RCA

Reference CCS Architecture

An initiative of the ERTMS users group and the EULYNX consortium

Concept: Architectural Design Plan Execution

Document id: RCA.Doc.49

Table of Contents

List of Figures

List of Tables

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

Version History

Version	Date	Author	Description
0.1	2021-09-20	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	2021-11-30	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
0.3	2022-09-09	René Schönemann, Martin Kemkemer, Jens Wieczorek, Gabriele Löber, Rouzbeh Boloukian-Roudsari, Ulrich Schöni and Marcus Völcker as members of the RCA-OPE-Cluster	Draft version for RCA core group review (RCA BL1 R0)

1. Indroduction

1.1. Release information

Basic document information:
RCA.Doc.49
Concept Architectural Design Plan Execution
Cenelec Phase: 1
Version: 0.3
RCA Baseline set: BL1R0
Approval date: 2022-09-30

1.2. 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

1.3. 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.

1.4. 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.

1.5. 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.*

1.6. Related documents

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

- RCA System Architecture [RCA.Doc.35]
- RCA Terms and Abstract Concepts [RCA.Doc.14]
- Concept: SCI-OP [RCA.Doc.31]
- Concept: SCI-CMD [RCA.Doc.70]
- Concept: Plan Execution [RCA.Doc.47]

2. Terms and abstract concepts

2.1. Terms related to SubSys Plan Execution

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

Term	Description
Phase	Refer to 3.3.2
System State	Refer to 3.3.2
Operational State	Refer to 3.3.2
Mode of Operation	Refer to 3.3.2
Robustness	Refer to 3.4.3.1
Fault tolerance	Refer to 3.4.3.1
High availability	Refer to 3.4.3.1
CAP Theorem	Refer to 3.4.3.1
Eventual Consistency	Refer to 3.4.3.1
Event Sourcing	Refer to 3.4.3.1

2.2. Abstract concepts related to SubSys Plan Execution

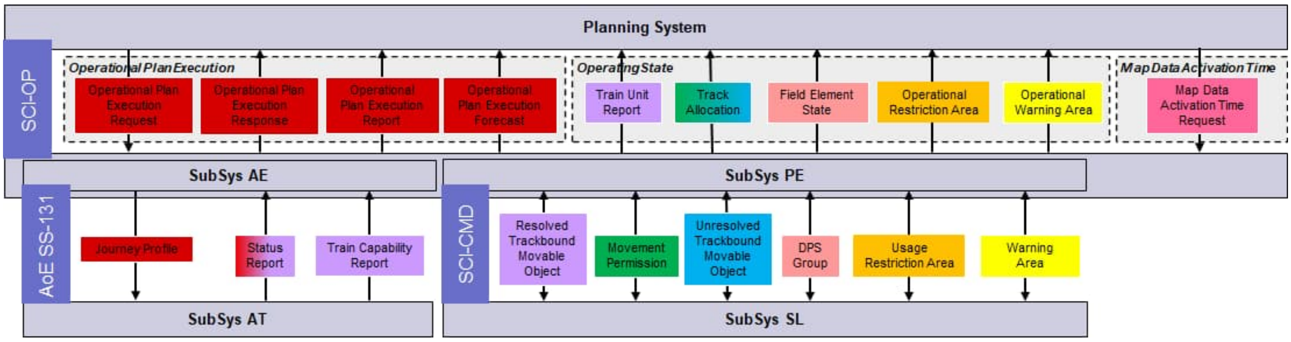


Figure 1: Overview of the main abstract concepts of SCI-OP and SCI-CMD

All definitions of abstract concepts related to SubSys Plan Execution can be found in RCA Terms and Abstract Concepts [RCA.Doc.14].

3. Functional concepts

3.1. Functional concept: Requesting of Operational Plans

This section will be further elaborated in future releases.

3.1.1. Objectives and System Requirements

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

3.1.2. 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.

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 (update 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 PE 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.

3.1.3. Design Principles

3.1.4. Operational Plan Operation Movement

3.1.4.1. 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 successive 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 Terms and Abstract Concepts [RCA.Doc.14])

➔ ***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 Terms and Abstract Concepts [RCA.Doc.14])

➔ ***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 Terms and Abstract Concepts [RCA.Doc.14])

3.1.4.2. 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. All subsequent scenarios are derived from this first scenario “Initial Plan”. Descriptions and illustrations are directly linked to scenario “Initial Plan”.

3.1.4.2.1. Scenario: Initial Plan (acceptance)

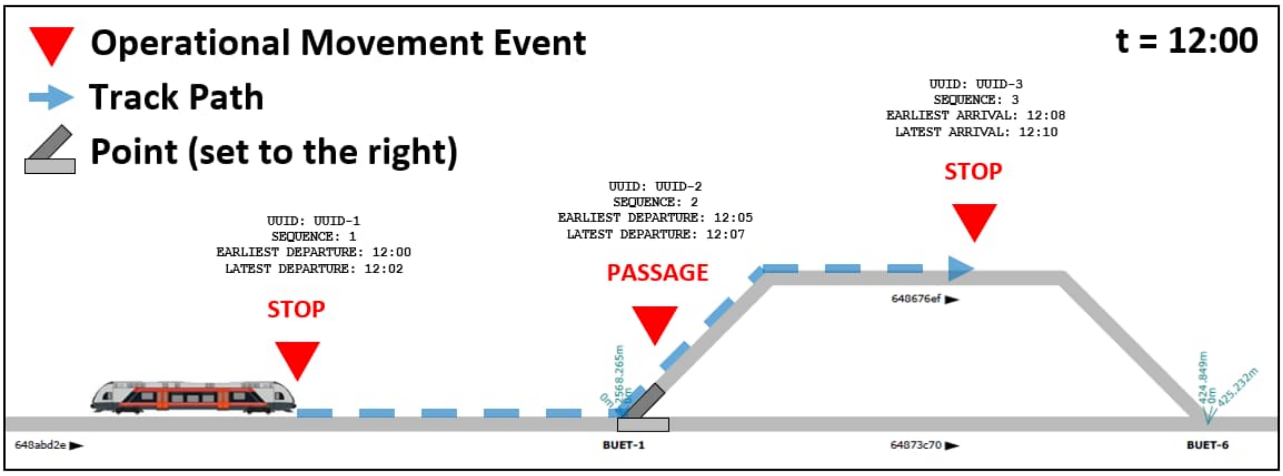


Figure 2: Schematic example of a feasible Operational Plan Operational Movement

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 defined feasibility checks (see 4.1.2.1.1) are all fulfilled.

Sequence Diagram

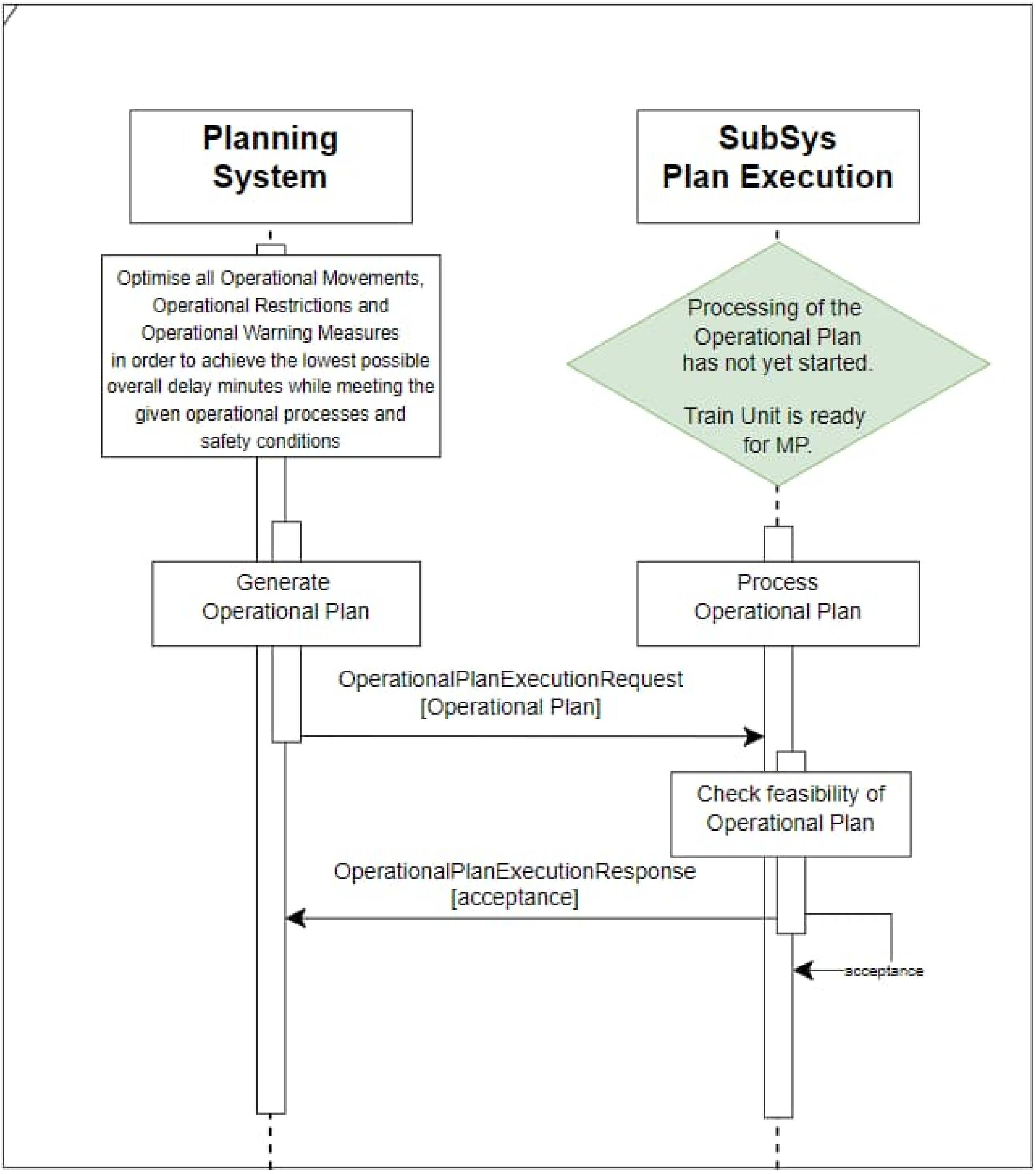

Planning System
SubSys Plan Execution
Optimise all Operational Movements, Operational Restrictions and Operational Warning Measures in order to achieve the lowest possible overall delay minutes while meeting the given operational processes and safety conditions
Processing of the Operational Plan has not yet started.
Train Unit is ready for MP.
Generate Operational Plan
Process Operational Plan
OperationalPlanExecutionRequest [Operational Plan]
Check feasibility of Operational Plan
OperationalPlanExecutionResponse [acceptance]
acceptance

Figure 3: Sequence Diagram: Initial Plan (acceptance)

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

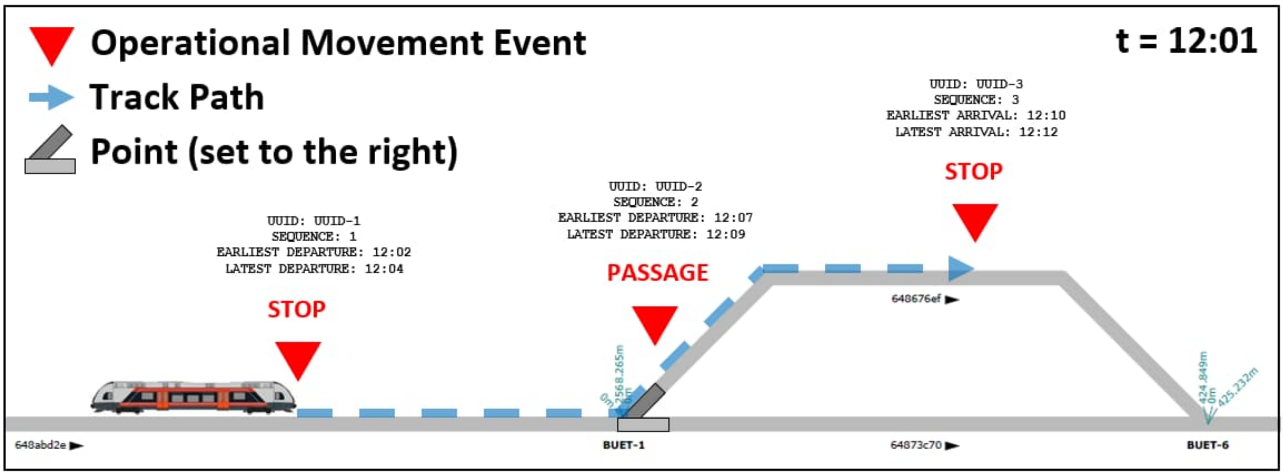


Figure 4: Schematic example of a feasible Operational Plan Operational Movement (all three Operational Movement Events have been rescheduled (+ 2 min) and have not been completed by SubSys Plan Execution)

Description of Scenario

The Scenario illustrates a complete update request of a previously requested Operational Plan Operational Movement (see 3.1.4.2.1), 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 3.1.4.2.1.). All feasibility checks (see **4.1.2.1.1**) are all fulfilled.

Sequence Diagram

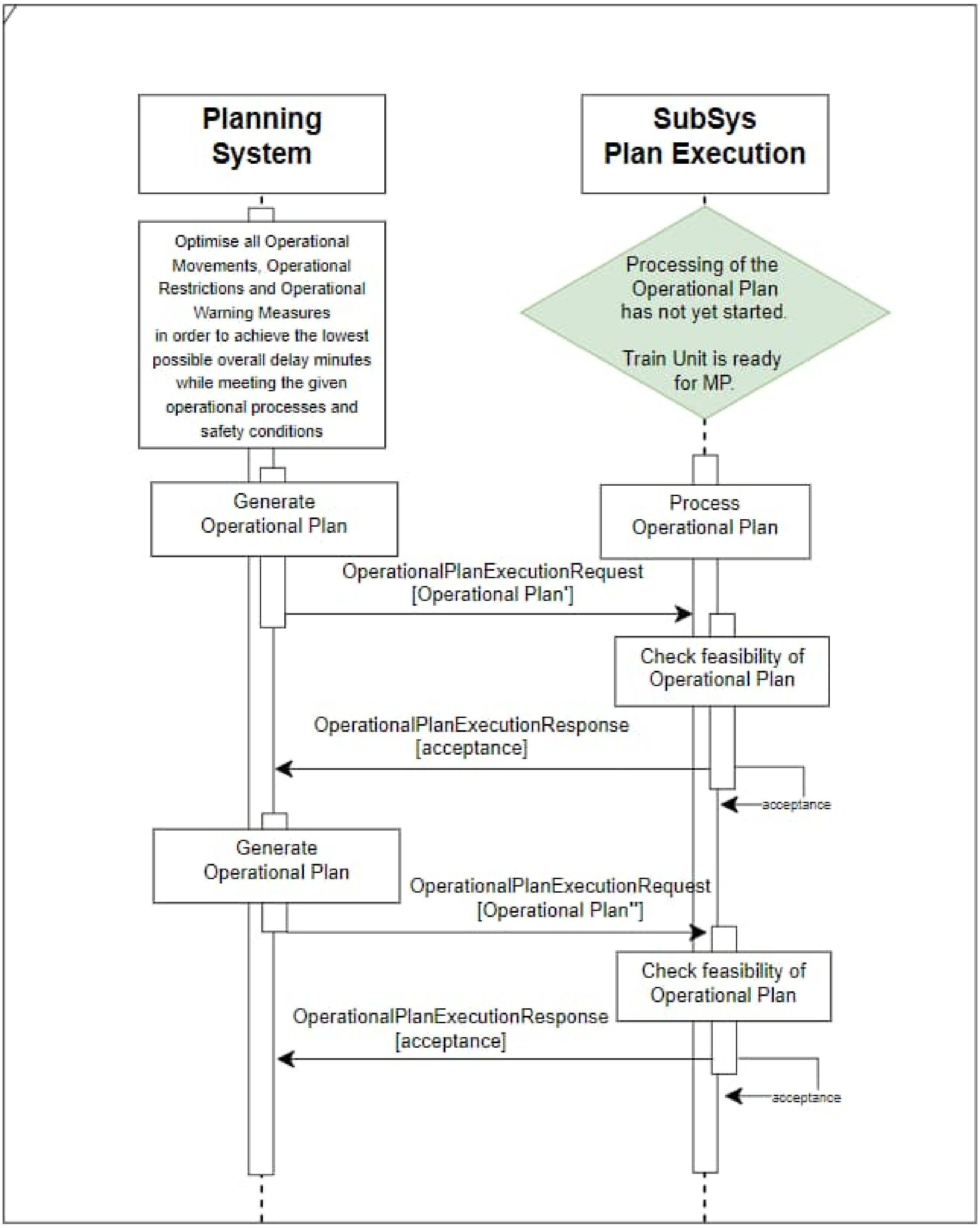

Planning System
SubSys Plan Execution
Optimise all Operational Movements, Operational Restrictions and Operational Warning Measures in order to achieve the lowest possible overall delay minutes while meeting the given operational processes and safety conditions
Processing of the Operational Plan has not yet started.
Train Unit is ready for MP.
Generate Operational Plan
Process Operational Plan
OperationalPlanExecutionRequest [Operational Plan']
Check feasibility of Operational Plan
OperationalPlanExecutionResponse [acceptance]
acceptance
Generate Operational Plan
OperationalPlanExecutionRequest [Operational Plan'']
Check feasibility of Operational Plan
OperationalPlanExecutionResponse [acceptance]
acceptance

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

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

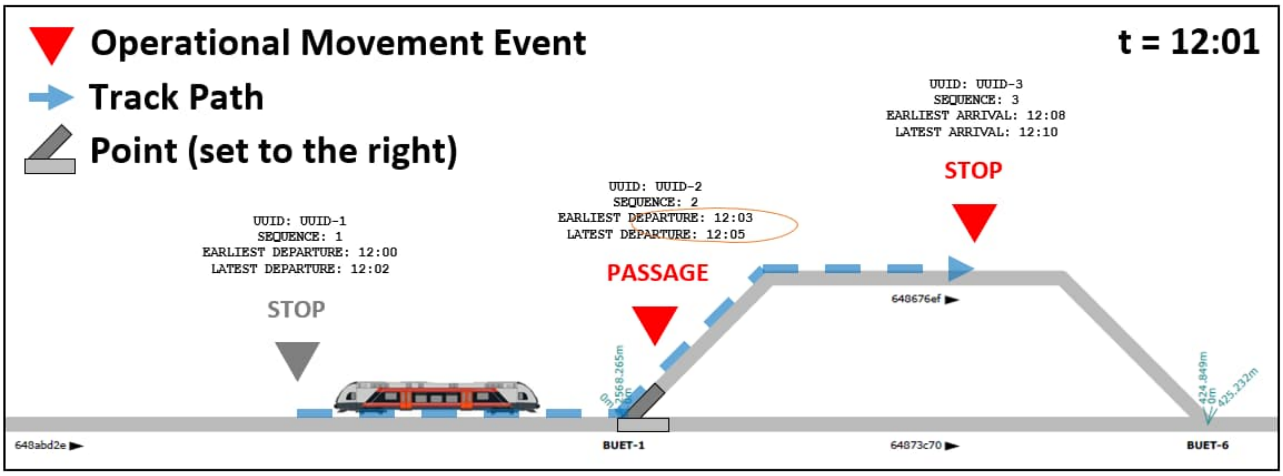


Figure 6: Schematic example of a feasible Operational Plan Operational Movement (first Stop Event has been completed, Passage Event has been rescheduled (- 1 min))

Description of Scenario

The Scenario illustrates an update request of a previously requested Operational Plan Operational Movement (see 3.1.4.2.1), 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 3.1.4.2.1.), but the first Stop Event would be ignored by SubSys Plan Execution. All feasibility checks (see **4.1.2.1.1**) are all fulfilled.

Sequence Diagram

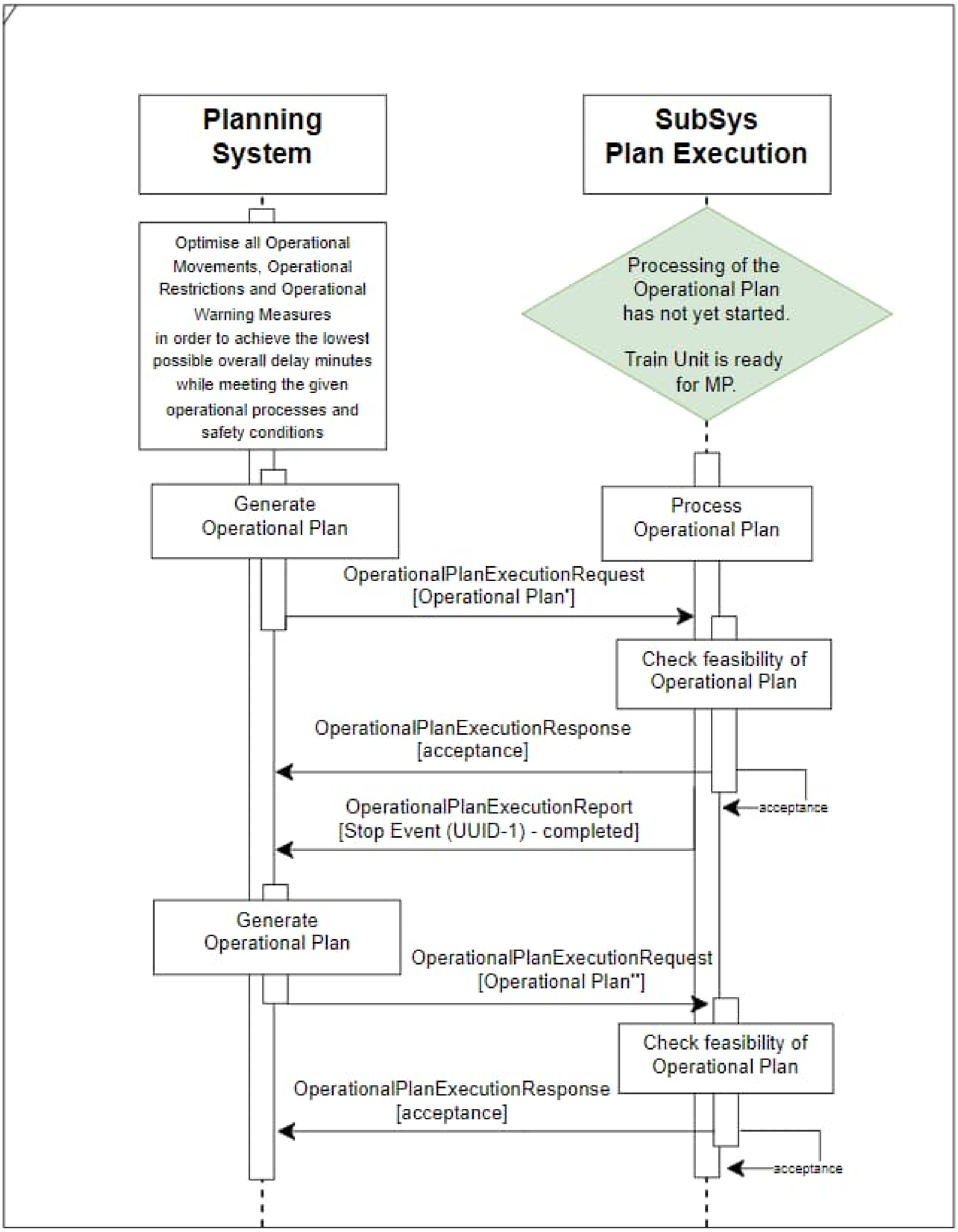

Planning System
SubSys Plan Execution
Optimise all Operational Movements, Operational Restrictions and Operational Warning Measures in order to achieve the lowest possible overall delay minutes while meeting the given operational processes and safety conditions
Processing of the Operational Plan has not yet started.
Train Unit is ready for MP.
Generate Operational Plan
Process Operational Plan
OperationalPlanExecutionRequest [Operational Plan']
Check feasibility of Operational Plan
OperationalPlanExecutionResponse [acceptance]
acceptance
OperationalPlanExecutionReport [Stop Event (UUID-1) - completed]
Generate Operational Plan
OperationalPlanExecutionRequest [Operational Plan'']
Check feasibility of Operational Plan
OperationalPlanExecutionResponse [acceptance]
acceptance

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

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

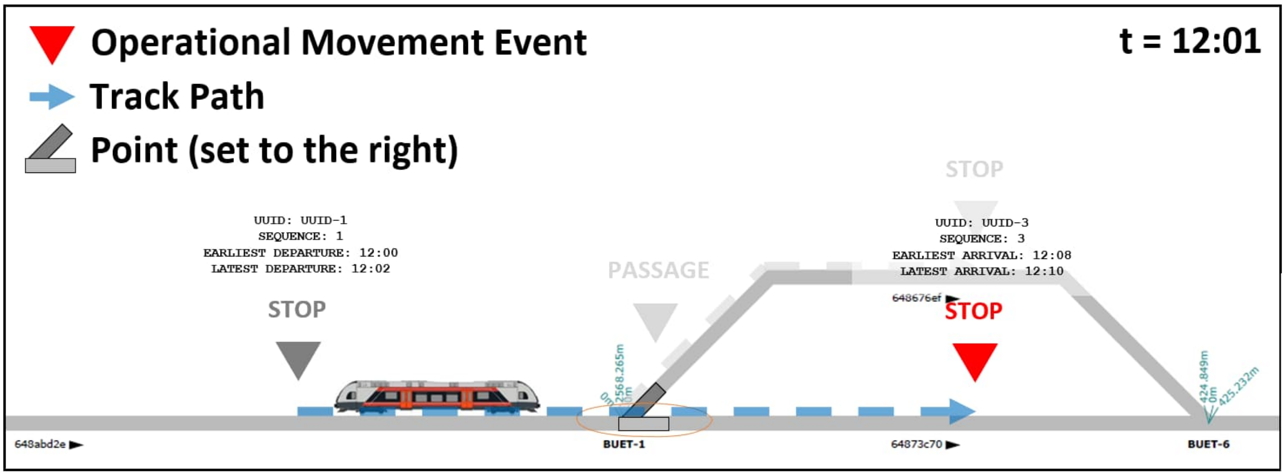


Figure 8: Schematic example of a feasible Operational Plan Operational Movement (first Stop Event has been completed, Passage Event has been removed, position of second Stop Event has been moved)

Description of Scenario

The Scenario illustrates an update of previously requested Operational Plan Operational Movement (see 3.1.4.2.1). 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 3.1.4.2.1.). All feasibility checks (see 4.1.2.1.1) are all fulfilled.

Sequence Diagram

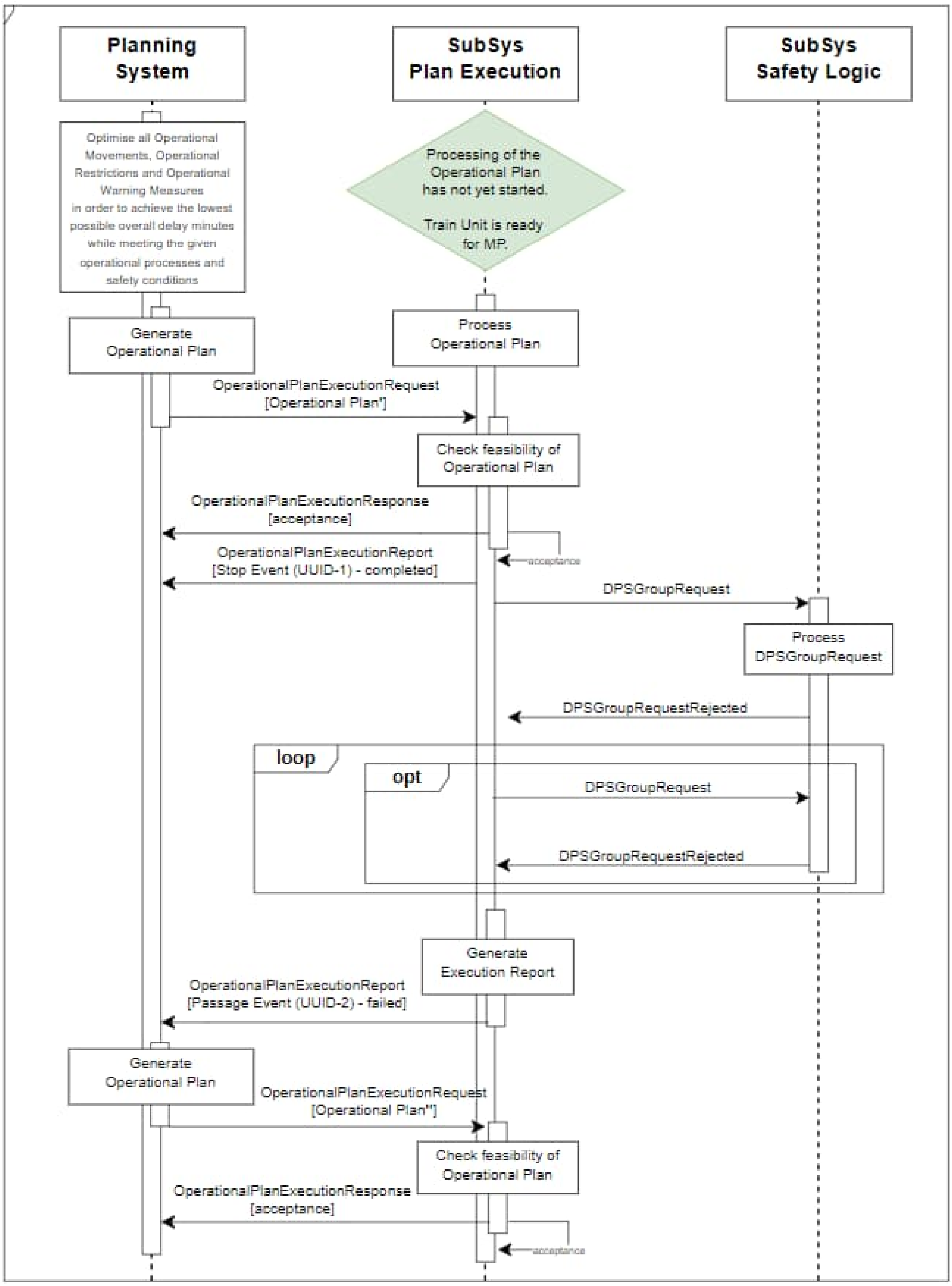

Planning System
SubSys Plan Execution
SubSys Safety Logic
Optimise all Operational Movements, Operational Restrictions and Operational Warning Measures in order to achieve the lowest possible overall delay minutes while meeting the given operational processes and safety conditions
Processing of the Operational Plan has not yet started.
Train Unit is ready for MP.
Generate Operational Plan
Process Operational Plan
OperationalPlanExecutionRequest [Operational Plan']
Check feasibility of Operational Plan
OperationalPlanExecutionResponse [acceptance]
acceptance
OperationalPlanExecutionReport [Stop Event (UUID-1) - completed]
DPSGroupRequest
Process DPSGroupRequest
DPSGroupRequestRejected
loop
opt
DPSGroupRequest
DPSGroupRequestRejected
Generate Execution Report
OperationalPlanExecutionReport [Passage Event (UUID-2) - failed]
Generate Operational Plan
OperationalPlanExecutionRequest [Operational Plan'']
Check feasibility of Operational Plan
OperationalPlanExecutionResponse [acceptance]
acceptance

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

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

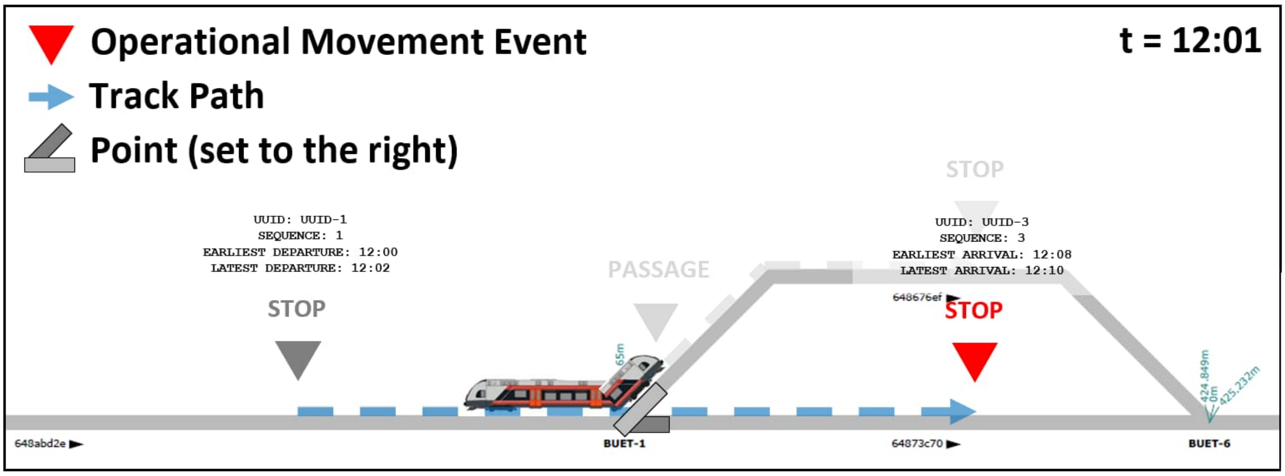


Figure 10: Schematic example of a non-feasible Operational Plan Operational Movement (first Stop Event has been completed, Passage Event has been removed, position of second Stop Event has been moved)

Description of Scenario

The Scenario illustrates an update of previously requested Operational Plan Operational Movement (see section 3.1.4.2.1). 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 3.1.4.2.1.), because the following of the feasibility check is not fulfilled.

Function	Subfunction	System Requirement
Process Operational Plan	Check feasibility of Operational Plan Operational Movement	• SubSys Plan Execution shall reject a requested update of an Operational Plan Operational Movement, if the referenced rMOB is partly located on the specified Track Path, but the front end of the referenced rMOB (in direction of travel) is not located on the Track Path.

Sequence Diagram

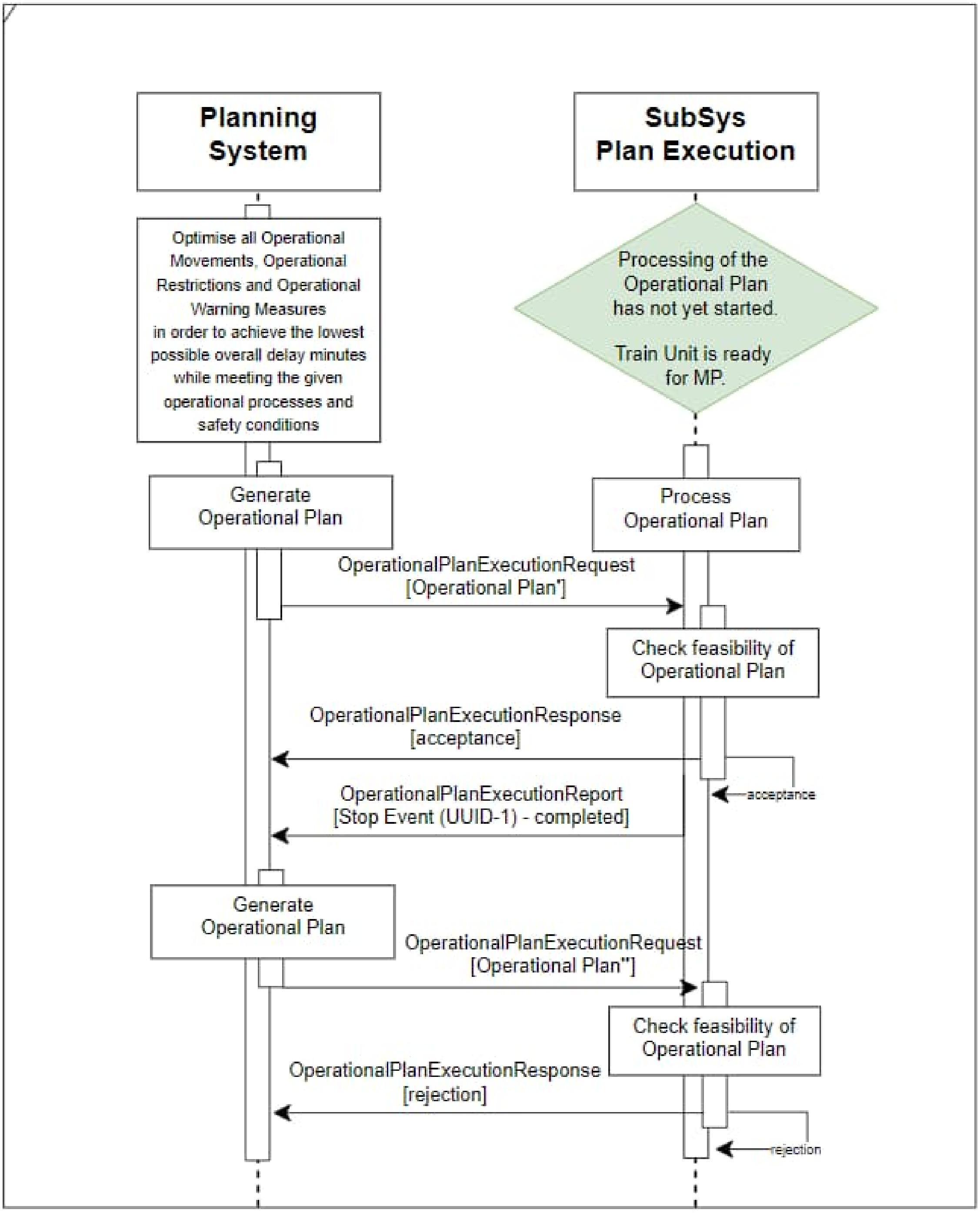

Planning System
SubSys Plan Execution
Optimise all Operational Movements, Operational Restrictions and Operational Warning Measures in order to achieve the lowest possible overall delay minutes while meeting the given operational processes and safety conditions
Processing of the Operational Plan has not yet started.
Train Unit is ready for MP.
Generate Operational Plan
Process Operational Plan
OperationalPlanExecutionRequest [Operational Plan']
Check feasibility of Operational Plan
OperationalPlanExecutionResponse [acceptance]
acceptance
OperationalPlanExecutionReport [Stop Event (UUID-1) - completed]
Generate Operational Plan
OperationalPlanExecutionRequest [Operational Plan'']
Check feasibility of Operational Plan
OperationalPlanExecutionResponse [rejection]
rejection

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

3.1.5. Operational Plan Operational Restriction

This section will be further elaborated in future releases.

3.1.6. Operational Plan Operational Warning Measure

This section will be further elaborated in future releases.

3.1.7. 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.
- Versioning format could be aligned to ISO 8601.

3.1.8. 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.
- 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.

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

This section will be further elaborated in future releases.

3.2.1. Objectives and System Requirements

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

3.2.2. Introduction

This concept describes the functionality according to which an MP is generated. The functionality defines how long the MP Extent is and at what time an MP is requested via SCI-CMD. It also describes at what time the driveability state of a DPS-Group (Field Element) is requested via SCI-CMD.

3.2.3. Functionality for calculation of MPs and triggering of requests via SCI-CMD

3.2.3.1. MP Limit Marker

➔ ***Abstract concept: MP Limit Marker***
A MP Limit Marker defines the maximum position on the Track Path to which the next MP can be extended upon request. A MP Limit Marker is described by one-to-many MP Limit Marker Constraints and is active until all its MP Limit Marker Constraints were resolved.

3.2.3.2. MP Limit Marker Constraint

➔ ***Abstract concept: MP Limit Marker Constraint***
A MP Limit Marker Constraint describes one constraint to be resolved before dissolving a MP Limit Marker.

Examples of MP Limit Marker Constraints (list is not exhaustive)

- A Field Element is not in the required state to implement the requested Track Path.
- A Constraint specified in the Operational Plan Operational Movement is not fulfilled
  - Earliest departure/arrival time
  - Train run sequence given by Operational Event Link.

3.2.3.3. Request Trigger

➔ **Abstract concept: Request Trigger**
A Request Trigger describes a point in time when a request should be sent via SCI-CMD from SubSys Plan Execution to the SubSys Safety Logic. A Request Trigger is described by one-to-many Request Trigger Constraints and is performed when all Request Trigger Constraints are resolved.

Examples of a Request Trigger:

- Trigger to request the driveability state of a DPS-Group (Field Element) via SCI-CMD

3.2.3.4. Request Trigger Constraint

➔ ***Abstract concept: Request Trigger Constraint***
A Request Trigger Constraint describes one constraint to be fulfilled before executing a Request Trigger.

Different types of Request Trigger Constraints:

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

3.2.3.5. Working principle: MP Limit Marker, Request Triggers, Request Trigger Constraints

The following event types result in updating of MP Limit Markers and Request Triggers:

- Update of Operational Plan Operational Movement (sent by the Planning System via SCI-OP or SubSys Workbench via SWI-PE).
- Update of rMOB state and position (send by SubSys Safety Logic via SCI-CMD).
- Update of MP (send by SubSys Safety Logic via SCI-CMD).
- Update of drivability state of DPS-Group (send by SubSys Safety Logic via SCI-CMD).
- Passed timing point (Operational Movement Event)

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

- For each event, check whether new MP Limit Markers and Request Triggers must be created or whether existing MP Limit Markers and Request Triggers need to be removed.
- For each event, check whether existing MP Limit Marker Constraints and Request Triggers are resolved.

- Execute all Request Triggers whose Request Trigger Constraints are resolved for the first time.
- Dissolve all MP Limit Markers whose MP Limit Marker Constraints are resolved for the first time.
- Calculate and request a MP update with the front extension of the MP to the next active MP Limit Marker.

3.2.4. Calculation of the MP Extent

- A MP Extent is always requested along the exact track specified in the Operational Plan Operational Movement. The MP Extent 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 Extent, then the MP Extent is requested with the maximum possible length (if the maximum possible length of a MP Extent 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)

3.2.4.1. Dissolving of a MP Limit Marker

Active MP Limit Marker	**Dissolved MP Limit Marker**
• A DPS-Group is not in the required drivability state to implement the requested Track Path.	• A DPS-Group is in the required state to implement the requested Track Path.
• Another MP or rMOB is on the requested Track Path.	• MP or rMOB have moved and are now behind another active MP Limit Marker. • The rMOB fulfils the required conditions for its own MP to overlap with another rMOB or MP.
• A Usage Restriction Area is on the requested Track Path.	• The Usage Restriction Area was removed. • The rMOB fulfils the required conditions for its own MP to overlap with the Usage Restriction Area.
• A Constraint specified in the Operational Plan Operational Movement is not fulfilled o Earliest departure/arrival time. o Train run sequence given by Operational Event Link.	• The Constraint specified in 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 rMOB has passed the position).

3.2.4.2. Triggering of MP Updates

The objective is:

- To extend the MP at front of the rMOB whenever possible

The following events will trigger SubSys Plan Execution to request a MP-Update.

- Message of position update of the rMOB
- The MP Limit Marker in front of a rMOB 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 extended?

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

3.2.5. Triggering of Field Element States

To implement the requested Track Path of an Operational Plan Operational Movement, SubSys Plan Execution needs to set the drivability state of DPS-Groups (Field Elements) on the Track Path into the required state.

Therefore, SubSys Plan Execution creates Request Triggers for all DPS-Groups on the Track Path. For an efficient implementation these Request Triggers are assigned with Request Trigger Constraints. At the time when all Request Trigger Constraints of the Request Triggers are resolved, the required driveability state of the DPS-Group is requested by SubSys Plan Execution via SCI-CMD.

3.2.5.1. Calculation of Pass Position Constraint

➔ ***Abstract concept: Pass Position Constraint***
The Pass Position Constraint defines the position on the track where the driveability state of a DPS-Group must be requested by SubSys Plan Execution at the latest when the rMOB passes this position to receive an MP over the DPS-Group without driving into the current MPs braking curve.

The calculation of the Pass Position Constraint depends on various factors:

- Position to which the rMOB can move if the corresponding DPS-Group is not in the required driveability state
- 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 rMOB comes to a stop at the end of the existing MP
- Distance travelled during the setting time (t_setting) in which the DPS-Group is brought into the required driveability 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 Pass Position Constraint (Trigger Position) not only the braking distance of the Physical 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 parameters of the Physical Train Unit.

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

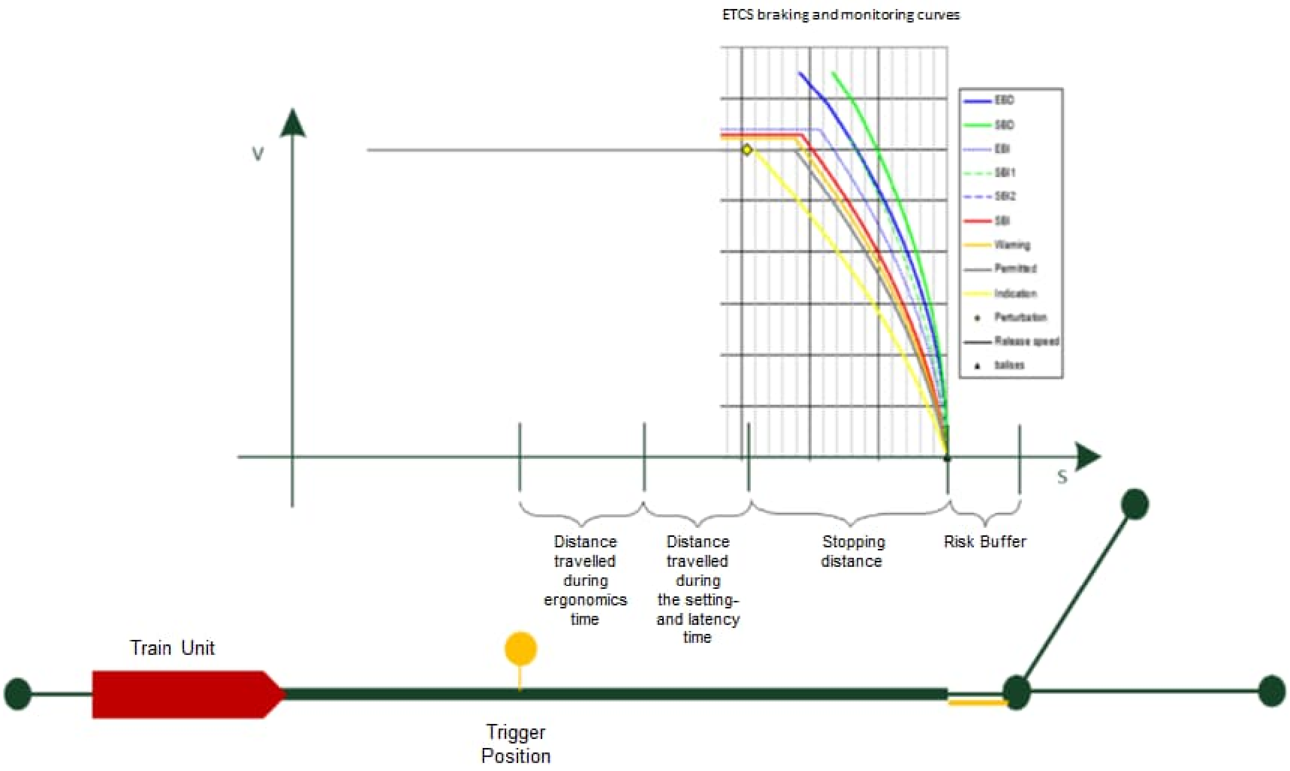


Figure 12: Example: breakdown of different distances required to calculate the Pass Position Constraint of a point

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 Pass Position Constraint only indicates the position where the first attempt is made to set the driveability state of the DPS-Group, provided that no other Request Trigger Constraints are present at the Request Trigger. If the DPS-Group is not allocated at this time, the request will be triggered. If the DPS-Group is allocated by another rMOB at the time the Request Trigger is not triggered, the request is stored in a queue and executed as soon as the DPS-Group is no longer allocated by another rMOB.

3.2.5.2. Calculation of Earliest Time Constraint

➔ ***Abstract concept: Earliest Time Constraint***
The Earliest Time Constraint defines the point in time that must be reached before a request shall be triggered from SubSys Plan Execution via SCI-CMD.

The Earliest Time Constraint is added as a further Request Trigger Constraint to the Pass Position Constraint where necessary. The Earliest Time Constraint is relevant if the Operational Plan Operational Movement has a timing constraint (Stop Event with scope EARLIEST_DEPARTURE/EARLIEST_ARRIVAL or a Passage Event with scope EARLIST_DEPARTURE) between the Pass Position Constraint and the DPS-Group.

The Earliest Time Constraint depends on the following factors:

- The earliest departure time (t_earliest_departure) of a stop.
- The earliest departure time (t_earliest_departure) of a passage.
- The setting time (t_setting) in which the DPS-Group is brought into the required drivability 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 Time Constraint 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

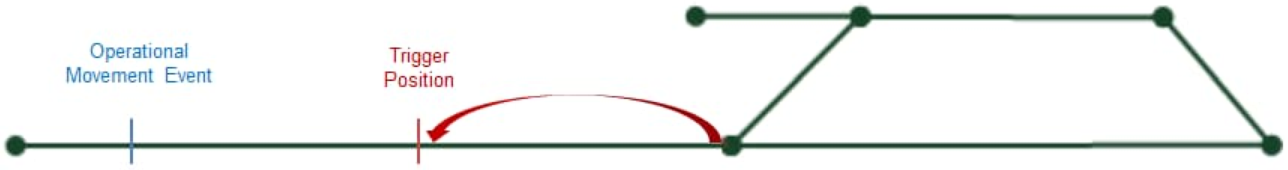
3.2.5.3. Calculation of Request Trigger Constraints

Based on Operational Plan Operational Movement, SubSys Plan Execution identifies all DPS-Groups that must be brought into a certain drivability state. Initially and with each update of the Operational Plan Operational Movement, the Request Trigger Constraints are updated for all identified DPS-Groups.

The different cases and the resulting Request Trigger Constraints are described below.

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

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

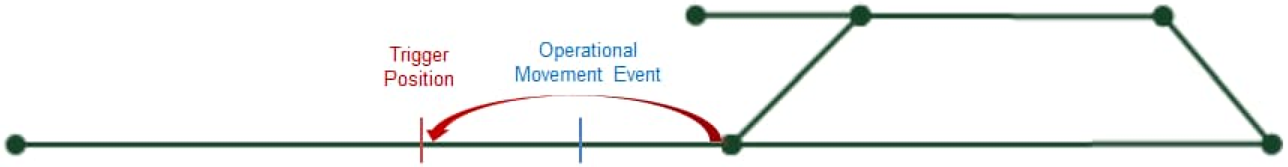


In this case, the following Request Trigger Constraint is assigned to the Request Trigger for the Field Element:

➔ Pass Position Constraint: Train Unit front end passed Trigger Position

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

There is a Stop Event from type EARLIEST_OPERATIONAL_DEPARTURE between the Trigger Position and the Field Element



In this case, the following Request Trigger Constraints are assigned to the Request Trigger for the Field Element:

➔ Pass Position Constraint: Train Unit front end passed Trigger Position

➔ Earliest Time Constraint: earliest operational departure expired

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

This section will be further elaborated in future releases.

3.3.1. Objectives and System Requirements

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

3.3.2. 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 characterises 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.*

3.3.3. 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.

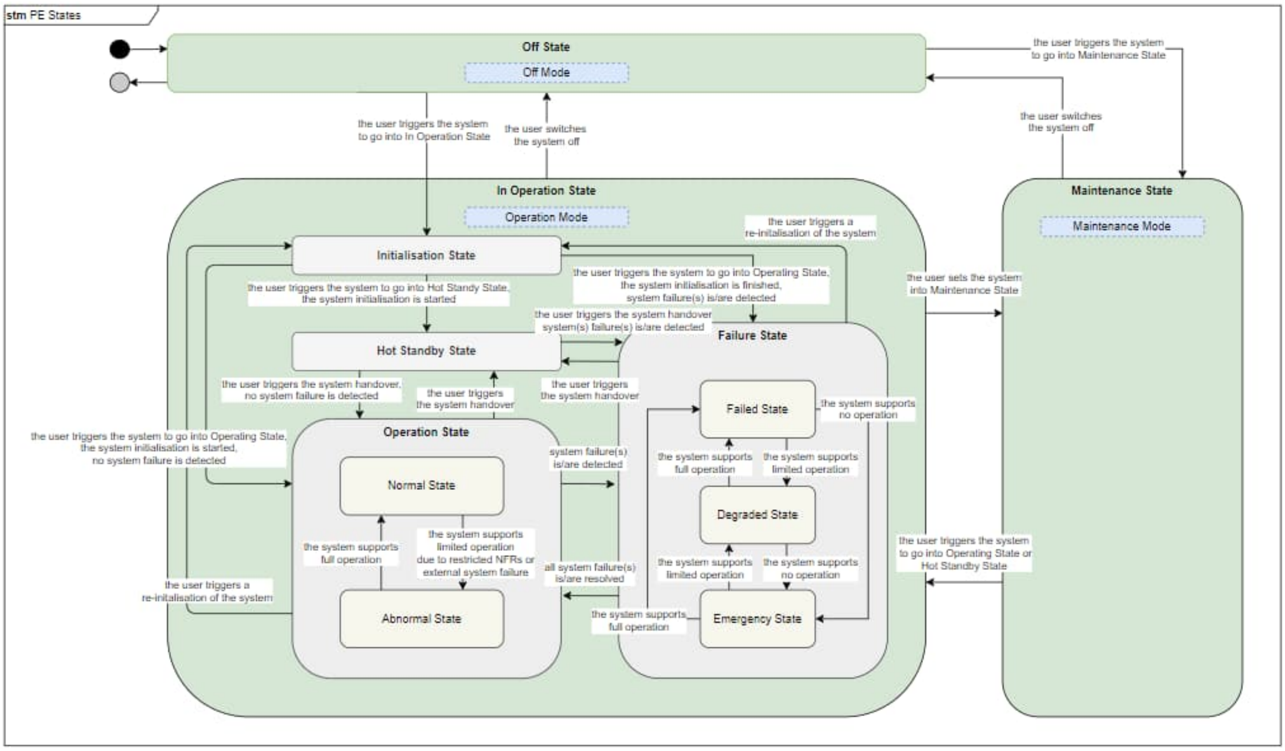

stm PE States
Off State
Off Mode
the user triggers the system to go into In Operation State
the user switches the system off
the user triggers the system to go into Maintenance State
the user switches the system off
In Operation State
Operation Mode
the user triggers a re-initalisation of the system
Initialisation State
the user triggers the system to go into Hot Standy State, the system initialisation is started
the user triggers the system to go into Operating State, the system initialisation is finished, system failure(s) is/are detected
the user triggers the system handover system(s) failure(s) is/are detected
Hot Standby State
Failure State
the user triggers the system handover, no system failure is detected
the user triggers the system handover
the user triggers the system handover
the user triggers the system to go into Operating State, the system initialisation is started, no system failure is detected
Operation State
Normal State
the system supports full operation
the system supports limited operation due to restricted NFRs or external system failure
Abnormal State
the user triggers a re-initalisation of the system
system failure(s) is/are detected
all system failure(s) is/are resolved
the system supports full operation
Failed State
the system supports no operation
the system supports full operation
the system supports limited operation
Degraded State
the system supports limited operation
the system supports no operation
Emergency State
the user sets the system into Maintenance State
the user triggers the system to go into Operating State or Hot Standby State
Maintenance State
Maintenance Mode

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

3.3.4. 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 - Provide and receive System State and Operational State	- SubSys Plan Execution shall perform the following functions once and one after the other
- 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

3.4. Functional concept: Robustness and High availability

This section will be further elaborated in future releases.

3.4.1. Objectives and System Requirements

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

3.4.2. 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**	**System 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 PE 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.

3.4.3. Description

3.4.3.1. 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.*

3.4.3.2. CAP Theorem

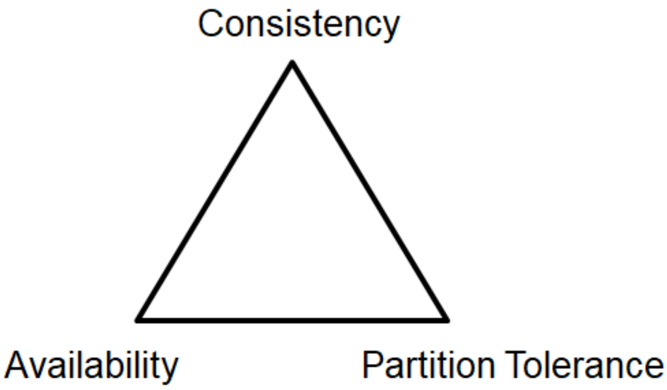


Figure 14: CAP Theorem

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.

3.4.3.3. 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) held, provided and exchanged by the different RCA SubSys within RCA.

Eventual Consistency means that consistency of data held 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. Functionality

Overview

Overview diagram on functionality will be added in future releases.

4.1. Automatic plan execution

4.1.1. Prerequisites

This function will be further elaborated in future releases.

Exported System Requirements to the Planning System

Exported System Requirements
The Planning System shall schedule any Operational Movement (e.g., train run, shunting movement). For the execution of one Operational Movement, the Planning System shall send an Operational Plan Execution Request containing an Operational Plan Operational Movement via SCI-OP to SubSys Plan Execution. The Planning System shall send a valid, conflict-free, track-precise, capacity-optimal Operational Plan Operational Movement based on a Map Data version that is system-wide available and valid at the time of the Operational Plan Execution.
The Planning System shall schedule any Operational Restriction Area (e.g., for track closures or speed restrictions). For the execution of one or many Operational Restriction Areas, the Planning System shall send an Operational Plan Execution Request containing an Operational Plan Operational Restriction via SCI-OP to SubSys Plan Execution. The Planning System shall send a valid, conflict-free, track-precise Operational Plan Operational Restriction based on a Map Data version that is system-wide available and valid at the time of the Operational Plan Execution.
The Planning System shall schedule any Operational Warning Area (e.g., for securing trackside construction works). For the execution of one or many Operational Warning Areas, the Planning System shall send an Operational Plan Execution Request containing an Operational Plan Operational Warning Measure via SCI-OP to SubSys Plan Execution. The Planning System shall send a valid, conflict-free, track-precise Operational Plan Operational Warning Measure based on a Map Data version that is system-wide available and valid at the time of the Operational Plan Execution.
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).

4.1.2. Operational Plan Management

4.1.2.1. Function: Process Operational Plan

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
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 (for details refer to 4.1.2.1.1, 4.1.2.1.2, 4.1.2.1.3) 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 an accepted Operational Plan, by sending different types of requests to control e.g., Drive Protection Sections (DPS), Movement Permission (MPs), Usage Restriction Areas (URA) and Warning Areas (WA) to SubSys Safety Logic via SCI-CMD. All requests issued by SubSys Plan Execution via SCI-CMD to SubSys Safety Logic shall respect the safety rules given by it. 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 flank protection and to bring the field elements into the required state by sending just in time DPSGroupRequests to SubSys Safety Logic. As soon as the desired state is reached, SubSys Plan Execution shall then request the MP (incl. 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 retry after a certain time. If this procedure is unsuccessful, SubSys Plan Execution shall provide a failure code to the Planning System via SCI-OP or to SubSys Workbench via SWI-PE in form of the Operational Plan Execution Report.
SubSys Plan Execution shall provide an Operational Plan Execution Reports for an accepted Operational Plan to the Planning System via SCI-OP or to SubSys Workbench via SWI-PE. The Execution Report is used by SubSys Plan Execution to inform about completed Operational Events as well as conflicts or failures related to Operational Events.
SubSys Plan Execution shall provide the most current and complete PE Operating State to the Planning System via SCI-OP or to SubSys Workbench via SWI-PE by sending updates. Any update event on the PE Operating State that occurs due to the implementation of an Operational Plan shall provide a reference to the relevant Operational Plan. Any update event on the PE Operating State that occurs unplanned and cannot be clearly assigned to an Operational Plan shall not provide a reference to an Operational Plan (because no related Operational Plan exists).
SubSys Plan Execution shall be able to implement frequent and short-term updates of Operational Plans. SubSys Plan Execution shall check the time feasibility of short-term requests and respond with an appropriate Operational Plan Execution Response if the request is no longer feasible.
SubSys Plan Execution shall not process any Operational Event which was already completed by it. Note: An Operational Movement Event is completed for SubSys Plan Execution, if a MP was granted by SubSys Safety Logic which allocates the position of the Operational Movement Event, allowing the departure, arrival, or passage of the Physical Train Unit from/at this Operational Movement Event and if the Physical Train Unit has started moving within this MP. SubSys Plan Execution can send an Operational Plan Execution Report for an Operational Movement Event that has already been completed and set the execution state to FAILED. This behaviour may occur, for example, when an MP has been granted, the Physical Train Unit starts moving, but the is MP revoked for some reason.
SubSys Plan Execution shall report a failure by an Operational Plan Execution Report, if the referenced rMOB is not existing within the PE Operating State.
SubSys Plan Execution shall report a failure by an Operational Plan Execution Report, if the referenced rMOB is not, even partly, located on the specified Track Path and the Track Path between the position of the front end of the referenced rMOB (in direction of travel) and the position of the first Operational Movement Event is not distinct.
SubSys Plan Execution shall report a failure by an Operational Plan Execution Report, if the referenced rMOB is partly located on the specified Track Path, but the front end of the referenced rMOB (in direction of travel) is not located on the Track Path.

4.1.2.1.1. Subfunction: Check Feasibility of Operational Plan Operational Movement

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall check each requested Operational Plan Operational Movement for its feasibility before executing the Operational Plan Operational Movement and shall provide the check result via SCI-OP or via SWI-PE to the originating requester.
SubSys Plan Execution shall reject a requested Operational Plan Operational Movement, if the specified sequence of Operational Movements Events and the resulting Track Path is not navigable, not track-precise, or not covered, even partly, by the Area of Control.
SubSys Plan Execution shall reject a requested update of an Operational Plan Operational Movement, if the referenced rMOB is not, even partly, located on the specified Track Path and the Track Path between the position of the front end of the referenced rMOB (in direction of travel) and the position of the first Operational Movement Event is not distinct.
SubSys Plan Execution shall reject a requested update of an Operational Plan Operational Movement, if the referenced rMOB is partly located on the specified Track Path, but the front end of the referenced rMOB (in direction of travel) is not located on the Track Path.

SubSys Plan Execution shall reject a requested update of an Operational Plan Operational Movement, if it does not contain at least one Operational Movement Event of the existing Operational Plan Operational Movement as an Operational Plan Event Reference.
SubSys Plan Execution shall execute the feasibility check of an Operational Plan Operational Movement only once and in timely manner after receiving the Operational Plan Execution Request. Note: If changes occur between the time of reception of the Operational Plan Execution Request and the time of execution of the specified Operational Plan which prevent the feasibility of an accepted Operational Plan Execution Request, SubSys Plan Execution shall report this by an Operational Plan Execution Report at the time of execution.
SubSys Plan Execution shall execute a requested Operational Plan Operational Movement that is feasible. SubSys Plan Execution shall not execute a requested Operational Plan Operational Movement that is not feasible.
SubSys Plan Execution shall provide the result of the feasibility check as Operational Plan Execution Response to the requester via SCI-OP or SWI-PE.

4.1.2.1.2. Subfunction: Check Feasibility of Operational Plan Operational Restriction

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
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.

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

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
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.

4.1.3. Driveability management

4.1.3.1. Function: Observe driveability of railway network

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall observe the DpsDriveabilityType updates of DPSs (NONE, LIMITED, FULL) provided by SubSys Safety Logic via SCI-CMD.
SubSys Plan Execution shall observe the DpsGroupStateType (e.g., READY, PROCESSING, DISTURBED) provided by SubSys Safety Logic 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 based on the DpsGroupUpdatedEvents provided by SubSys Safety Logic 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.

4.1.3.2. Function: Provide driveability of railway network

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
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.

4.1.3.3. Function: Control driveability of railway network

This function will be further elaborated in future releases.

Description:

System Requirements

System Requirement
SubSys Plan Execution shall command all Field Elements changes giving or ensuring driveability of the railway network by sending DPSGroupState requests to SubSys Safety Logic via SCI-CMD.

4.1.3.3.1. Subfunction: Calculate Trigger Points for driveability requests

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall calculate the optimal trigger points for driveability requests to implement the given Operational Plan with the least possible utilisation of the planned capacity.
SubSys Plan Execution shall at least consider the following aspects when calculating the trigger points for driveability requests (list not exhaustive). • Required setting times (any system times necessary for reaching the requested state) for DPS Groups (field elements) • Position, speed and MP of relevant rMOB

4.1.4. Operational Movement management

4.1.4.1. Function: Observe Operational Movement of Physical Train Units

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall observe the updates of Resolved Trackbound Movable Objects provided by SubSys Safety Logic 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 SubSys Safety Logic via SCI-CMD.
SubSys Plan Execution shall observe the updates of MPs provided by SubSys Safety Logic 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.

4.1.4.2. Function: Provide Operational Movement of Physical Train Units

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
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 (type MOVEMENT) 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 (type OCCUPANCY) 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.

4.1.4.3. Function: Control Operational Movement of Physical Train Units

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall be able to request a MP to allow the Physical Train Unit to move to the planned position on the railway network.
For the implementation of an Operational Movement, SubSys Plan Execution shall control the driveability of railway network in the Area of Control by generating DPSGroupState requests and send them just in time to SubSys Safety Logic via SCI-CMD, to make the best possible utilisation of the planned capacity of the railway network.
For the implementation of an Operational Movement, SubSys Plan Execution shall control the movement of Physical Train Units in the Area of Control by generating MP requests and send them just-in-time to SubSys Safety Logic via SCI-CMD, to make the best possible utilisation of the planned capacity of the railway network. A MP allocates the Track Path for the Operational Movement as well as the track sections and field elements that ensure the safety of the Operational Movement (see risk buffer and risk paths).

SubSys Plan Execution shall send timely requests to SubSys Safety Logic to create, to lengthen, to shorten (only at front end) or to adjust the speed profile or mode profile of a MP. Timely means that requests are made early enough that a Physical Train Unit never enters its braking curve due to an MP not requested in time.
SubSys Plan Execution shall automatically calculate the content and trigger point for any request sent to SubSys Safety Logic, 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 SubSys Safety Logic manages all the underlying additional transition logic for SubSys Plan Execution.
SubSys Plan Execution shall only request extensions (and reductions) of a MP without gaps on the railway network, without overlaps on the railway network and in the correct chronological and geographical order. However, SubSys Plan Execution can request MPs that overlap with another MP belonging to another rMOB, in specific situations, 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 request an upgrade or downgrade of an existing MP (e.g., with a higher or lower speed profile).
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 a 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).
SubSys Plan Execution shall calculate the maximum speed profile of a MP based on the information in the Operational Plan, the PE Operating State, and the properties of the underlying topology.
SubSys Plan Execution shall create a work list for MP requests, as SubSys Safety Logic does not queue future MP requests from SubSys Plan Execution but expects SubSys Plan Execution to send the request individually at the right time.

4.1.4.3.1. Subfunction: Calculate MP

This function will be further elaborated in future releases.

Description

tbd

System Requirements

Functional Requirement
SubSys Plan Execution shall calculate the appropriate MP Extent, Risk Buffer and Risk Paths for the calculation of the MP.
SubSys Plan Execution shall consider the currently valid and active Map Data for the calculation of the MP (e.g., allowed max speed of allocated railway network, non-stopping Areas, diamond crossing with single slip must always be completely clear or completely allocated).
SubSys Plan Execution shall consider the temporary restriction of the railway network (URA) for the calculation of the 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.)
SubSys Plan Execution shall consider the characteristics of the Train Unit for the calculation of the MP.
SubSys Plan Execution shall consider the maximum allowed length of a MP for the calculation of the MP length.
SubSys Plan Execution shall consider MP Limit Markers (e.g., due to other MP, MOB, state of DPS-Groups, URA) for the calculation of the MP.
SubSys Plan Execution shall shorten the MP at front when the Train Unit is in standstill to deallocate the railway network in front of the Train Unit.

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 Physical Train Unit).
SubSys Plan Execution shall calculate a MP which complies with the safety rules of SubSys Safety Logic (a rejection may lead to an unscheduled stop. Unscheduled stops shall be avoided).

4.1.4.3.1.1. Subfunction: Calculate MP Extent

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
For the implementation of an MP request, SubSys Plan Execution shall determine a safe and appropriate MP Extent.
SubSys Plan Execution shall calculate a MP Extent which complies with the static restrictions, temporary restrictions, and existing allocations of the referenced Track Path.
SubSys Plan Execution shall calculate a MP Extent which complies with the restrictions, position, and extent of the referenced rMOB.
SubSys Plan Execution shall calculate a MP Extent which complies with the driveability of the referenced track path.
SubSys Plan Execution shall calculate a MP Extent which allows the Physical Train Unit to run without interruption during the planned movement.
SubSys Plan Execution shall calculate a MP Extent which allocates as less Track Path as possible under the given Operational Plan Operational Movement.
SubSys Plan Execution shall calculate a MP Extent which complies with the Operational Movement Event specified in the Operational Plan Operational Movement.
SubSys Plan Execution shall calculate a MP Extent which follows the Track Path specified in the Operational Plan Operational Movement.

4.1.4.3.1.2. Subfunction: Calculate Risk Buffer

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall calculate the required Risk Buffer and consider it when requesting a MP.
SubSys Plan Execution shall determine the required Risk Buffer 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.
SubSys Plan Execution shall consider that 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 (which are to be defined in SubSys SL).

4.1.4.3.1.3. Subfunction: Calculate Risk Paths

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall calculate the required Risk Paths and consider it when requesting a MP.
SubSys Plan Execution shall determine the necessary type of flank protection (track-enforced flank protection or flank protection provided by reservation of Allocation Sections) when 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 cases where track enforced flank protection is required, SubSys Plan Execution shall first send the necessary DPS flank protection requests to SubSys Safety Logic 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 cases where flank protection provided by reservation of Allocation Sections is sufficient, SubSys Plan Execution may request an MP with reduced speed and Risk Paths that do not cover the protecting field elements, but only the relevant Allocation Sections, so that the Allocation Sections cannot be used by another MP.

4.1.4.4. Subfunction: Calculate Trigger Points for MP requests

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall calculate the optimal time to request a MP.
SubSys Plan Execution shall calculate the optimal time to trigger the MP requests at SubSys Safety Logic to ensure the capacity-efficient minimum headway between Physical Train Units and thus achieve the best possible utilisation 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 rMOB • Point in time when driveability of required Track Path will be given

4.1.5. Operational Restriction management

4.1.5.1. Function: Observe Usage Restriction Areas on railway network

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall observe the updates of URA (planned and unplanned) provided by SubSys Safety Logic 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.

4.1.5.2. Function: Provide Operational Restriction Areas to the railway network

This function will be further elaborated in future releases.

Description

System Requirements

System Requirement
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 and via SWI-PE.

4.1.5.3. Function: Control Operational Restriction Areas on railway network

This function will be further elaborated in future releases.

Description

System Requirements

System Requirement
SubSys Plan Execution shall implement an Operational Restriction Area on the railway network by generating one to many URA requests and shall send it to SubSys Safety Logic via SCI-CMD to be implemented at the specified time.
SubSys Plan Execution shall be able to request the temporal beginning and end of a planned URA separately at the right time. Note: Before an URA is removed by SubSys Safety Logic, the removal shall be confirmed by an Authorised Trackside Person).
SubSys Plan Execution shall be able to request URAs for any Track Edge Section in the Area of Control. The following aspects shall be considered: • The extent of an URA is always defined by the referenced Track Edge Section(s). • The direction of the Track Edge Sections on which the URA is placed on may be bidirectional or unidirectional to represent Operational Restrictions which are valid in one or both directions of travel. • URAs may overlap.
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 to those.

4.1.5.3.1. Subfunction: Define safequard measures

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall implement a planned Operational Restriction Area by defining the required safeguard measures. In addition to restricting the actual Operational Restriction Area by an Usage Restriction Area, this can mean, for example, that point closures outside the Operational Restriction Area shall be defined in order to protect the Operational Restriction Area against unauthorized entry

4.1.6. Operational Warning Measure management

4.1.6.1. Function: Observe Warning Areas on railway network

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall observe the updates of Warning Areas (WA) provided by SubSys Safety Logic 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.

4.1.6.2. Function: Provide Operational Warning Areas on railway network

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
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.

4.1.6.3. Function: Control Operational Warning Areas on railway network

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall implement an Operational Warning Area on the railway network by generating one to many Warning Area requests and shall send them at the specified time to SubSys Safety Logic via SCI-CMD to be implemented. The following aspects shall be considered: • SubSys Plan Execution shall be able to request the temporal beginning and 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 extent 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.
An OWA may have a dependency (e.g. all or non-creation/deletion) to other OWAs or ORAs and may therefore be linked.

4.2. Manual plan execution

4.2.1. Function: Manual plan execution

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall provide 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 (list not exhaustive):

• Circulate and lubricate switches. Note: Assignment of functionality requires a detailed system and risk analysis. • Perform mapping of an identified but not localised 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 MP to a defined position where Physical Train Unit can be safely localised (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. Function may be assigned to proposed RCA Subsystem IPM-TS • Performing an emergency stop for a specific Physical Train Unit. Note: Assignment of functionality requires a detailed system and risk analysis. Function may be assigned to proposed RCA Subsystem IPM-TS • 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 of railway infrastructure to manage Operational Movement (for example to move Passenger Train Units to the nearest station) • Simple manual operation of railway infrastructure to manage Operational Warning Areas • Simple manual operation of railway infrastructure to manage Operational Restriction Areas
SubSys Workbench shall process the PE Operating State and the Execution Report sent by SubSys Plan Execution via SWI-PE and display the results 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.

4.3. Device and Configuration Management

4.3.1. Function: Import Configuration Data

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall import all required configuration data from the SubSys Device & Config Management via the SMI-PE.
SubSys Plan Execution shall check that the import process was successful.
SubSys Plan Execution shall acknowledge the successful import of configuration data to the SubSys Device & Config Management via SMI (Standard Maintenance Interface).

4.3.2. Function: Activate Map Data

This function will be further elaborated in future releases.

Description:

tbd

System Requirements

System Requirement
SubSys Plan Execution shall support the activation of distributed Map Data.

4.4. Diagnostics and Monitoring

4.4.1. Function: Send diagnostics data

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall register and log system data and send them to SubSys Diagnostics & Monitoring via SDI (Standard Diagnostic and Monitoring Interface)

4.5. Authentication und Authorisation

4.5.1. Function: Use authentication and authorisation services

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall use services for authenticating and authorising human users and SubSys, depending on the RCA security architecture to be implemented and separated with different role definitions.

4.6. Robustness

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

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall support different System States, Operational States and Modes of Operation 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 any specific RCA SubSys.

4.6.2. Function: Determine System State and Operational State

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall determine its own System State and Operational State due to 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

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

This function will be further elaborated in future releases.

Description

tbd

System Requirements

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

4.6.4. Function: Calculate integrity and check internal data model

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall calculate the integrity of its internal data model based on the Configuration Data and Operational Data received via its interfaces.
SubSys Plan Execution shall check the integrity of the calculated internal data model.
If inconsistencies were determined, SubSys Plan Execution shall synchronise and re-synchronise the Configuration and Operational Data with the neighbouring systems.

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

This function will be further elaborated in future releases.

Description

tbd

System Requirements

System Requirement
SubSys Plan Execution shall be able to synchronise the Operational and Configuration Data sent and received over its interfaces if an inconsistency in the internal data model was detected: This includes: • Configuration Data o Synchronisation sent and received via SMI • Operational Data o Synchronisation sent and received via SCI-OP o Synchronisation sent and received via SWI-PE o Synchronisation sent and received via SCI-CMD o Synchronisation sent and received via SHI-PE

5. Concept: High level migration scenarios with Plan Execution

5.1. Objective and System Requirements

The underlying 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] 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], 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.

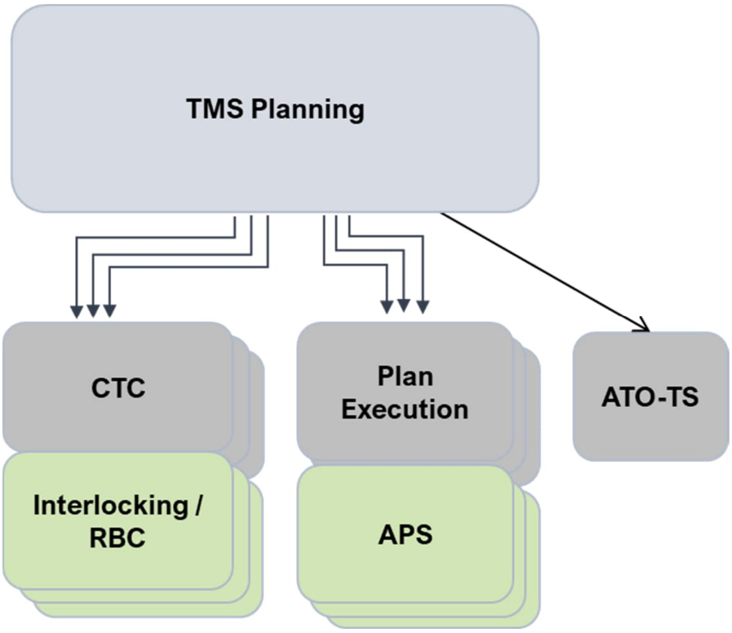


Figure 15: General connection of the RCA architecture to a timetable system

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.

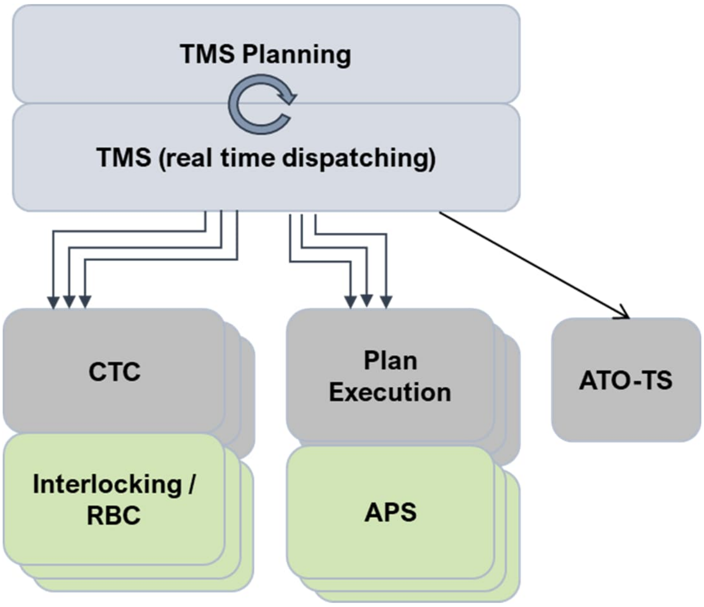


Figure 16: General connection of the RCA architecture to a dispatching system

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.

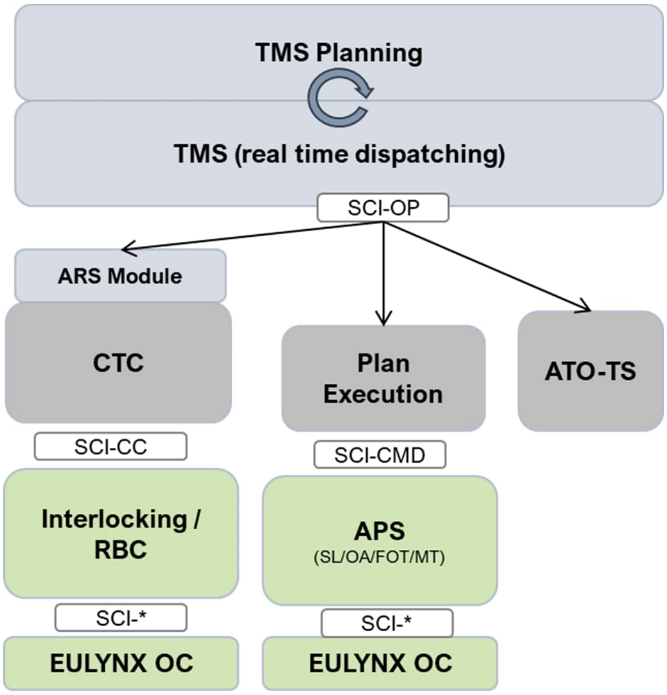


Figure 17: Specific connection of the RCA architecture to a dispatching system

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 synchronised 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.

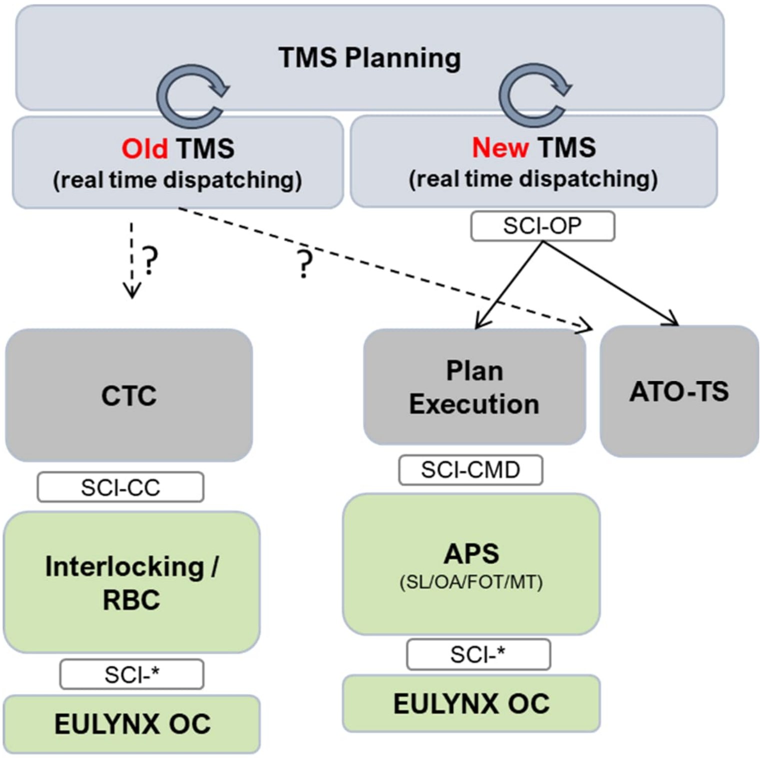


Figure 18: Specific connection of the RCA architecture to a dispatching system

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

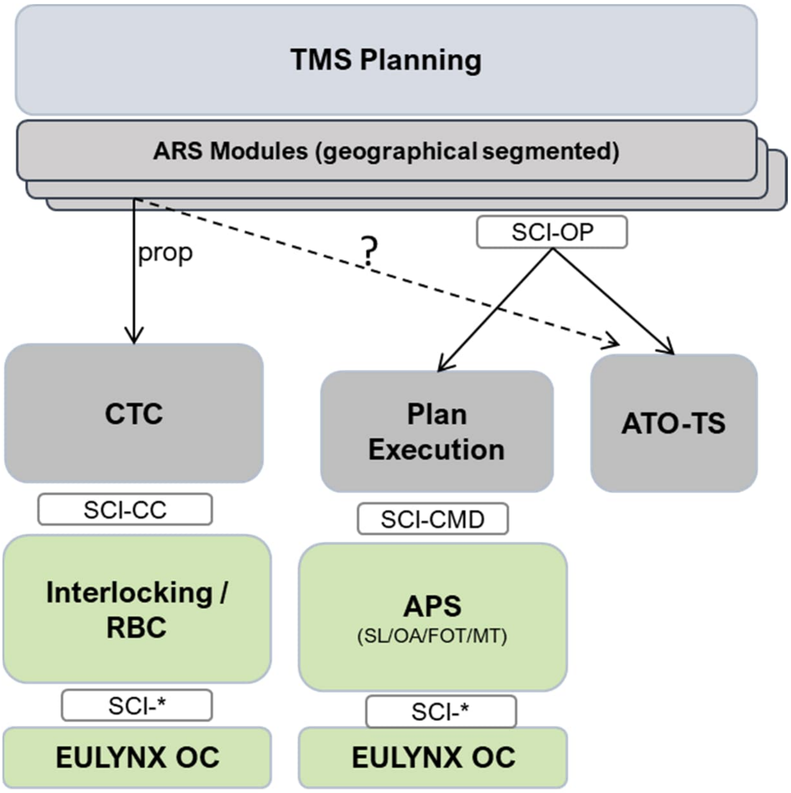


Figure 19: Integration via an ARS module below the timetable system

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.

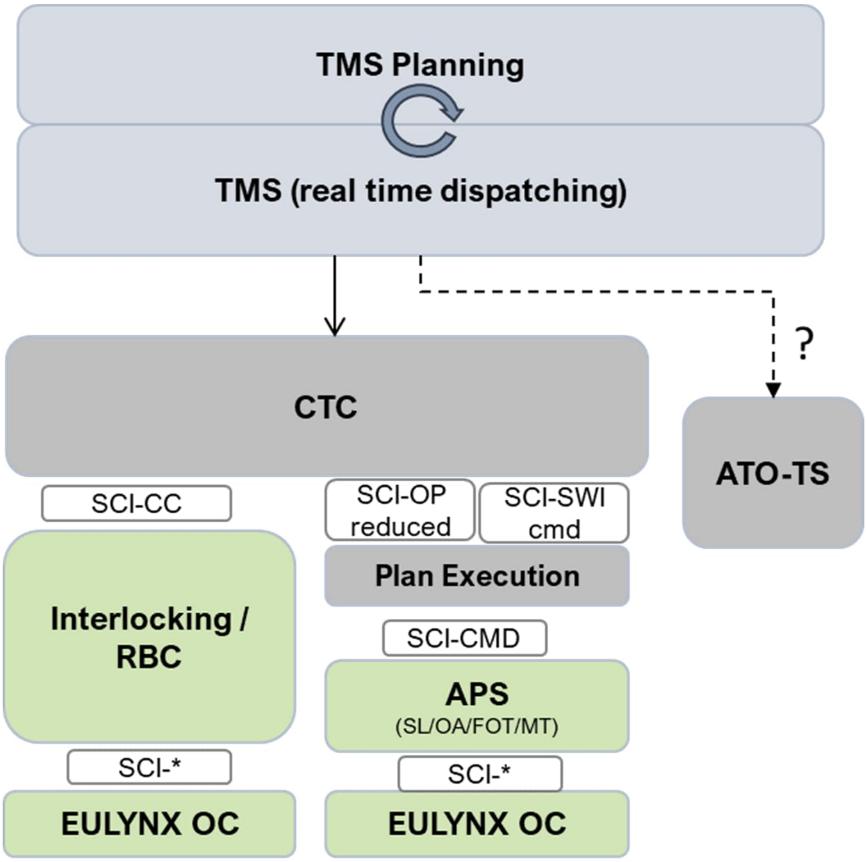


Figure 20: Extension of the existing CTC layer through a connection to SubSys Plan Execution

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	PE Operating State vs Operational State	Distinguish between PE Operating State and Operational State by naming them more explicitly
2	Naming of functions: “report” instead of “provide”	Perhaps the verb 'report' would be better suited as opposed to 'provide', which could be interpreted as executing an action to ensure the '**drivability status**'.
3	Subfunction “Calculate Risk Paths”	Two enquiry points for clarification: 1. Where trains are moving away from the conflict (under their own MP of course) at the time of setting a MP, shouldn't this be viewed as a legitimate move without impacting the overall capacity of the railway. (This is foreseen on APS Concepts as the Risk Path being considered would end at the back of a train which is moving away from conflict). Note this is important because we are trying to ensure the running of trains near one another. 2. When it is said that any possible Safe solution can be chosen by SubSys PE, how does APS know how to go about checking if the path is legitimate in order to later grant the MP?
4	Handling of commands via SCI-OP and SWI-PE	Define the principle how parallel and conflicting commands from Workbench or Planning System are handled. For example, “different priority or “first come first serve” or “depending on specific commands and use cases”.
5	MP Limit Marker	The current concept does not consider requesting the MP extent as long as required, but as short as possible. Could possibly resolved with additional MP Limit Marker Constraints.
6	Exported Requirements	Analyse and specify exported requirements for other RCA SubSys