends an acknowledgement to the DR-PUB-TS, indicating the version of the Domain Data that has been processed.

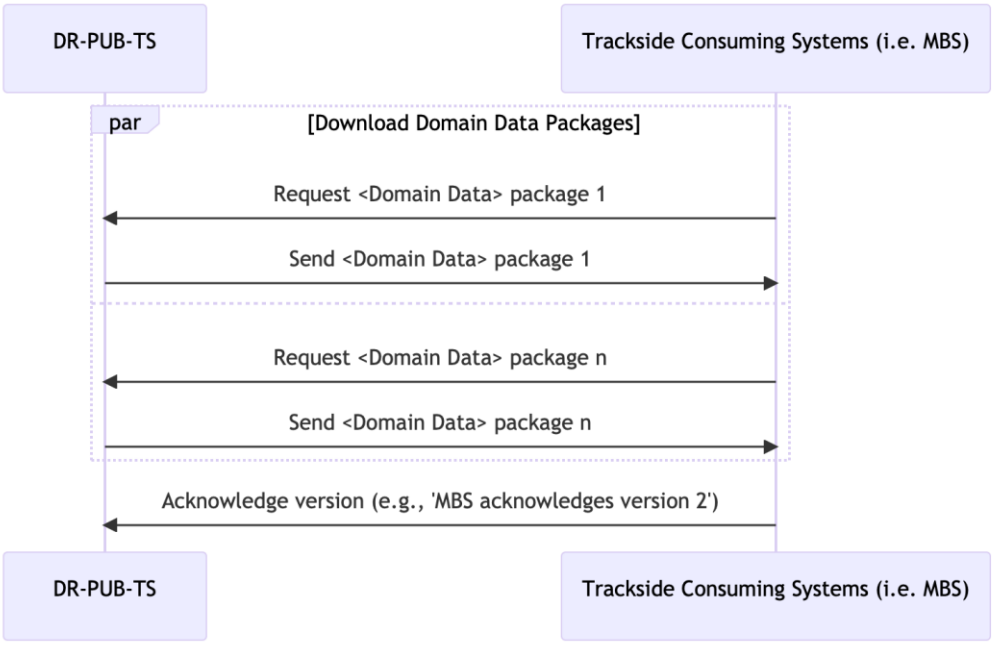

Figure 7: Acquire Domain Data

Notes:

- The specific set of Domain Data packages to download for the Consuming System (here: MBS) is to be decided.
- The parallel download of the packages is an option and can also be done in a sequential way.
- Acknowledgement should be sent by the consuming subsystem after downloading all the relevant packages AND after processing them (caching, analysing, synchronising with other services of the systems, etc.)
- The implementation currently does not cover the version check mechanism defined in chapter 4.1.1.1 yet.

The following table provides the message mapping information for the generic scenario defined in Chapter 4.1.1.1 and the implementation scenario defined here.

Table 18: Mapping information

Generic Scenario (as defined in 4.1.1.1): Messages	Implementation Scenario (as defined in 5.1.1): Queries
Domain Data request	Request <Domain Data> package 1..n
Domain Data	Send <Domain Data> package 1..n
Domain Data Acknowledgement	Acknowledge version

Domain Data Version Acknowledgement	(Not implemented yet)

The descriptions for the queries mentioned in the scenario are defined in chapter 5.3.

5.1.2 Distribute a new version of Domain Data (push)

The scenario Distribution of a new version of Domain Data to the Trackside Consuming Systems, as defined in 4.1.1.2, is partially realised for the current scope of Release 2 as follows:

Pre-Condition: The DR method 'PostNewVersionAvailable' made the latest Domain Data version on the DR-PUB-TS server available for all subscribed or requesting consumers.

It is assumed that the Trackside Consuming System (i.e., MBS) are subscribed to the data items on the DR-PUB-TS side for which it needs the latest data available updates[3]. Once the Trackside Consuming Systems (i.e., MBS) have been informed of the updates, the subscribed Trackside Consuming Systems are pushed with the data updates.

Alternatively, the data is pulled on request of the consumer, similar to the scenario described in 5.1.1.

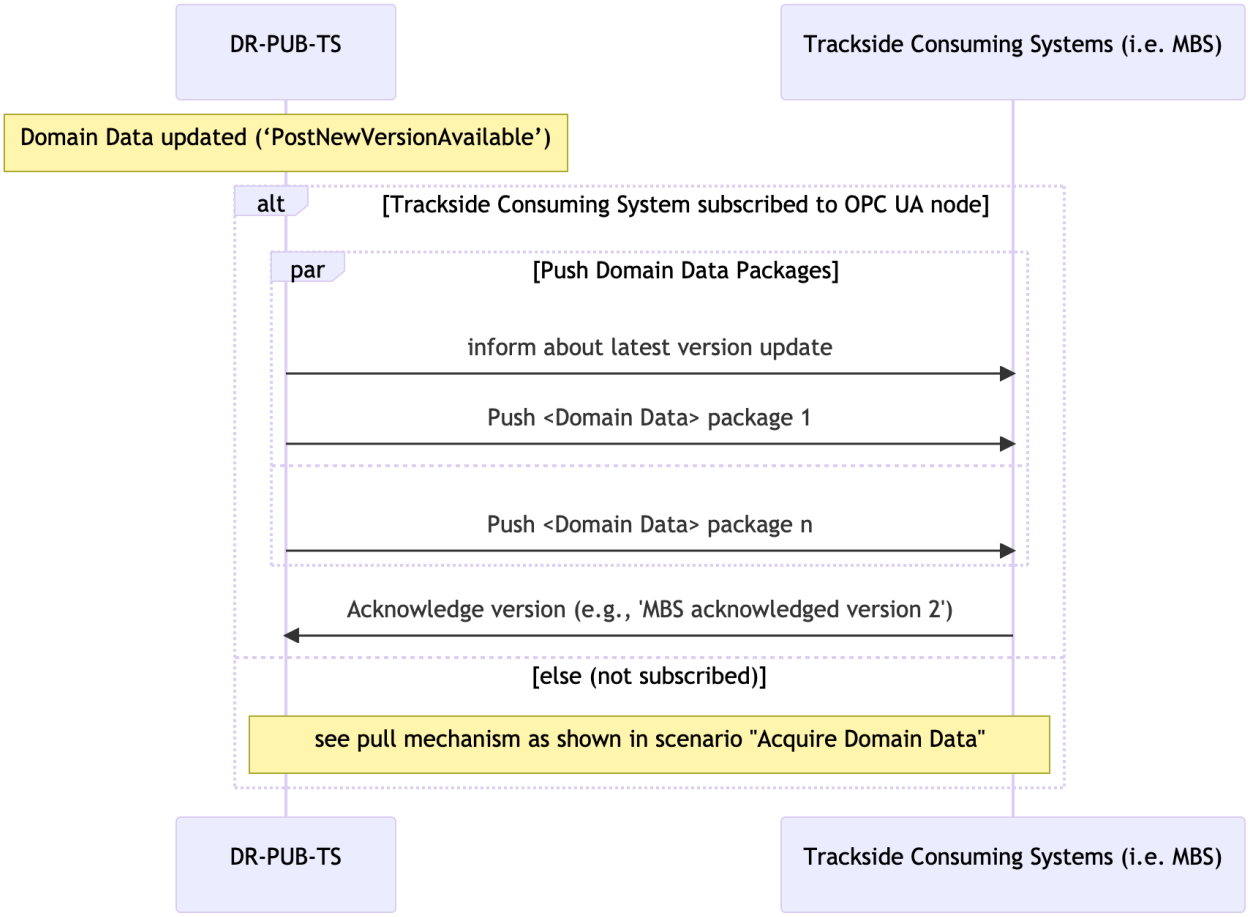

Figure 8: Distribute a new version of Domain Data

[3] Since information regarding updates is only triggered to the subscribed Trackside Consuming Systems, it is imperative that systems subscribe to the "latest" version nodes.

Notes:

- The implementation currently does not cover the pre-distribution of Domain Data to TMS, and distribution based on the time window as defined in Chapter 4.1.1.2.

The following table provides the message mapping information for the generic scenario defined in Chapter 4.1.1.2 and the implementation scenario defined here.

Table 19: Mapping information

Generic Scenario (as defined in 4.1.1.2): Messages	Implementation Scenario (as defined in 5.1.2): queries
Domain Data	Push <Domain Data> package 1..n
Domain Data acknowledgement	Acknowledge version
Domain Data activation time	(Not implemented yet)

The descriptions for the queries mentioned in the scenario are defined in Chapter 5.3.

5.2 OPC-UA INTERFACE STRUCTURE

This chapter defines the OPC-UA interface structure between DR-PUB-TS and MBS. The interface structure is currently populated with several data packages from the CCS/TMS Data model. The CCS/TMS data model from System Pillar[4] structures the data models into Domain Data packages corresponding to their use case[5], e.g., like Infrastructure (INFRA), Engineering for ETCS L2 (ENG), Train Protection (TP), ETCS SS026 packets (SS026). Ergo, the OPC-UA interface structure also follow the same data model structure to populate the nodes. The DR-PUB-TS OPC-UA server has a structured hierarchy of nodes which represent the Domain Data versions and the 'Acknowledgements' of the Consuming Systems (e.g., MBS), as shown below.

[4] Provides a link to copy of model v0.3 in Project place. For actual model in Polarion refer to this link.

[5] The package structure is preliminary and subject to change

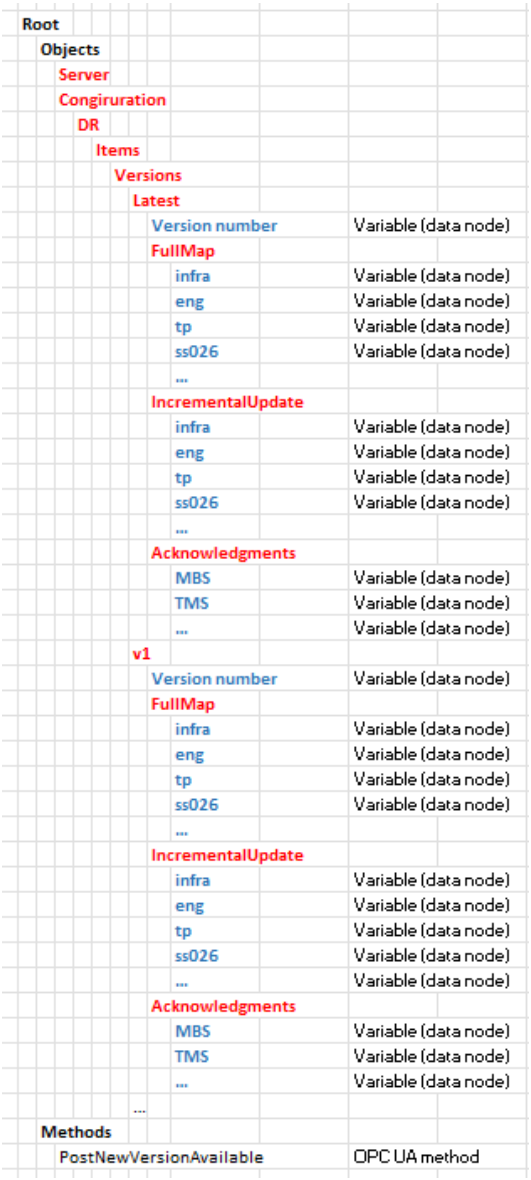

Figure 9: OPC-UA server structure

5.3 OPC-UA METHODS AND QUERIES

This chapter defines the different methods used for the OPC-UA specification between DR-PUB-TS and MBS. In addition, descriptions for queries like push, request, acknowledge, and send are provided in this chapter.

5.3.1 **Method 'PostNewVersionAvailable'**

This method adds new Domain Data versions to the DR-PUB-TS server.

Table 20: Method: Adding new Domain Data version

Method name
`PostNewVersionAvailable`

Arguments			
Argument name	**Data Type**	**Description**	**Release Number**
version	Int64	Version number to make available to consuming systems	2

Return value type

JSON object representing result (if new version is not accepted, the reason is provided in the 'message' field)

```
{
 "accepted": True,
 "message": "Ok"
}
```

Method 'PostNewVersionAvailable':

1. Check if the version already exists on the DR-PUB-TS server. If it exists, it should return the result 'false' with the message 'version already exists'.
2. Reads the complete Domain Data and/or incremental update data from the provided engineering/configuration files structured in several Domain Data packages
3. Creates a new version node under Versions if it's a new version.
4. Returns a Boolean acknowledgement upon completion.

Note: This method does not push the data to the subscribed Consuming Systems. However, as a side effect of this method all subscribed consumers to the updated version will be notified regarding this update in the following process.

5.3.2 Query 'Request <Domain Data> package'

To request a Domain Data version, MBS requests the required node as per the OPC-UA node structure defined in Chapter 5.2. The structure of the request itself depends on used programming language.

5.3.3 Query 'Send <Domain Data> package'

On request, DR-PUB-TS sends the Domain Data version to the requestor based on the requested node as per the OPC-UA structure defined in Chapter 5.2.

5.3.4 Query 'Push <Domain Data> package'

DR-PUB-TS pushes the updated Domain Data version to the subscribed consuming systems whenever the 'PostNewVersionAvailable' method has been performed. The content of the push query corresponds to the nodes/data packages subscribed by the consumer.

5.3.5 Query 'Acknowledge version'

To acknowledge a Domain Data version, MBS will modify a variable within the 'Acknowledgement' node. This modification will ascertain DR-PUB-TS that the consuming system has

downloaded/received and processed according to internal domain logic. Refer to 'Acknowledgement' node defined in Chapter 5.2.

5.4 OPC-UA INFORMATION MODEL

The OPC-UA information model used to publish Domain Data from DR to the Moving Block System shall use version 0.3 of the CCS/TMS data model.

5.5 SECURITY

5.5.1 Authentication

For the MBD Release 2, Security Mode=None.

Future considerations:

OPC UA has built-in support for authentication using X.509 certificates.

Authentication using X.509 certificates involves using digital certificates to prove the identity of a client or server. Here's a breakdown of how authentication using X.509 certificates works in the context of OPC UA:

5.5.1.1 Certificate Creation

Before authentication can occur, both the server and the client must have their own X.509 certificates. This involves:

- Certificate Authority (CA): A trusted third party that issues certificates. Both the client and server need to trust the CA that issues their certificates.
- Certificate Creation: The client and server create a Certificate Signing Request (CSR) and submit it to the CA. The CA then returns a signed X.509 certificate.
- Private Key: Along with the X.509 certificate, the client and server also have their private keys, which must remain secret.

5.5.1.2 Trust Establishment

For authentication to be successful, the client and server must trust each other's certificates. This involves:

- Certificate Exchange: Typically, the client and server exchange their certificates before authentication.
- Trust List: Both the client and server maintain a list of trusted certificates. To establish trust, they will add each other's certificates to their respective trust lists. Any certificate not in the list would be considered untrusted.

5.5.1.3 Authentication Process

When a client tries to connect to a server:

- Certificate Presentation: The client presents its X.509 certificate to the server.
- Certificate Verification: The server checks:
    - If the client's certificate is on its trust list.

- If the certificate is valid (not expired and hasn't been revoked).
- If the certificate was issued by a trusted CA.

- Private Key Usage: The client uses its private key to sign a piece of data, and the server uses the client's public key (from the certificate) to verify the signature. This proves that the client is the legitimate owner of the certificate because only they should have the corresponding private key.
- Mutual Authentication: Similarly, the server will also present its certificate to the client, and the client will perform the same verification steps. This ensures both parties are legitimate.

5.5.1.4 Session Encryption

Once authentication is successful, the client and server can establish an encrypted session. They'll typically use asymmetric encryption (from the X.509 certificates) to exchange symmetric keys, which are then used to encrypt the actual data transfer. This ensures data confidentiality and integrity.

5.5.2 **Authorisation**

Role-based authorization is used to allow specific data writing permissions and method calls based on client's role.

5.5.2.1 Roles

Role	Permissions
IM Data System	**Methods**: 'PostNewVersionAvailable' **Read+Write data permissions**: All data nodes under 'Configuration' node
Domain Data Consuming System	**Permissions:** **Read:** All data nodes under 'Versions' node **Write:** Data node under Acknowledgment corresponding to this system node (i.e., 'MBS' data node for MBS Consuming System), for all Domain Data versions

5.6 AUDIT

Keeping audit logs for the calls to the DR-PUB-TS server is not expected to affect the interface as required information (client name and network address, timestamp, and call result) can be extracted by DR-PUB-TS server itself, so no need to include it into client calls.

Specifically, OPC UA provides a way to get session information, which includes details about the client, such as its endpoint URL, session name, and user identity (when authentication is implemented). Specific mechanisms to implement logging for DR-PUB-TS are described in the OPC UA standard: https://reference.opcfoundation.org/Core/Part4/v104/docs/6.5

6 OPEN POINTS

1. Extension of the Functional Interface Specification for other logical interfaces identified in Chapter 4.
2. Implementations specifications for interfaces to other trackside and on-board consuming systems
3. Define “bad weather” scenarios for all the use cases.
4. Clarify the topology restriction approach as per the Topology Data Management concept document from WP13/44/45
5. Clarify the partial Domain Data Updates management as per the Topology Data Management concept document from WP13/44/45
6. Do the Trackside Consuming Systems need explicit activation of Domain Data when a consuming system has received a Domain Data post restart as per 4.1.1.1.
7. How should the Domain Data in DR be structured.
8. Elaborate the external requirements towards consuming systems regarding preloading (what to check?), version check to request Domain Data (for scenario 4.1.1.1), activation (what to ensure?).

7 CONCLUSIONS

The deliverable D27.2 currently focuses only on a subset of the logical interfaces (focusing on the Trackside Consuming Systems) identified in Chapter 4. As a part of the functional interface specification, the different interface scenarios and the messages exchanged between DR-PUB-TS, the Trackside Consuming Systems, and the external actors are elucidated. These scenarios depict the functional interactions along with exchanged messages between the systems.

Over the course of specification and review processes, some functional/technical aspects arose which, due to time constraints, could not be included in the deliverable D27.2 but were defined as a list of open points (see Chapter 6). These open points will be addressed in the next iterations of this deliverable in close collaboration with the specification and development work done in WP44/WP45.

REFERENCES

[1] System Pillar Transversal CCS: CCS TMS Data Model.

[2] D27.1 – Set of requirements on the Digital Register in R2DATO

[3] Communications Interface between TMS and CCS, “TMS <> CCS” (also known as “SCI-OP”)