RCA

Reference CCS Architecture

An initiative of the ERTMS users group and the EULYNX consortium

Concept: Architectural Design for Plan Execution

Preliminary issue

Document id: RCA.Doc.49

Table of Contents

Version History

0.1	20.09.2021	René Schönemann, Martin Kemkemer, Jens Wieczorek, Gabriele Löber, Rouzbeh Boloukian-Roudsari and Marcus Völcker as members of the RCA-OPE-Cluster	Preliminary version for RCA BL0 R3
0.2	30.11.2021	René Schönemann, Martin Kemkemer, Jens Wieczorek, Gabriele Löber, Rouzbeh Boloukian-Roudsari and Marcus Völcker as members of the RCA-OPE-Cluster	Release version for RCA BL0 R3

Release information

Basic document information:

RCA.Doc.49

Concept Architectural Design Plan Execution

Cenelec Phase: 1

Version: 0.3 (0.A)

RCA Baseline set: 0

Approval date: 30.11.2021

Disclaimer

This issue is a preliminary version of this document. The content of this document reflects the current ongoing specification work of RCA. Formal requirements management and change management will be introduced in future iterations. The content may be unfinished, will likely contain errors and can be changed without prior notice.

Imprint

Publisher:

RCA (an initiative of the ERTMS Users Group and EULYNX Consortium)

Support and Feedback

For feedback, or if you have trouble accessing the material, please contact rca@eulynx.eu

1. Introduction

1.1. Purpose of the document

This document is the architectural design of RCA SubSys Plan Execution (PE). It describes the functionality, functional concepts, abstract concepts, design principles and functional requirements of SubSys Plan Execution.

1.2. Maturity and Related Topics

The concept is still work in progress. Whenever it is already known that a section needs further elaboration, this is marked with red italic notes as this: *This section will be further elaborated in future releases.*

This document does not currently consider the division of the SCI-CMD into SCI-CMD, SCI-OS, SCI-PS as envisaged in the RCA Architecture Poster, because the motivation and working principle of the interface division has not yet been sufficiently described.

1.3. Related documents

The following related RCA documents provide further information and build on this concept:

- RCA System Architecture, RCA.Doc.35
- RCA Glossary, RCA.Doc.4
- RCA Domain Knowledge Specification, RCA.Doc.18
- Concept: Standard Communication Interface Operational Plan, RCA.Doc.31
- Concept: Plan Execution, RCA.Doc.47

2. Abstract concepts and Terms

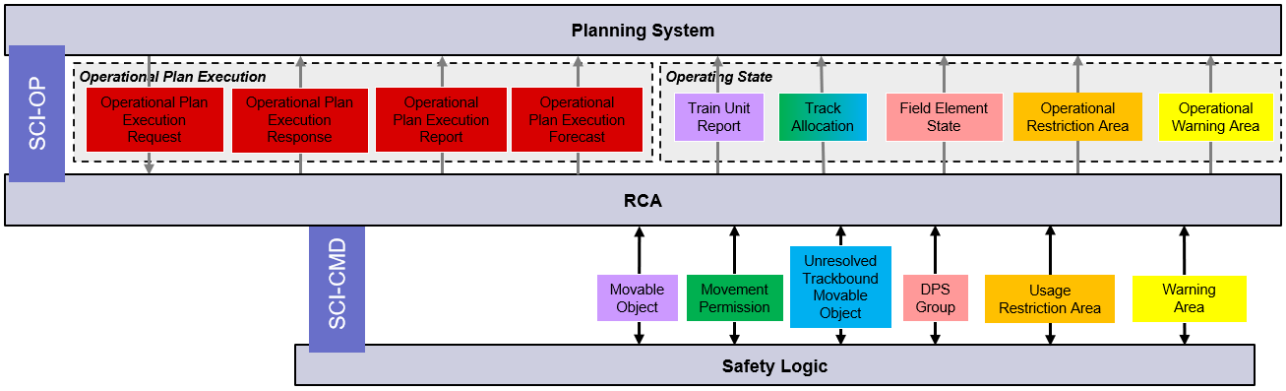


Figure 1: Overview of the main abstract concepts of SCI-OP and SCI-CMD

2.1. Abstract concepts of SCI-OP

The following table lists the abstract concepts defined for SCI-OP and for which the OPE Cluster is responsible. These abstract concepts can be found in:

- *RCA Domain Knowledge Specification, RCA.Doc.18*
- Concept: Standard Communication Interface Operational Plan, RCA.Doc.31

Abstract Concept
Operational Plan
Operational Event
Operational Plan Event Link
Operational Movement
Operational Plan Operational Movement
Operational Segment
Operational Train Unit
Operational Consist
Operational Movement Event
Track Path
Passage Event
Stop Event
Other Event
Operational Plan Operational Restriction
Operational Restriction
Operational Restriction Area
Operational Restriction Event
Start ORA Event
End ORA Event
Operational Plan Operational Warning Measure
Operational Warning Area
Operational Warning Measure
Operational Warning Measure Event
Start OWA Event
End OWA Event

Operating State
Operational Plan Execution Request
Operational Plan Execution Report
Operational Plan Execution Response
Operational Plan Execution Forecast
Field Element State
Operational Movement
Train Unit Report
Track Allocation

2.2. Abstract concepts of SCI-CMD

The following table lists the abstract concepts defined for SCI-CMD and for which the APS Cluster is responsible.
These abstract concepts can be found in:

- R*CA Domain Knowledge Specification, RCA.Doc.18*

Abstract Concept
Drive Protection Section
Drive Protection Section Group
Resolved Trackbound Movable Object
Unresolved Trackbound Movable Object
Movement Permission
Risk Buffer
Risk Path
Usage Restriction Area
Warning Area

2.3. Abstract concepts of SMI

This section will be further elaborated in future releases.
The listed abstract concepts must be clarified within Digital Map Cluster and must be added to model.

The following table lists the abstract concepts defined for SCI-SMI and for which the Digital Map Cluster is responsible.
These abstract concepts can also be found in:

- R*CA Domain Knowledge Specification, RCA.Doc.18*

Abstract Concept	**Description**

Configuration Data (Proposal)	➔ ***Abstract concept: Configuration Data***
	Configuration Data consists of:
	• Parameter (data):
	  ○ Technical parameters, e.g. identification of the system which shall be connected of the same stack and neighbouring stacks
	  ○ Configuration parameters, e.g. country specific safety rules (e.g. Requirements for Risk Path and Risk Buffer)
	• Map Data

Operational Data (Proposal)	➔ ***Abstract concept: Operational Data*** Operational Data is transmitted between production systems as part of messages, commands, requests, etc.. Operational Data can refer to or contain required parts of Map Data. In addition, Operational Data can contain temporary states or properties of infrastructure and its elements, that is overlaid on the Map Data.
Map Data	Refer to R*CA Domain Knowledge Specification, RCA.Doc.18*
Engineering Data	Refer to R*CA Domain Knowledge Specification, RCA.Doc.18*

2.4. Abstract concepts of SubSys Plan Execution

This section will be further elaborated in future releases.
The listed abstract concepts must be clarified and must be added to model

The following table lists the abstract concepts defined for SubSys Plan Execution and for which the OPE-Cluster is responsible.
These abstract concepts can also be found in:

- R*CA Domain Knowledge Specification, RCA.Doc.18*

Abstract Concept	Description
Operational Plan Event Reference	Refer to 4.1.5.1
MP Limit Marker	Refer to 4.2.5.1
Trigger	Refer to 4.2.5.2
Constraint	Refer to 4.2.5.3
Trigger Position	Refer to 4.2.7.1
Earliest Trigger Time	Refer to 4.2.7.2

2.5. Terms introduced by OPE-Cluster

This section will be further elaborated in future releases.
The listed terms must be clarified and must be added to model.

The following table lists the terms that have been defined for cross-cutting aspects of SubSys Plan Execution. These terms are currently defined by the OPE cluster but are intended to become common RCA terms.
These terms can also be found in:

- RCA Glossary, RCA.Doc.4

Term	Description
Phase	Refer to 4.3.2
System State	Refer to 4.3.2
Operational State	Refer to 4.3.2
Mode of Operation	Refer to 4.3.2
Robustness	Refer to 4.4.4.1
Fault tolerance	Refer to 4.4.4.1
High availability	Refer to 4.4.4.1
CAP Theorem	Refer to 4.4.4.1
Eventual Consistency	Refer to 4.4.4.1
Event Sourcing	Refer to 4.4.4.1

3. Functionality

3.1. Plan execution

3.1.1. Operational Plan management

3.1.1.1. Function: Process Operational Plan

Description of functionality:

- SubSys Plan Execution shall check each Operational Plan which was sent by either the Planning System via SCI-OP, or by SubSys Workbench via SWI-PE, for its feasibility and it shall reject the Operational Plan if it is not feasible.
- SubSys Plan Execution shall send an Operational Pan Execution Response (acceptance or rejection) via SCI-OP or SWI-PE for every requested Operational Plan.
- SubSys Plan Execution shall process and implement the accepted Operational Plan, by sending various requests to APS via SCI-CMD. All requests issued by SubSys Plan Execution via SCI-CMD to APS shall issue requests that respect the safety rules given by APS.
    - For the implementation of an Operational Movement:
        - SubSys Plan Execution shall control the driveability of railway network by generating 1 to many DPSGroupState requests and send them just-in-time to APS, to make the best possible utilisation of the planned capacity of the railway network.
        - SubSys Plan Execution shall control the movement of Physical Train Units by generating 1 to many MP requests and send them just-in-time to APS, to make the best possible utilisation of the planned capacity of the railway network.
    - For the implementation of an Operational Restriction:
        - SubSys Plan Execution shall implement an Operational Restriction Area on the railway network by generating a URA request and shall send it to APS in time (to be created/removed at the time specified in the Operational Plan Operational Restriction).
    - For the implementation of an Operational Warning Measure:
        - SubSys Plan Execution shall implement an Operational Warning Area on the railway network by generating a Warning Area request and shall send it to APS in time (to be created/removed at the time specified in the Operational Plan Operational Warning Measure).
- For an Operational Plan Operational Movement this means e.g. that SubSys Plan Execution shall identify the affected field elements (like points, level crossing, etc.) to set the specified Track Path and to eventually request flank protection and attempts to bring them into the required state by sending DPSGroupState requests to APS. As soon as the desired state is reached, SubSys Plan Execution shall then request the MP (incl. the Risk Paths and Risk Buffer).
- If SubSys Plan Execution requests are rejected by APS, SubSys Plan Execution shall at most (depending on the received failure code) try to correct its request or request again after a certain time. If this procedure is unsuccessful, SubSys Plan Execution shall provide the failure code received from APS to the Planning System via the Execution Report and shall finally also consider further Operational Events to be implemented as FAILED and inform the Planning System of this in form of the Execution Report.
- SubSys Plan Execution shall provide Execution Reports for accepted Operational Plan via SCI-OP or to SubSys Workbench via SWI-PE.
    - The Execution Report is used by SubSys Plan Execution to inform the Planning System about existing conflicts with reference to the Operational Events given in the Operational Plan. (further functionality and details need to be clarified)
- SubSys Plan Execution shall provide the most current, consistent, and complete Operating State (including all planned and unplanned events) to the Planning System via SCI-OP or to SubSys Workbench via SWI-PE by sending Operating State Update Events.

    - Any update event on the Operating State that occurs due to the implementation of an Operational Plan does provide a reference to the relevant Operational Plan.
    - Any update event on the Operating State that occurs unplanned and cannot be clearly assigned to an Operational Plan does not provide a reference to an Operational Plan (because no Operational Plan exists).
- SubSys Plan Execution shall have to deal with frequent updates of Operational Plans, which are already in execution by SubSys Plan Execution. An update of a whole or partial Operational Plan shall be possible under certain constraint (e.g. that the Operational Plan is not updated for Operational Events which are already being processes or completed)

3.1.1.1.1. Subfunction: Check Feasibility of Operational Plan Operational Movement

Description of functionality:

SubSys Plan Execution shall check each Operational Plan Operational Movement which was sent by either the Planning System via SCI-OP, or by SubSys Workbench via SWI-PE, for its feasibility and it shall reject the Operational Plan Operational Movement if it is not feasible.

3.1.1.1.2. Subfunction: Check Feasibility of Operational Plan Operational Restriction

Description of functionality:

SubSys Plan Execution shall check each Operational Plan Operational Restriction which was sent by either the Planning System via SCI-OP, or by SubSys Workbench via SWI-PE, for its feasibility and it shall reject the Operational Plan Operational Restriction if it is not feasible.

3.1.1.1.3. Subfunction: Check Feasibility of Operational Plan Operational Warning Measure

Description of functionality:

SubSys Plan Execution shall check each Operational Plan Operational Warning Measure which was sent by either the Planning System via SCI-OP, or by SubSys Workbench via SWI-PE, for its feasibility and it shall reject the Operational Plan Operational Warning Measure if it is not feasible.

3.2. Automatic plan execution

General notes:

- An Operational Plan is a track-precise schedule (target) with additional temporal and logical conditions based on a defined topology.
- The timely requesting of Field Element States (DPSGroupState) and requesting of MPs is based on a comparison of the target situation and the current represented operational situation (Operating State). It depends on the Operating State, e.g. position and/or speed of the respective Train Unit, the situation of other Train Units, e.g. preceding or crossing Train Units and the relevant DPS-Groups (driveability state, conditions, properties).
- SubSys Plan Execution expects the information in the Operational Plan (e.g. referenced map data or rolling stock data) always to be formally correct.
- The Planning System as well as any other RCA SubSys like SubSys Plan Execution or SubSys Safety Logic are expected to reference the identical Map Data.

3.2.1. Driveability management

3.2.1.1. Function: Observe driveability of railway network

Description of functionality:

- SubSys Plan Execution shall observe the driveability state updates of DPSs (NONE, LIMITED, FULL) provided by APS via SCI-CMD.
- SubSys Plan Execution shall observe the state updates of DPS Groups (e.g. READY, PROCESSING, FAILURE) provided by APS via SCI-CMD.

    - Example for PROCESSING: point lost its end position because the point is turning due to a DPSGroupState request.
- SubSys Plan Execution shall observe the flank protection state updates of DPS Groups (TRUE, FALSE) provided by APS via SCI-CMD.
- SubSys Plan Execution shall keep the observed driveability and flank protection state of railway network in an internal data model.
- SubSys Plan Execution shall observe the reference to the corresponding Operational Plan, if available, for each DPS Group update sent via SCI-CMD.

3.2.1.2. Function: Provide driveability of railway network

Description of functionality:

- SubSys Plan Execution shall provide the state updates of DPS Groups in form of Field Element State updates to the Planning System via SCI-OP and to SubSys Workbench via SWI-PE.
- SubSys Plan Execution shall provide the reference to the corresponding Operational Plan, if available, for each Field Element State update sent via SCI-OP.

3.2.1.3. Function: Control driveability of railway network

Description of functionality:

- SubSys Plan Execution shall control all field elements giving or ensuring driveability of the railway network by sending DPSGroupState requests to APS via SCI-CMD. All the following field elements (list is not exhaustive) shall, for example, be controlled by DPSGroupState requests: point, single- or double-slip points, level crossing, derailer, gates, lifting ramps or lift bridges.
- To minimize the allocation of the available infrastructure, SubSys Plan Execution shall send DPSGroupState requests to APS via SCI-CMD at the optimal point in time considering any system times necessary for reaching the demanded state.

3.2.1.3.1. Subfunction: Calculation of Trigger Points for driveability requests

Description of functionality:

- SubSys Plan Execution shall calculate the optimal time to trigger the DPSGroupState requests at APS in order to implement the given Operational Plan with the least possible utilisation of the planned capacity.
- SubSys Plan Execution shall calculate the optimal trigger points requests individually for each execution request of an Operational Plan (initial or update).
- SubSys Plan Execution shall at least consider the following aspects when calculating the trigger points (list not exhaustive).
    - Required setting times for DPS Groups (field elements).
    - Position, Speed and Movement Permission of relevant Train Unit

3.2.2. Operational Movement management

3.2.2.1. Function: Observe Operational Movement of Physical Train Units

Description of functionality:

- SubSys Plan Execution shall observe the updates of Resolved Trackbound Movable Objects provided by APS via SCI-CMD.
- SubSys Plan Execution shall keep the observed Resolved Trackbound Movable Objects in an internal data model.
- SubSys Plan Execution shall observe the reference to the corresponding Operational Plan, if available, for each Resolved Trackbound Movable Object update sent via SCI-CMD.
- SubSys Plan Execution shall observe the updates of Unresolved Trackbound Movable Objects provided by APS via SCI-CMD.
- SubSys Plan Execution shall observe the updates of MPs provided by APS via SCI-CMD.

- SubSys Plan Execution shall keep the observed MPs in an internal data model.
- SubSys Plan Execution shall observe the reference to the corresponding Operational Plan, if available, for each MP update sent via SCI-CMD.

3.2.2.2. Function: Provide Operational Movement of Physical Train Units

Description of functionality:

- SubSys Plan Execution shall provide updates of Resolved Trackbound Movable Objects in form of Train Unit Report updates to the Planning System via SCI-OP and to SubSys Workbench via SWI-PE.
- SubSys Plan Execution shall provide updates of MP in form of Track Allocation updates to the Planning System via SCI-OP and to SubSys Workbench via SWI-PE.
- SubSys Plan Execution shall provide updates of Unresolved Trackbound Movable Objects in form of Track Allocation updates to the Planning System via SCI-OP and to SubSys Workbench via SWI-PE.
- SubSys Plan Execution shall provide the reference to the applicable Operational Plan, if available, for each Track Allocation update sent via the SCI-OP.
- SubSys Plan Execution shall provide the reference to the applicable Operational Plan, if available, for each Train Unit Report update sent via the SCI-OP.

3.2.2.3. Function: Control Operational Movement of Physical Train Units

Description of functionality:

- SubSys Plan Execution shall control the movement of Physical Train Units in the Area of Control by allocating Track Paths in form of MPs.
- SubSys Plan Execution shall send timely requests to APS to create, to lengthen, to shorten at front (exceptionally) or to adjust the speed of a MP. Timely means that the requests are sent early enough so that a Physical Train Unit does not run into its braking curve.
- SubSys Plan Execution shall automatically calculate the content and trigger point for any request sent to APS, based on the information given in the Operational Plans.
- SubSys Plan Execution shall request a MP only in the Area of Control. For transition of Physical Train Units to or from other Areas of Control, SubSys Plan Execution shall have to grant or reject handover with adjacent SubSys Plan Execution. It is assumed that APS manages all the underlying additional transition logic for SubSys Plan Execution.
- In principle, SubSys Plan Execution shall only request extensions (and reductions) of a MP without gaps, without overlaps and in the correct chronological and geographical order. However, SubSys Plan Execution can also request MPs that overlap with another MP e.g. for joining or splitting. The overlap of MPs is only allowed under certain conditions (e.g. drive on-sight). This MP overlap is in general possible at any location in the Area of Control.
- SubSys Plan Execution shall be able to adapt a MP speed profile for a higher speed or for a lower speed (shall be granted by driver, resp. ATO).
- SubSys Plan Execution shall be able to request MP even though the driveability of the Track Path is limited e.g. due to limited state of a DPS (e.g. barrier opened). The MP shall then be requested under certain conditions, which can be defined by country-specific rules like lower speed and drive on-sight.
- SubSys Plan Execution shall calculate the optimal extent of an MP and the optimal trigger time for the MP request and therefore ensures the closest possible Physical Train Unit sequence.
- SubSys Plan Execution is responsible to request the shortening of the MP at front (e.g. in case that a Physical Train Unit was reported to be ready for departure, the MP was extended, but the Physical Train Unit could not depart for some reason) or extension of the MP at front (regular use case).
- APS is responsible for the shorting of the MP at the rear.
- SubSys Plan Execution shall calculate the maximum speed profile of an MP based on the information in the Operational Plan, the Operating State and the properties of the underlying topology.
- SubSys Plan Execution shall create a work list for MP requests, as APS does not queue future MP requests from SubSys Plan Execution but expects SubSys Plan Execution to send the request individually at the right time.

3.2.2.3.1. Subfunction: Calculation of MP

Description of functionality:

- SubSys Plan Execution shall continuously determine the optimal characteristics of an MP. Therefore, SubSys Plan Execution shall at least consider the following aspects (list is not exhaustive):
  - The MP shall be always long enough to allow the Physical Train Unit to run continuously.
  - The MP shall be just long (front and rear) enough to reserve as less railway capacity as possible.
  - The Operational Events (e.g. scheduled departure time, scheduled stop position on Track Path) of the Operational Plan shall be regarded.
  - The properties of railway network used by the Track Path (e.g. permissible maximum speed of field elements, non-stopping areas).
  - The properties and position of the Resolved Trackbound Movable Object.
  - The restrictions of current APS Operating State (e.g. other MPs, URAs, not suitable state of DPS-Groups, etc.) on the Track Path.

3.2.2.3.2. Subfunction: Calculation of Risk Buffer

Description of functionality:

- For the creation of an MP the calculation of a Risk Buffer is necessary.
- The required Risk Buffer shall be determined dynamically and individually for each MP request.
- SubSys Plan Execution shall calculate the Risk Buffer based on same rules as SubSys Safety Logic.
- SubSys Plan Execution shall consider the maximum speed profile of the MP and the properties of the railway network allocated by the MP to calculate the Risk Buffer.
- An overlap of a Risk Buffer with another MP (incl. Risk Buffer), Resolved Trackbound Movable Object or an Unresolved Trackbound Movable Object is only allowed under certain conditions (e.g. movement modes of the Train Control System: limited speed, drive on-sight).

3.2.2.3.3. Subfunction: Calculation of Risk Paths

Description of functionality:

- SubSys Plan Execution shall determine the necessary type of flank protection (track-enforced flank protection or light flank protection) before requesting a MP.
- SubSys Plan Execution shall determine the necessary type of flank protection based on the maximum speed of the MP and the properties of the railway network allocated by the MP.
- In case that a track enforced flank protection is required, SubSys Plan Execution shall first send the necessary DPS flank protection requests to APS in order to protect the MP (e.g. to protect an MP by a point in diverging position or by a derailer in position “on”). Then, SubSys Plan Execution shall request an MP whose Risk Paths cover the Track Paths up to the protecting elements so that their condition cannot be changed by another request and the Track Paths in between cannot be used by another MP. Since the risk paths shall be determined by SubSys Plan Execution, any possible safe solution ensuring track enforced flank protection can be chosen by SubSys Plan Execution.
- In case of light flank protection is sufficient, SubSys Plan Execution shall request an MP with limited speed and Risk Paths that do not cover the protecting field elements, but only the relevant Allocation Sections, so they cannot be used by another MP.

3.2.2.3.4. Subfunction: Calculation of Trigger Points for MP requests

Description of functionality:

- SubSys Plan Execution shall calculate the optimal time to trigger the MP requests at APS to ensure the closest possible Physical Train Unit sequence and therefore make the best possible use of the planned capacity of the existing rail network.
- SubSys Plan Execution shall calculate the optimal trigger points individually for each execution request of an Operational Plan (initial or update).

- SubSys Plan Execution shall consider the following aspects when calculating the trigger points (list not exhaustive):
    - Points in time of Operational Events (e.g. scheduled departure time) in the Operational Plan
    - Provided position of Physical Train Unit
    - Point in time when driveability of required Track Path will be given.

3.2.3. Operational Restriction management

3.2.3.1. Function: Observe Usage Restriction Areas on railway network

Description of functionality:

- SubSys Plan Execution shall observe the updates of URA (planned and unplanned) provided by APS via SCI-CMD.
- SubSys Plan Execution shall keep the observed URAs on railway network in an internal data model.
- SubSys Plan Execution shall calculate the reference to the corresponding Operational Plan, if available, for each Usage restriction Area update sent via SCI-CMD.

3.2.3.2. Function: Provide Operational Restriction Areas on railway network

Description of functionality:

- SubSys Plan Execution shall provide URA updates in form of Operational Restriction Area updates to the Planning System via SCI-OP and to SubSys Workbench via SWI-PE.
- SubSys Plan Execution shall provide the reference to the Operational Plan, if available, for each Operational Restriction Area update sent via SCI-OP.

3.2.3.3. Function: Control Operational Restriction Areas on railway network

Description of functionality:

- Any planned Operational Restriction Area (e.g. for closures or speed restrictions) is scheduled by the Planning System and sent to SubSys Plan Execution by means of an Operational Plan. Based on the Operational Plan, SubSys Plan Execution shall request one to many URAs from APS according to the given Operational Events.
    - SubSys Plan Execution shall be able to request the temporal beginning and end of a planned URA separately at the right time (before an URA is removed by SubSys Safety Logic the removal has to be confirmed by an Authorized Trackside Person).
    - SubSys Plan Execution shall be able to request URAs for any Track Edge Section in the Area of Control.
        - The extend of an URA is always defined by the referenced Track Edge Section(s).
        - The direction of the Track Edge Sections of the URA may be bidirectional or unidirectional to represent Operational Restrictions which are valid in one or both directions of travel.
        - URAs may overlap each other.
- An Operational Restriction Area (ORA) may have dependency (e.g. all or non-creation/deletion) to other ORAs or Operational Warning Areas (OWAs) and may therefore be linked.

3.2.4. Operational Warning Measure management

3.2.4.1. Function: Observe Warning Areas on railway network

Description of functionality:

- SubSys Plan Execution shall observe the updates of Warning Areas (WA) provided by APS via SCI-CMD.
- SubSys Plan Execution shall keep the observed WAs on railway network in an internal data model.
- SubSys Plan Execution shall calculate the reference to the corresponding Operational Plan, if available, for each WA update sent via SCI-CMD.

3.2.4.2. Function: Provide Operational Warning Areas on railway network

Description of functionality:

- SubSys Plan Execution shall provide WA updates in form of OWA updates to the Planning System via SCI-OP and to SubSys Workbench via SWI-PE.
- SubSys Plan Execution shall provide the reference to the Operational Plan, if available, for each OWA update sent via SCI-OP.

3.2.4.3. Function: Control Operational Warning Areas on railway network

Description of functionality:

- Any OWA is planned by the Planning System and sent to SubSys Plan Execution in an Operational Plan. Based on the Operational Plan, SubSys Plan Execution shall request one to many WAs from APS according to the given Operational Events.
    - SubSys Plan Execution shall be able to request the temporal beginning and the end of a planned WA.
    - SubSys Plan Execution shall be able to request WAs for any Track Edge Sections in the Area of Control.
    - The extend of a WA is defined by a list of referenced Track Edge Section(s)
    - The direction of validity of a WA may be bidirectional or unidirectional.
    - WAs may overlap each other.
- An OWA may have a dependency (e.g. all or non-creation/deletion) to other OWAs or ORAs and may therefore be linked.
- The operational activation of an OWA is done by an Authorized Trackside Personal.

3.3. Manual plan execution

3.3.1. Function: Manual plan execution

Description of functionality:

- SubSys Plan Execution shall implement an interface (SWI-PE) to SubSys Workbench for manual interaction of Railway Operators with SubSys Plan Execution.
- The functionality which shall be provided by SubSys Plan Execution to SubSys Workbench is used for manual activities which are not planned by the Planning System. This includes the following use cases:
    - Circulate and lubricate switch.
      Note: Assignment of functionality requires a detailed system and risk analysis.
    - Perform mapping of an identified but not localized Physical Train Unit to an existing Unresolved Trackbound Movable Object (e.g. revised train, locomotive without cold-movement-detection or cold-movement-detection = true) and move with a restricted Movement Permission to a defined position where Physical Train Unit can be safely localized (e.g. at next balise).
    - Perform an emergency stop or restricted operation for all Physical Train Units in a defined area (e.g. due to tunnel fire, persons near the track).
      Note: Assignment of functionality requires a detailed system and risk analysis.
    - Performing an emergency stop for a specific Physical Train Unit.
      Note: Assignment of functionality requires a detailed system and risk analysis.
    - Operational movement into and out-off industrial areas.
- The functionalities which shall be provided by SubSys Plan Execution to SubSys Workbench is used for "Fallback level for the Planning System" which includes the following use case:
    - Simple manual operation to transport passengers to the nearest station
- SubSys Workbench shall process the Operating State and Execution Report sent by SubSys Plan Execution via SWI-PE and display it on a graphical user interface.
- SubSys Workbench shall process user inputs and send manually created or adapted Operational Plans for execution to SubSys Plan Execution via SWI-PE.

3.4. Device and Configuration Management (DCM)

3.4.1. Function: Import Configuration Data

Description of functionality:

- SubSys Plan Execution shall import all required configuration data from the SubSys Device & Config Management via the SMI-PE.
  Configuration Data consist of the following contents:
  - Technical parameters
    - E.g. identification of the system which shall be connected (the Planning System, SubSys Plan Execution, APS (SubSys Safety Logic/ SubSys Object Aggregation) of the same stack and neighbouring stacks
  - Configuration parameters
    - E.g. country specific safety rules (e.g. Requirements for Risk Path and Risk Buffer)
    - E.g. geographical extent of the Area of Control
  - Map Data
    - Topology data describing the geographical and logical aspects of the infrastructure of the rail network. SubSys Plan Execution shall require Map Data for the Area of Control and the Area of Supervision.
- SubSys Plan Execution shall check that the importing process was successful
- SubSys Plan Execution shall acknowledge the successful importing of configuration data to the SubSys Device & Config Management via the SMI (Standard Maintenance Interface)

3.4.2. Function: Activate Map Data

Description of functionality:

- SubSys Plan Execution shall support activation of imported Map Data.

3.5. Diagnostics and Monitoring (DM)

3.5.1. Function: Send diagnostics data

Description of functionality:

- SubSys Plan Execution shall register and log system data and sends it to SubSys Diagnostics & Monitoring via SDI (Standard Diagnostic and Monitoring Interface)

3.6. Authentication und Authorisation

3.6.1. Function: Use authentication and authorisation services

Description of functionality:

- SubSys Plan Execution shall use services for authenticating and authorising human user and SubSys, depending on the RCA security architecture to be implemented and divided with several role definitions.

3.7. Robustness

This section will be further elaborated in future releases.

3.7.1. Function: Support different System States, Operational States and Modes of Operation

This section will be further elaborated in future releases.

Description of functionality:

- SubSys Plan Execution shall support different System States, Operational States and Modes of Operation in order to support different operational use cases and system capabilities.
- SubSys Plan Execution shall provide updates about its System State and Operational State.

- SubSys Plan Execution shall request the System State and Operational State of RCA SubSys.

3.7.2. Function: Determine System State and Operational State

This section will be further elaborated in future releases.

Description of functionality:

- SubSys Plan Execution shall determine its own System State and Operational State due the following criteria:
  - The state of its internal data model
  - Detected system failures
  - Detected limitations in functionality
  - Detected limitations due to non-functional aspects
  - The System State and Operational State of neighbouring systems

3.7.3. Function: Provide and receive System State and Operational State

This section will be further elaborated in future releases.

Description of functionality:

- SubSys Plan Execution shall provide updates about its System State and Operational State to the connected systems.
- SubSys Plan Execution shall receive and process updates about the System State and Operational State of the connected systems.

3.7.4. Function: Calculate and check internal data model

This section will be further elaborated in future releases.

Description of functionality:

- SubSys Plan Execution shall calculate its internal data model based on the Configuration Data and Operational Data received via its interfaces
- SubSys Plan Execution shall check the consistency of the calculated internal data model.

3.7.5. Function: Synchronise the state of Configuration Data and Operational Data

This section will be further elaborated in future releases.

Description of functionality:

- SubSys Plan Execution shall be able to synchronise the Operational- and Configuration Data send and received at its interfaces in case of an inconsistent internal data model.
  This includes:
  - Configuration Data
    - Synchronisation send and received via SMI
  - Operational Data
    - Synchronisation send and received via SCI-OP
    - Synchronisation send and received via SWI-PE
    - Synchronisation send and received via SCI-CMD
    - Synchronisation send and received via SHI-PE

4. Functional concepts

4.1. Functional concept: Requesting of Operational Plans

This section will be further elaborated in future releases.

4.1.1. Objectives and System Requirements

The applicable objectives and system requirements can be found in *Concept: Plan Execution, RCA.Doc.47*, chapter: *Objectives and System Requirements*.

4.1.2. Introduction

This concept focuses on the functioning of the SCI-OP and SubSys Plan Execution and SubSys ATO Execution, when requesting the execution of new or updated Operational Plans.

4.1.3. Motivation

The Planning System sends an Operational Plan Execution Request for one or more of the following reasons:

- An Operational Movement, Operational Restriction or Operational Warning Measure has been planned and shall be executed.
- The Operational Plan was updated because the scheduled times for an Operational Movement, Operational Restriction or Operational Warning Measure have been replanned.
- The Operational Plan was updated because the execution sequence between different Operational Movements, Operational Restrictions or Operational Warning Measures has been replanned.
- The Operational Plan was updated because the Operational Restriction Area or Operational Warning Area has been replanned (extent, updated Operational Restrictions, updated Operational Warning Measures).
- The Operational Plan was updated because Operational Events have been replanned (e.g. number of Operational Events, type of Operational Events).
- The Operational Plan was updated because the Track Path to be used for an Operational Movement has been replanned.
- The Operational Plan was updated because the Operational Train Unit and/or Operational Consists have been replanned.
- The Operational Plan was updated because of conflicts detected by the Planning System in the current Operating State or because of Execution Failures or Execution Warnings reported by SubSys Plan Execution or SubSys ATO Execution.
- The Operational Plan was updated because the former version of the Operational Plan was rejected by SubSys Plan Execution and/or SubSys ATO Execution.

The main motivation for sending a new or updated Operational Plan via an Operational_plan_execution_request is that the Planning System optimises all Operational Movements, Operational Restrictions and Operational Warning Measures in order to achieve best KPIs (Key performance indicators) e.g. the lowest possible overall delay minutes, shortest travel times, maximum percentage of kept connections, lowest energy consumption etc. in the entire AoC while meeting the given operational processes and safety conditions.

SubSys Plan Execution, on the other hand, will execute all received and accepted Operational Plans as long as no updates are provided by the Planning System. It will strictly adhere to the given Operational Plans and will not optimise them, even if the Operational Events cannot be executed in the given time window and are therefore delayed.

4.1.4. Design Principles

Design Principle

The Planning System shall identify the need to update an Operational Plan (e.g. due to optimization, existing conflicts within one or between different Operational Plans, operational processes or safety conditions).

Design Principle

SubSys Plan Execution shall provide the operational situation, the completion of Operational Events and, if applicable, failures or warnings regarding the Operational Events via SCI-OP to the Planning System and via SWI-PE to SubSys Workbench.

Design Principle

SubSys ATO Execution shall provide information about the state of a Train Unit, the completion of Operational Events and, if applicable, failures or warnings regarding the Operational Events and forecasts regarding upcoming Operational Movement Events via SCI-OP to the Planning System.

4.1.5. Operational Plan Operation Movement

4.1.5.1. Abstract Concept

➔ ***Abstract concept: Operational Plan Event Reference***
An Operational Plan Event Reference is a marker on one Operational Movement Event of an Operational Plan Operational Movement. It serves as a reference between two subsequent versions of the same Operational Plan Operational Movement to assure gap-free and unambiguous transition to the new version of an Operational Plan Operational Movement.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

➔ ***Abstract Concept: Execution Failure***
An Execution Failure describes that at least one step for the execution of an Operational Event cannot be successfully implemented by the executing RCA SubSys. The Execution Failure, detected by SubSys Plan Execution or SubSys ATO Execution during the execution of an Operational Event, cannot be resolved by the corresponding SubSys and is therefore reported as an Operational Plan Execution Report via SCI-OP to the Planning System.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

➔ ***Abstract Concept: Execution Warning***
An Execution Warning describes that at least one step for the execution of an Operational Event cannot be implemented as planned by the executing RCA SubSys. The Execution Warning, detected by SubSys Plan Execution or SubSys ATO Execution during the execution of an Operational Event, can be neglected by the corresponding SubSys but is reported as an Operational Plan Execution Report via SCI-OP to the Planning System.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

4.1.5.2. Functional Requirements

This section will be further elaborated in future releases.

Function	Subfunction	Functional Requirement
Process Operational Plan	Check feasibility of Operational Plan Operational Movement	The Operational Plan Execution Request for an Operational Plan Operational Movement shall be rejected by SubSys Plan Execution: • if there is an Operational Movement Event which is not positioned on the Track Path.
Process Operational Plan	Check feasibility of Operational Plan Operational Movement	The Operational Plan Execution Request for an Operational Plan Operational Movement shall be rejected by SubSys Plan Execution: • if the Track Path is not a gap-free and track-specific route on the railway network.

Process Operational Plan	Check feasibility of Operational Plan Operational Movement	The Operational Plan Execution Request for an Operational Plan Operational Movement shall be rejected by SubSys Plan Execution: • if the sequence of Operational Movements Events is not navigable for a Train Unit on basis of the defined Track Path and the defined Stop Activities
Process Operational Plan	Check feasibility of Operational Plan Operational Movement	The Operational Plan Execution Request for an Operational Plan Operational Movement shall be rejected by SubSys Plan Execution: • if the Operational Plan Operational Movement references any Map Data Element(s) that are not included in the Map Data Version currently activated in the RCA system. • if the Operational Plan Operational Movement references any Map Data Element(s) that are included in the Map Data Version currently preloaded (List of differing Map Data elements) in the RCA system.
Process Operational Plan	Check feasibility of Operational Plan Operational Movement	The Operational Plan Execution Request for an **update** of an already requested Operational Plan Operational Movement shall be rejected by SubSys Plan Execution: • if the Train Unit is not (also not partly) located on the Track Path, but the route between the position of the front end of the Train Unit (in direction of travel) and the position of the first Operational Movement Event is not distinct • if the Train Unit is partly located on the Track Path, but the front end of the Train Unit (in direction of travel) is not located on the Track Path
Process Operational Plan	Check feasibility of Operational Plan Operational Movement	The Operational Plan Execution Request for an **update** of an already requested Operational Plan Operational Movement shall be rejected by SubSys Plan Execution: • if the update of the Operational Plan Operational Movement does not contain at least one Operational Movement Event of the already requested Operational Plan Operational Movement as an Operational Plan Event Reference.
Process Operational Plan	-	Any Operational Event which was already completed by SubSys Plan Execution shall not be processed by SubSys Plan Execution anymore. Additional information: • An Operational Movement Event is completed for SubSys Plan Execution, if a Movement Permission was granted by SubSys Safety Logic which allocates the position of the Operational Movement Event, allowing the departure, arrival, or passage of the Train Unit from/at this Operational Movement Event, and if the Train Unit has started moving within this Movement Permission.
Process Operational Plan	Check feasibility of Operational Plan Operational Movement	The feasibility check of the Operational Plan sent with Operational Plan Execution Request is done once and in timely manner after receipt of the Operational Plan Execution Request via SCI-OP by SubSys Plan Execution. As a result of the feasibility check, an Operational Plan Execution Reponse is sent by SubSys Plan Execution via SCI-OP. If conflicts arise between the time of receipt of the request and the start of execution of the Operational Plan, these are reported by an Operational Plan Execution Report by SubSys Plan Execution at the start of execution.

Process Operational Plan	-	At the time when SubSys Plan Execution starts executing an Operational Plan Operational Movement, a conflict for the first Operational Movement Event shall be reported by SubSys Plan Execution: • if the Train Unit is not (also not partly) located on the Track Path, but the route between the position of the front end of the Train Unit (in direction of travel) and the position of the first Operational Movement Event is not distinct • if the Train Unit is partly located on the Track Path, but the front end of the Train Unit (in direction of travel) is not located on the Track Path

4.1.5.3. Basic Scenarios

Hereinafter several scenarios for updating of an Operational Plan Operational Movement are described and illustrated. Introductory the procedure for sending the initial Operational Plan Operational Movement without the need for an update is defined. Alle subsequent scenarios are derived from this first scenario “Initial Plan”. Descriptions and illustrations are directly linked to scenario “Initial Plan”.

4.1.5.3.1. Scenario: Initial Plan (acceptance)

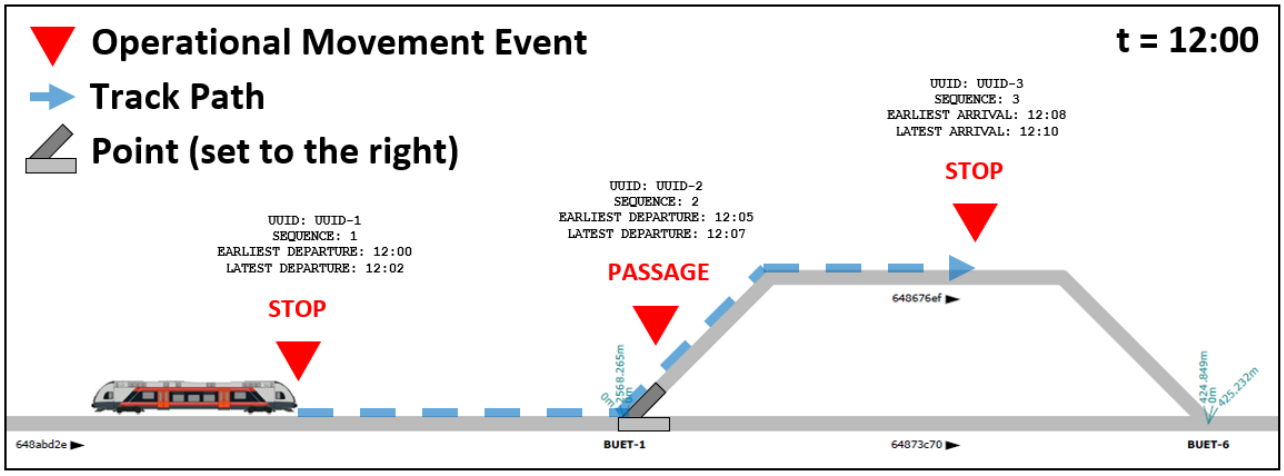


Figure 2: Schematic example of a feasible Operational Plan Operational Movement

Description of Scenario

The Scenario illustrates an initially requested Operational Plan Operational Movement which needs to be executed by SubSys Plan Execution and SubSys ATO Execution.

Feasibility Check

The shown example of an Operational Plan Operational Movement will be accepted by SubSys Plan Execution if initially requested via SCI-OP, because all the above defined feasibility checks (see 4.1.5.2) are all fulfilled.

Sequence Diagram

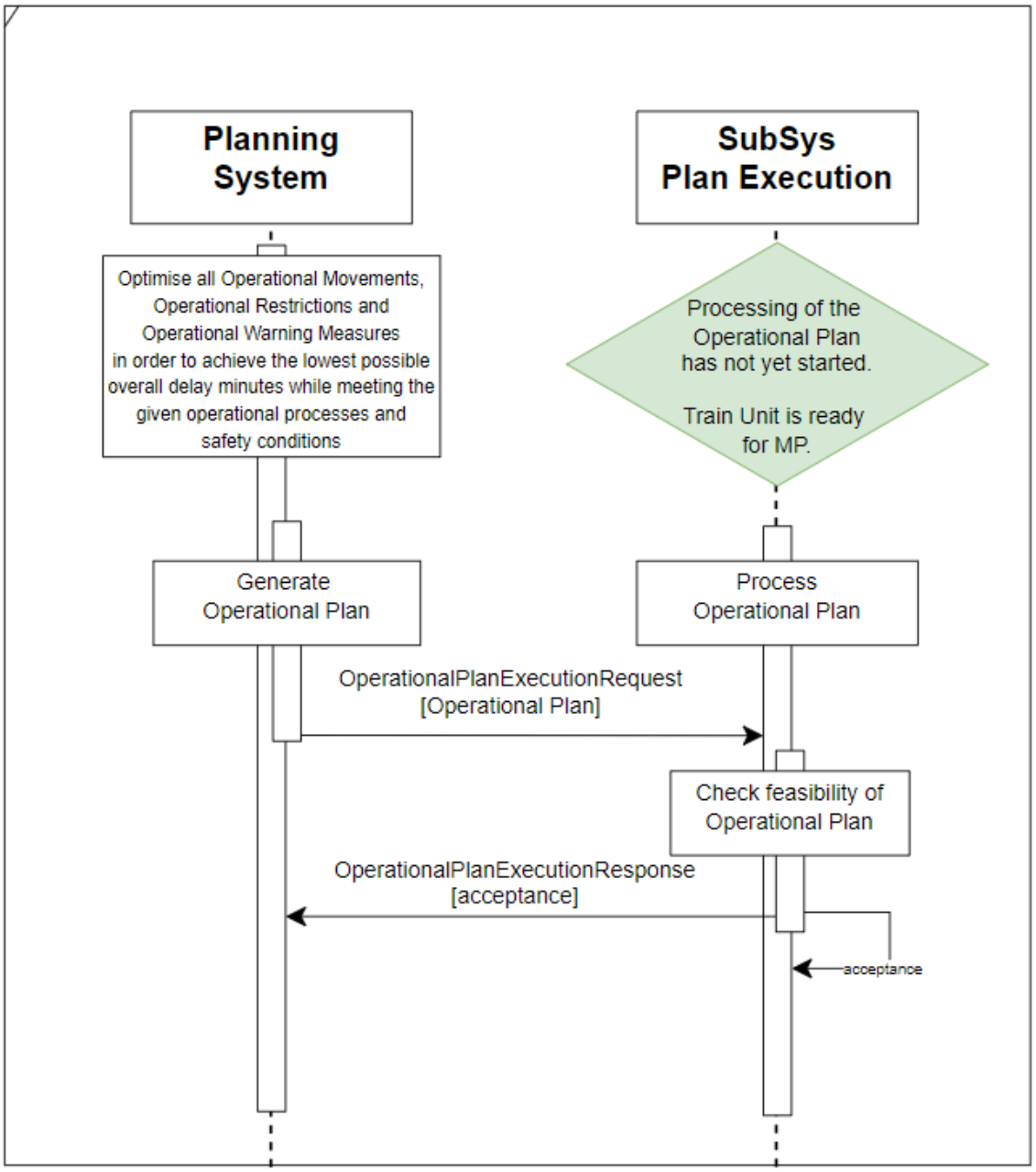


Figure 3: Sequence Diagram: Initial Plan (acceptance)

4.1.5.3.2. Scenario: Update of complete Plan, all Events rescheduled (acceptance)

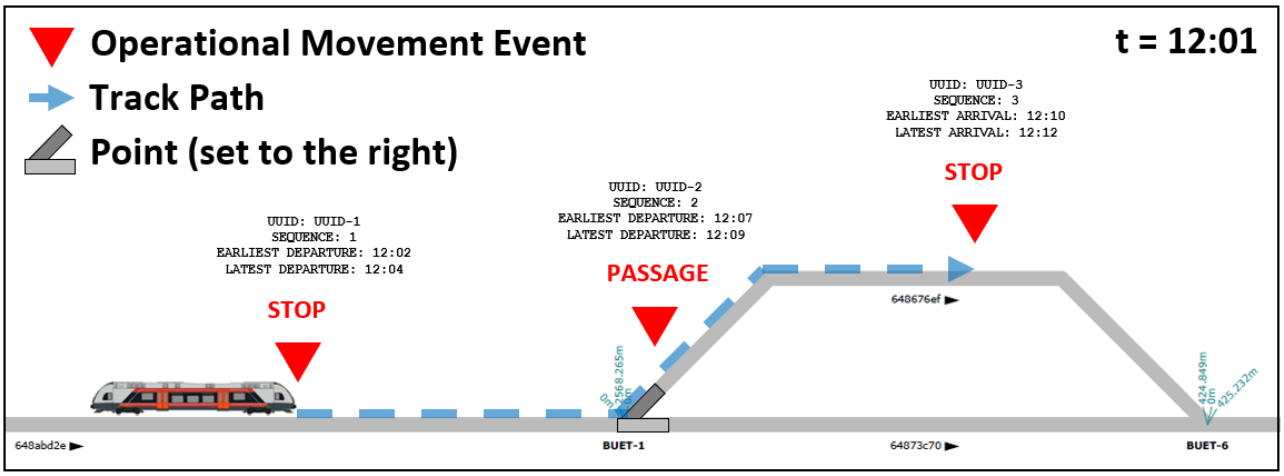


Figure 4: Schematic example of a feasible Operational Plan Operational Movement (all three Operational Movement Events have been rescheduled (+ 2 min) and have not been completed by SubSys Plan Execution)

Description of Scenario

The Scenario illustrates a complete update request of a previously requested Operational Plan Operational Movement (see 4.1.5.3.1), where Operational Event Times were rescheduled to optimise the Operational Plan Operational Movement. No event has already been completed by SubSys Plan Execution. The Train Unit is ready for MP but has not yet started the run.

Feasibility Check

The shown example of an Operational Plan Operational Movement will be accepted by SubSys Plan Execution if requested via SCI-OP as an update request of the previous version of that Operational Plan Operational Movement (see example at 4.1.5.3.1.). All feasibility checks (see 4.1.5.2) are all fulfilled.

Sequence Diagram

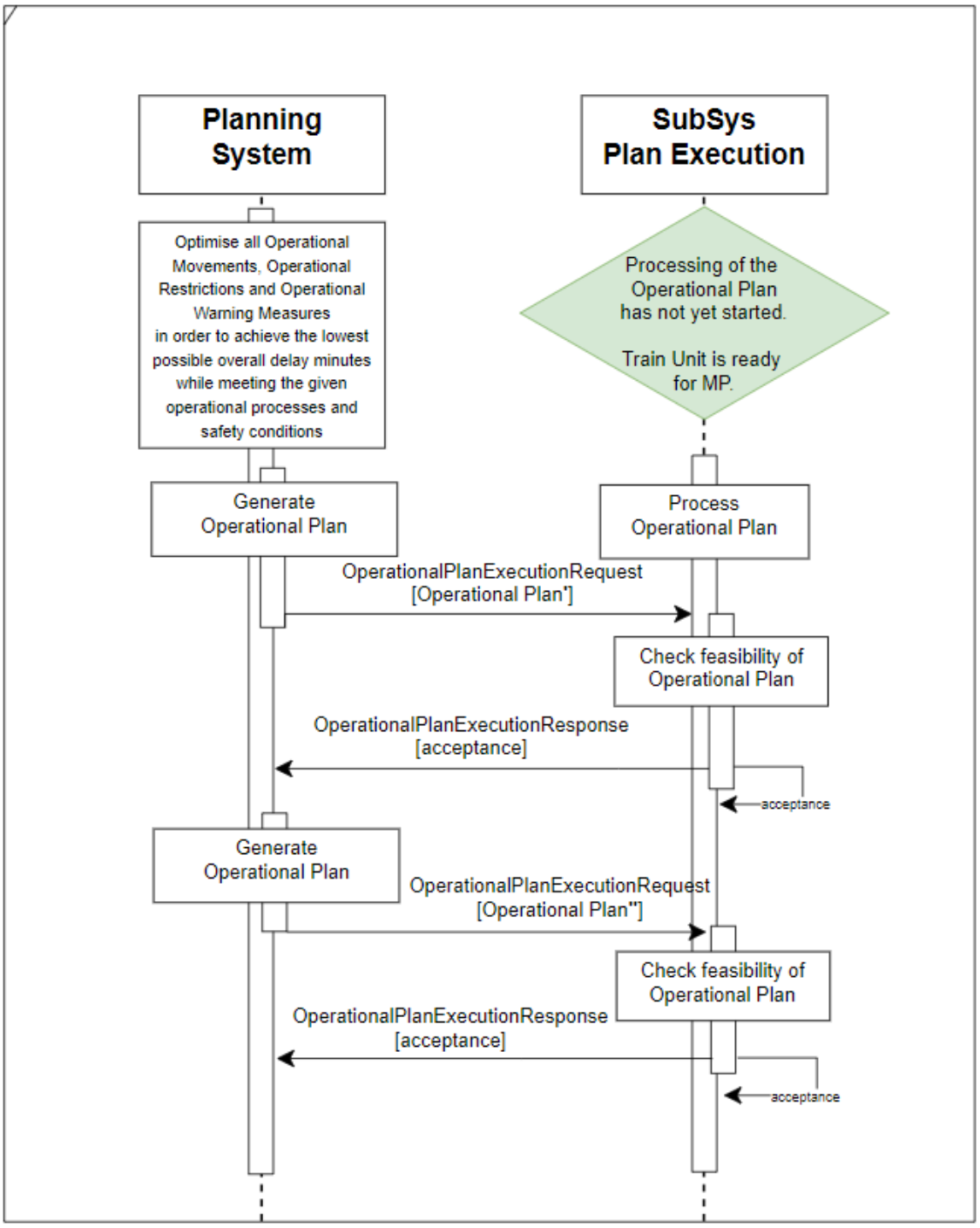


Figure 5: Sequence Diagram: Update of complete Plan, all Events rescheduled (acceptance)

4.1.5.3.3. Scenario: Update of partial Plan, Passage Event rescheduled (acceptance)

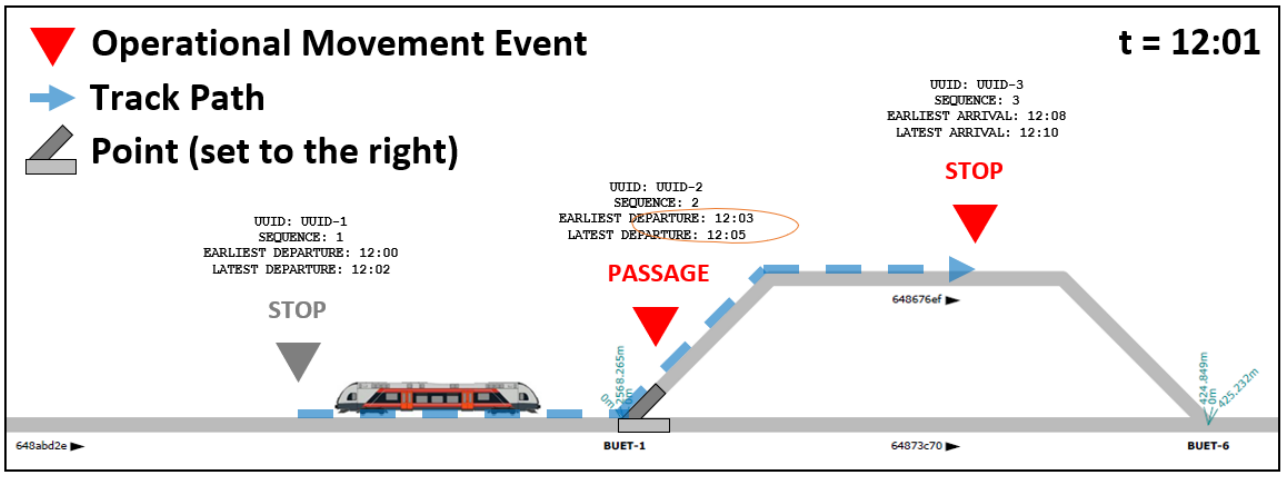


Figure 6: Schematic example of a feasible Operational Plan Operational Movement (first Stop Event has been completed, Passage Event has been rescheduled (- 1 min))

Description of Scenario

The Scenario illustrates an update request of a previously requested Operational Plan Operational Movement (see 4.1.5.3.1), where Operational Event Times of the Passage Event were prescheduled (for example in order not to block another outgoing Train Unit). SubSys Plan Execution already completed the first Stop Event and the Train Unit therefore moved forward on the Track Edge.

Feasibility Check

The shown example of an Operational Plan Operational Movement will be accepted by SubSys Plan Execution if requested via SCI-OP as an update request of the previous version of that Operational Plan Operational Movement (see example at 4.1.5.3.1.), but the first Stop Event would be ignored by SubSys Plan Execution. All feasibility checks (see 4.1.5.2) are all fulfilled.

Sequence Diagram

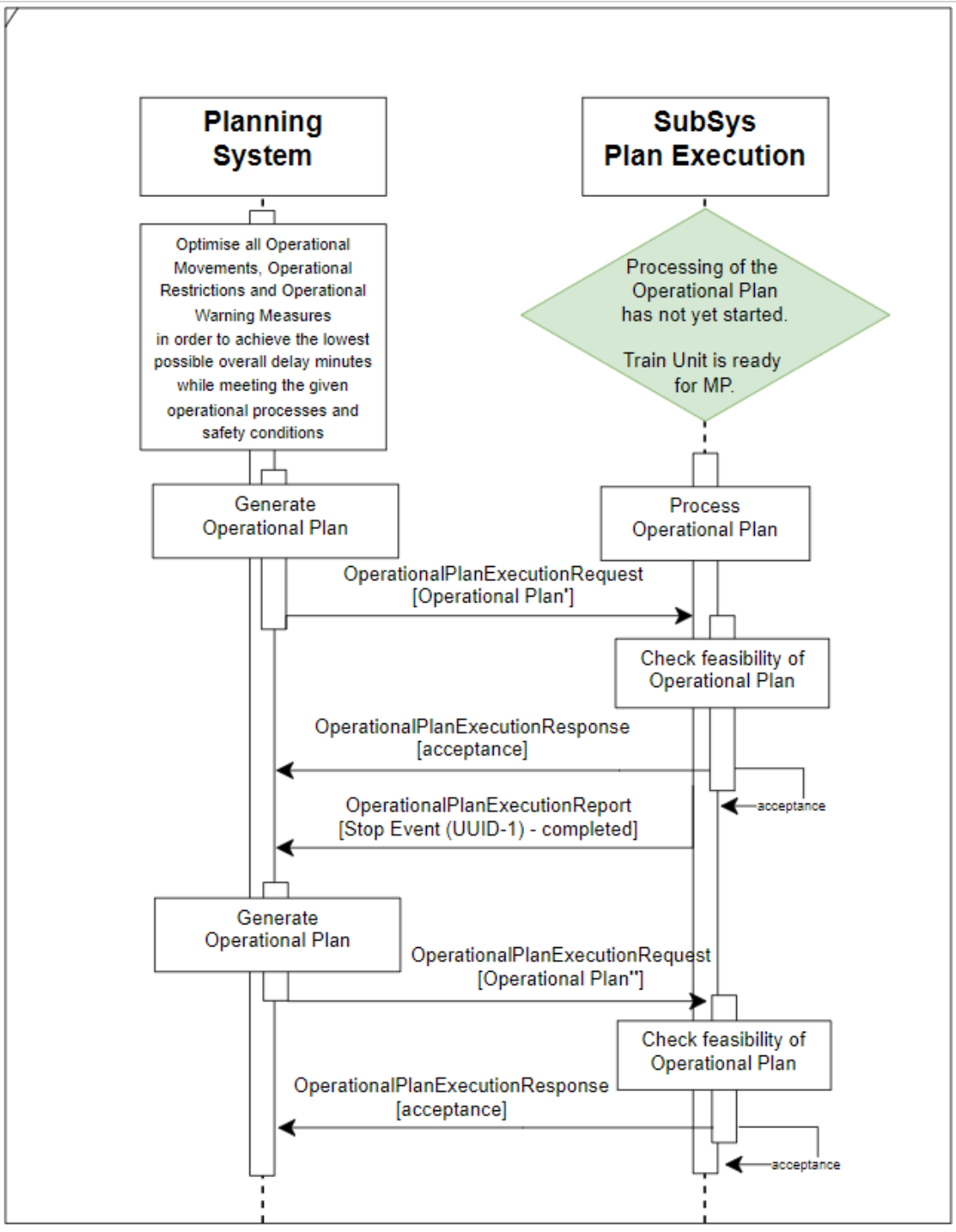


Figure 7: Sequence Diagram: Update of partial Plan, Passage Event rescheduled (acceptance)

4.1.5.3.4. Scenario: Update of partial Plan, different Track Path (acceptance)

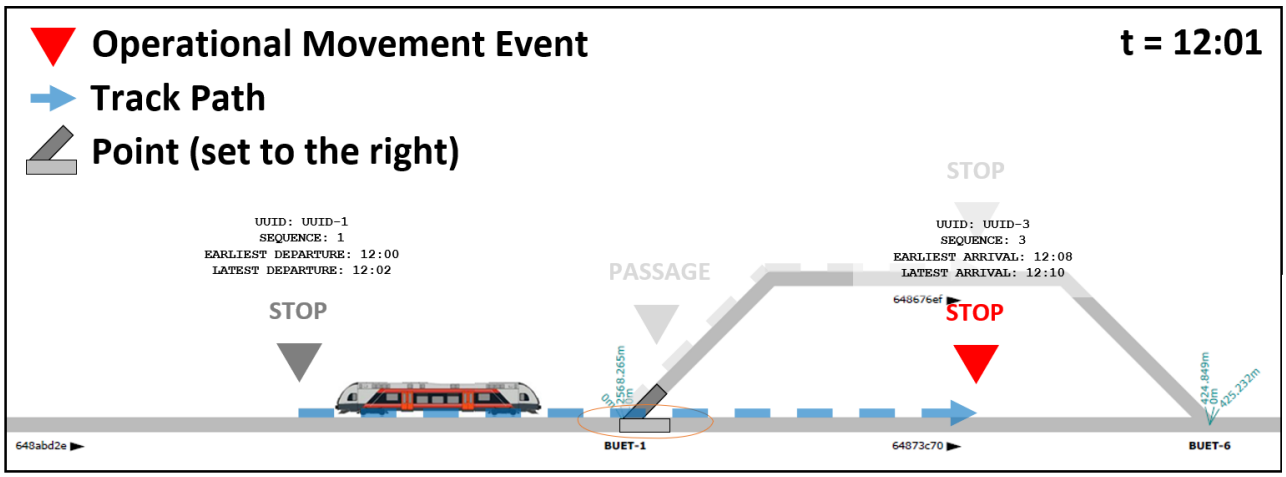


Figure 8: Schematic example of a feasible Operational Plan Operational Movement (first Stop Event has been completed, Passage Event has been removed, position of second Stop Event has been moved)

Description of Scenario

The Scenario illustrates an update of previously requested Operational Plan Operational Movement (see 4.1.5.3.1). SubSys Plan Execution already completed the first Stop Event and the Train Unit therefore moved forward on the Track Edge. A conflict arises due to a point (BUET-1) that cannot be set to the left due to a failure. This conflict is resolved by updating the Operational Plan Operational Movement, with the Track Path leading to another track.

Feasibility Check

The shown example of an Operational Plan Operational Movement will be accepted by SubSys Plan Execution if requested via SCI-OP as an update request of the before requested Operational Plan Operational Movement (see example in section 4.1.5.3.1.). All feasibility checks (see 4.1.5.2) are all fulfilled.

Sequence Diagram

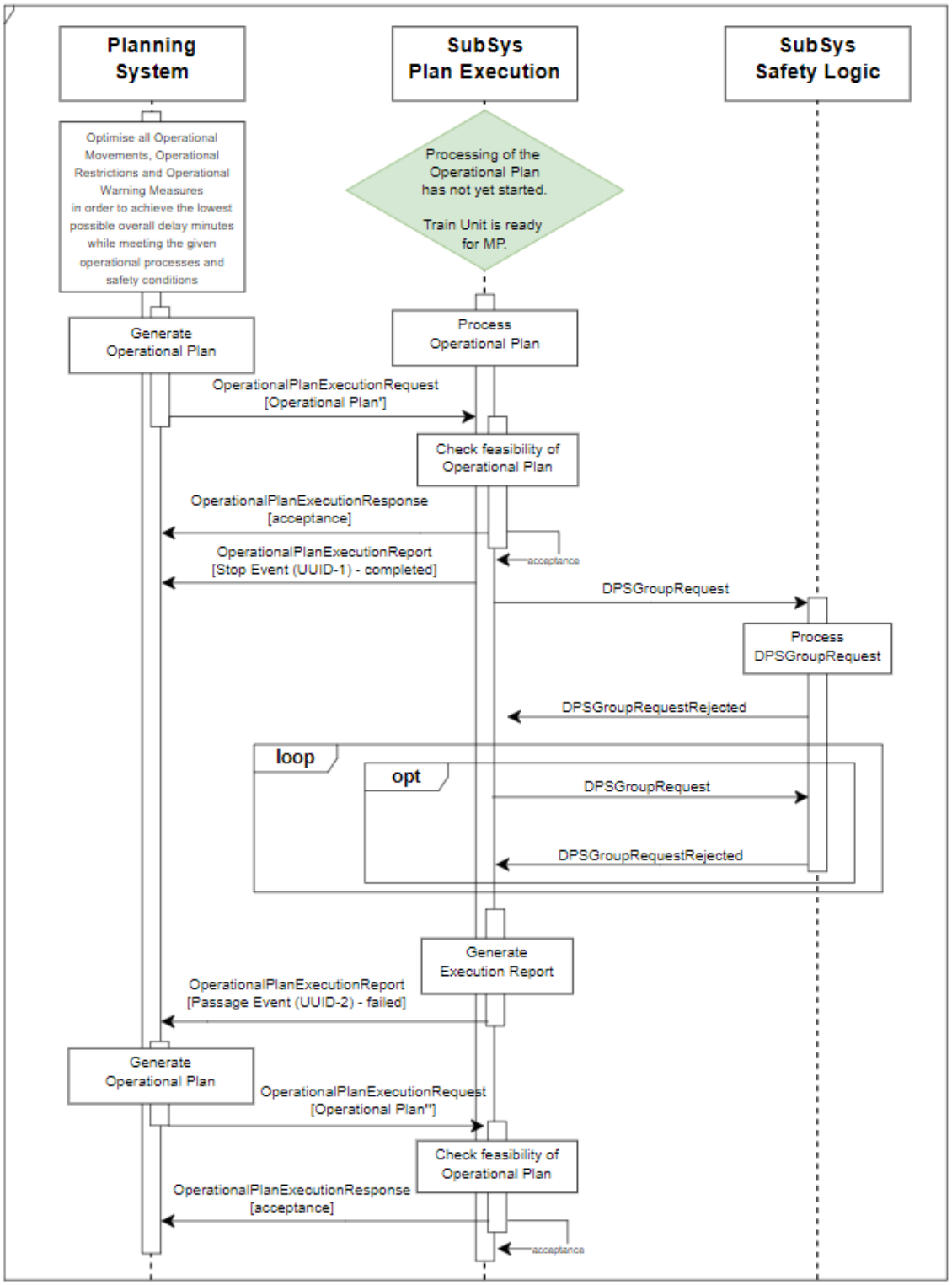


Figure 9: Sequence Diagram: Update of partial Plan, different Track Path (acceptance)

4.1.5.3.5. Scenario: Update of a partial Plan, different Track Path (rejection)

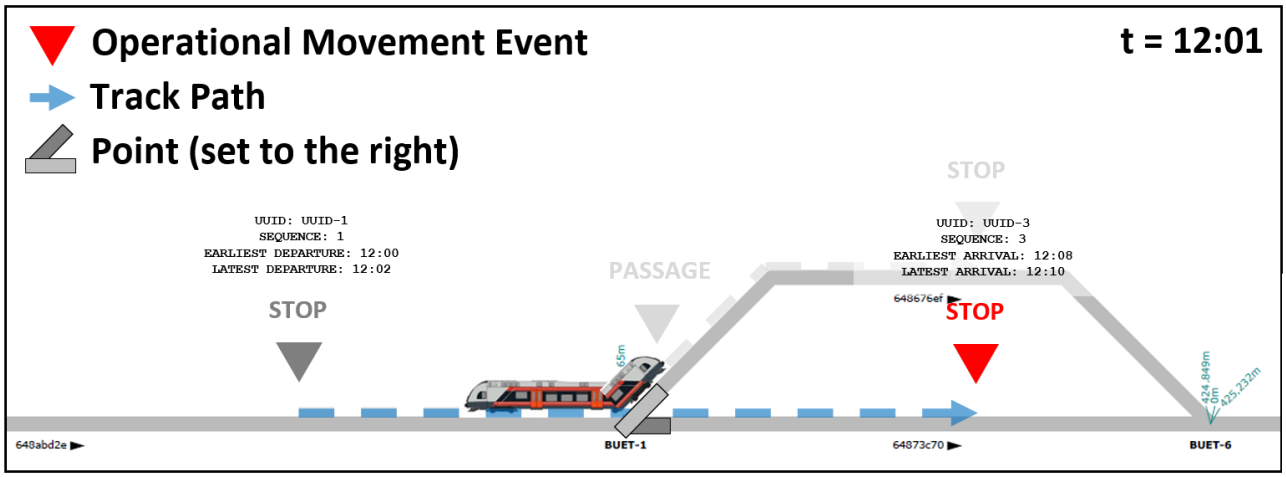


Figure 10: Schematic example of a non-feasible Operational Plan Operational Movement (first Stop Event has been completed, Passage Event has been removed, position of second Stop Event has been moved)

Description of Scenario

The Scenario illustrates an update of previously requested Operational Plan Operational Movement (see section 4.1.5.3.1). SubSys Plan Execution already completed the first Stop Event. The Train Unit therefore moved forward on the upper Track Edge. The updated request shall be rejected Operational Plan Operational Movement since the Train Unit has already passed the point.

Feasibility Check

The shown example of an Operational Plan Operational Movement will be rejected by SubSys Plan Execution if requested via SCI-OP as an update request of the before requested Operational Plan Operational Movement (see example at 4.1.5.3.1.), because the following of the feasibility checks (see 4.1.5.2) is not fulfilled.

Function	Subfunction	Functional Requirement
Process Operational Plan	Check feasibility of Operational Plan Operational Movement	The Operational Plan Execution Request for an **update** of an already requested Operational Plan Operational Movement shall be rejected by SubSys Plan Execution: • if the Train Unit is not (also not partly) located on the Track Path, but the route between the position of the front end of the Train Unit (in direction of travel) and the position of the first Operational Movement Event is not distinct • if the Train Unit is partly located on the Track Path, but the front end of the Train Unit (in direction of travel) is not located on the Track Path

Sequence Diagram

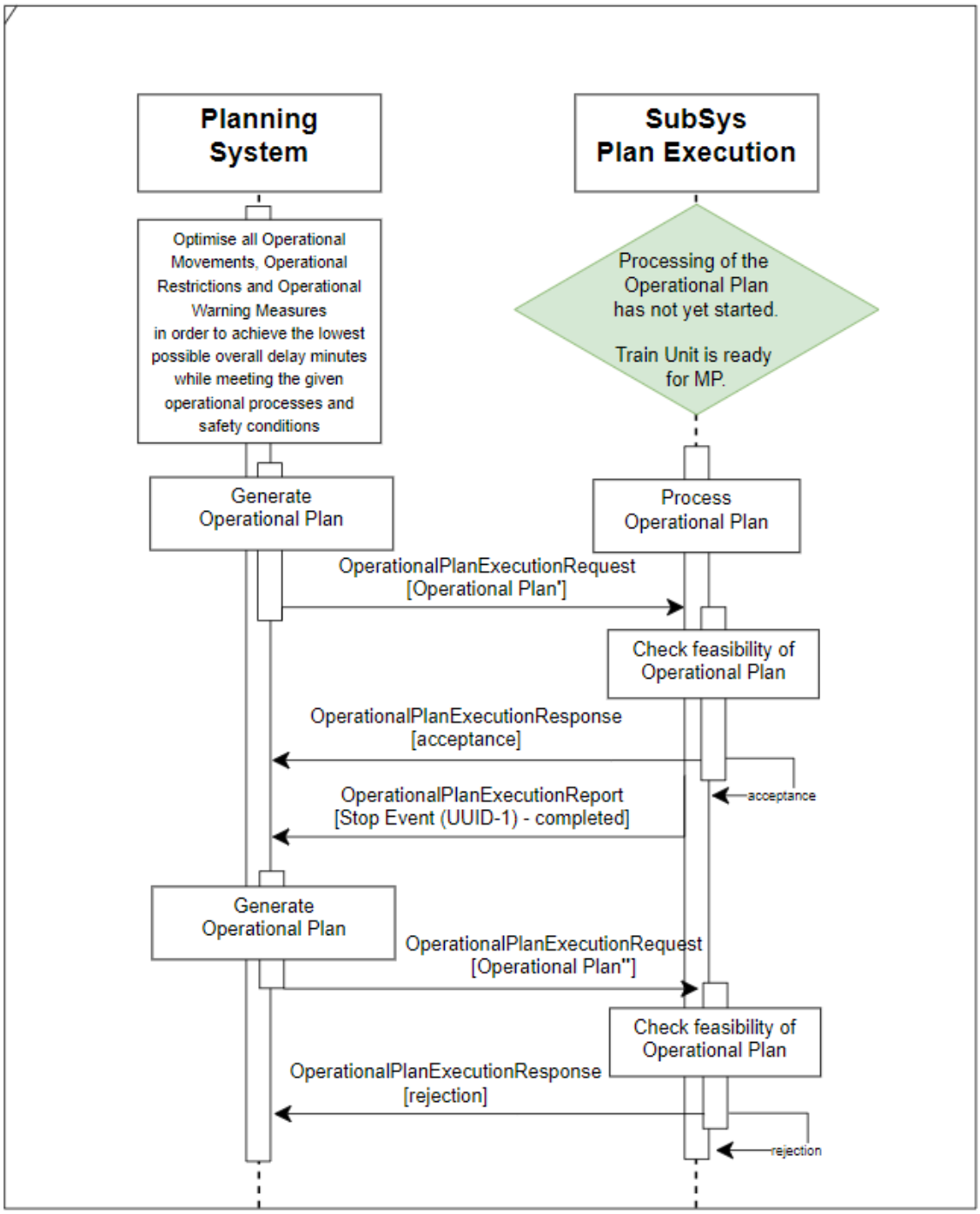


Figure 11: Sequence Diagram: Update of a partial Plan, different Track Path (rejection)

4.1.6. Operational Plan Operational Restriction

4.1.7. Operational Plan Operational Warning Measure

4.1.8. Versioning of Operational Plans

- A versioning of Operational Plans sent via SCI-OP is needed to determine the currently valid version of an Operational Plan. It shall be transparent to the Planning System which version of an Operational Plan is currently being executed by SubSys Plan Execution and SubSys ATO Execution in order to identify potential conflicts regarding different versions due to rejections of Operational Plan by one or the other RCA SubSys, asynchronous messaging or lost messages.

- Each Operational Plan shall be given an initial version number, which is incremented when the Operational Plan is updated.
- SubSys Plan Execution and SubSys ATO Execution always process only the currently valid version of an Operational Plan. In case of an accepted update of an Operational Plan, the previous version of the Operational Plan is replaced by the updated one.
- SubSys Plan Execution and SubSys ATO Execution execute their currently valid version of the Operational Plan separately. The Planning System is responsible for ensuring that both RCA SubSys work with the same Operational Plan or a consistent version of the Operational Plan.
- Versioning format could be aligned to ISO 8601.

4.1.9. Optimal time and frequency to send an Operational Plan Execution Request via SCI-OP

- An Operational Plan Execution Request shall be sent by the Planning System via SCI-OP as often as necessary and as infrequently as possible.
- An Operational Plan Execution Request is sent for the reasons, motivation and use cases specified in section 4.1.3.
- An Operational Plan Execution Request is sent, for an initial Operational Plan or if any changes have been made to the Operational Plan. This excludes, for example, periodically requesting the Operational Plan within a fixed defined time window.
- An Operational Plan Execution Request shall be sent by the Planning System via SCI-OP as early as necessary and as late as possible.
- The most sufficient time for sending an Operational Plan Execution Request is determined by the Planning System. It is assumed that the optimiser in the Planning System is checking the actual operational situation continuously and depending on train speed, station sequences, routing decisions, and further parameters, is calculating the optimal time for each Operational Plan Execution Request. Sending the request too early would mean the Operational Plan potentially might be adapted again, too late would mean that the update is late for its implementation which could lead to delays and additional conflicts in the rail traffic.

4.2. Functional concept: Calculation of MP and triggering of requests via SCI-CMD

This section will be further elaborated in future releases.

4.2.1. Objectives and System Requirements

The applicable objectives and system requirements can be found in *Concept: Plan Execution, RCA.Doc.47*, chapter: *Objectives and System Requirements*.

4.2.2. Introduction

This concept describes the logic according to which a Movement Permission (MP) is generated. The logic defines how long a MP is and when a MP is requested. It also describes when the states of a DPS-Group are requested.

4.2.3. Design Principles

Design Principle

Every movement of a Physical Train Unit shall be planned with an Operational Plan Operational Movement.

Rationale: For effective schedule optimisation the Planning System needs to control every movement and every track usage. To use the tracks in the most efficient way, the Planning System creates Operational Plans and tracks their execution.

Design Principle

SubSys Plan Execution shall ensure the most efficient utilisation of the planned capacity of the railway network by the optimal implementation of Operational Plans sent by the Planning System.

Rationale: Enabling the closest possible sequence of train movements (moving block) and ensuring that the train can run continuously without unscheduled stops.

4.2.4. Functional Requirements

This section will be further elaborated in future releases.

Function	Subfunction	Functional Requirement
Control Operational Movement of Physical Train Units	-	SubSys Plan Execution shall be able to request a MP to allow the Train Unit to move to the planned position on the railway network.
Control Operational Movement of Physical Train Units	Calculation of MP	SubSys Plan Execution shall calculate the length of the MP. • SubSys Plan Execution shall ensure that the MP is long enough for the train unit to run continuously. • SubSys Plan Execution shall ensure that the MP is only as long as necessary to allocate as little of the railway network as possible.
Control Operational Movement of Physical Train Units	Calculation of Trigger Points for MP Request	SubSys Plan Execution shall calculate the optimal time to request a MP.
Control driveability of railway network	Calculation of Trigger Points for driveability requests	SubSys Plan Execution shall calculate the optimal time for request the states of an DPS-Group.
Control Operational Movement of Physical Train Units	Calculation of Risk Buffer	SubSys Plan Execution shall calculate the required Risk Buffer and consider it when requesting a MP.
Control Operational Movement of Physical Train Units	Calculation of Risk Paths	SubSys Plan Execution shall calculate the required Risk Paths and consider it when requesting a MP.
Control Operational Movement of Physical Train Units	Calculation of MP	SubSys Plan Execution shall consider the Map Data for the calculation of the MP (e.g. max speed of allocated railway network, non-stopping Areas, diamond crossing with single slip must always be completely clear or completely allocated)
Control Operational Movement of Physical Train Units	Calculation of MP	SubSys Plan Execution shall consider the temporary restriction of the railway network (URA) for the calculation of the MP
Control Operational Movement of Physical Train Units	Calculation of MP	SubSys Plan Execution shall consider the position of a planned stop on the railway network for the calculation of the MP. (Train Unit must be able to move to the planned stop position without being stopped by the ETCS on-board equipment due to odometry inaccuracy)
Control Operational Movement of Physical Train Units	Calculation of MP	SubSys Plan Execution shall consider the characteristics of the Train Unit for the calculation of the MP
Control Operational Movement of Physical Train Units	Calculation of MP	SubSys Plan Execution shall consider the maximum allowed length of a MP for the calculation of the MP length.
Control Operational Movement of Physical Train Units	Calculation of MP	SubSys Plan Execution shall consider MP Limit Markers as other MP, MOB, state of DPS-Groups, URA for the calculation of the MP

Control Operational Movement of Physical Train Units	Calculation of MP	SubSys Plan Execution shall shorten the MP at front when the Train Unit stops in order to deallocate the railway network in front of the Train Unit.
Control Operational Movement of Physical Train Units	Calculation of MP	SubSys Plan Execution shall be able to detect an unscheduled stop and therefore shorten the MP at front (the Train Unit receives an initial MP after a stop but can't depart due to a failure on the Train Unit).
Control Operational Movement of Physical Train Units	Calculation of MP	SubSys Plan Execution shall continuously shorten the MP at rear when the Train Unit moves in order to deallocate the railway network behind the Train Unit. (to be clarified if this is done by SubSys Plan Execution or SubSys Safety Logic)
Control Operational Movement of Physical Train Units	Calculation of MP	SubSys Plan Execution shall minimize the probability that a MP request is rejected (a rejection may lead to an unscheduled stop. Unscheduled stops shall be avoided).
Process Operational Plan	-	SubSys Plan Execution shall be able to process even short-term updates of Operational Plans

4.2.5. Functionality for calculation of MPs and triggering of requests

4.2.5.1. MP Limit Marker

➔ ***Abstract concept: MP Limit Marker***
A MP Limit Marker represents the possible limit of a future MP. Each MP Limit Marker defines a potential stopping position on a Track Edge. A MP Limit Marker can have Constraints and is active exactly until all its Constraints are fulfilled. The extent of a MP can reach at most to the next active MP Limit Marker along the Track Path. A MP Limit Marker once dissolved never becomes active again.

Examples of MP Limit Markers:

- A Field Element is not in the required state to implement the requested Track Path.
- Another MP or MOB is on the requested Track Path.
- A Usage Restriction Area is on the requested Track Path
- A Constraint given by the Operational Plan Operational Movement is not given
    - earliest departure/arrival time
    - train run sequence given by Operational Event Link.

4.2.5.2. Trigger

➔ **Abstract concept: Trigger**
A Trigger represents an action which is triggered by SubSys Plan Execution under certain constrains to implement the Operational Plan Operational Movement. A trigger is executed exactly at the time when all its constraints are fulfilled for the first time. Once a trigger has been executed, it is not executed again.

Examples of a Trigger:

- Trigger to request the driveability state of a Field Element

4.2.5.3. Constraint

➔ ***Abstract concept: Constraint***
A Constraint describes a condition to be fulfilled before executing a Trigger or conditions to be fulfilled before dissolving a MP Limit Marker. Constraint can therefore be applied to both MP Limit Markers and Triggers. Constraints can be combined with each other and also nested.

Different types of Constraints:

Constraint type	Description
Pass Position Constraint	A Train Unit must pass a certain position
Stop Position Constraint:	A Train Unit must stop at a certain position
Field Element State Constraint	A Field Element must reach a certain state
Timing Constraint	A certain point in time is reached
Linking Constraint	A Train Unit must wait for another Train Unit to arrive before passing, arriving or departing from a certain position

4.2.5.4. Working principle: MP Limit Marker, Triggers, Constraints

The following events types trigger the updating of MP Limit Markers and Triggers:

- Update of Operational Plan Operational Movement (sent by the Planning System via SCI-OP).
- Update of MOB state and position (send by APS via SCI-CMD).
- Update of MP (send by APS via SCI-CMD).
- Update of Field Element State (send by APS via SCI-CMD).
- Passed timing point.

The following procedure is executed when updating MP Limit Markers and Triggers:

- For each event, check whether new MP Limit Markers and Triggers (including their Constraints) must be created or whether existing MP Limit Markers and Triggers need to be removed.
- For each event, check whether existing constraints of MP Limit Markers and Triggers are fulfilled.
- Execute all Triggers whose Constraints are fulfilled for the first time.
- Dissolve all MP Limit Markers whose Constraints are fulfilled for the first time.
- Calculate and request a MP update with the front extension of the MP to the next active MP Limit Marker.

4.2.6. Calculation of the MP Extent

- A MP is always requested along the exact track specified in the Operational Plan Operational Movement. It is always requested up to the next active MP Limit Marker.
- If a MP Limit Marker is dissolved, the MP is requested to be extended to the next MP Limit Marker or to the maximum possible MP length.
- If a MP Limit Marker is shifted, the MP is requested to be extended to the shifted position of the MP Limit Marker.
- If the MP Limit Marker is further away than the maximum possible length of a MP, then the MP is requested with the maximum possible length (if the maximum possible length of an MP is a constant or a dynamic value is not defined yet, it also might correspond to the maximum possible length of an ETCS Movement Authority)

4.2.6.1. Dissolving of a MP Limit Marker

Active MP Limit Marker	Dissolved MP Limit Marker
• A Field Element is not in the required state to implement the requested Track Path.	• A Field Element in the required state to implement the requested Track Path.
• Another MP or MOB is on the requested Track Path.	• MP or MOB have moved and are now behind another active MP Limit Marker. • The Train Unit fulfils the required conditions for its own MP to overlap with another MOB or MP.
• A Usage Restriction Area is on the requested Track Path.	• The Usage Restriction Area was removed. • The Train Unit fulfils the required conditions for its own MP to overlap with the Usage Restriction Area.

• A Constraint given by the Operational Plan Operational Movement is not fulfilled o earliest departure/arrival time. o train run sequence given by Operational Event Link.	• The Constraint given by the Operational Plan Operational Movement is fulfilled. o the earliest departure time has been reached. o the Constraint of the Operational Event Link is fulfilled (e.g. the expected Train Unit has passed the position).

4.2.6.2. Triggering of MP Updates

The objective is:

- to extend the MP at front of the Train Unit whenever possible
- to shorten the MP at the rear of the Train Unit whenever possible (Track Path section behind the moving Train Unit which allocated by MP)

The following events will trigger SubSys Plan Execution to request a MP-Update (extend and/or shorten).

- Message of position update of the Train Unit
- The MP Limit Marker in front of a Train Unit has shifted or was dissolved
- Time has elapsed. The earliest departure time (Stop Event or Passage Event) has been reached

The following is therefore checked for each of the above events:

- Can the MP be shortened?
- Can the MP be extended?

If one or both cases apply, then SubSys Plan Execution shall request a MP-Update.

4.2.7. Triggering of Field Element States

To implement the requested Track Path of an Operational Plan Operational Movement, SubSys Plan Execution needs to set the Field Elements on the Track Path into the required state. Therefore, SubSys Plan Execution creates triggers for all Field Elements on the Track Path. For an efficient implementation these triggers are assigned with Constraints. At the time when all Constraints of the triggers are fulfilled, the required Field Element state is requested by SubSys Plan Execution.

4.2.7.1. Calculation of the Trigger Position

➔ ***Abstract concept: Trigger Position***
The Trigger Position defines the position where a Field Element State must be requested by SubSys Plan Execution at the latest in order for a Train Unit to receive a MP over the Field Element without driving into the current MPs braking curve.

The calculation of the Trigger Position depends on various factors:

- Position to which the Train Unit can move if the corresponding Field Element is not in the required state (Position of Field Element minus length of the required Risk Buffer of the MP).
- Stopping distance as the distance between the point at which the train driver is requested by the supervising system to initiate the braking process so that the Train Unit comes to a stop at the end of the existing MP
- Distance travelled during the setting time (t_setting) in which the Field Element is brought into the required state.
- Distance travelled during the system latency time (t_latency) for processing the request and for granting the extended MP
- Distance travelled during the ergonomics time (t_ergonomics) for the train driver (i.e. the MP should already be updated a certain time before reaching the braking curve, so that the train driver has enough time to perceive the updated MP).

The above breakdown makes it clear that for the calculation of the Trigger Points not only the braking distance of the Train Unit but also the intervention and warning behaviour of the supervising system (ETCS), must be

taken into account. Since this behaviour of ETCS is also standardised, the calculation can also be carried out with the necessary accuracy in SubSys Plan Execution, knowing the necessary Train Unit parameters.

The following picture conceptually shows the breakdown at the example of a point:

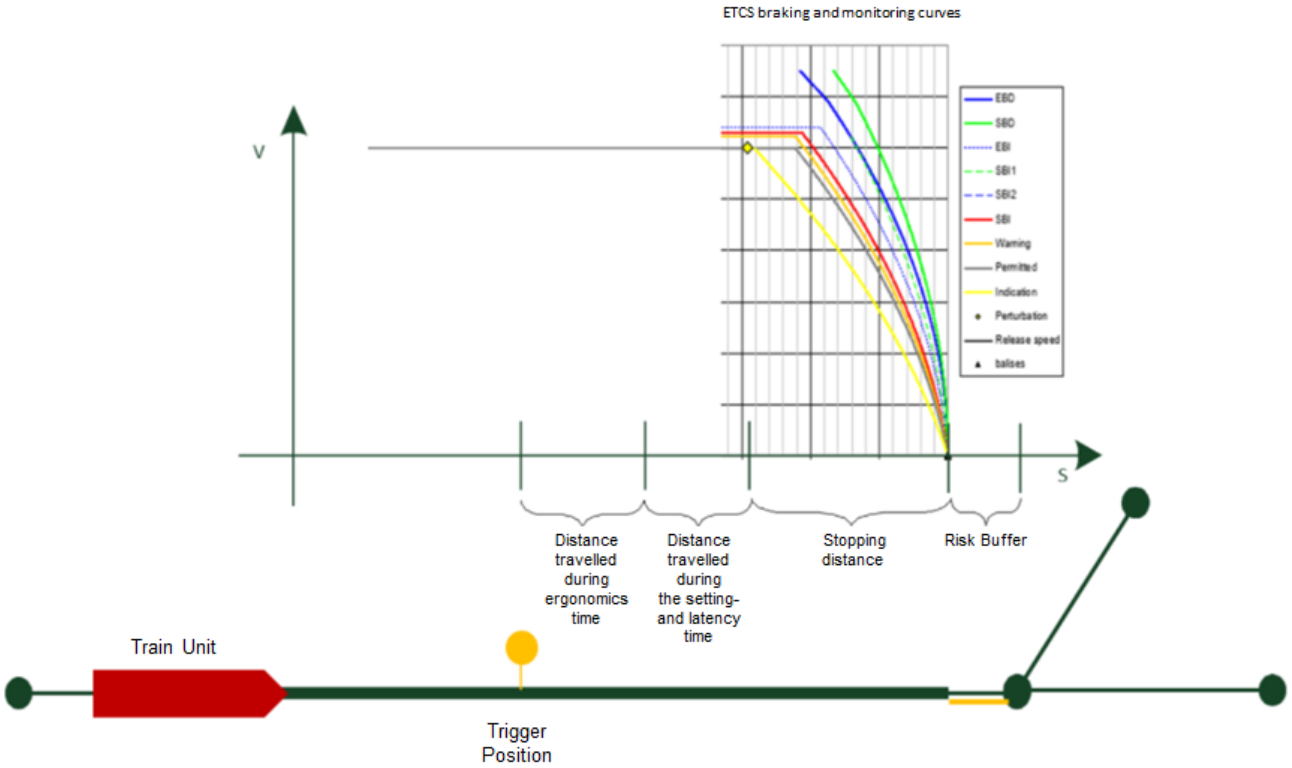


Figure 12: Example: breakdown of different distances required to calculate the Trigger Position of a point

NOTE: The above graph assumes a constant Vmax. If Vmax is not constant, the integral v(t) must be determined. As an alternative to Vmax, Vopt from the Operational Plan Operational Movement could also be used.

The Trigger Position only indicates the position where the first attempt is made to set the Field Element, provided that no other constraints are present at the Trigger. If the Field Element not allocated at this time, the request will be triggered. If the Field Element is allocated by another Train Unit at the time the Trigger is triggered, the request is stored in a queue and executed as soon as the Field Element is no longer allocated by the other Train Unit.

4.2.7.2. Calculation of Earliest Trigger Time

➔ ***Abstract concept: Earliest Trigger Time***
The Earliest Trigger Time defines the time before a Field Element may not be requested.

The Earliest Trigger Time is added as a further Constraint to the Trigger Position where necessary. The Earliest Trigger Time is always relevant if the Operational Plan Operational Movement has a timing constraint (Stop Event with scope EARLIST_DEPARTURE/EARLIST_ARRIVAL or a Passage Event with scope EARLIST_DEPARTURE) between the Trigger Position and the Field Element.

The Earliest Tigger Time depends on the following factors:

- The earliest departure time (t_earliest_departure) of the Train Unit for a stop.
- The earliest departure time (t_earliest_departure) of the Train Unit for a passage.
- The setting time (t_setting) in which the Field Element is brought into the required state
- The system latency time (t_latency) for processing the request and for granting the extended MP

- The ergonomics time (t_ergonomics) for the train driver (i.e. the MP should already be updated a certain time before reaching the braking curve, so that the train driver has enough time to perceive the updated MP).

The Earliest Trigger Time of a Field Element results from:

- For Stop Event:
  t_earliestTriggerTime = t_ earliest_departure - T_setting - t_latency - t_ergonomics
- For Passage Event:
  t_earliestTriggerTime = t_ earliest_departure - T_setting - t_latency - t_ergonomics

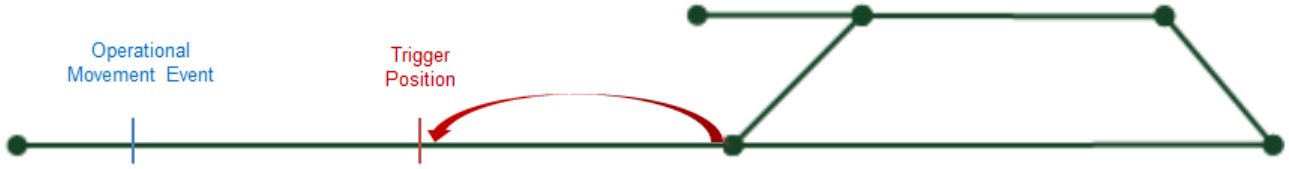
4.2.7.3. Calculation of Constraints

Based on Operational Plan Operational Movement, SubSys Plan Execution identifies all Field Elements that must be brought into a certain state. Initially and with each update of the Operational Plan Operational Movement, the Constraints are updated for all identified Field Element.

The different cases and the resulting Constraints are described below.

Case 1: No Operational Movement Event between Trigger Position and Field Element

There is no Operational Movement Event between the Trigger Position and the Field Element.



In this case, the following Constraint is assigned to the Trigger for the Field Element:

➔ Pass Position Constraint: Head of Train Unit passed Trigger Position

Case 2: Operational Movement Event is between Trigger Position and Field Element

There is a Stop Event with scope EARLIST_DEPARTURE/EARLIST_ARRIVAL or a Passage Event with scope EARLIST_ DEPARTURE between the Trigger Position and the Field Element

In this case, a Trigger for the Field Element with the following constraint is defined:

➔ Pass Position Constraint: Head of Train Unit passed Trigger Position
➔ Timing Constraint: Earliest Trigger Time

4.3. Functional concept: System States, Operational States and Modes of Operation

This section will be further elaborated in future releases.

4.3.1. Objectives and System Requirements

The applicable objectives and system requirements can be found in *Concept: Plan Execution, RCA.Doc.47*, chapter: *Objectives and System Requirements*.

4.3.2. Introduction

To face the complexity of the SubSys Plan Execution, it is decomposed into different states and operating modes, which are described in the following chapters.

The following definitions, adapted from System Engineering sources, are proposed:

Term	Definition
Phase	➔ ***Term: Phase*** *A label applied to the segments of the System/ Product Life Cycle or Mission Life Cycle of a System or Product.*
System State	➔ ***Term: System State*** *"An attribute that represents the current logistical employment, availability, or performance-based condition of an Enterprise asset such as a system, product, or service. System State examples include: (in) Storage; (in) Deployment; (in) Operation, (in) Maintenance, (in) Retirement, (in) Disposal" (Adaption of Wasson, 2014, p. 3).*
Operational State	➔ ***Term: Operational State*** *"An attribute that characterizes the operational status, readiness, availability, or condition of a system, product, or service at a specific instant in time to conduct or continue a mission. For example, a system or product is active or inactive; operational/operating (On), or non-operating (Off); failed; awaiting maintenance" (Adaption of Wasson, 2014, p. 3).*
Mode of Operation	➔ ***Term: Mode of Operation*** *An abstract label applied to a user-selectable (or a predefined trigger in case of autonomous or fully automated systems) option that enables a set of System Capabilities to be employed in conjunction with enterprise processes and procedures to monitor, command and control (MC2) of an enterprise or engineered system, product, or service to achieve a specific mission outcome, objectives, and levels of performance.*

4.3.3. Functional Requirements

This section will be further elaborated in future releases.

SubSys Plan Execution shall support different System States and Operational States and shall provide these states to connected systems. It shall also receive the System States and Operational States of connected systems.

Function	Subfunction	Functional Requirement
Support different System States, Operational States and Modes of Operation	-	tbd
Determine System State and Operational State	-	tbd
Provide and receive System State and Operational State	-	tbd

4.3.4. Description

This chapter describes the intended System States, Operational States and Modes of Operation of SubSys Plan Execution.

The following table gives an overview on the System State, Operational States and supported Modes of Operation of SubSys Plan Execution. For the specification of SubSys Plan Execution, only the states of the Phase *Operation* are relevant and are therefore more detailed.

Phase	System State	System Substate	Operational State	Supported Mode of Operation	Description of State
Concept					
Development					
Production					
Operation					
	Off State			Off Mode	▪ SubSys Plan Execution is switched off
	In Operation State			Operation Mode	▪ SubSys Plan Execution continuously synchronises the state of Configuration Data and Operational Data ▪ SubSys Plan Execution determines System State and Operational State ▪ SubSys Plan Execution provides and receives System State and Operational State ▪ SubSys Plan Execution supports activation of Map Data
		Hot Standby State			▪ SubSys Plan Execution is in hot standby ▪ SubSys Plan Execution supports handover of Operation to system instance which is in hot standby ▪ SubSys Plan Execution supports no operation
		Initialisation State			▪ SubSys Plan Execution imports Configuration Data (technical parameters, configuration parameters, other external data needed for operation like e.g. Map Data)
		Operation State			
			Normal State		▪ No system failure is detected ▪ SubSys Plan Execution supports full operation
			Abnormal State		▪ No system failure is detected ▪ SubSys Plan Execution supports limited operation (NFRs are reduced, External system failure was detected)
		Failure State			
			Failed State		▪ System failure(s) is/was detected ▪ SubSys Plan Execution supports full operation
			Degraded State		▪ System failure(s) is/was detected ▪ SubSys Plan Execution supports limited operation
			Emergency State		▪ System failure(s) is/was detected ▪ SubSys Plan Execution supports only emergency operation

	Maintenance State			Maintenance Mode	▪ SubSys Plan Execution supports maintenance
Retirement					
Support					

The following table gives an overview of the Modes of Operation.

Modes of Operation	Description of Mode of Operation
Off Mode	▪ User can trigger SubSys Plan Execution to go into In Operation State or Maintenance State
Operation Mode	▪ User can operate the SubSys Plan Execution according to the systems Operational State (e.g. full, limited, none) ▪ User can trigger the SubSys Plan Execution to switch off ▪ User can trigger the SubSys Plan Execution to go into Maintenance State ▪ User can trigger the SubSys Plan Execution handover to system instance which is in hot standby
Maintenance Mode	▪ User can maintain SubSys Plan Execution ▪ User can trigger the SubSys Plan Execution to switch off ▪ User can trigger the SubSys Plan Execution to go into Operation State

The following state machine diagram illustrates the System States, Operational States and Modes of Operation of SubSys Plan Execution and their transitions.

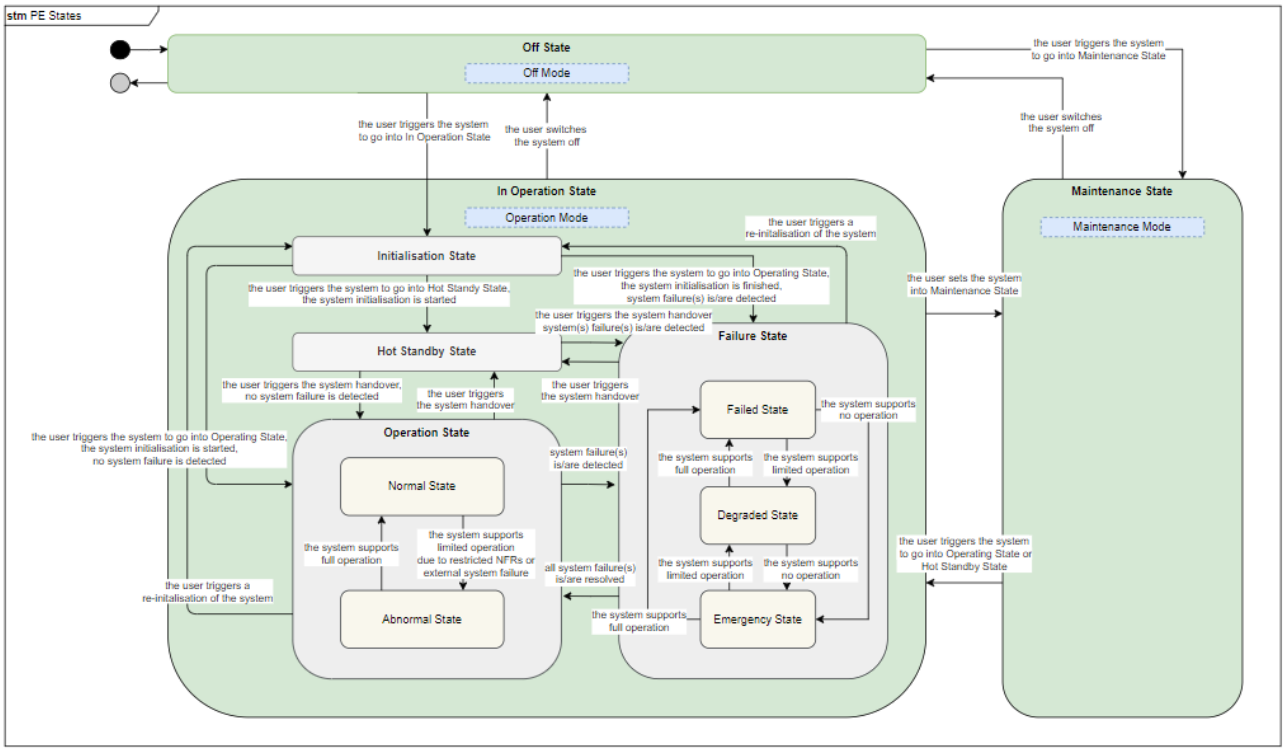


Figure 13: Overview on System States, Operational State and Modes of Operation of SubSys Plan Execution

4.3.5. Mapping of functions to System States

System State	Mapped Functions	Description
• Initialisation State	• Import Configuration Data • Synchronise the state of Configuration Data and Operational Data • Calculate and check internal data model • Determine System State and Operational State	• SubSys Plan Execution shall perform the following functions once and one after the other

	• Provide and receive System State and Operational State	
• Hot Standby State • Operation State • Failure State	• Import Configuration Data	• Function may be triggered by SubSys Device & Config Management when providing a new version of Configuration Data • Function will be performed in a background process
	• Synchronise the state of Configuration Data and Operational Data • Calculate and check internal data model • Determine System State and Operational State • Provide and receive System State and Operational State	• PE performs the following functions in a time-controlled and sequential manner • Functions will be performed in a background process

4.4. Functional concept: Robustness and High availability

This section will be further elaborated in future releases.

4.4.1. Objectives and System Requirements

The applicable objectives and system requirements can be found in *Concept: Plan Execution, RCA.Doc.47*, chapter: *Objectives and System Requirements*.

4.4.2. Non-functional Requirements

This section will be further elaborated in future releases.

This chapter describes a basic concept to address the non-functional requirements (NFR) Robustness, Fault tolerance and Availability of SubSys Plan Execution.

The following use cases regarding these NFRs must be addressed by SubSys Plan Execution:

- Behaviour of SubSys Plan Execution in case of failure of SubSys Plan Execution or adjacent systems
- Behaviour of SubSys Plan Execution in case of lost, invalid or malformed inputs on the interfaces
- Behaviour of SubSys Plan Execution in case of detected inconsistencies in its data
- Behaviour of SubSys Plan Execution during different System States and Operational States of system life cycle

The table describe the requirements regarding the listed use cases.

Use Case	Error Conditions	Functional Requirement
Behaviour of SubSys Plan Execution in case of failure of SubSys Plan Execution or adjacent systems	Failover of Planning System	SubSys Plan Execution shall retain undelivered messages and shall deliver them as soon as the Planning System becomes again available. It is assumed that the Planning System is not a highly available system (> 99.95%). The availability, fault tolerance and robustness of SubSys Plan Execution shall be dimensioned accordingly higher to ensure uninterrupted rail traffic.
Behaviour of SubSys Plan Execution in case of lost, invalid or malformed inputs on the interfaces	Handling of lost messages	SubSys Plan Execution shall be capable of detecting the loss of messages which it was sending via SCI-OP. The quality of services is expected to be “at least once delivery”. Unsuccessfully delivered messages shall be delivered when the situation allows for it again. Appropriate capabilities in SubSys Plan Execution are required for this.

Behaviour of SubSys Plan Execution in case of lost, invalid or malformed inputs on the interfaces	Handling of invalid messages: - SubSys Plan Execution receives malformed Operational Plan from Planning System - SubSys Plan Execution receives Operational Plan containing implausible data from Planning System - SubSys Plan Execution receives implausibly versioned Operational Plan from Planning System	SubSys Plan Execution shall be capable of detecting invalid messages and respond to the sending System with an appropriate reply. The reply shall only contain at least • a notification of rejection and • a description and localisation of the fault as precise as possible. At the same time, SubSys Plan Execution shall be able to understand responses from other RCA SubSys (e.g. SubSys Safety Logic) regarding its own invalid messages, recognise the error, correct the error, and resend the corrected message.
Behaviour of SubSys Plan Execution in case of detected inconsistencies in its data	Operational Data out of sync	SubSys Plan Execution shall contain functionalities to synchronise Operational Data. Therefore, SubSys Plan Execution shall be capable: • to send the complete Operating State to the Planning System
Behaviour of SubSys Plan Execution in case of detected inconsistencies in its data	Configuration Data out of sync	SubSys Plan Execution shall contain functionalities to synchronise Configuration Data. Therefore, SubSys Plan Execution shall be capable: • to import complete Configuration Data from defined sources.
Behaviour of SubSys Plan Execution in case of detected inconsistencies in its data	Map Data out of sync	SubSys Plan Execution shall contain functionalities to ensure that it is operating with the latest valid Map Data version. Therefore, SubSys Plan Execution shall be capable to import Map Data from defined sources.
Behaviour of SubSys Plan Execution during different System States and Operational States of system life cycle	-	• SubSys Plan Execution shall have a defined behaviour during system start-up o Off State → In Operation State / Initialisation State • SubSys Plan Execution shall have a defined behaviour during system initialisation o Initialisation State → Operation State • SubSys Plan Execution shall have a defined behaviour during handover of operation from one system instance to another system instance o Hot Standby State → Operation State • SubSys Plan Execution shall have a defined behaviour when recovering from Failure State o Failure State → Operation State • SubSys Plan Execution shall have a defined behaviour if it goes back to operation after planned maintenance o Maintenance State → In Operation State

		• SubSys Plan Execution shall be able to exchange System- / Operational State with the Planning System and adjacent RCA SubSys.

4.4.3. Functional Requirements

This section will be further elaborated in future releases.

SubSys Plan Execution shall calculate and check its internal data model based on the Configuration- and Operational Data received via its interfaces. In case of determined inconsistencies, it shall be able to synchronize and re-synchronize the state of Configuration- and Operational Data with the connected system.

Function	Subfunction	Functional Requirement
Calculate and check internal data model	-	tbd
Synchronise the state of Configuration Data and Operational	-	tbd

4.4.4. Description

4.4.4.1. Terms

Following term and definitions are proposed:

Term	Definition
Robustness	➔ ***Term Robustness*** *In computer science, robustness is the ability of a computer system to cope with errors during execution and cope with erroneous input. In fault-tolerant computer systems, programs that are considered robust are designed to continue operation despite an error, exception, or invalid input, instead of crashing completely.*
Fault tolerance	➔ ***Term Fault tolerance*** *Is the property that enables a system to continue operating properly in the event of the failure of one or more faults within some of its components. If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively designed system, in which even a small failure can cause total breakdown.*
High availability	➔ ***Term: High availability*** *High availability (HA) describes the ability of a system to guarantee operation with a high probability (often 99.99% or better) despite the failure of one of its components. In contrast to fault tolerance, operation can be interrupted in the event of a fault.*
CAP Theorem	➔ ***Term: CAP Theorem*** *The CAP Theorem or Brewers Theorem states that in a distributed system it is impossible to simultaneously guarantee the three properties Consistency, Availability and Partition Tolerance.*
Eventual Consistency	➔ ***Term: Eventual Consistency*** *Eventual consistency is a consistency model used in distributed computing to achieve high availability that informally guarantees that, if no new updates are made to a given data item, eventually all accesses to that item will return the last updated value.*

Event Sourcing	➔ ***Term: Event Sourcing*** *Event sourcing is a persistence mechanism in which the current state of an entity is not stored directly, but in the form of a chronological list of all its changes (domain events) starting with its creation, the so-called event stream.*

4.4.4.2. CAP Theorem

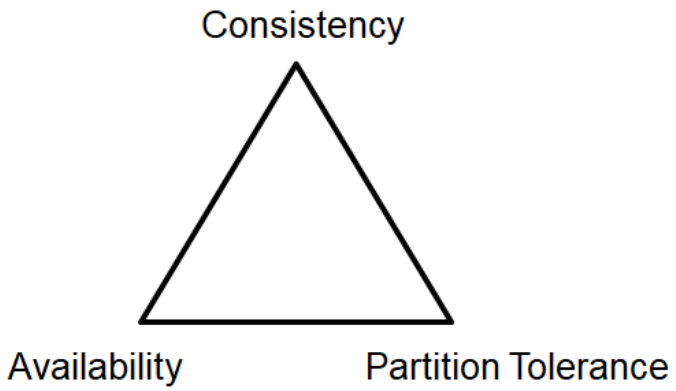


Figure 14: CAP Theorem

According to the CAP Theorem, a distributed system can satisfy two of the following properties simultaneously, but not all three.

- **Consistency:**
  The system is consistent in the sense of the CAP Theorem if all nodes of the distributed system see the same data. In contrast to Eventual Consistency, the term Consistency used here is also called Immediate Consistency or Strong Consistency.
- **Availability:**
  The system is available when it responds to all requests.
- **Partition tolerance:**
  The system is partition tolerant if it remains functional if communication with some of the nodes is interrupted.

4.4.4.3. Assumption

The distributed system RCA is assumed to use asynchronous messaging, to be highly available and to guarantee partition tolerance to be fault tolerant and robust. Consequently, the RCA system, according to the CAP Theorem, will “only” be able to guarantee Eventual Consistency for the data (especially Operational Data) hold, provided and exchanged by the different RCA SubSys within RCA.

Eventual Consistency means that consistency of data hold by the different RCA SubSys will not be guaranteed at any time, but that it will be provided eventually. To guarantee this Eventual Consistency, RCA shall provide certain synchronisation mechanism used by its SubSys to detect inconsistent, asynchronous data and to re-synchronise this data to guarantee its consistency.

Such a synchronisation mechanisms could use Event Sourcing as an architectural pattern to enable RCA SubSys to bridge the potential gap between their internally hold data and the data hold by other/adjacent RCA SubSys.

To implement such a synchronisation mechanisms on base of Event Sourcing could for example mean, that at fixed times (e.g. every 30s), a snapshot of the event stream is created for each interface containing all messages/events provided within that time period. These snapshots can then be used by the consuming SubSys to re-synchronise their internal data by re-consuming all missing snapshots since the failure occurred.

4.5. Functional Concept: Alarms regarding hazardous situations

It is currently being assessed whether the listed Objectives, System Requirements and Functions Requirements are within or outside the scope of SubSys Plan Execution.

This section will be further elaborated in future releases.

4.5.1. Objective and System Requirements

The applicable objectives and system requirements can be found in *Concept: Plan Execution, RCA.Doc.47*, chapter: *Objectives and System Requirements*.

4.5.2. Functional Requirements:

- SubSys Plan Execution shall receive, aggregate and provide alarms on hazardous situations that require only a basic integrity safety level.
- SubSys Plan Execution shall provide a generic (non-country specific) interface (hereafter referred to as PE Alarm Interface) that can be used to connect a proprietary alarm system.
- The PE Alarm Interface shall be capable of receiving the following categories of alarms (Hint: Both categories of alarms can be reported by stationary sensors or sensors located on the train unit)
    - Train specific alarms (e.g. hot box detection, fouling of gauge).
    - Location-specific alarms (e.g. avalanche / landslide sensors, short circuit in catenary sections).
- The PE Alarm Interface shall not implement country specific details on alarms but provide possible extension points.
- SubSys Plan Execution shall aggregate alarms to domain objects of the Operating State and pass them to the Planning System via SCI-OP and to SubSys Workbench via SWI-PE.
- SubSys Plan Execution shall take mitigating measures for hazardous situations that require only a basic integrity safety level.
- Depending on the position of the trains and the position/area affected by the alarm, SubSys Plan Execution shall take automated mitigating actions, such as
    - Request the emergency brake of a Physical Train Unit
    - Request the limitation the maximum speed of a Physical Train Unit
    - Request the extension or shortening of a MP
    - Request the safety responsibility mode of a train unit to "on sight”
    - Request a different drivability or flank protection state of a DPS
    - Request the creation of a Usage Restriction Area
    - Implement the Operational Plan only up to the respective intervention station and not beyond.

Out of scope:

- Alarms and hazards requiring a higher safety integrity level than basic integrity shall be reported by safety critical SubSys such as Object Controller (OC) and processed by APS and mitigated by SubSys SM Safety Manager in APS.
- Both the alarms and the mitigation actions to be performed are assumed to be country specific but shall not be implemented country specific in SubSys Plan Execution or the PE Alarm Interface.

Further information:

- Today's alarms reported by systems such as SBB-ZKE (Train Observe Units) or SBB-TAG (Tunnel Automatisation Gotthard) only require SIL-0, as the actual risk of a hazard is already low and can be reduced to an acceptable level with a SIL-0 application.
- It is assumed that it makes sense to automate certain mitigating actions in SubSys Plan Execution and not delegate every alarm to a human safety actor at the SubSys Plan Execution GUI of SubSys Workbench.

- It is assumed that mitigating actions will not be handled in the Planning System, as SubSys Plan Execution can react faster and directly to alarms. The Planning System is responsible for rescheduling train movements and planning further measures to circumvent or resolve the situation.

5. Concept: High level migration scenarios with Plan Execution

5.1. Objective and System Requirements

The applicable objectives and system requirements can be found in *Concept: Plan Execution, RCA.Doc.47*, chapter: *Objectives and System Requirements.*

5.2. Introduction

The RCA document "RCA Migration"(RCA.Doc.28 Version: 2.1) describes the different migration scenarios for different railways in Europe. The migration concept shows the different migration steps as so-called plateaus. The time periods in which the different railway infrastructure operators reach these migration plateaus are just as different as the individual technical migration steps.

These technical migration steps are so different mainly because each country has used different, non-uniform technical Centralised Traffic Control systems (CTC) at the train control technology level in the past and different strategies performing migration triggered via Traffic Management System first and/or the Signalling equipment like exchange of legacy interlocking and Radio Block Centre (RBC).

In a migration step between the defined plateaus (see "RCA Migration", RCA.Doc.28 Version: 2.1), long phases of hybrid situations arise for periods of 15-25 years for this reason. A coexistence of the different plateaus, as described in the migration concept, forms the transitions of the plateaus for a long time.

From the perspective of the Infrastructure Manager (IM), this long transition phase raises the question of how these hybrid situations can best be managed. For the migration out of the existing legacy systems, each country has its own individual path due to its specifically grown structure. Nevertheless, the central question is where and on which technical level a migration bracket of the old and the new systems can take place that enables the (IM to carry out a cost-effective and harmonised migration step.

In the following, we will look at some variants that address this issue on a high-level perspective. Whether and to what extent individual countries choose other migration scenarios for themselves is, of course, up to each country.

5.3. Migration scenarios in hybrid states

In general, a common scenario of the migration phase in hybrid system states is to separate old and new systems strictly by geographical boundaries. However, the further up the architecture of a railway production system is moved in the hierarchy in the architecture layers, the integration of existing legacy technology and new RCA building blocks occurs.

Basically, three levels can be identified for this integration:

1. the timetable layer
2. the dispatching layer
3. the CTC layer

5.3.1. Integration via the timetable layer

The scope of the timetable system refers to the planning of offers (1-3 years before the train journey), the ordering processes and the planning/derivation of the daily timetable (24/48 hours before the journey). The constant adaptation to changing topology and the non-exact knowledge of microscopic topology and rolling stock characteristics over a period of > 2 years until the actual journey (24h) already present the timetable software with very complex tasks.

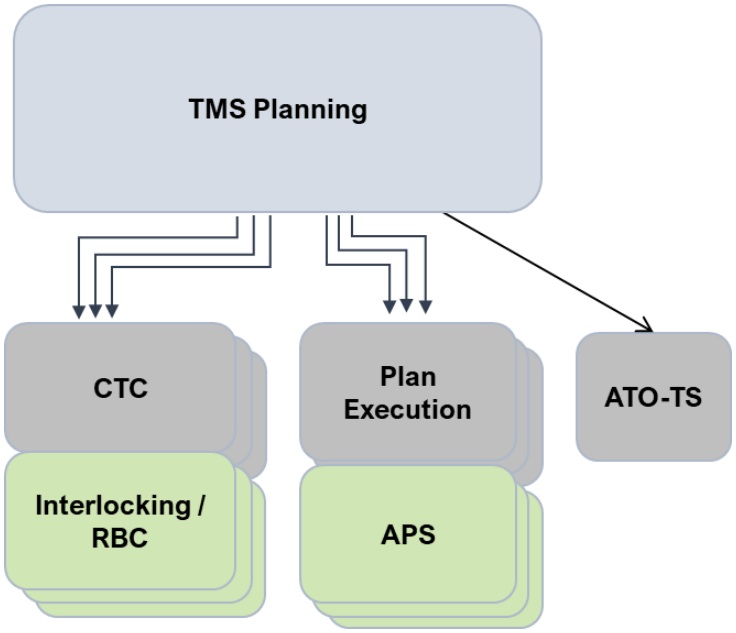


Figure 15: General connection of the RCA architecture to a timetable system

An integration via the timetable layer requires a good ARS (Automatic Route Setting) layer after the generation and permanent adaptation to the real time events, which automates the train routing derived from the current daily planning (see also 5.3.4).

All in all, the scope that a timetable system must cover in order to function as an integration layer appears to be enormous. At all levels in the architecture below the timetable, two worlds of operations and work processes would exist that would have to deal with completely different operating philosophies and systems in a hybrid system phase of more than 15 years. Therefore, integration via the timetable layer does not appear to be very practicable on closer inspection.

5.3.2. Integration via the dispatching Layer

In today's RCA representation, the TMS is found as a large system block outside the standardised RCA architecture. The scope of TMS is enormous, ranging from a complete timetable system to the control of CTC or interlocking systems.

This results in a large scope of TMS, which cannot be solved in one system, but also requires fine architectural layering. Regardless of the granular layering of the TMS, the architecture of the TMS plays a central role in the migration capability of the different plateaus from the migration concept, which can make the long periods of hybrid transition phases possible.

5.3.3. Migration to the differently defined plateaus of the RCA takes place in the TMS or in the CTC level

This is not only technically significant, but also of elementary importance for the essential work of the Infrastructure Manager. For not only do the systems have to exchange themselves technically in transition phases of more than 20 years, but the operational processes also have to be uniform in these phases if possible and not fundamentally different. In order to master the different technical system states and to control them as integratedly and centrally as possible via suitable systems, only systems that are able to

a. abstract different technologies centrally
b. provide uniform user interfaces
c. cover the entire rail network of the infrastructure manager in one system in order to avoid media breaks

will cover the needed functionality in hybrid states.

One of the two levels that can fulfil these conditions at all is the system Dispatching in real time in the TMS. The implementation of a daily updated timetable in a dispatching system, which monitors the real time train movements, takes over forecasts and conflict detection as well as their solution, is a central component for the management of dense train traffic systems.

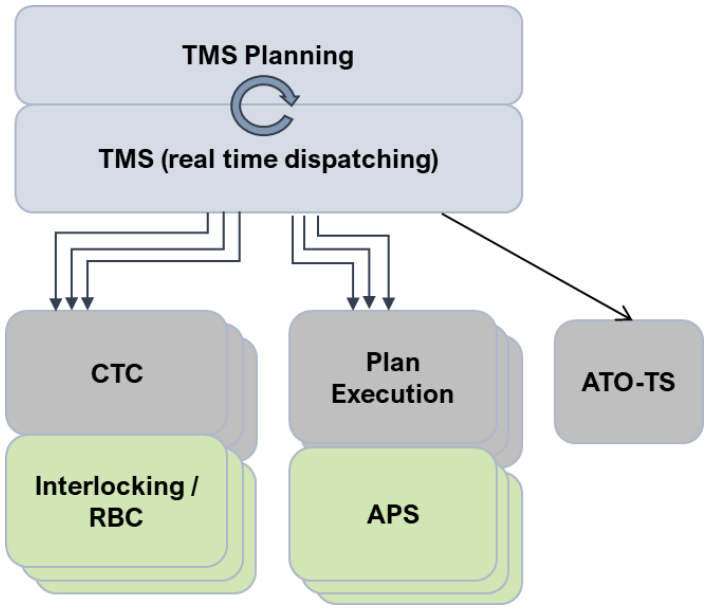


Figure 16: General connection of the RCA architecture to a dispatching system

Although the methods and measures in a real time dispatching system are similar to timetable construction, the goals (optimisation function) of both systems are fundamentally different.

The basic idea of a TMS dispatching system is deviation management with the optimisation criterion timetable. For this purpose, it works as a real time system and, if possible, on a daily updated micro topology, which largely corresponds to that of the underlying interlocking systems. The optimisation criterion of the timetable is the capacity of the entire network.

In contrast to integration via the timetable, the dispatching component of the TMS controls the train traffic in real time, works with up-to-date topology, current rolling stock properties and is therefore predestined to be a central switch for the migration to APS.

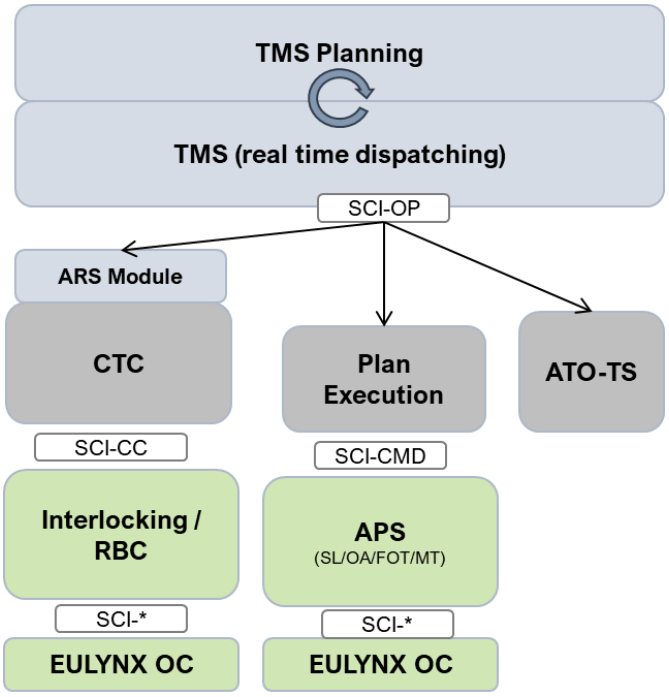


Figure 17: Specific connection of the RCA architecture to a dispatching system

A closer look at this migration option shows that the important connection to any ATO systems can also be controlled by the dispatching TMS system. Thus, the SCI-OP interface, whose most important component is the train's running speed profile, which is accurate to the second, can also be transferred to the corresponding ATO-TS modules. Only if the speed profile information for an ATO system and the output of an interlocking system, in particular APS/Plan Execution, is synchronized and has exactly the same information, ATO can be operated in a sustainable and optimal way in a dense network.

In individual cases, the testing of ATO (GOA-2 to GOA-4) is currently operated in Europe as isolated solutions, which can possibly also be adapted to the current operational situation on individual routes with appropriate converters of the timetable data. As isolated solutions, however, these projects do not exist for the long-term strategy of operating the entire rail network with ATO, because they are generally tailored to the individual application and do not scale. In addition, the present ATO solution might be referenced to legacy/current operational procedures and technical solution in signalling being changed withing the next years. Only a comprehensive TMS dispatching system based on redefined operational procedures, which monitors and controls the entire rail network, can enable a large-scale roll-out of ATO in the future.

The decisive factor in this architecture variant is that the entire network is controlled by a single TMS dispatching layer and conveys identical, optimised (energy-optimised and conflict-adjusted) driving profiles to all participating executive systems via the homogeneity of a single interface (SCI-OP).

The control of the existing legacy interlocking systems is also carried out from the TMS dispatching layer by means of an "automatic route setting" (ARS) module, which converts the travel profile of the SCI-OP interface to the block-based logic of routes. The complexity of such an ARS module is high and if it is not already available to control legacy interlockings, then it remains for everyone to consider whether the development of a country-specific ARS module to control from the TMS is necessary.

For Infrastructure Managers who already have integrated systems that use ARS from the TMS dispatching level to directly control existing interlocking systems, upgrading the TMS dispatching system to a system that is ATO and moving block capable is the ideal migration path to master the hybrid phase towards a complete RCA architecture.

In particular, the uniform user interfaces for harmonised operating processes, the possibly already uniformly existing modules for conflict resolution, ATO control and energy-optimised operation speak in favour of a network-wide integration variant in the TMS dispatching layer.

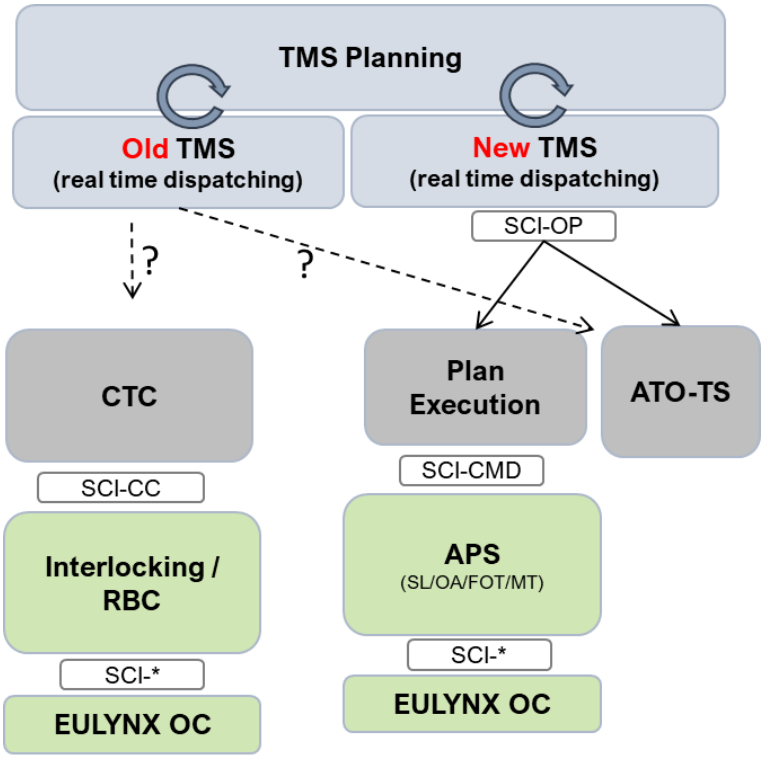


Figure 18: Specific connection of the RCA architecture to a dispatching system

Infrastructure Managers who do not have a full-featured TMS dispatching layer today or in the foreseeable future may consider the option of developing a new TMS dispatching system. This way, the new TMS controls the new RCA architecture and the legacy systems (including “old TMS”) remain in their current state.

The separation takes place with two different dispatching systems (TMS), which means two separate operating worlds, both in terms of technology and all operating processes. Ultimately, the integration then takes place at the highest level in the timetabling system (see above). Whether an infrastructure manager can implement this strict separation between new RCA architecture and existing systems over a period of at least 20 years for the full expansion within his organisation and the working methods of the dispatchers and schedulers remains at least questionable. Therefore, if a well-developed TMS dispatching system exists today, it should be examined whether the expansion to a system that can support the future ATO systems being standardised and the RCA architecture is not the optimal migration component.

Especially from the point of view of homogeneous control of the network from a uniform system as well as the possibility of uniform user interaction would be the advantages already mentioned. Nevertheless, Real Time Dispatching Systems like TMS of this quality are not to be found at all Infrastructure Managers in Europe or to be set up there in a short time.

5.3.4. Integration via the CTC layer

Migration via the TMS layer (Dispatching real time TMS) requires a very advanced TMS. In particular, the real time capabilities such as network-wide train sequence prediction, precise conflict detection on micro topology as well as the dynamic adaptation of running train speed profiles for the homogeneous output of these to ATO (Train CAB) and the RCA module "Plan Execution" (PE).

For infrastructure managers (IM) with strong CTC structures, which have been optimised over a long period of time and have ARS (Automatic Route Setting) capabilities, a valid migration path can also be shown, which can bridge the long hybrid phases of different architectures in a uniform way.

Infrastructure managers with highly developed CTC landscapes also have the majority of operating personnel located in this layer. So that operational and work processes are not torn into two separate worlds and philosophies, the CTC layer must take over the integration for these infrastructure managers.

Two possible scenarios can be identified for migration from the CTC layer:

1. integration via an ARS module below the timetable system
2. integration from an existing CTC system

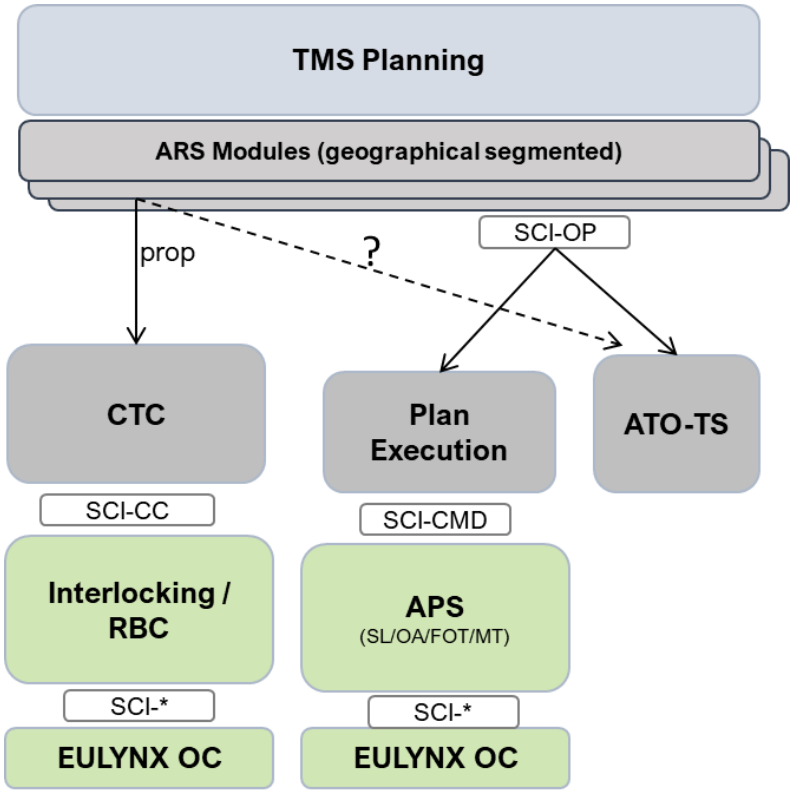


Figure 19: Integration via an ARS module below the timetable system

If the Infrastructure Manager already has a well-developed ARS module, which possibly controls geographically segmented different CTC systems, the idea of docking the new RCA architecture to this architecture layer is obvious.

It should be noted that the generation of RCA SCI-OP compatible driving profiles from a block-based logic of an ARS module is not easily possible. In addition to the control of "Plan Execution" via the SCI-OP interface, the ARS module must also synchronously transmit the driving profiles to a possible ATO system. Since today's ARS modules are all capable of block-based logic, there is a risk that the capacity advantages of optimally controlled moving block systems (PE/APS) will be lost when they are controlled from an ARS module.

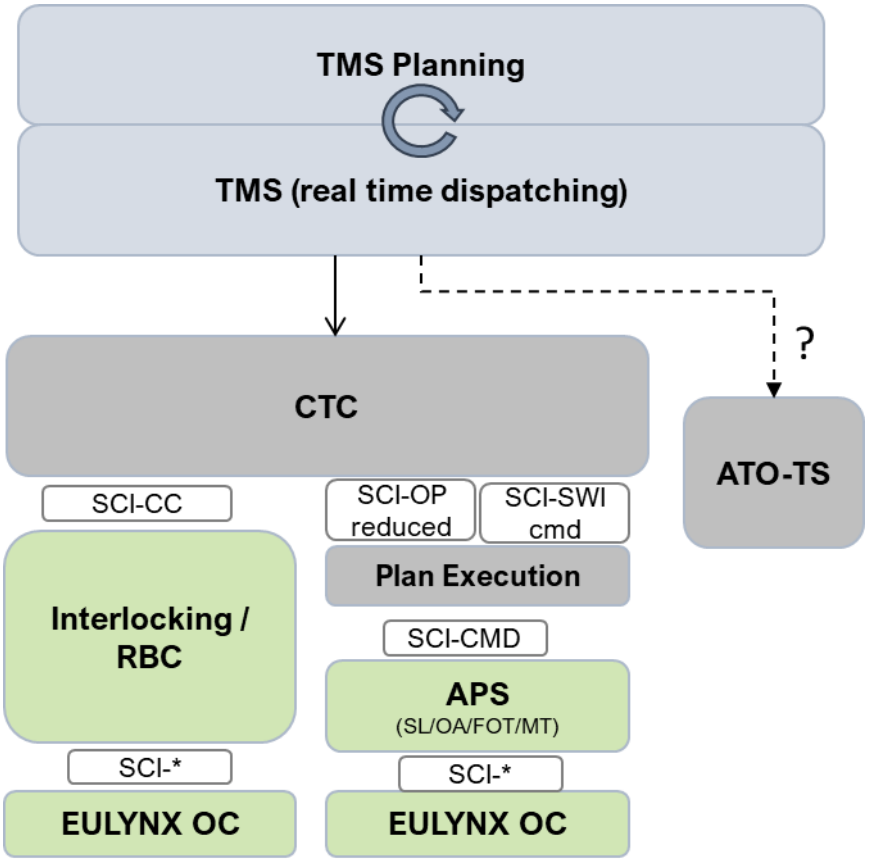


Figure 20: Extension of the existing CTC layer through a connection to SubSys Plan Execution

For Infrastructure Managers with highly developed CTC systems that do not have a strong TMS dispatching system, the RCA can be connected by extending the existing CTC systems.

In this case, the current logic of route requests is directly converted into a moving permission request and sent to the "Plan Execution" system for immediate execution by means of a reduced SCI-OP interface request. As with the variant via control by an ARS module, the capacity advantage of moving block may also be lost here due to the block-based logic of the CTC system. Direct commands from the existing CTC systems can be transferred directly to SubSys Plan Execution via the SWI-PE command interface. In this way, individual operations from the user interfaces of the CTC systems to the PE/APS system are also possible.

Even though capacity losses must be expected with this variant, it represents a transparent and valid solution for a hybrid state. Without having to change the essential operating philosophies in the CTC system, the expansion can be carried out downwards from the CTC system while maintaining the current system landscape. However, the question remains open as to how efficient ATO control is to be achieved with this model. Without efficient time-controlled management of the entire railway network, it will be difficult to achieve the advantages of moving block control systems with a CTC migration variant.

5.4. Conclusion

In principle, two migration models can be developed in hybrid states.

It is obvious that the timetable system is ruled out as the migration bracket for the hybrid phase (see above). What remains as a migration layer for the hybrid states is the real time layer of TMS (dispatching) and the CTC layer (control technology systems).

The migration is, as stated in the migration concept of the RCA, very much determined by the national architectures of existing railway technology systems. Each RCA member country has, on closer inspection, different system architectures due to different historical growth.

This ranges from very TMS-focused infrastructure managers with a corresponding automated CTC function (ARS), to IMs that have no or no dominant time-controlled traffic management system where the CTC layer has to carry the migration.

For Infrastructure Managers with a strong TMS dispatching layer, it is strongly recommended to extend the TMS dispatching layer homogeneously for the RCA. Uniform control of the PE/APS, legacy interlockings and ATO systems with one interface (SCI-OP) appears to be of great advantage.

For infrastructure managers with strong CTC architectures, the connection of "plan execution" with a reduced SCI-OP interface and direct commands through the SWI-PE interface is possible.

Whichever variant is chosen by the Infrastructure Manager, it is possible to work in hybrid system states over a long period of time with the introduction of moving block systems integrated according to the RCA, thus enabling a smooth migration.

6. Issues

Nr.	Title	Description
1	..	..