ERJU SYSTEM PILLAR

T3-SystemArchitectureDescription

SYSTEM ARCHITECTURE DESCRIPTION

Author(s)	Völcker Marcus (I-NAT-GST-CCS-EXT - Extern) , Uhlich Sebastian (I-NAT-GST-CCS-EXT - Extern) , Krista Matthias (I-NAT-GST-CCS-EXT - Extern) , NANNI Marco
Abstract	The purpose of this System Architecture Description is to define the top-level functional breakdown structure of the CMS and TMS subsystems according to the general principles, specifically considering how the so-called “Railway System” has been conceived, and in line with the concepts reported.
Config Item	System Architecture Description
Document ID	30 Deliverables/T3-SystemArchitectureDescription#714338 T3-SystemArchitectureDescription
Classification	Public
Status	In Progress (educated draft, discussion in domain nearly finished)
Version	1.0
Revision	714332
Last Change Date	24.09.2025
Copyright	Brussels: Europe’s Rail Joint Undertaking, 2025

This work is currently a work in progress. The content presented is subject to change as it undergoes further review, refinement, and development. Please do not consider this version as final or authoritative.

INFO: History table is not displayed, because this document is in status **doc_inprogress**.
RULE: History table is not displayed, in statuses: { draft doc_open doc_inprogress doc_contentApproval doc_contentDecision }
CONTACT: For more information contact Administrator

DISCLAIMER

This draft is to be intended as an Annex to the System Concept document and is issued as part of the planned deliverables of the first three months activity.

1 History of Changes

Nr.:	Changes:	Leader/ Authors:
00.01	First draft for internal revision.	M. Nanni
00.02	Review and Integration of additional sections	M. Voelcker
00.03	Second draft for internal revision	S. Uhlich
00.04	Draft for task 3 internal revision	S. Uhlich
00.04.1	Changes to task 3 comments	S. Uhlich
00.05	Submission to CG	S. Uhlich
00.06	Finalization of Building Block view	S. Uhlich
00.07	Finalization of Runtime View (Logical Architecture)	S. Uhlich
00.08	Finalization of Deployment View, Performance improvements.	S. Uhlich
00.09	Final Review SP3	S. Uhlich
00.10	Minor changes to MG Review	S. Uhlich
00.11	• Added links from SAD to TAsk 1 Operate Train Capability Report, according to the agreements taken with Task 1 and SPCG • Added an exchange from TMS to CMS to aligne both systems for relevantdeviations impacting CMS time horizon • Removed occurrences of type "Reference" for those WI where this type was misused according to the Reference Usage Guideline published by EET team.	M. Nanni

2 Introduction

2.1 System Context

Several initiatives (OCORA, LINX4RAIL, RCA, Shift2Rail, …) have been launched to shape the future of Europe's Rail, with the ambitious goal to imagine and build the evolution of the actual railway system towards a fully digitalized, integrated, automated, interoperable, modular new system. The System Pillars project is in line with these initiatives and promotes and supervises the design of an integrated a CCS/TMS/CMS Architecture. **[**SPT3TMS-10437 **]**

In this context and taking into account the relevant results of the previous activities, this document describes the Architectural Design of the Capacity / Traffic Management System (CMS/TMS), which is one of the planned deliverables of the Task 3 CMS/TMS Domain, one of the tasks defined inside the System Pillars project organization. The design shall take into account and proof to reach the Common Business Objectives given as input for all System Pillars activities. **[**SPT3TMS-13744 **]**

2.2 Purpose

The purpose of this System Architecture Description is to define the top-level functional breakdown structure of the CMS and TMS subsystems according to the general principles defined in [Ref 1], specifically considering how the so-called "Railway System" has been conceived, and in line with the concepts reported in [Ref 4].
This document addresses the design of the architecture which supports the full set of functions apportioned to CMS and TMS, according to what defined in [Ref 4] and the block diagram depicted in [Ref 5], which shall be expanded into a greater level of detail.
The functional description will focus on the tasks apportioned to the several logical components, indicating which macro functions are in charge of them, without recalling here any implementation detail, and indicating the main constraints and interfaces.
The document must be read together with [Ref 6], which collects and organizes the functional system requirements are listed.

More specifically, this document focuses on the following aspects of the overall architecture:

- The breakdown structure into the CMS and TMS building blocks.
- Description of the CMS and TMS components.
- Interfaces among the components, focusing on main data flows only.
- Allocation of the functions over the components.
- Identification of backend and GUI components.

This is a live document which shall evolve along the duration of the System Pillars activities, increasing the level of details consistently with the achievements coming from the workgroup design activity. **[**SPT3TMS-10436 **]**

2.3 Content

Apart from the Introduction, this document is structured into the following different sections:

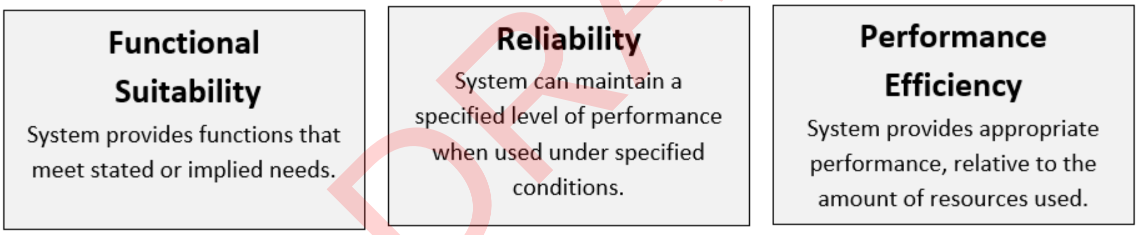
- **Building Block View** (see sect. 5): this section describes the main parts of the system including an introduction of domains, components, and exchanges.
- **Runtime View (Logical Architecture)** (see sect. 6): this section describes concrete behaviour and interactions of the system's building blocks in form of important use cases or features, interactions at critical external interfaces, operation and administration and error and exception scenarios.
- **Deployment View** (see sect. 7): this section describes technical infrastructure used to execute the system, with infrastructure elements like geographical locations, environments, computers, processors, and channels.
- **Cross-cutting Concepts** (see sect. 8): this section describes overall, principal regulations and solution ideas that are relevant in multiple parts (= cross-cutting) of the system.

This document is based on arc42: An architecture standard providing templates for documentation and communication of software and system architectures. Arc42 was published by Dr. Peter Hruschka and Dr. Gernot Starke in 2014 and is since then an implemented industry standard for architecture documentation. For more information: https://arc42.org/

Further, the document follows the concept of SEMP.
For more information: *SPPROCESS/SEMP Annex D Processes/SEMP process 04-Logical Architecture : 714338*

2.4 Quality Goals

According to ISO 25010 standard, the quality goals for the architecture can be described as the following. The fulfillment is of highest importance to the major stakeholders.



[SPT3TMS-10435 **]**

In accordance with the defined milestones for 2024, the following release dates apply:

Chapter:	**Milestone date:**
Building Block view	12/01/2024
Runtime view	09/02/2024
Deployment view	08/03/2024
Cross-Cutting Concepts	10/05/2024

[SPT3TMS-10434 **]**

2.5 References

- [Ref 1] - Annex 1 CCS/TMS Architectural principles - rev 00.05

[SPT3TMS-16192 **]**

- [Ref 2] - CCS/TMS/CMS Systems Architecture – rev 00.04

[SPT3TMS-16194 **]**

- [Ref 3] - WP2 - Common Business Objectives – rev 00.04

[SPT3TMS-16186 **]**

- [Ref 4] - SYSTEM CONCEPT R2 EUROPE RAIL CMS & TMS

[SPT3TMS-16191 **]**

- [Ref 5] – CMS & TMS System Definition

[SPT3TMS-16190 **]**

- [Ref 6] – CMS Functional Requirements, TMS Functional Requirements

[SPT3TMS-16171 **]**

3 Architecture Constraints

The system relies on several architecture constraints which have a relevant impact to the efficiency and success of the overall. Inside the System Pillars context, a specific task, namely Task 2: Architecture and Release Coordination, is accountable for defining architectural guidelines to be followed by other tasks, to design a consistent and homogeneous railway environment able to implement the Operational Vision depicted for TMS/CMS and the achievements of the underlying Common Business Objectives apportioned to this domain. Afterwards, the key Common Business Objectives are given, along with the analysis which proves the correctness of the architectural choices and their capability to reach the assigned goals. **[**SPT3TMS-10704 **]**

The following table illustrates the architecture constraints:

ID:	Constraint:	Rationale:	Concept:
CSTR-1	The system must be designed to be elastic.	Based on load, or other factors, elasticity is the main factor of fine tuning the systems performance, availability, and resource usage. Elasticity can be realized by operating multiple stateless instances of each component, a mechanism for Service naming, registration, and discovery along with routing and load-balancing of high load.	8.1.2 - Elasticity
CSTR-2	A failure of a single instance should have minimal impact on the system	A failure of one instances of a certain component should only impact a single system function - the rest of the system must operate without impact. Components must be resilient, loosely coupled with other components. Component must be “minimal but complete”. Components should offer a complete function, with minimal dependencies (loose coupling) to other components of the system.	8.1 - Availability Concepts
CSTR-3	The system must be designed to be real-time.	Quick work cycles (response times, transaction time, etc) are crucial to achieve a real time behaviour.	8.3 - Real time Concepts
CSTR-4	The system must accommodate updates and upgrades without downtime (e.g., hardware, software, content updates)	To ensure availability, load balancing is crucial.	8.1 - Availability Concepts
CSTR-5	A consistent data model structure / language will be used.	Consistency on data model level reduces complexity and failures.	8.6.5.2 - Database Versioning
CSTR-6	Data models support various versions.	Important aspect to fulfil specific requirements for different consumers. With versioning backward compatibility can be achieved.	8.6.5 - Versioning
CSTR-7	One responsibility/ ownership of a data model.	Shifting responsibilities/ownerships lead to inconsistencies in complex environments.	8.5.2.1 - Database design
CSTR	Each data model can be clearly identified by a		

ID:	Constraint:	Rationale:	Concept:
-8	unique identifier (business/technical key).	Data transaction must be comprehensive and meaningful.	Will be added in later revision
CS TR -9	Components communicate only with its own persistence layers (Saga pattern).	Concurrent transactions lead to inconsistencies and are known to be error prone.	8.5.3 - Concurrency
CS TR -10	Components are stateless and independent without the knowledge of other components or instances.	The communication between the components is event-driven, asynchronous and exchange item based via Publish/Subscribe pattern.	8.5.1.2 - Message Queues
CS TR -11	The System must be horizontally scalable when applicable.	The system must be able to distribute load (e.g., via a load balancer) and manage available instances (e.g., via Service discovery).	8.1.1 - Scalability
CS TR -12	Referential integrity must be realized with business keys instead of technical keys when sharing data outside of its domain.	Technical IDs should never be exposed outside of the domain.	Will be added in later revision

[SPT3TMS-10440 **]**

4 Operational Requirements

Operational Requirements (OR) are crucial for the CMS & TMS architecture as they influence security, runtime, scalability, maintainability, and robustness of the system. **[**SPT3TMS-10505 **]**

The following table illustrates the operational requirements:

ID:	Requirement:	Rationale:	Concept:
OR S-1	Logging	Real-time and historical logging data visualize the behaviour of the system. For debugging purposes it's important to store logging data for a certain timespan. However, the increase of logging data has a significant impact on disk space and must be balanced with log rotations. A logging solution is responsible for the aggregation of logging data, distributed over all CMS / TMS components.	8.6.3 - Monitoring
OR S-2	Metrics	Inspection of the system's condition in combination with logging and monitoring.	8.6.3 - Monitoring
OR S-3	High availability	To ensure availability, CMS/TMS components must be scalable, fault tolerant, robust and have quick start/stop times. Multiple instances of CMS/TMS components serve the throughput of data. To manage several instances, service discovery and load balancing becomes irreplaceable.	8.1 - Availability Concepts
OR S-4	Security	To minimize any risk and harm to the system, a security concept is essential in any term.	8.8 - Security Concepts
OR S-5	Safety	No safety requirements applicable as far as known yet.	-

[SPT3TMS-10705 **]**

5 Building Block View

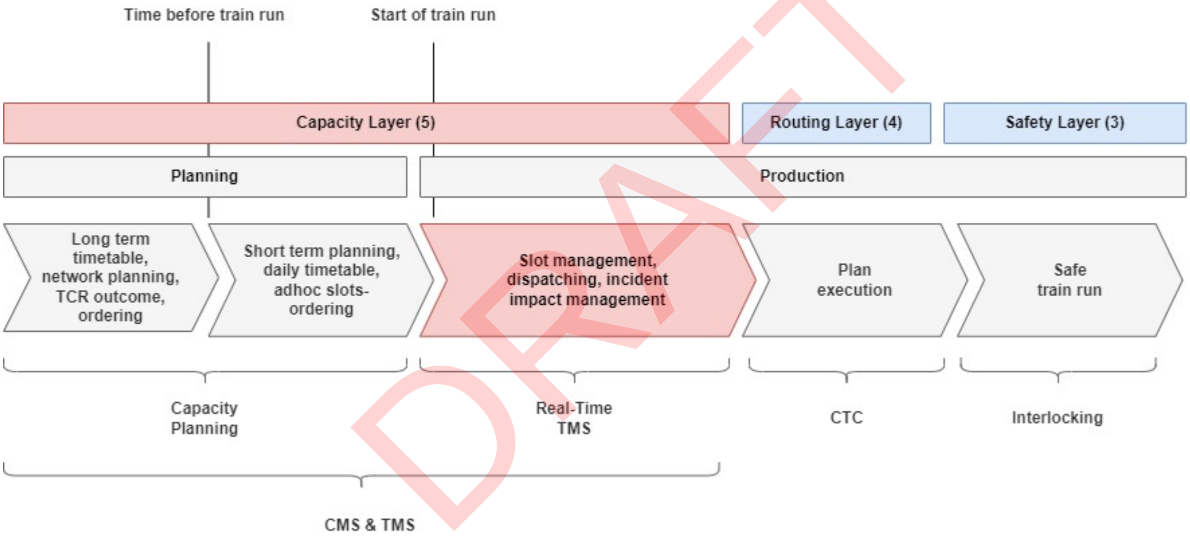
Today's railway operation is influenced by multiple factors which have a critical impact on its efficiency, punctuality, and carbon footprint. Railway operation is characterized by different organizations which focus on its individual tasks/duties such as providing the infrastructure (RIM) or providing the locomotive and the wagons (ROC). Efficient railway operation requires a smooth coordination between all involving organizations. With the CMS & TMS system Pillars project the coordination is being made: Its main goal is a conflict free operation. **[**SPT3TMS-10441 **]**

Every conflict-free operation is based on the following phases:

1. Capacity planning - in advance
2. Capacity production - during execution

The order of the phases is highly important. Every railway operation starts with the capacity planning phase and ends with the capacity production phase. **[**SPT3TMS-10707 **]**

The following figure illustrates the timeline of railway operations:



[SPT3TMS-10442 **]**

5.1 Domains

The previously mentioned phases are distinguished within the system into domains - the Capacity planning phase is covered by the CMS domain; Capacity production phase is covered by the TMS domain. The distinction between the domains is caused by specific business areas of expertise and shifted railway operation timelines. Both aspects lead to highly individual functional scopes. A domain's objective is highly individual. However, domains utilize cross cutting functionalities. **[**SPT3TMS-10444 **]**

In summary the system consists of the following domains:

Domain:	Objective:
CMS	Business area of capacity planning phase
TMS	Business area of capacity production phase
XFN	Utilization of cross functional concerns between CMS and TMS domain

[SPT3TMS-10706 **]**

The following table illustrates the key features of the respective domain: **[**SPT3TMS-10446 **]**

CMS domain:	TMS domain:	XFN domain:
Long-term capacity planning	Slot management	Topology and TCR data
Mid-term capacity planning	Dispatching	Sectional runtime calculation
TCR outcome	Incident Impact Management	
Ordering		
Short term planning		
Daily timetable		
Ad hoc slots-ordering		

[SPT3TMS-13740 **]**

5.2 Components

Domain components are the logical representation of the domain features. Domain components are further named as "components" and encapsulate the business requirements in contribution to the overall objective. **[**SPT3TMS-13739 **]**

Components consists of different characteristics which are described as the following:

- **Tasks:** How will the system contribute to the system. Refer to: 5.2.1 - Component Tasks
- **Types:** When will the component interact with the system. Refer to: 5.2.2 - Component Types

[SPT3TMS-10447 **]**

The following table illustrates the respective components:

Component:	Task:	Domain:
Topology Master Data Validation & Import	DATA-ACQUISITION	CMS

Component:	Task:	Domain:
Topology various time horizons	DATA-ACQUISITION	
Path mgmt. & processing	CALCULATION	
HMI	INTERACTION	
Manual Path Construction	CALCULATION	
Manual Path Conflict Detection	CALCULATION	
Automatic Path Construction	CALCULATION	
Capacity Plan Export	CALCULATION	
Topology Master Data Validation & Import	DATA-ACQUISITION	TMS
Daily Topology	DATA-ACQUISITION	
Capacity plan and decision processing	CALCULATION	
Deviation Detection	CALCULATION	
Forecasting	CALCULATION	
Real-Time Conflict Detection	CALCULATION	
HMI	INTERACTION	
Manual Conflict Solution	CALCULATION	
Automatic Conflict Solution	CALCULATION	
Automatic Connection Management	CALCULATION	
Topology Master Data	DATA-ACQUISITION	XFN
Topology various horizons	CALCULATION	
Sectional Runtime calculation	CALCULATION	
Train Unit Master Data	CALCULATION	

[SPT3TMS-10708 **]**

5.2.1 Component Tasks

The following table describes the different component tasks:

Task:	Description:
INTERACTION	Serves as a user interface to interact with the system to provide custom user (dispatcher) decisions
CALCULATION	Processes of information, either from external sources or user interactions.
DATA-ACQUISITION	Provision of data from external sources

[SPT3TMS-13735 **]**

5.2.2 Component Types

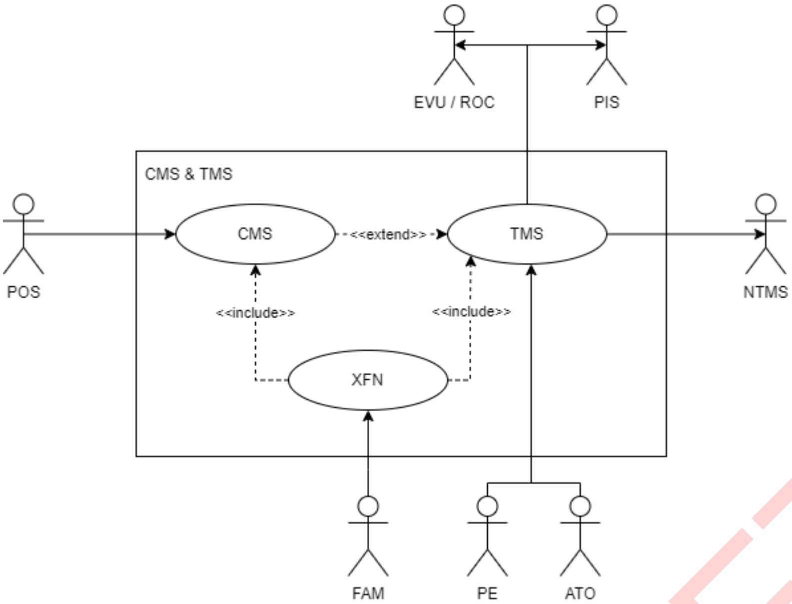
The following table describes the different component types:

Type:	Description:
MANUAL	Manual components running based on specific interactions provided by HMI.
AUTOMATIC	Automatic components running independent of user interactions such as custom business rules (outdated topology, new capacity plans, etc)

[SPT3TMS-13736 **]**

5.3 Exchanges

Interfaces are the connection between different systems with the aim of critical information exchange. The system relies on interfaces supplied by external providers (called actors) and internal components. **[**SPT3TMS-10448 **]**



In general, the main interfaces of the system are divided into north bound communication (ROC and PIS) and south bound communication (Plan Execution System (PES) and ATO Trackside (ATO-TS)). The communication of north and south bound interfaces is very important for the overall functionality of the system. Any interruption on interface level may lead to issues in the capacity planning and capacity production process. **[**SPT3TMS-13227 **]**

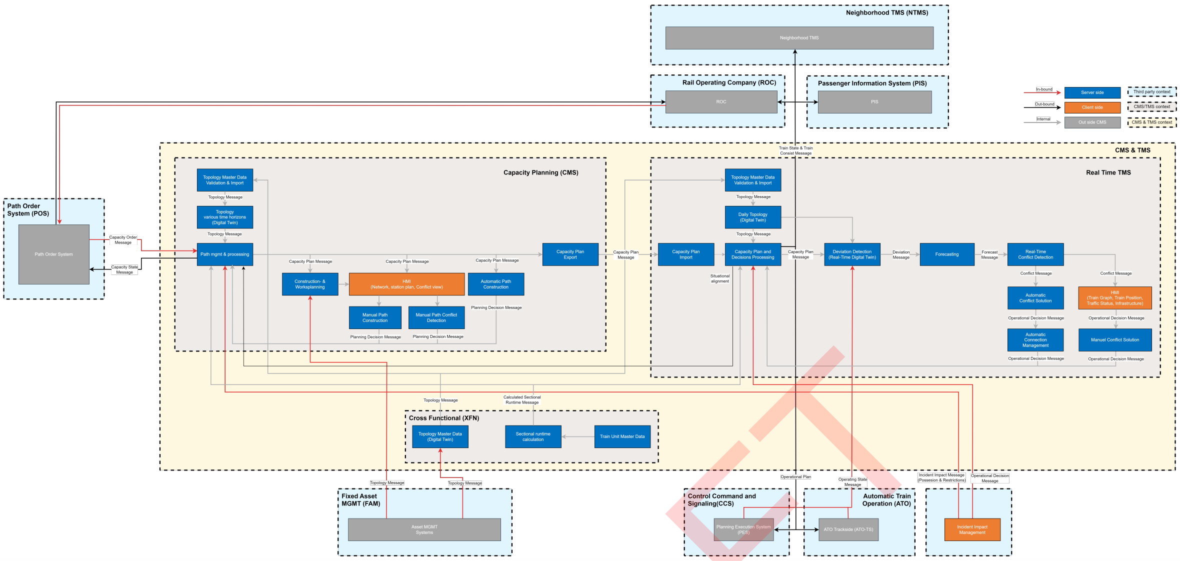
The following table illustrates the actors of the system: **[**SPT3TMS-10450 **]**

Actor:	**Objective:**
POS	Exchange of capacity requests
XFN	Provision of reference data
ATO Trackside (ATO-TS)	Plan execution time related
Plan Execution (PE)	Plan execution distance related
ROC	Provision of train data (CDN & composition)
PIS	Information base of railway punctuality & states
NTMS	Exchange operational plan to other TMS implementations (relevant for cross border traffic)
FAM	Provision of reference data

[SPT3TMS-11059 **]**

As the system is based on many independent components the communication between the components is an important aspect of the system. The communication is also based on exchanges. **[**SPT3TMS-10451 **]**

The following figure illustrates the dataflow between components and actors of the system: **[**SPT3TMS-13896 **]**



[SPT3TMS-16170 **]**

The following table illustrates the overall functional and component exchanges of the system:

Component A:	**Component B:**	**Flow:**	**Scope:**
Path Order System (POS)	Path mgmt. & processing	Bidirectional	CMS
Path mgmt. & processing	Capacity Plan Export	Unidirectional	
	HMI		
	Topology various horizons		
	Sectional runtime calculation		
HMI	Manual Path Construction		
	Manual Path Conflict Solution		
Manual Path Construction	Path mgmt. & processing		
Manual Path Conflict Detection	Path mgmt. & processing		
Capacity Plan Export	Capacity plan & decisions processing		TMS
Daily Topology	Capacity plan & decisions processing		
Capacity plan & decisions processing	Deviation Detection		

Component A:	Component B:	Flow:	Scope:
	Sectional runtime calculation		
	ROC (TMS-HMI read-only)	Bidirectional	
	PIS	Unidirectional	
	ATO Trackside (ATO-TS)		
	Plan Execution System (PES)		
	NTMS		
	Automatic Connection Management		
	Path mgmt & Processing		
Deviation Detection	Forecasting & RT Conflict Detection		
Plan Execution (PE)	Deviation Detection		
ATO Trackside (ATO-TS)	Deviation Detection		
Forecasting	Real-Time Conflict Detection		
Real-Time Conflict Detection	HMI		
	Automatic Conflict solution		
HMI	Manual Conflict Solution		
Manual Conflict Solution	Capacity plan & decisions processing		
Automatic Conflict Management	Capacity plan & decisions processing		
Incident Impact Management	Capacity plan & decisions processing		
Topology Master data	FAM		XFN
Topology various horizons	Topology Master data		
Train Unit Master Data	Sectional runtime calculation		

[SPT3TMS-10453 **]**

Furthermore, exchange items are shared between several systems.
It is of highest interest to harmonize external exchange items as any changes may lead to interruptions or incompatibilities with actors or domains. Some exchange items are harmonize already (e.g. Train State & Train Consist via TAF/TAP TSI). However if changes appears to the standardization, those changes must

be applies to the respective domain accordingly.
The following table illustrates the scope of each Exchange item:

Exchange item:	**Scope:**	**Actor:**
Topology	FUNCTIONAL-EXCHANGE	FAM
Train State & Train Consist	FUNCTIONAL-EXCHANGE	ROC
Operation State	FUNCTIONAL-EXCHANGE	CCS
Capacity State	FUNCTIONAL-EXCHANGE	POS
Capacity Order	FUNCTIONAL-EXCHANGE	POS
Capacity Plan	COMPONENT-EXCHANGE	CMS
Operational Plan	COMPONENT-EXCHANGE	TMS
Calculated Sectional Runtime	COMPONENT-EXCHANGE	CMS & TMS
Operational Decisions	COMPONENT-EXCHANGE	TMS
Situational Alignment	COMPONENT-EXCHANGE	CMS
Planning Decisions	COMPONENT-EXCHANGE	CMS
Forecast	COMPONENT-EXCHANGE	TMS
Deviations	COMPONENT-EXCHANGE	TMS
Conflicts	COMPONENT-EXCHANGE	TMS

[SPT3TMS-15273 **]**

The following matrix illustrates the overall communication of the CMS domain incl. the respective exchange item type: **[**SPT3TMS-10703 **]**

CMS	Topology Master Data Validation	Topology various time horizons	Path mgmt. & processing	HMI	Manuel Path Construction	Manuel Path Conflict Detection	Automatic Path Construction	Capacity plan creation	Path Order System	XNF/Topology Master Data	XNF/Sectional Runtime calculation	ROC	TMS
Topology Master Data Validation &		Topology Message											
Topology various time horizons			Topology Message										
Path mgmt. & processing				Capacity Plan Message	Capacity Plan Message	Capacity Plan Message	Capacity Plan Message	Capacity Plan Message	Capacity State Message				
HMI					Capacity Plan Message	Capacity Plan Message							
Manuel Path Construction			Planning Decisions Message										
Manuel Path Conflict Detection			Planning Decisions Message										
Automatic Path Construction			Planning Decisions Message										
Capacity plan creation													Capacity Plan Message
Path Order System			Capacity Order Message									Capacity State Message	
XNF/Topology Master Data	Topology Message	Topology Message											
XNF/Sectional Runtime calculation			Calculated Sectional Runtime Message										
ROC									Capacity Order				
TMS													

[SPT3TMS-15275 **]**

Legend:

CMS DOMAIN

EXTERNAL DOMAIN

EXTERNAL SYSTEM

[SPT3TMS-10802 **]**

The following matrix illustrates the overall communication of the TMS domain incl. the respective exchange item type: **[**SPT3TMS-10702 **]**

TMS	Topology Master Data Validation	Daily Topology	Capacity Plan and decision Processing	Deviation Detection	Forecasting	Real-Time Conflict Detection	HMI	Automatic Conflict Solution	Incident Impact Management	NTMS	PIS	ROC	CCS/Plan Execution System (PES)	CCS/AE	XNF / Topology Master	XNF / Sectional Runtime	Automatic Connection Management	CMS	Manual Conflict Solution
XNF / Topology Master Data	Topology Message																		
XNF / Sectional Runtime			Calculated Sectional Runtime Message																
Automatic Connection			Operational Decisions Message																
Topology Master Data Validation & Import		Topology Message																	
Daily Topology			Topology Message																
Capacity Plan and decision Processing				Operational Plan Message	Operational Plan Message		Operational Plan Message	Operational Plan Message	Operational Plan Message	Operational Plan Message	Operational Plan Message	Operational Plan Message (via TSI/TAF-TAP)	Operational Plan Message (via SCI-TMS-CCS)	Operational Plan Message (via SCI-TMS-CCS)		Operational Plan Message		Situational Alignment	
Deviation Detection					Deviation Message		Deviations Positions	Deviations Positions	Deviations Positions										
Forecasting						Forecast Message					Forecast Message	Forecast Message (via TSI/TAF-TAP)							
Real-Time Conflict Detection							Conflict Message	Conflict Message											
HMI																			Operational Decisions Message
Automatic Conflict Solution			Operational Decisions Message														Operational Decisions Message		
Incident Impact Management			Operational Decisions Message																
NTMS			Train State & Train Consist Message																
PIS																			
ROC			Train State & Train Consist Message																
CCS/Plan Execution System (PES)			Operation State Message	Operation State Message															
CCS/AE			Operation State Message	Operation State Message															
CMS			Capacity Plan Message																
Manual Conflict Solution			Operational Decisions Message																

Legend:

TMS DOMAIN
EXTERNAL DOMAIN
EXTERNAL SYSTEM

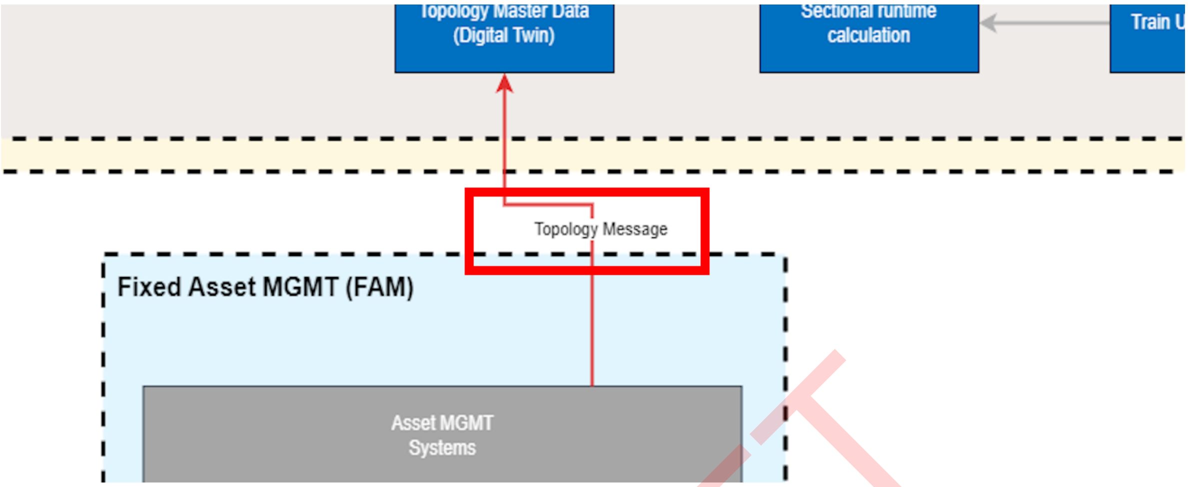
[SPT3TMS-10801 **]**

5.3.1 Exchange Items

5.3.1.1 Topology Exchange Item

SCOPE: FUNCTIONAL-EXCHANGE ACTOR: FAM

The Topology Exchange item is located between FAM and XFN: **[**SPT3TMS-10810 **]**



[SPT3TMS-10412 **]**

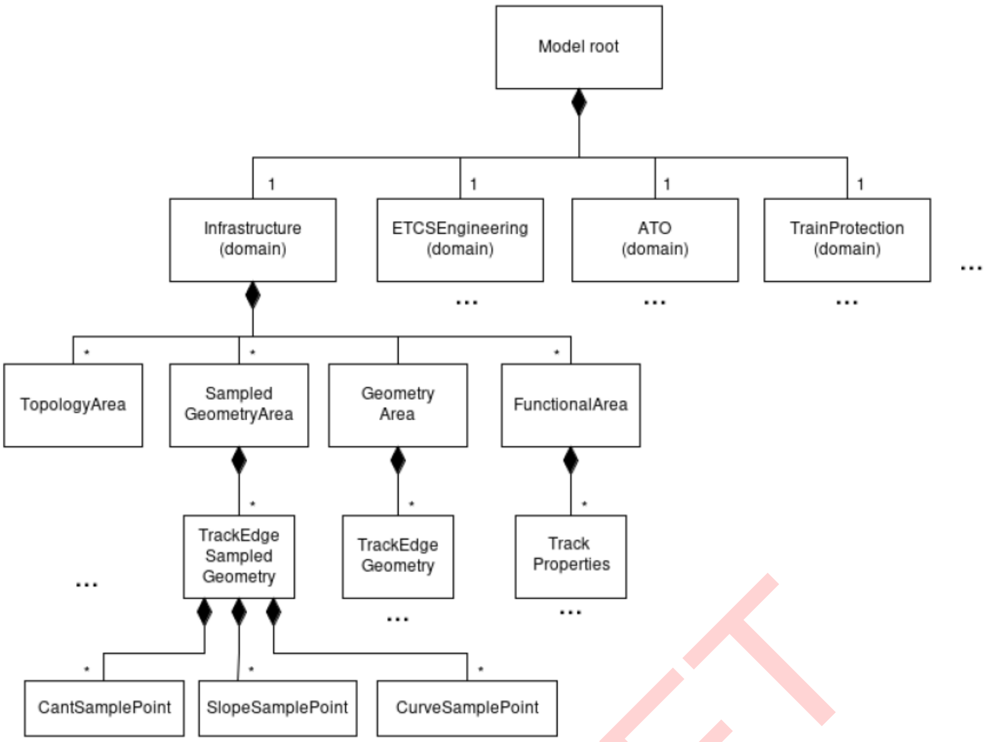
The Topology Exchange item is a complex representation of the Topology model of the respective infrastructure Manager (IM). The Topology Exchange item is capable to represent the following types: **[**SPT3TMS-10414 **]**

- Track Properties
- Speed profiles
- Stop locations
- ETCS Marker
- Platform
- Operational Point
- Tunnel
- Bridge
- Underpass
- Track Vacancy Proving Section (TVPS)
- Level Crossing
- National border
- Buffer Stop

[SPT3TMS-10413 **]**

The Topology Exchange item is a composition of several exchange items that constitute the topology data model mostly related to Infrastructure information, as shown in the figure below. Every Exchange item can be divided into exchange items (e.g., TopologyArea, GeometryArea, etc.) depending on its needs. **[**SPT3TMS-10416 **]**

The figure below depicts an example Exchange item structure with the current included content:



[SPT3TMS-10411 **]**

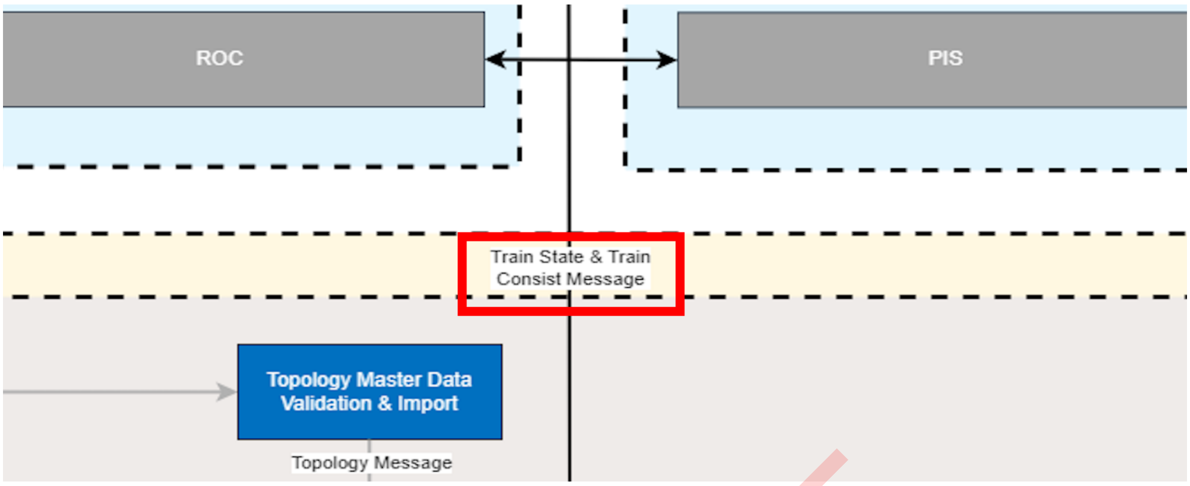
References:

TCCS - Data Model_10_INFRA **[**SPT3TMS-16220 **]**

TCCS - Data Model_00_Guide **[**SPT3TMS-16219 **]**

5.3.1.2 Train State & Train Consist Exchange Item

SCOPE: FUNCTIONAL-EXCHANGE ACTOR: ROC



[SPT3TMS-10462]

The Train State & Train Consist Exchange item is located between ROC and TMS. and is a subset of several exchange items, including the following: [SPT3TMS-10711]

- Path Request
- Path Details
- Consignment Note Data
- Train Composition
- Train Accepted
- Train Not Suitable
- Train Ready
- Train Running Information
- Train Running Forecast
- Train Position
- Train Running Interruption
- Delay Justification

[SPT3TMS-10507]

All Exchange items comply to the specification of TAF/TAP TSI. Please refer to the system definition to read more about the details of the Exchange item. [SPT3TMS-10506]

More information about the usage of this Exchange item can be found in the chapter 6.2 - Interface logical view . [SPT3TMS-16214]

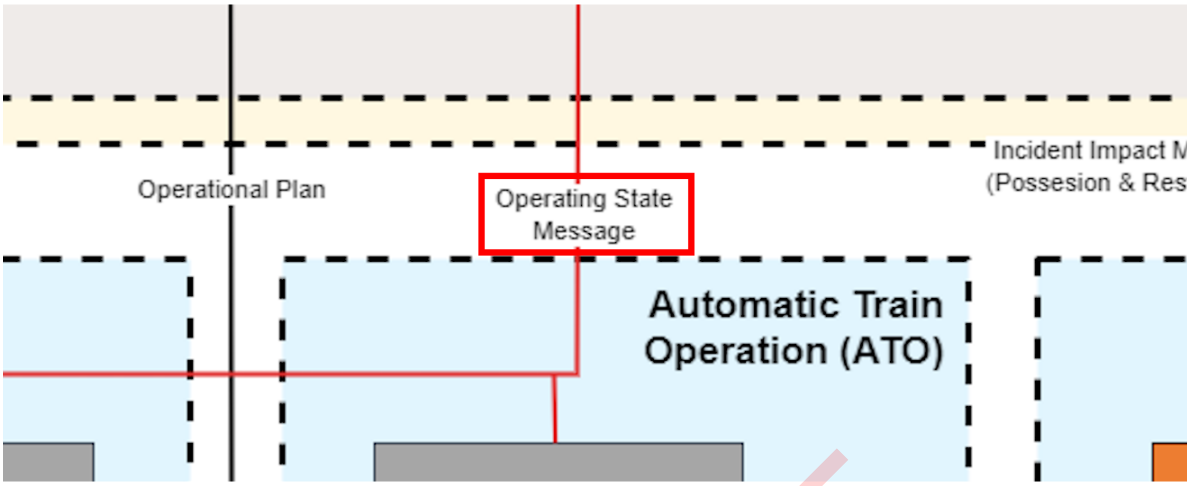
Reference:

SPT3-CMS_TMS_20_Workspace_CMS_TMS_System_Definition_V0_5.pdf, P. 23 [SPT3TMS-16213]

RU/IM COMMUNICATION APPLICATION GUIDE: https://www.era.europa.eu/system/files/2022-11/TAP%20B%2056%20-%20RU_IM%20Communication%20Application%20Guide.pdf [SPT3TMS-10417]

5.3.1.3 Operating State Exchange Item

SCOPE: FUNCTIONAL-EXCHANGE ACTOR: CCS



[SPT3TMS-10461]

The Operating State Exchange item is located between CCS and TMS.
The Operating State Exchange item is a subset of several Exchange items, including the following:
[SPT3TMS-10721]

- Train Unit Report
- Track Ocuupation State
- DPS Group State
- Restriction Area State
- Warning Area State

[SPT3TMS-10514]

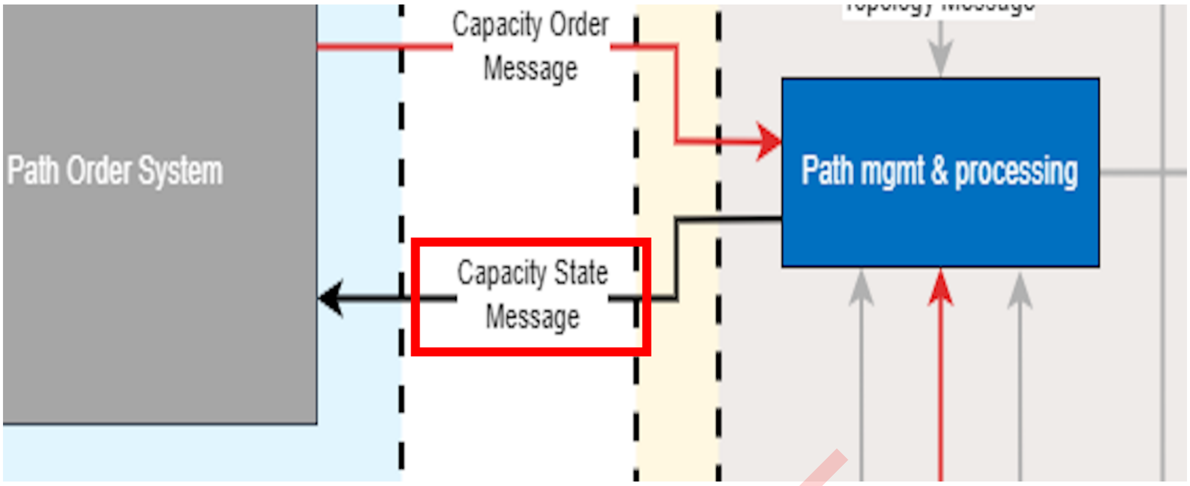
More information about the usage of this Exchange item can be found in the chapter 6.2.3 - TMS-CCS Interface .

Reference:

T3-ConceptInterfaceTMSCCS [SPT3TMS-16215]

5.3.1.4 Capacity State Exchange Item

SCOPE: FUNCTIONAL-EXCHANGE ACTOR: POS



[SPT3TMS-10464]

The Capacity State Exchange item is located between POS and CMS.
The Capacity State Exchange item is a subset of several Exchange items available for the following: [SPT3TMS-10723]

- Path Confirmed / Path Details Refused
- Path Not Available
- Answer Not Possible
- Path Cancelled

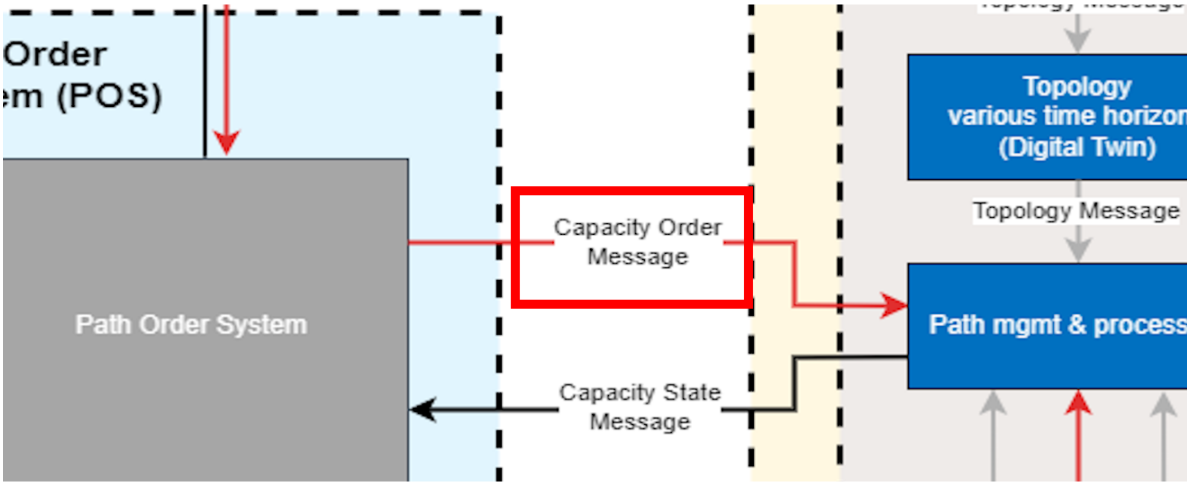
[SPT3TMS-10523]

Reference:

TAP B 56 - RU_IM Communication Application Guide.pdf, P. 94 [SPT3TMS-16197]

5.3.1.5 Capacity Order Exchange Item

SCOPE: FUNCTIONAL-EXCHANGE ACTOR: POS



[SPT3TMS-10463 **]**

The Capacity Order Exchange item is located between POS and CMS.
The Capacity Order Exchange item is a subset of several Exchange items available for the following: **[**SPT3TMS-10722 **]**

- Path Request
- Answer Not Possible
- Path Details
- Path Confirmed / Path Details Refused
- Path Cancelled
- Path Not Available
- Utilization Notification
- Dossier
- Receipt Confirmation

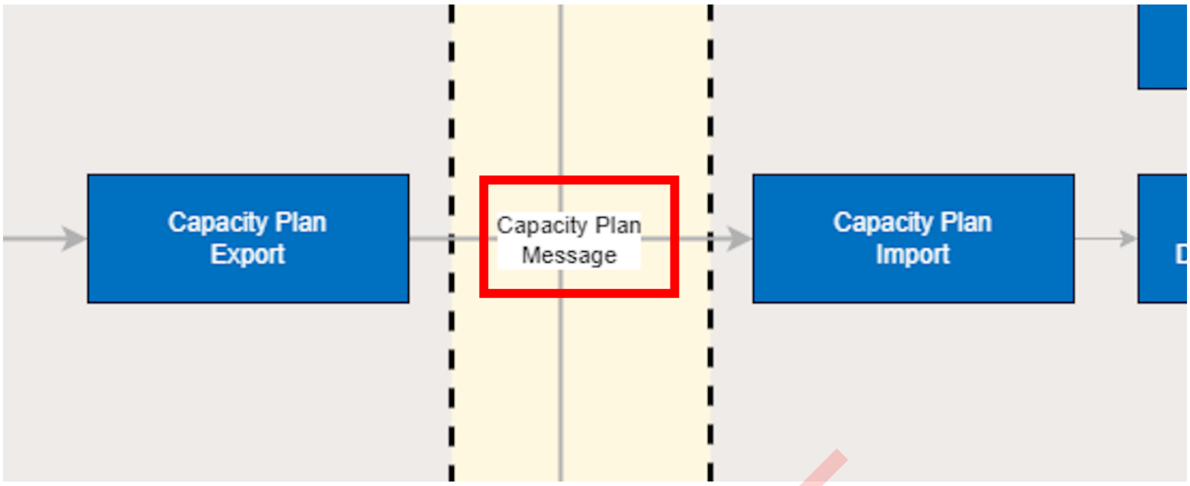
[SPT3TMS-10516 **]**

Reference:

TAP B 56 - RU_IM Communication Application Guide.pdf, P. 94 **[**SPT3TMS-16196 **]**

5.3.1.6 Capacity Plan Exchange Item

SCOPE: COMPONENT-EXCHANGE ACTOR: CMS



[SPT3TMS-10455 **]**

The Capacity Plan Exchange item is located between CMS and TMS. **[**SPT3TMS-10725 **]**

The Capacity Plan Exchange item is a subset of several Exchange items available for the following:

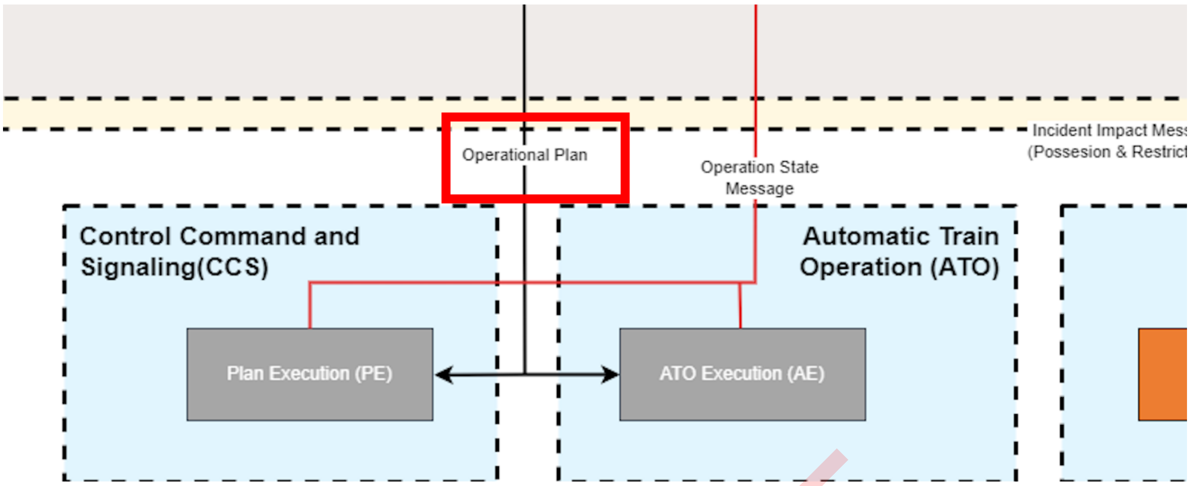
- Train Timetable
- Train Links
- Track Reservations (Parking, Shunting etc.)
- Temporary Capacity Restrictions (TCR)
- Crew related information (RU)
- Rolling Stock / Load related information (RU)

Main attributes will be continued at a later stage. **[**SPT3TMS-10739 **]**

Reference:
D3.2.2.2 Concept for CapacityPlanning-CapacityProduction interface_003.pdf, P.5 **[**SPT3TMS-16180 **]**

5.3.1.7 Operational Plan Exchange Item

SCOPE: COMPONENT-EXCHANGE ACTOR: TMS



[SPT3TMS-10454 **]**

The Operational Plan Exchange item is located between Capacity plan and decisions processing and Deviation Detection. **[**SPT3TMS-10726 **]**

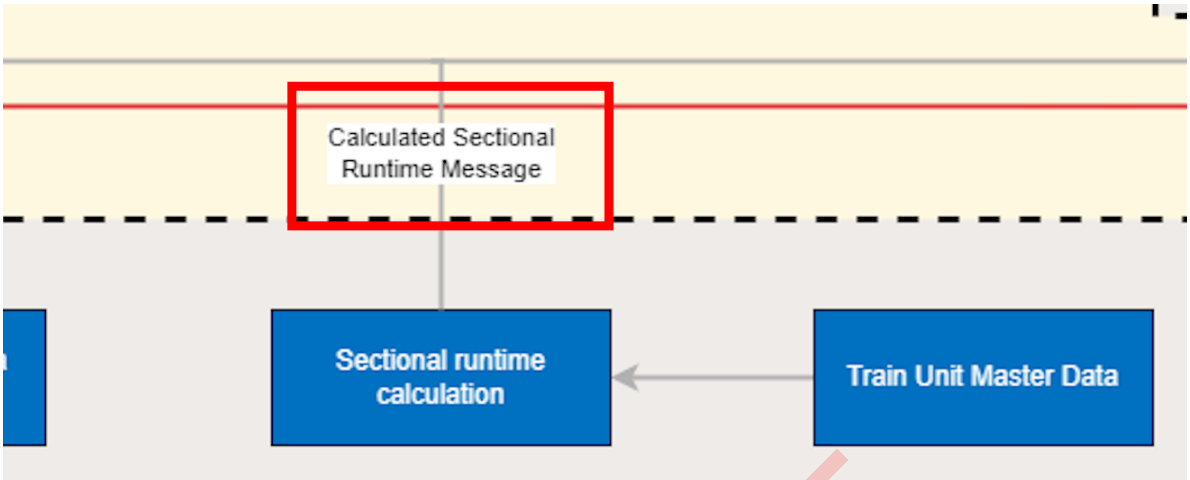
The Operational Plan Exchange item is a subset of several Exchange items available for the following:

- Operational Plan Movement (incl. nested classes)
- Movement Events (incl. nested classes)
- Movement Restrictions (incl. nested classes)

Reference: Concept Interface TMS-CCS **[**SPT3TMS-14652 **]**

5.3.1.8 Calculated Sectional Runtime Exchange Item

SCOPE: COMPONENT-EXCHANGE ACTOR: CMS & TMS



[SPT3TMS-10459 **]**

The Calculated Sectional Runtime Exchange item is located between XFN and CMS/TMS. **[**SPT3TMS-10727 **]**

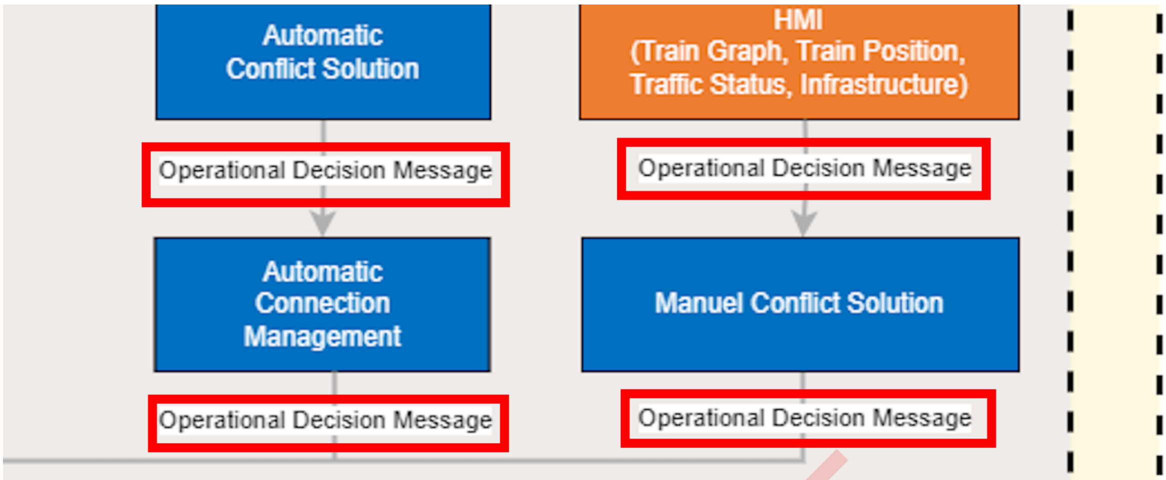
The following table describes the attributes of the runtime Exchange item:

Attribute:	**Description:**
id	Identifier for runtime (for e.g. trackEdgeId or trackEdgeSegment).
startingPoint	Information about the starting point of the runtime calculation
endingPoint	Information about the ending point of the runtime calculation
paths	List of paths
totalDistance	Total distance in km
totalRuntime	Total runtime in hh:mm:ss

Main attributes will be continued at a later stage. **[**SPT3TMS-10728 **]**

5.3.1.9 Operational Decisions Exchange Item

SCOPE: COMPONENT-EXCHANGE ACTOR: TMS



[SPT3TMS-10457 **]**

The Operational Decisions Exchange item is located between Capacity plan and decisions processing, HMI, Automatic Conflict solution, Automatic Connection Management, and Incident Impact Management. Operational Decision Exchange items are subject to one decision. **[**SPT3TMS-10731 **]**

The following table describes the attributes of the operational decisions Exchange item:

Attribute:	**Description:**
id	Identifier of the daily topology element
operationalPlan	Reference to original operational plan
type	Type of decision
description	Description of the decisions.
validFrom	Decision is valid from given timestamp. If null, the decision is valid from now on.
validTo	Decision is valid to given timestamp. If null, the decision is valid until further notice.

Main attributes will be continued at a later stage. **[**SPT3TMS-10732 **]**

The following decision types are available:

Type:	**Description:**
ROUTE_CHANGE	Change of route
TRACK_CHANGE	Change of track
PLATFORM_CHANGE	Change of platform

Type:	Description:
ARRIVAL_TIME_CHANGE	Change of arrival time
DEPARTURE_TIME_CHANGE	Change of departure time
DRIVING_STRATEGY_CHANGE	Change of driving strategy
CANCELLATION	Cancellation of a train (parts)
EXTRA_TRAIN	Extra train
REDIRECTION_BYPASS	Redirection / bypass
CONNECTION_CHANGE	Change of connections
STOP_SKIP	Skip a stop
ROLLING_STOCK_CHANGE	Change of resource allocation (rolling stock)
CHANGE_OF_TRAIN_SEQUENCE	Change of train sequences.
ACTIVATING_SINGLE_TRAIN_FOR_FORECAST	Consider train for forecast
INACTIVATING_SINGLE_TRAIN_FOR_FORECAST	Suppress train in forecast

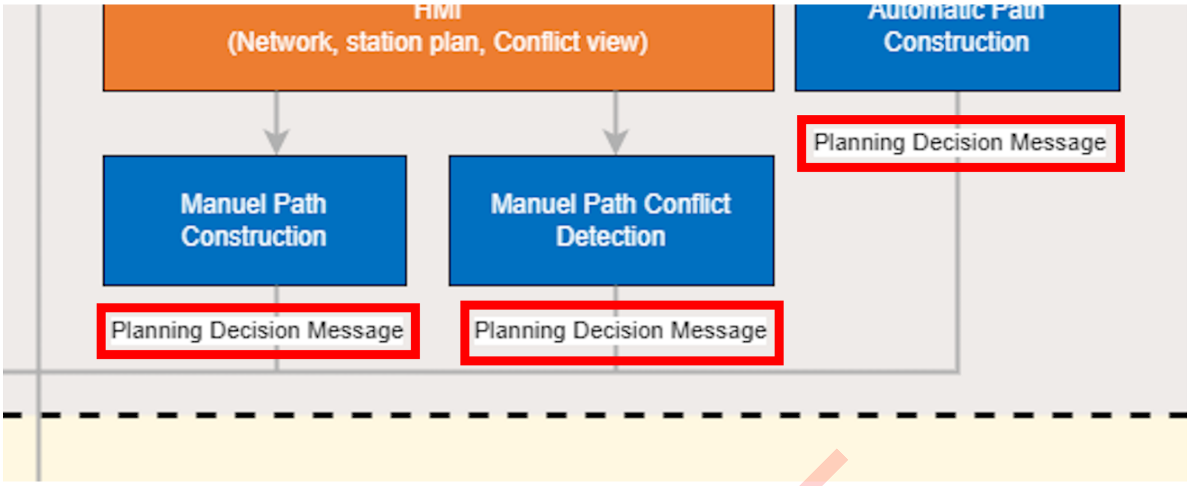
[SPT3TMS-10803 **]**

Reference:

SPT3-CMS_TMS_20_Workspace_CMS_TMS_System_Concept_R2_V1_5.pdf, P. 47 **[**SPT3TMS-16199 **]**

5.3.1.10 Planning Decisions Exchange Item

SCOPE: COMPONENT-EXCHANGE ACTOR: CMS



[SPT3TMS-15281 **]**

The Planning Decisions Exchange item is located between Manual Path Construction, Manual Path Conflict Detection, Automatic Path Construction and Path mgmt & processing.

Planning Decisions Exchange items are subject to one decision. **[**SPT3TMS-10806 **]**

The following table describes the attributes of the Planning Decisions Exchange item:

Attribute:	**Description:**
id	Identifier of the daily topology element
capacityPlan	Reference to original capacity plan
type	Type of decision
description	Description of the decisions.
validFrom	Decision is valid from given timestamp. If null, the decision is valid from now on.
validTo	Decision is valid to given timestamp. If null, the decision is valid until further notice.

Main attributes will be continued at a later stage. **[**SPT3TMS-10805 **]**

The following decision types are available:

Type:	**Description:**
ROUTE_CHANGE	Change of route
TRACK_CHANGE	Change of track
PLATFORM_CHANGE	Change of platform

Type:	Description:
ARRIVAL_TIME_CHANGE	Change of arrival time
DEPARTURE_TIME_CHANGE	Change of departure time
DRIVING_STRATEGY_CHANGE	Change of driving strategy
CANCELLATION	Cancellation of a train (parts)
EXTRA_TRAIN	Extra train
REDIRECTION_BYPASS	Redirection / bypass
CONNECTION_CHANGE	Change of connections
STOP_SKIP	Skip a stop
ROLLING_STOCK_CHANGE	Change of resource allocation (rolling stock)
CHANGE_OF_TRAIN_SEQUENCE	Change of train sequences.
ACTIVATING_SINGLE_TRAIN_FOR_FORECAST	Consider train for forecast
INACTIVATING_SINGLE_TRAIN_FOR_FORECAST	Suppress train in forecast

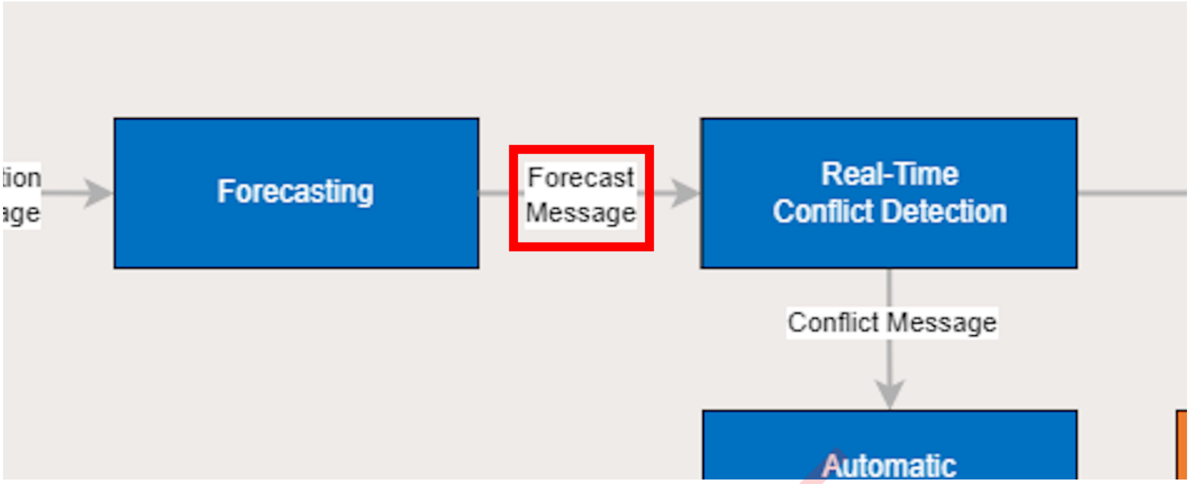
[SPT3TMS-10808 **]**

Reference:

SPT3-CMS_TMS_20_Workspace_CMS_TMS_System_Concept_R2_V1_5.pdf, P. 47 **[**SPT3TMS-16198 **]**

5.3.1.11 Forecast Exchange Item

SCOPE: COMPONENT-EXCHANGE ACTOR: TMS



[SPT3TMS-10465 **]**

The Forecast Exchange item is located between Forecasting & RT Conflict Detection and Automatic Conflict solution and Automatic Connection Management.

Forecast Exchange items are subject to one individual train run referring to the whole journey from current to future. **[**SPT3TMS-10733 **]**

The following table describes the attributes of the forecast Exchange item:

Attribute:	**Description:**
Id	Identifier of the forecast
trainId	Train identifier for which the following forecast will apply.
location	Forecasted path precise location of the train
forecast	Forecast in seconds (where plus (+) illustrates delays, minus (-) earlier as expected)

[SPT3TMS-10525 **]**

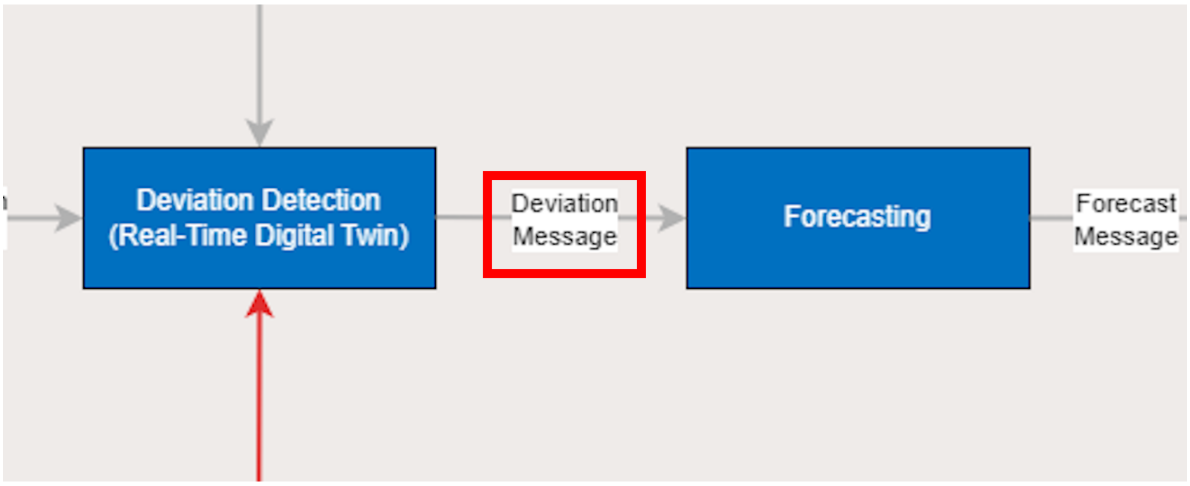
Example:

- 00316, Bern Gleis 4, +0
- 30316, Olten Gleis 7, +10
- 70316, Zürich Gleis 31, -5
- 70316, Bern, Signal, D4, +

[SPT3TMS-10535 **]**

5.3.1.12 Deviations Exchange Item

SCOPE: COMPONENT-EXCHANGE ACTOR: TMS



[SPT3TMS-10460 **]**

The Deviations Exchange item is located between Deviation Detection and Forecasting & RT Conflict Detection: Deviation Exchange items are subject to one individual train run referring to the current situation. **[**SPT3TMS-10734 **]**

The following table describes the attributes of the deviations Exchange item:

Attribute:	**Description:**
id	Identifier of the deviation
trainId	Train identifier for which the following forecast will apply.
location	Path precise location of the deviation
deviation	Deviations - represents a broad range of circumstances whenever production plan doesn't reflect to capacity plan. May be time related, changes in platform, etc.

[SPT3TMS-10533 **]**

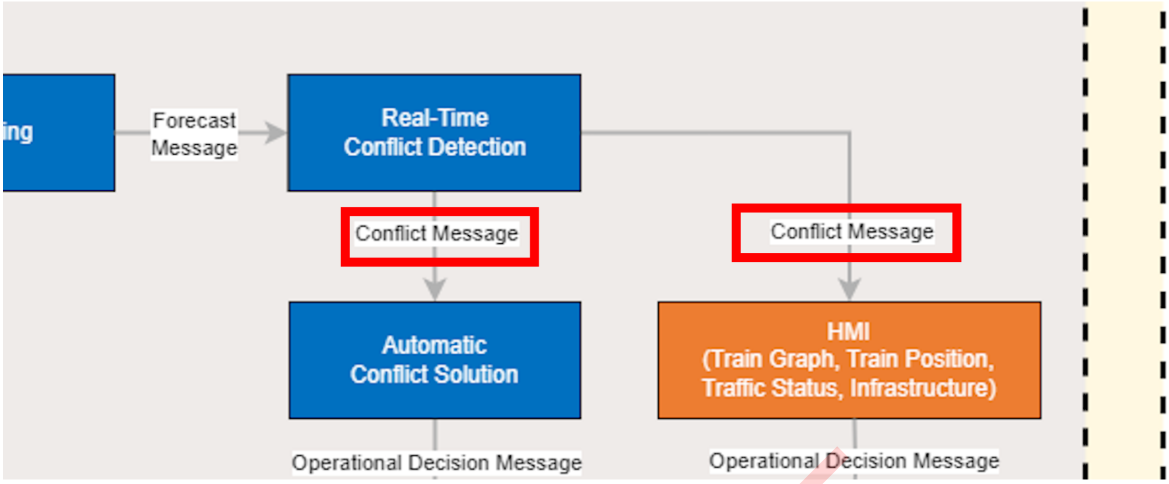
Example:

- 00316, Bern Gleis 4, +0
- 30316, Olten Gleis 1, +10
- 70316, Zürich Gleis 2, -5
- 70316, Bern, Signal, D4, +3

[SPT3TMS-10539 **]**

5.3.1.13 Conflicts Exchange Item

SCOPE: COMPONENT-EXCHANGE ACTOR: TMS



[SPT3TMS-10458 **]**

The Conflict Exchange item is located between Forecasting & RT Conflict Detection and Automatic Conflict solution. Conflict Exchange items are subject to one specific location referring to forecasted conflicts in the (near) future. **[**SPT3TMS-10735 **]**

The following table describes the attributes of the conflicts Exchange item:

Attribute:	**Description:**
Id	Identifier of the conflict
location	Path precise location of the conflict
type	Conflict type
trainIds	List of train identifiers involved in the conflict
timestamp	Timestamp of the conflict

[SPT3TMS-10540 **]**

The following conflict types are available:

Type:	**Description:**
CROSSING_CONFLICT	Conflict based on train crossings
OVERTAKING_CONFLICT	Conflict based on train overtaking
HEADWAY_CONFLICT	Conflict based on train headway
JUNCTION_CONFLICT	Conflict based on train junction
JOINING_CONFLICT	Conflict based on train joining
STATION_ENTRY_EXIT_CONFLICT	Conflict based on station entry exit

Type:	Description:
OTHER_PATH_EXCLUSIONS_CONFLICT	Conflict based on other path exclusions
TRAIN_CAPABILITY_CONFLICT	Conflict based on train capability

[SPT3TMS-10712 **]**

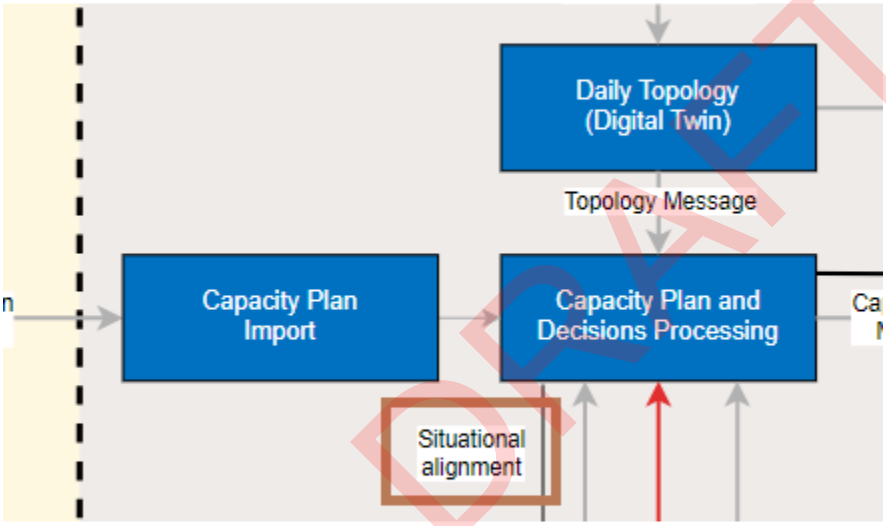
Example:

- Bern Gleis 4, 2023-09-20 13:29:00, STATION_ENTRY_EXIT_CONFLICT, 00371, 21945

[SPT3TMS-10537 **]**

5.3.1.14 Situational Alignment Exchange Item

SCOPE: COMPONENT-EXCHANGE ACTOR: TMS



[SPT3TMS-16151 **]**

The Situational Alignment Exchange item is located between Capacity Plan and Decision Processing and Path Management and Processing. Situational Alignment items are foreseen to align CMS whan some unforeseen events occur, which change the traffic context in an unplanned way for a period which impacts on CMS time horizon and could prevent a proper short or medium-term planning for CMS.
[SPT3TMS-16153 **]**

At the time being, only unforeseen restrictions are planned to be communicated to CMS. No Operational Plan Operational Movements are envisaged tobe modified by TMS, which instead shall notify this need to CMS and CMS shall generate and publish an update to the Capacity Plan.

The following table describes the kind of the message Exchange item:

Attribute:	Description:
Operational Restriction	An unforeseen restriction had to be planned by TMS and is sent CMS for alignment. CMS shall consider this restriction will keep it as it is or close it

Attribute:	Description:
	opening a new planned one with an estimated duration
Operational Movement Id	Identifier of the Operational movement suggested to be removed
Contingengy Plan Id	Identifier of the Contingency plan suggested to be applied

6 Logical Architecture (Runtime View)

The following chapters describe concrete behavior and interactions of the system's building blocks.
[SPT3TMS-10544 **]**

The Runtime View is structured into the following sub chapters:

- **Domain logical view:** describes the logical interaction and workflow inside CMS & TMS scope.
- **Interface logical view:** describes the logical interaction and workflow outside CMS & TMS scope.
- **Scenario view:** describes concrete scenarios, involving multiple domains and interfaces.

[SPT3TMS-15280 **]**

6.1 Domain logical view

The following chapters describe the interaction and workflow inside domain scope.
The Domain logical view is structured into the following sub chapters:

- **Actors:** Describes all interacting actors of the component incl. components and external systems.
- **Exchange items:** Describes inbound and outbound Exchange items to the component.
- **Interactions:** Describes all bidirectional and unidirectional interactions of the component.
- **Workflow:** Describes the concrete workflow of the component, step by step.
- **Workflow triggers:** Describes the point in time when the component starts to perform its tasks.

[SPT3TMS-15283 **]**

The domain logical view highlights different component types. Those component types can be described as the following:

Type:	Description:
SERVER	Logical server side component are the base for all calculations but may not be able to represent the results in a user-friendly way to the end-user. Server side components interact with client side components therefore.
CLIENT	Representational client component are the base for all representations to the end-user in a user-friendly manner. Client components actively communicate with server side components to exchange date. Client components are installed / running on end-user devices.

[SPT3TMS-15282 **]**

6.1.1 Capacity Planning (CMS)

CMS may consists in the future of a planning ability to generate one rolling plan (immediate future to many years ahead, ad-hoc traffic or regular timetable), manage all types of track usages (stabling, train movements, construction, maintenance, occupation for other reasons, reduced/changed track usage conditions, etc.), manage all “planning partners” affecting the plan such as national/international RIMs and ROCs and dependent supporting services (e.g., energy, telecommunication, etc.), construction, maintenance, resource providers, planners for activities adjacent to the network, network access facilities, shunting yards, terminals, stations, connected transport systems, and any other actors on whom the plan depends (or vice versa). **[**SPT3TMS-10547 **]**

The following table illustrates the CMS components:

Component:	**Type:**	**Artifact:**
Topology Master Data Validation & Import	SERVER	TOPOLOGY
Topology various time horizons	SERVER	VARIOUS-TIME-HORIZONS-TOPOLOGY
Path mgmt. & processing	SERVER	CAPACITY-PLAN
HMI	CLIENT	-
Manual Path Construction	SERVER	PLANNING-DECISIONS
Manual Path Conflict Detection	SERVER	PLANNING-DECISIONS
Automatic Path Construction	SERVER	PLANNING-DECISIONS
Capacity Plan Export	SERVER	CAPACITY-PLAN

[SPT3TMS-15277 **]**

6.1.1.1 Topology Master Data Validation & Import

TYPE: SERVER ACTOR: CMS ARTIFACT: TOPOLOGY

This component imports topology data into the CMS context by considering specific CMS validation rules. With the import of Topology data, the CMS domain decouples the Topology management.
[SPT3TMS-15372 **]**

6.1.1.1.1 Actors

The following actors are directly involved in the components communication:

Actor:	**Direction:**	**Scope:**	**Action:**
Topology Master Data	INBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.
Topology various time horizons	OUTBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.

[SPT3TMS-15276 **]**

6.1.1.1.2 Exchange items

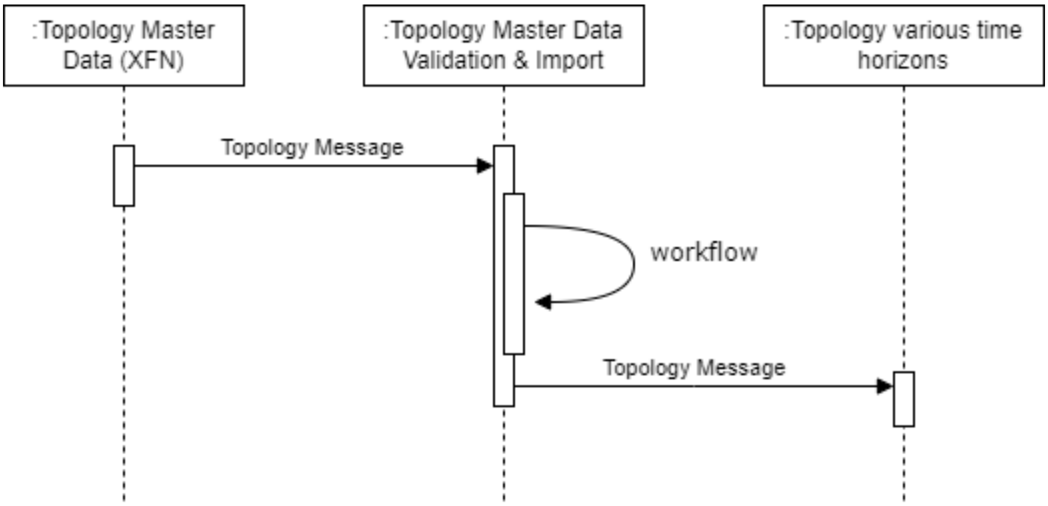
The following Exchange items are involved in the components process:

Exchange item:	**Direction:**
Topology	INBOUND
Topology	OUTBOUND

[SPT3TMS-15279 **]**

6.1.1.1.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15307 **]**

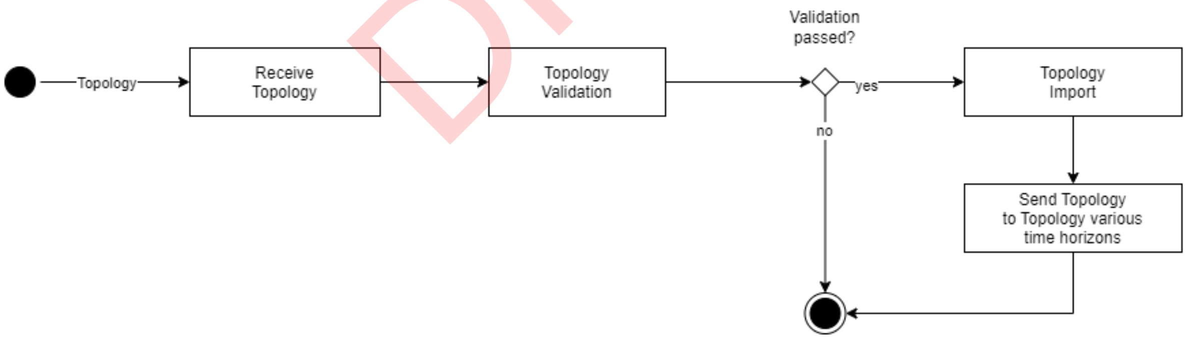
6.1.1.1.4 Workflow

The workflow consists of the following tasks:

- **Receive Topology:** from Topology Master Data (XFN).
- **Execute Validation:** to verify and protect the system from poor data quality provided from external sources.
- **Import Topology:** into the CMS domain to avoid dependencies to other domains.
- **Share Topology:** to other components.

[SPT3TMS-15375 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15306 **]**

6.1.1.1.4.1 Workflow triggers

The following table describes the workflow triggers:

Trigger:	**Business-Event:**	**Action:**
Addition or modification to Topology	An addition or modification to Topology must be considered in CMS.	Initiate Topology Validation & Import

[SPT3TMS-15464 **]**

6.1.1.2 Topology various time horizons

TYPE: SERVER ACTOR: CMS ARTIFACT: VARIOUS-TIME-HORIZONS-TOPOLOGY

This component imports topology data into the CMS context by considering various time horizons. Topology data are crucial for precise calculations. As Capacity Plans are considered for different timings (e.g. in 10 years), changes reflected in the Topology must be considered. Therefore, various time horizons of the Topology must be calculated to include them into the calculation. With different time frames of the Topology, the component will provide the result of the calculations to the Path mgmt. & processing component. **[**SPT3TMS-10546 **]**

6.1.1.2.1 Actors

The following actors are directly involved in the components communication:

Actor:	**Direction:**	**Scope:**	**Action:**
Topology various time horizons	INBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.
Path mgmt. & processing	OUTBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.

[SPT3TMS-15450 **]**

6.1.1.2.2 Exchange items

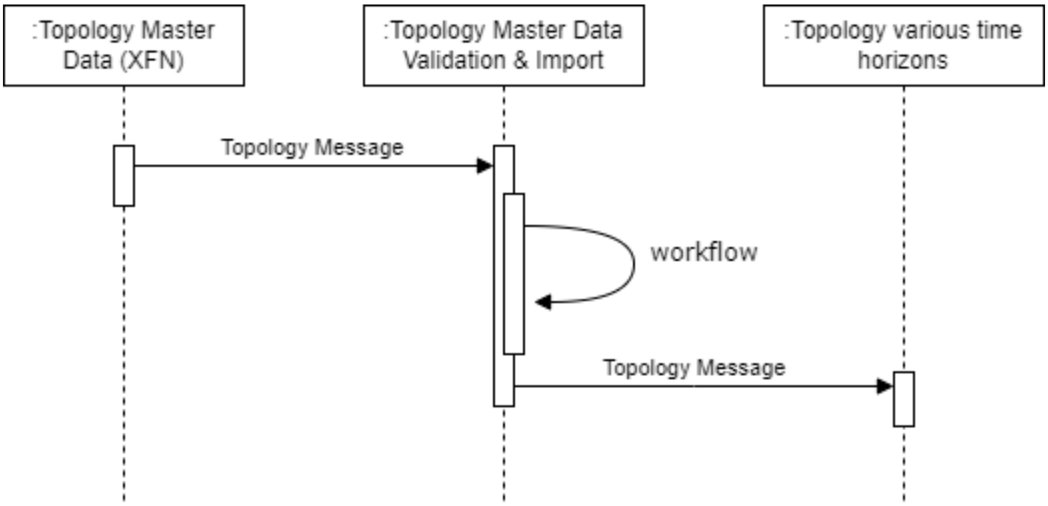
The following Exchange items are involved in the components process:

Exchange item:	**Direction:**
Topology	INBOUND
Topology	OUTBOUND

[SPT3TMS-15278 **]**

6.1.1.2.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15309]

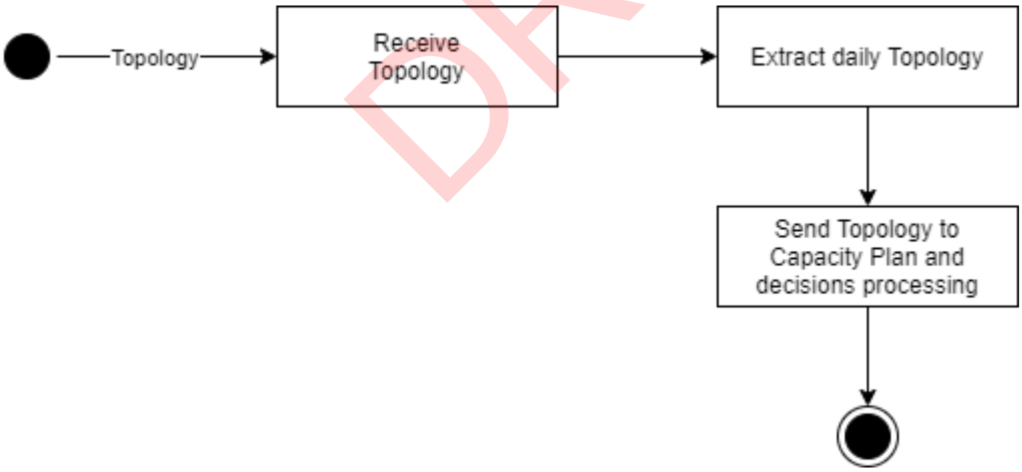
6.1.1.2.4 Workflow

The workflow consists of the following tasks:

- **Receive Topology:** from Topology Master Data Validation & Import.
- **Extract Topology:** for various time horizons
- **Share Topology:** to other components.

[SPT3TMS-15374]

The following activity diagram illustrates the workflow:



[SPT3TMS-15308]

6.1.1.2.4.1 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
Addition or modification to Topology	An addition or modification to Topology must be considered in current Topology various time horizons.	Initiate Topology extraction

[SPT3TMS-15463]

6.1.1.3 Path mgmt. & processing

TYPE: SERVER ACTOR: CMS ARTIFACT: CAPACITY-PLAN

The Path mgmt. & processing component represents the central point of the CMS architecture and is responsible to manage Capacity Orders and Capacity Plans. **[**SPT3TMS-10545 **]**

6.1.1.3.1 Actors

The following actors are directly involved in the components communication:

Actor:	**Direction:**	**Scope:**	**Action:**
Path Order System (POS)	INBOUND	FUNCTIONAL-EXCHANGE	Provision of Capacity orders via Capacity Order Exchange item.
Path Order System (POS)	OUTBOUND	FUNCTIONAL-EXCHANGE	Capacity State via Capacity State Exchange item.
Incident Impact Management	INBOUND	FUNCTIONAL-EXCHANGE	Provision of Incidents via Incident Impact Exchange item.
Topology various horizons	INBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.
Automatic Path Construction	INBOUND	COMPONENT-EXCHANGE	Provision of Planning Decisions via Planning Decisions Exchange item.
Automatic Path Construction	OUTBOUND	COMPONENT-EXCHANGE	Provision of Capacity Plan via Capacity Plan Exchange item.
Sectional Runtime Calculation	INBOUND	COMPONENT-EXCHANGE	Provision of sectional runtime calculations via Calculated Sectional Runtime Exchange item.
Capacity Export	OUTBOUND	COMPONENT-EXCHANGE	Capacity Plan Export via Capacity Plan Exchange item
HMI	OUTBOUND	COMPONENT-EXCHANGE	Provision of Capacity Plan via Capacity Plan Exchange item.

[SPT3TMS-15449 **]**

6.1.1.3.2 Exchange items

The following Exchange items are involved in the components process:

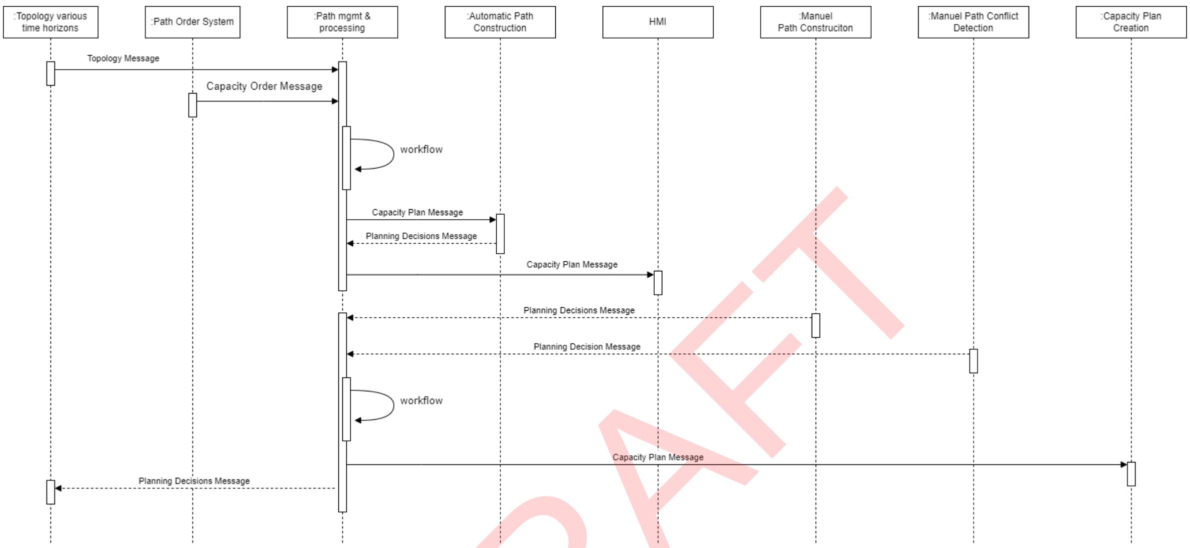
Exchange item:	**Direction:**
Capacity Order	INBOUND
Incident Impact	INBOUND
Topology	INBOUND
Planning Decisions	INBOUND
Calculated Sectional Runtime	INBOUND

Exchange item:	Direction:
Capacity State	OUTBOUND
Capacity Plan	OUTBOUND

[SPT3TMS-15284 **]**

6.1.1.3.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15303 **]**

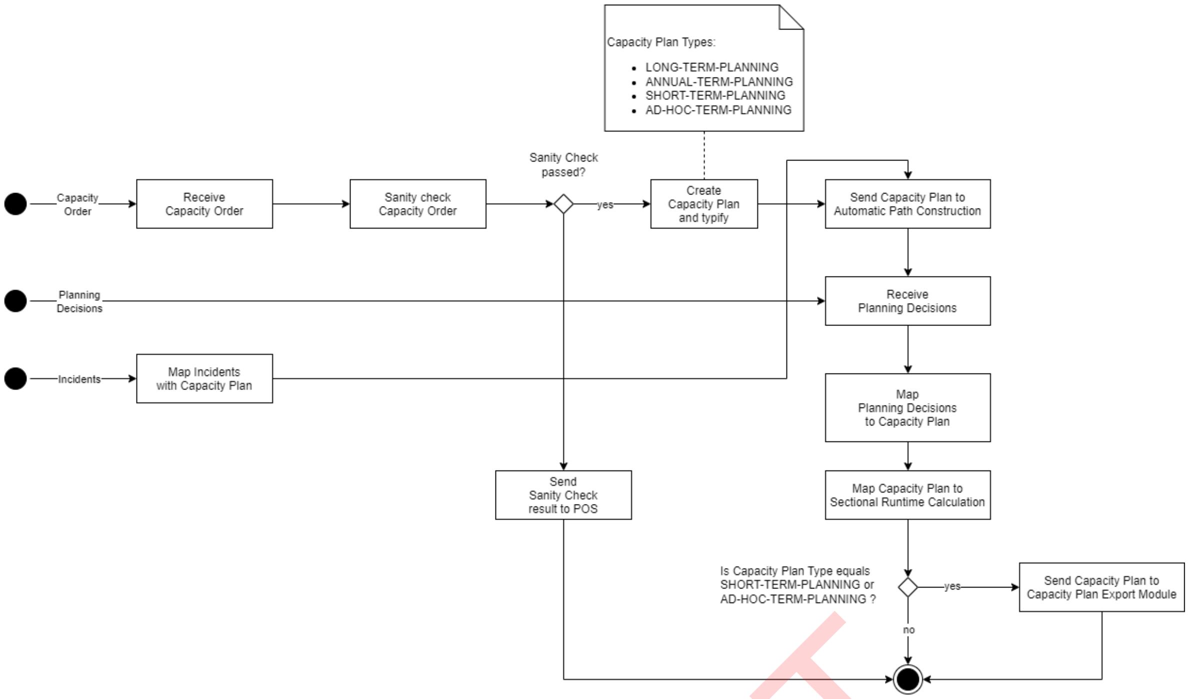
6.1.1.3.4 Workflow

The workflow consists of the following tasks:

- **Receive Capacity Orders:** from Path Order System (POS).
- **Execute sanity checks:** verify and protect the system from poor data quality provided from external sources.
- **Manage Capacity Orders and Capacity Plans:** Initiate capacity planning and manage decisions.
- **Share Capacity Plans:** to other components.

[SPT3TMS-15369 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15302 **]**

6.1.1.3.4.1 Sanity Checks

The component is highly dependent on the quality of data and may be impacted by poor data quality from Path Order System (POS). Therefore, the provision of Capacity Orders (via Capacity Order Exchange item) is subject of sanity checks to mitigate the risk of poor data quality.
The following fields are considered by the sanity check:

- **Train identifier:** Check for conventions such as Train identifier and mandatory fields.
- **Feasibility:** is the given Capacity order producible under given circumstances (considering topology and train properties)

Irrespective, if a field may not pass the verification, the sanity check is considered to be failed.
Consequently, respective capacity orders are not considered by CMS - the state will be communicated to Path Order System (POS). **[**SPT3TMS-15368 **]**

6.1.1.3.4.2 Capacity Plan Types

Capacity Plan Types describes the current state of the Capacity Plan with respect to different time frames and corresponding actions. Capacity Plan Types vary from long-term (5 years ahead) to ad-hoc planning actions (1-6 hours ahead).
The following table describes the possible Capacity Plan Types:

Type:	**Time frame:**	**Action(s):**
LONG-TERM-PLANNING	5 years before production	Concretize Capacity Plan
ANNUAL-TERM-PLANNING	1 year before production	Concretize Capacity Plan
SHORT-TERM-PLANNING	24 hr before production	Concretize Capacity Plan
AD-HOC-TERM-PLANNING	1 – 6 hr before production	Export Capacity Plan to TMS

[SPT3TMS-15371 **]**

6.1.1.3.4.3 Workflow triggers

The following table describes the workflow triggers:

Trigger:	**Business-Event:**	**Action:**
Capacity Orders	A new capacity order must be considered.	Manage Capacity Orders.
Planning Decisions	A planning decision must be considered.	Manage Planning Decisions.
Incidents	Incidents must be considered.	Verify the impact of Incidents in regards to all Capacity Plans.

[SPT3TMS-15370 **]**

6.1.1.4 Human Machine Interface (HMI)

TYPE: CLIENT ACTOR: CMS

HMI is a visualization tool for Capacity Planner to interact with Capacity Plans. The functional scope of HMI mainly consists of planning train runs in an efficient way by dispatching the desired train run, resolving predicted conflicts and improve planned train runs if necessary. Human intervention remains an important part of the overall architecture and may overrule any automatic set decision. **[**SPT3TMS-15381 **]**

6.1.1.4.1 View types

HMI consists of the following view types which can be used as a stand-alone view but also in combination:

- **Network plan:** Visualization of the railway network
- **Station plan:** Visualization of the train stations
- **Conflict view:** Visualization of conflicts
- **Traffic View:** Visualization of planned traffic
- **Forecast View:** Visualization of planning forecasts

[SPT3TMS-15389 **]**

6.1.1.4.2 Actors

The following actors are directly involved in the components communication:

Actor:	**Direction:**	**Scope:**	**Action:**
Path mgmt & processing	INBOUND	COMPONENT-EXCHANGE	Provision of Capacity Plans
Manual Path Construction	OUTBOUND	COMPONENT-EXCHANGE	Provision of Planning Decisions.
Manual Path Conflict Detection	OUTBOUND	COMPONENT-EXCHANGE	Provision of Planning Decisions.

* Besides the named actors above, the component will receive Topology information. Please refer to Topology Import Events. **[**SPT3TMS-15452 **]**

6.1.1.4.3 Exchange items

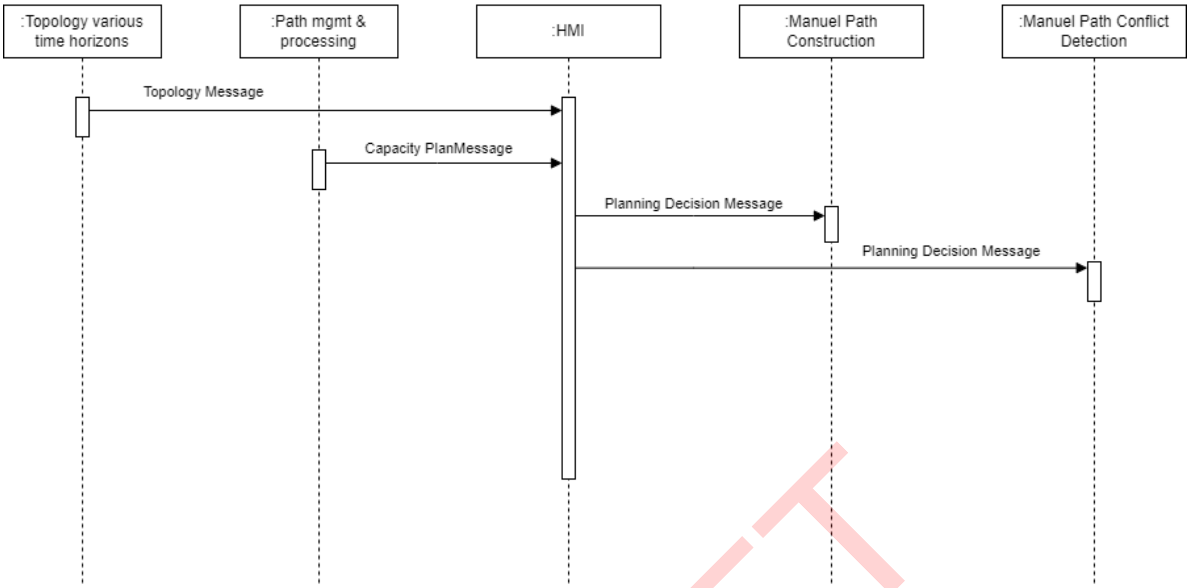
The following Exchange items are involved in the components process:

Exchange item:	**Direction:**
Capacity Plan	INBOUND
Planning Decision	OUTBOUND

[SPT3TMS-15288 **]**

6.1.1.4.4 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15305 **]**

6.1.1.4.5 Workflow

HMI is not based on a workflow. **[**SPT3TMS-15387 **]**

6.1.1.4.5.1 Workflow triggers

HMI is fully functional anytime - there is no trigger. **[**SPT3TMS-15386 **]**

6.1.1.5 Manual Path Construction

TYPE: SERVER ACTOR: CMS ARTIFACT: PLANNING-DECISIONS

Manual Path Construction's main field of responsibility relies in the calculation of paths from point to point for a specific Capacity Plan, consider Topology, Train specifications and limitations to provide various options to HMI. Manual Path construction insists of user interactions by HMI; A Capacity Planner receives various options from point to point and shall then select the best feasible option. Once selected, the component further communicates the Planning Decisions to other components. **[**SPT3TMS-15385 **]**

6.1.1.5.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
HMI	INBOUND	COMPONENT-EXCHANGE	Provision of Planning Decisions.
HMI	OUTBOUND	COMPONENT-EXCHANGE	Provision of Planning Decisions.
Path mgmt & processing	OUTBOUND	COMPONENT-EXCHANGE	Provision of Planning Decisions.

* Besides the named actors above, the component will receive Topology information. Please refer to Topology Import Events. **[**SPT3TMS-15451 **]**

6.1.1.5.2 Exchange items

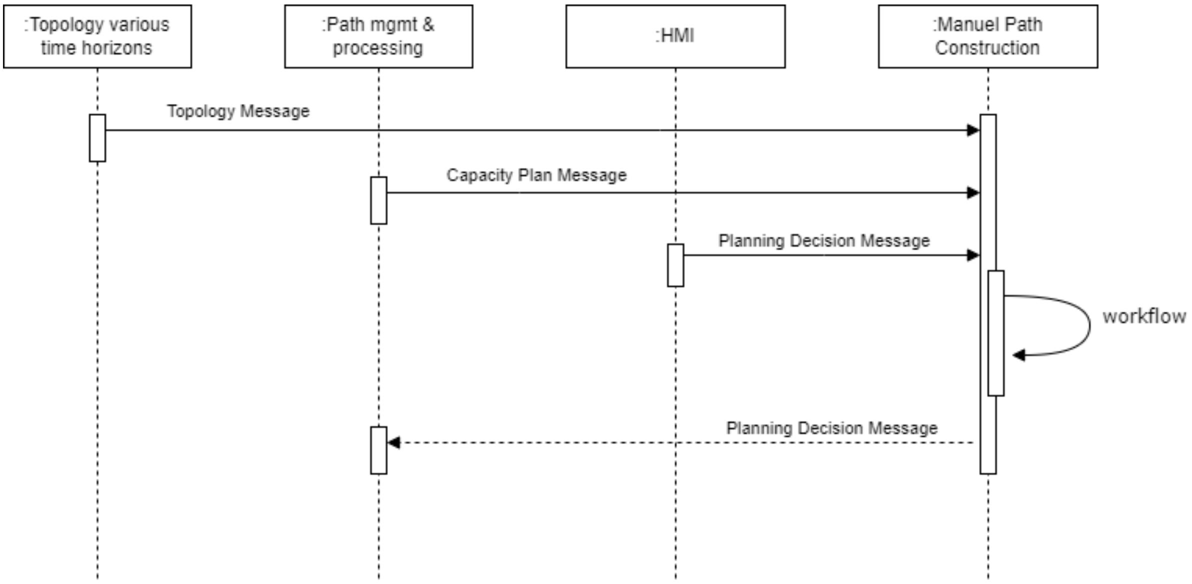
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Capacity Plan	INBOUND
Planning Decision	INBOUND
Planning Decision	OUTBOUND

[SPT3TMS-15287 **]**

6.1.1.5.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15304 **]**

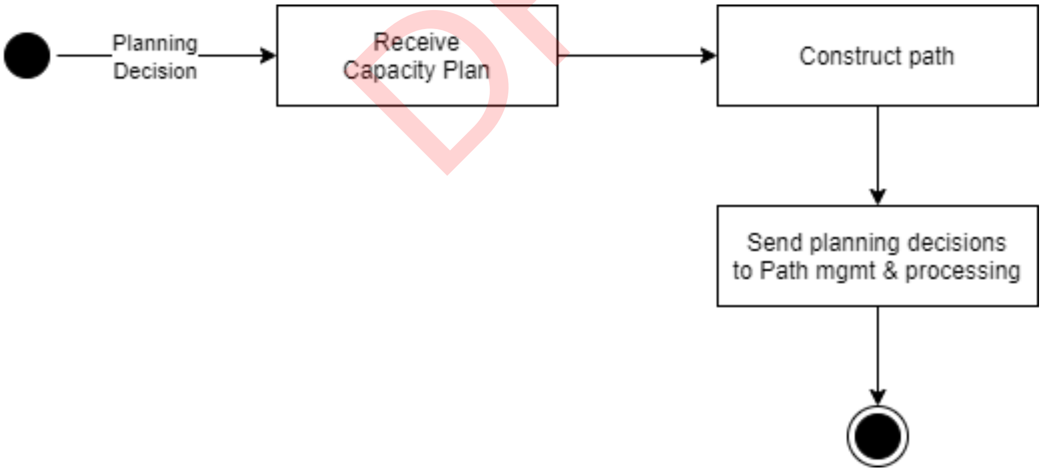
6.1.1.5.4 Workflow

The workflow consists of the following tasks:

- **Receive Capacity Plans:** from Path mgmt. & processing.
- **Construct path:** considering topology and train properties.
- **Share Planning Decisions:** to Path mgmt. & processing.

[SPT3TMS-15394 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15310 **]**

6.1.1.5.4.1 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
HMI interaction	A HMI interaction requested the construction of a train run from point to point	Construct of path from point to point

[SPT3TMS-15447 **]**

6.1.1.6 Manual Path Conflict Detection

TYPE: SERVER ACTOR: CMS ARTIFACT: PLANNING-DECISIONS

Manual Path Conflict Detection's main field of responsibility relies in the identification of conflicts from manual calculated path constructions. The component is responsible to indicate potential conflicts between all Capacity Plans to HMI. A dispatcher resolves the indicated conflicts then issuing a Planning Decision - the component will then further communicate the Planning Decisions to other components. **[**SPT3TMS-15393 **]**

6.1.1.6.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
HMI	INBOUND	COMPONENT-EXCHANGE	Provision of Planning Decisions.
HMI	OUTBOUND	COMPONENT-EXCHANGE	Provision of Planning Decisions.
Path mgmt & processing	OUTBOUND	COMPONENT-EXCHANGE	Provision of Planning Decisions.

* Besides the named actors above, the component will receive Topology information. Please refer to Topology Import Events. **[**SPT3TMS-15448 **]**

6.1.1.6.2 Exchange items

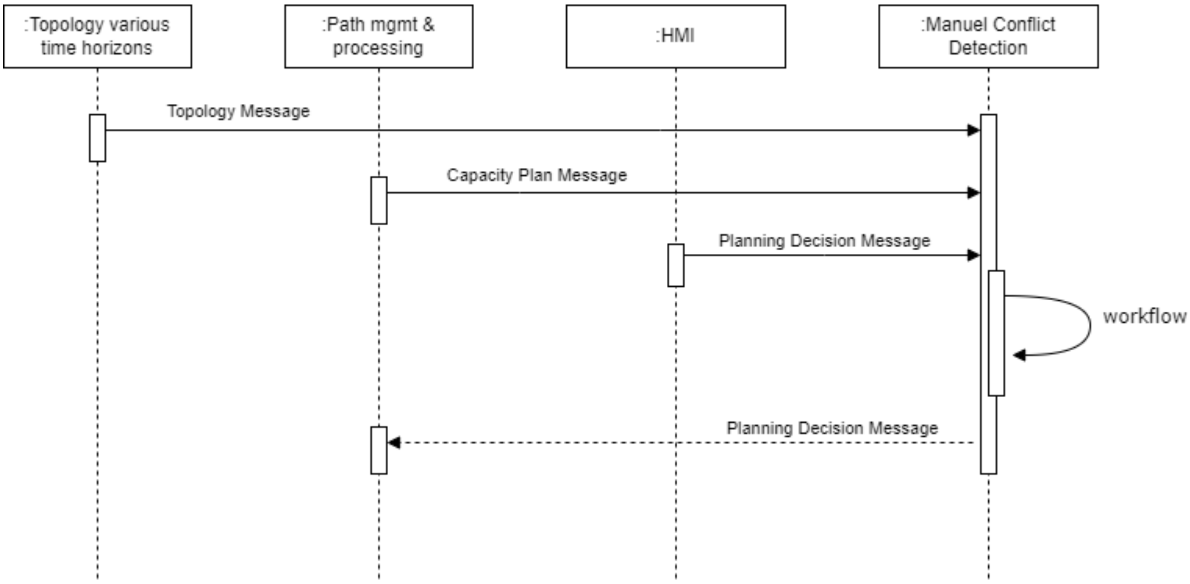
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Capacity Plan	INBOUND
Planning Decision	INBOUND
Planning Decision	OUTBOUND

[SPT3TMS-15286 **]**

6.1.1.6.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15314 **]**

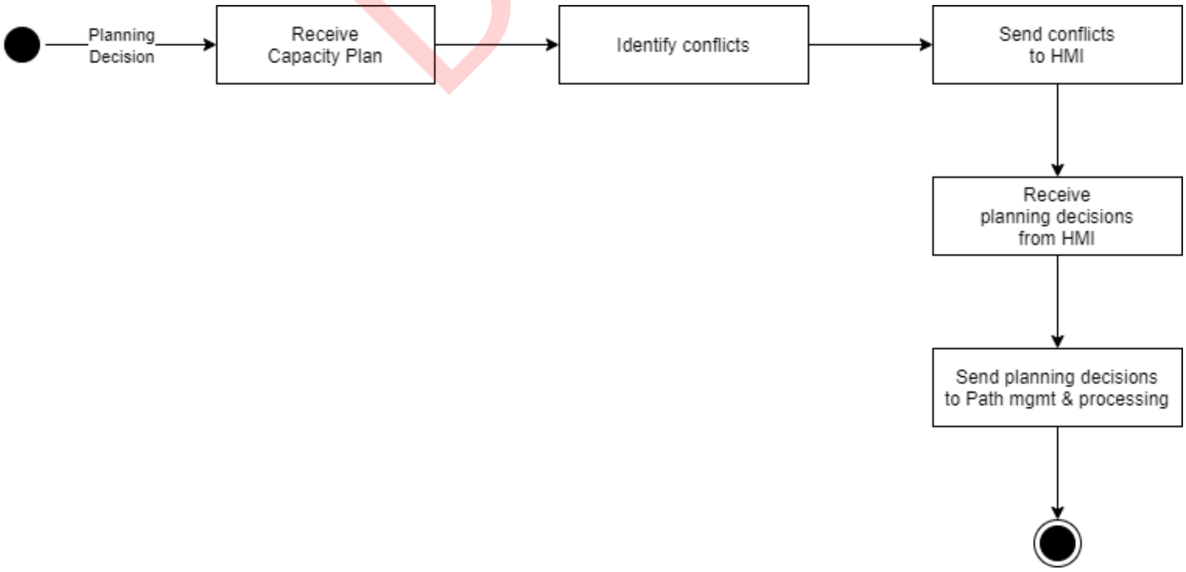
6.1.1.6.4 Workflow

The workflow consists of the following tasks:

- **Receive Capacity Plans:** from Path mgmt. & processing.
- **Identify conflicts:** compare Capacity Plans with the same location and timing.
- **Receive Planning Decisions:** from HMI to resolve the conflict.
- **Share Planning Decisions:** to Path mgmt. & processing.

[SPT3TMS-15392 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15313 **]**

6.1.1.6.4.1 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
HMI interaction	A HMI interaction requested the construction of a train run from point to point which must be checked for potential conflicts.	Verify path construction and indicate conflicts if applicable.

[SPT3TMS-15391 **]**

6.1.1.7 Automatic Path Construction

TYPE: SERVER ACTOR: CMS ARTIFACT: PLANNING-DECISIONS

Automatic Path construction is based on algorithms, to consider the most effective pathway by involving train specifications and limitations. The functionality relies on topology data. The component will make sure to receive those data via the Topology various horizons component. Then, the automatic path construction can be processed. **[**SPT3TMS-15396 **]**

6.1.1.7.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
Path mgmt & processing	INBOUND	COMPONENT-EXCHANGE	Provision of Capacity Plans
Path mgmt & processing	OUTBOUND	COMPONENT-EXCHANGE	Provision of Planning Decisions.

* Besides the named actors above, the component will receive Topology information. Please refer to Topology Import Events. **[**SPT3TMS-15454 **]**

6.1.1.7.2 Exchange items

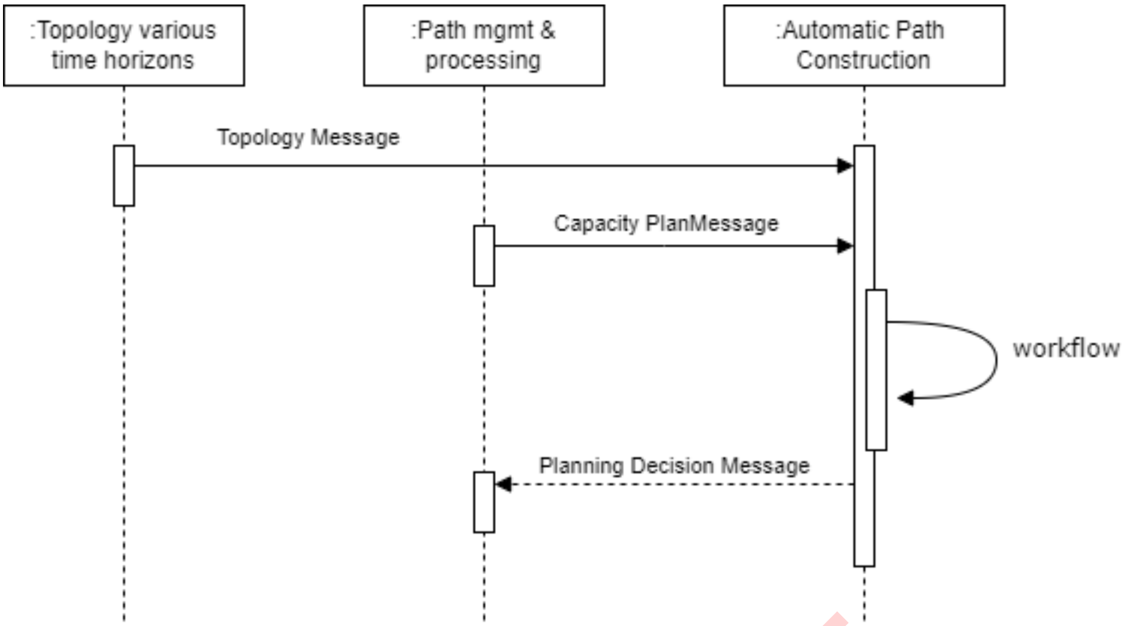
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Capacity Plan	INBOUND
Planning Decision	OUTBOUND

[SPT3TMS-15285 **]**

6.1.1.7.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15312 **]**

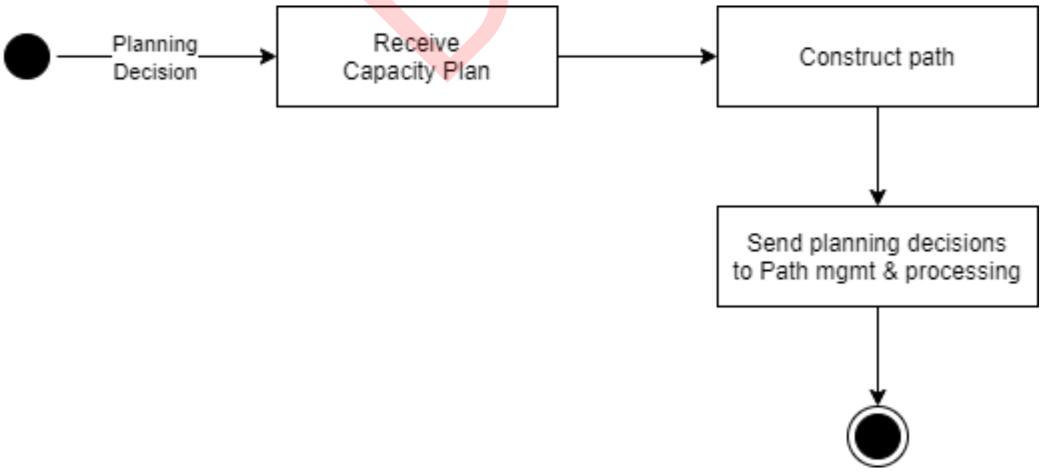
6.1.1.7.4 Workflow

The workflow consists of the following tasks:

- **Receive Capacity Plans:** from Path mgmt. & processing.
- **Construct path:** considering topology and train properties.
- **Share Planning Decisions:** to Path mgmt. & processing.

[SPT3TMS-15395 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15311 **]**

6.1.1.7.4.1 Workflow triggers

The following table describes the workflow triggers:

Trigger:	**Business-Event:**	**Action:**
Additions or modifications to Capacity Plans	An addition or modification to Capacity Plans must be considered in Automatic Path Construction.	Initiate the construction of paths.

[SPT3TMS-15397 **]**

6.1.1.8 Capacity Plan Export

TYPE: SERVER ACTOR: CMS ARTIFACT: CAPACITY-PLAN

The Capacity Plan Export component is the latest step to collect the Capacity Plan. Therefore, planned operations within 1h upcoming operation time are aggregated and finalized. Once the capacity plan is created the component will transfer the Capacity Plan to TMS. With the transfer to TMS the Capacity Plan will not be considered by the CMS anymore. **[**SPT3TMS-15401 **]**

6.1.1.8.1 Actors

The following actors are directly involved in the components communication:

Actor:	**Direction:**	**Scope:**	**Action:**
Path mgmt & processing	INBOUND	COMPONENT-EXCHANGE	Provision of Capacity Plans
Capacity Plan and decision processing	OUTBOUND	COMPONENT-EXCHANGE	Provision of Capacity Plans

[SPT3TMS-15458 **]**

6.1.1.8.2 Exchange items

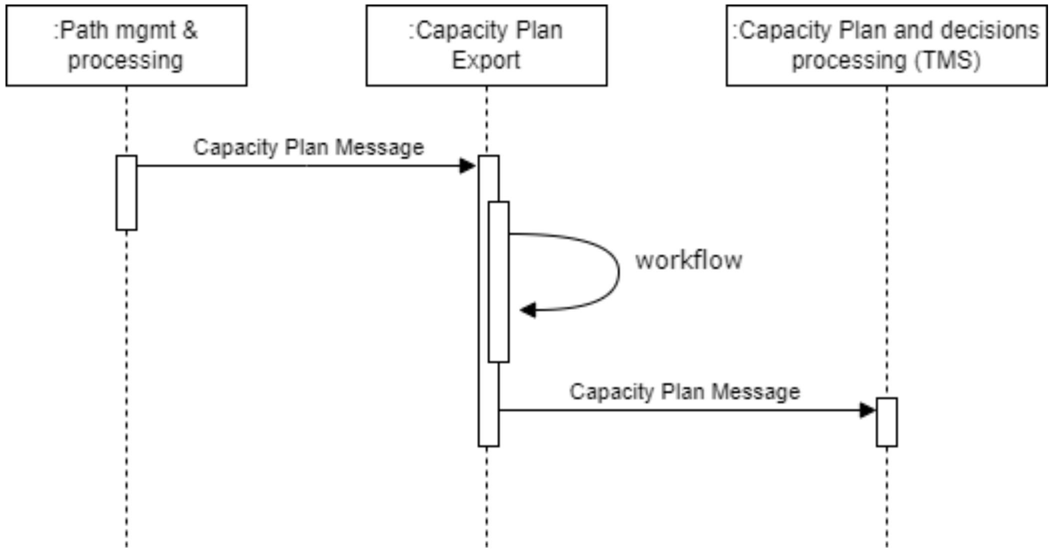
The following Exchange items are involved in the components process:

Exchange item:	**Direction:**
Capacity Plan	INBOUND
Capacity Plan	OUTBOUND

[SPT3TMS-15292 **]**

6.1.1.8.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15318 **]**

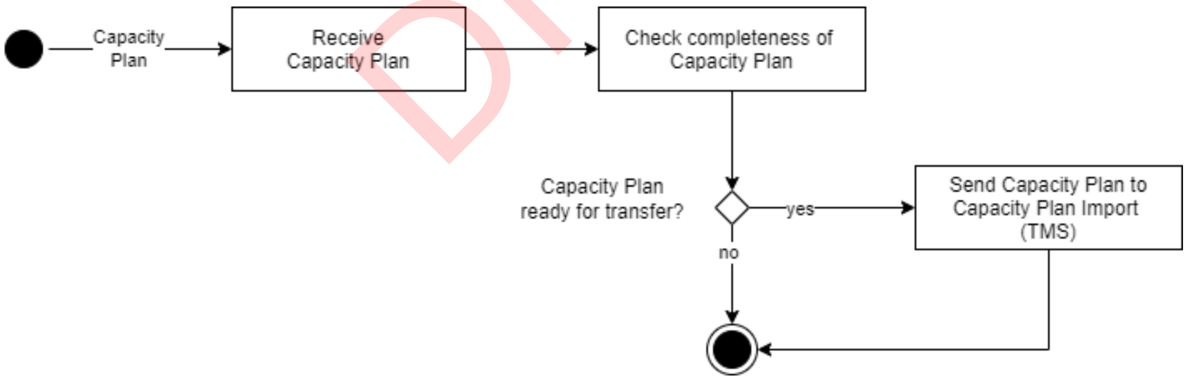
6.1.1.8.4 Workflow

The workflow consists of the following tasks:

- **Receive Capacity Plans:** from Path mgmt. & processing.
- **Check completeness:** verify the completeness of Capacity Plans.
- **Transfer Capacity Plans:** to Capacity Plan and decisions processing (TMS).

[SPT3TMS-15400 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15317 **]**

6.1.1.8.4.1 Workflow triggers

The following table describes the workflow triggers:

Trigger:	**Business-Event:**	**Action:**
Capacity Plan is ready to be transfered to TMS.	The Capacity Plan is near production and must be transfered to TMS therefore.	Transfer Capacity Plan to TMS.

[SPT3TMS-15399 **]**

6.1.2 Capacity Production (TMS)

TMS is the domain responsible for all production activities including producing an operational plan, based upon the RT- operational state and RT-operational events of a railway infrastructure provider.
[SPT3TMS-15398 **]**

The following table illustrates the TMS components:

Component:	**Type:**	**Artifact:**
Topology Master Data Validation & Import	SERVER	TOPOLOGY
Daily Topology	SERVER	DAILY-TOPOLOGY
Capacity plan and decision processing	SERVER	OPERATIONAL-PLAN
Deviation Detection	SERVER	DEVIATION
Forecasting	SERVER	FORECAST
Real-Time Conflict Detection	SERVER	CONFLICT
HMI	CLIENT	-
Manual Conflict Solution	SERVER	CONFLICT-SOLUTION
Automatic Conflict Solution	SERVER	CONFLICT-SOLUTION
Automatic Connection Management	SERVER	CONFLICT-SOLUTION

[SPT3TMS-15405 **]**

6.1.2.1 Topology Master Data Validation & Import

TYPE: SERVER ACTOR: TMS ARTIFACT: TOPOLOGY

This component imports topology data into the TMS context by considering specific TMS validation rules. With the import of Topology data, the TMS domain decouples the Topology management.
[SPT3TMS-15404 **]**

6.1.2.1.1 Actors

The following actors are directly involved in the components communication:

Actor:	**Direction:**	**Scope:**	**Action:**
Topology Master Data (XFN)	INBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.
Daily Topology	OUTBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.

[SPT3TMS-15457 **]**

6.1.2.1.2 Exchange items

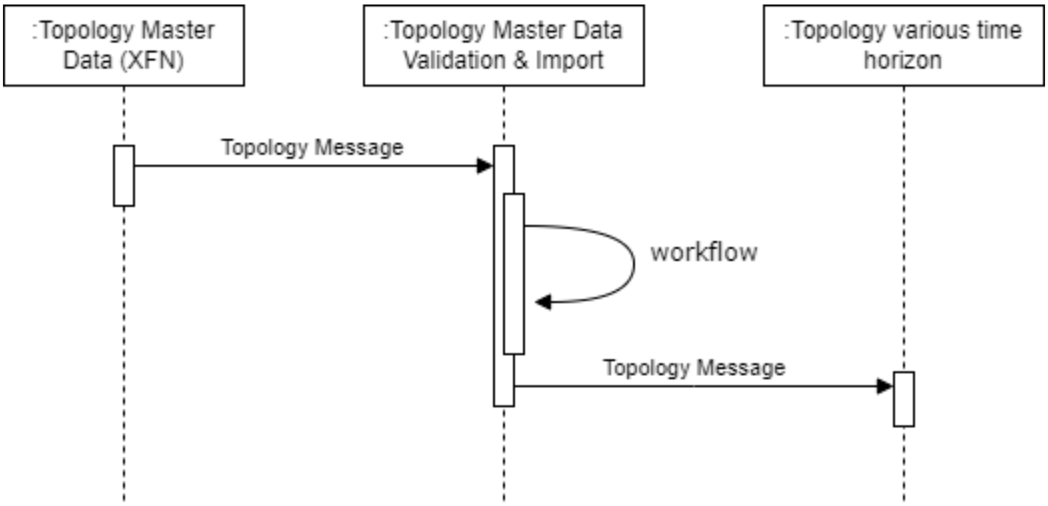
The following Exchange items are involved in the components process:

Exchange item:	**Direction:**
Topology	INBOUND
Topology	OUTBOUND

[SPT3TMS-15291 **]**

6.1.2.1.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15316]

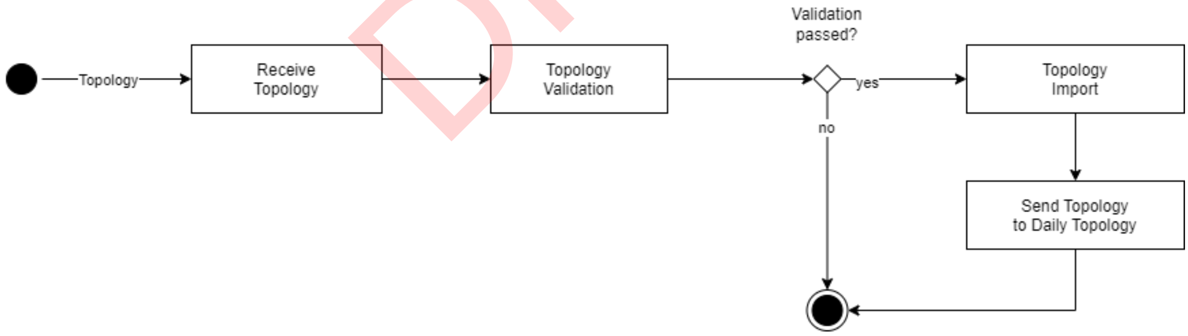
6.1.2.1.4 Workflow

The workflow consists of the following tasks:

- **Receive Topology:** from Topology Master Data (XFN).
- **Execute Validation:** to verify and protect the system from poor data quality provided from external sources.
- **Import Topology:** into the TMS domain to avoid dependencies to other domains.
- **Share Topology:** to other components.

[SPT3TMS-15403]

The following activity diagram illustrates the workflow:



[SPT3TMS-15315]

6.1.2.1.4.1 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
Addition or modification to Topology	An addition or modification to Topology must be considered in TMS.	Initiate Topology Validation & Import

[SPT3TMS-15465]

6.1.2.2 Daily Topology

TYPE: SERVER ACTOR: TMS ARTIFACT: DAILY-TOPOLOGY

This component imports topology data into the TMS context. Topology data are crucial for precise calculations. The component will calculate the topology for today and provides the result of calculations to the Capacity Plan & decisions processing component. **[**SPT3TMS-11041 **]**

6.1.2.2.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
Topology various time horizons	INBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.
Capacity Plan and decisions processing	OUTBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.

[SPT3TMS-15456 **]**

6.1.2.2.2 Exchange items

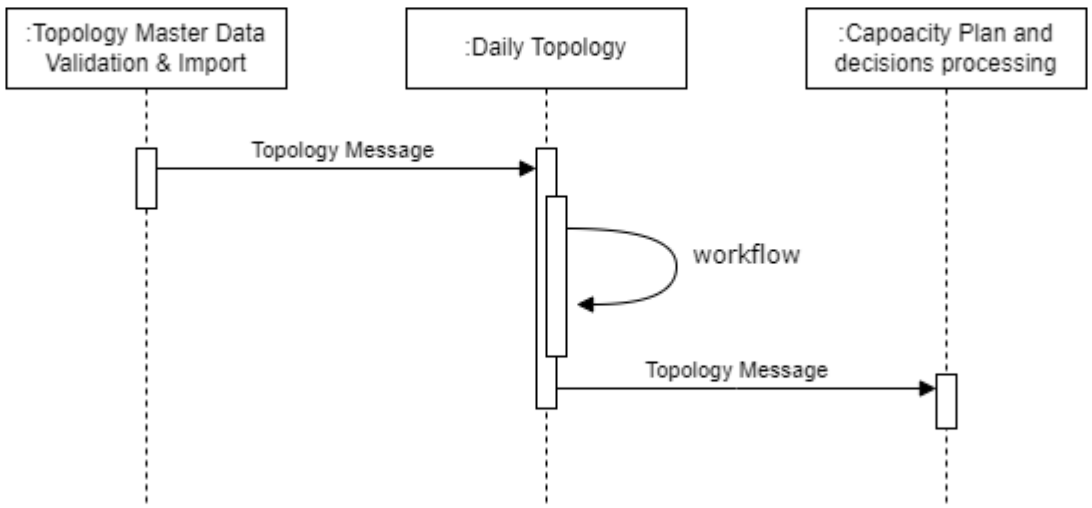
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Topology	INBOUND
Topology	OUTBOUND

[SPT3TMS-15290 **]**

6.1.2.2.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15320 **]**

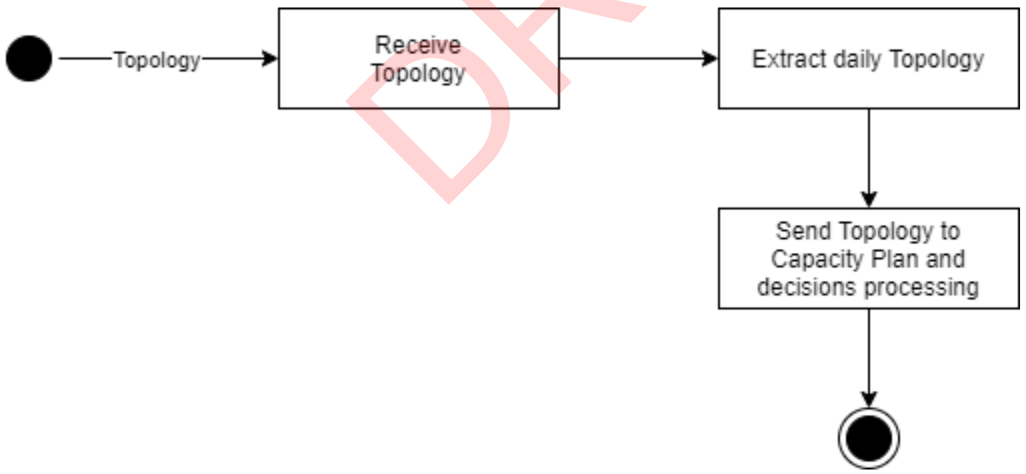
6.1.2.2.4 Workflow

The workflow consists of the following tasks:

- **Receive Topology:** from Topology Master Data Validation & Import.
- **Extract Topology:** for the current day only.
- **Share Topology:** to other components.

[SPT3TMS-15402 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15319 **]**

6.1.2.2.4.1 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
Addition or modification to Topology	An addition or modification to Topology must be considered in current daily Topology.	Initiate Topology extraction
Nightly		

Trigger:	Business-Event:	Action:
	Daily additions and modifications must be considered in the Topology.	Initiate Topology extraction

[SPT3TMS-15406 **]**

6.1.2.3 Capacity plan and decision processing

TYPE: SERVER ACTOR: TMS ARTIFACT: OPERATIONAL-PLAN

This component is the central component of the TMS architecture. It transforms the agreed Capacity Plan to an Operational Plan after confirming the feasibility (by sanity check) of the operational plan. Daily topology data are used to concertize the actual operation. **[**SPT3TMS-10552 **]**

6.1.2.3.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
Control Command and Signaling (CCS)	INBOUND	FUNCTIONAL-EXCHANGE	Provision of Operational State via Operational State Exchange item.
Incident Impact Management	INBOUND	FUNCTIONAL-EXCHANGE	Provision of Operational Decisions via Operational Decision Exchange item.
Capacity Plan Export (CMS)	INBOUND	COMPONENT-EXCHANGE	Provision of Capacity Plans via Capacity Plan Exchange item.
Daily Topology	INBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.
Sectional Runtime Calculation	INBOUND	COMPONENT-EXCHANGE	Provision of sectional runtime calculations via Calculated Sectional Runtime Exchange item.
Automatic Connection Management	INBOUND	COMPONENT-EXCHANGE	Provision of Operational Decisions via Operational Decision Exchange item.
Manual Conflict Solution	INBOUND	COMPONENT-EXCHANGE	Provision of Operational Decisions via Operational Decision Exchange item.
Deviation Detection	OUTBOUND	COMPONENT-EXCHANGE	Follow-up process via Capacity Plan Exchange item.
Path mgmt & processing	OUTBOUND	COMPONENT-EXCHANGE	Notification of infra changes which have an impact on CMS planning or suggestions to be evaluated to be applied

[SPT3TMS-15455 **]**

6.1.2.3.2 Exchange items

The following Exchange items are involved in the components process:

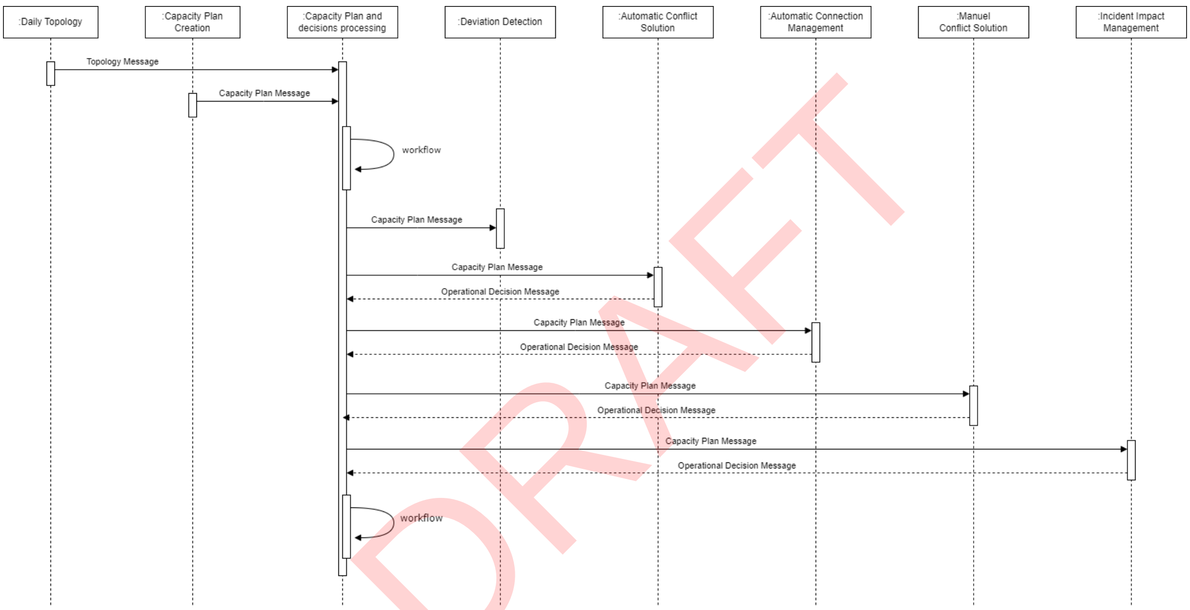
Exchange item:	Direction:
Operational State	INBOUND
Operational Decision	INBOUND

Exchange item:	Direction:
Capacity Plan	INBOUND
Topology	INBOUND
Calculated Sectional Runtime	INBOUND
Capacity Plan	OUTBOUND

[SPT3TMS-15467 **]**

6.1.2.3.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15321 **]**

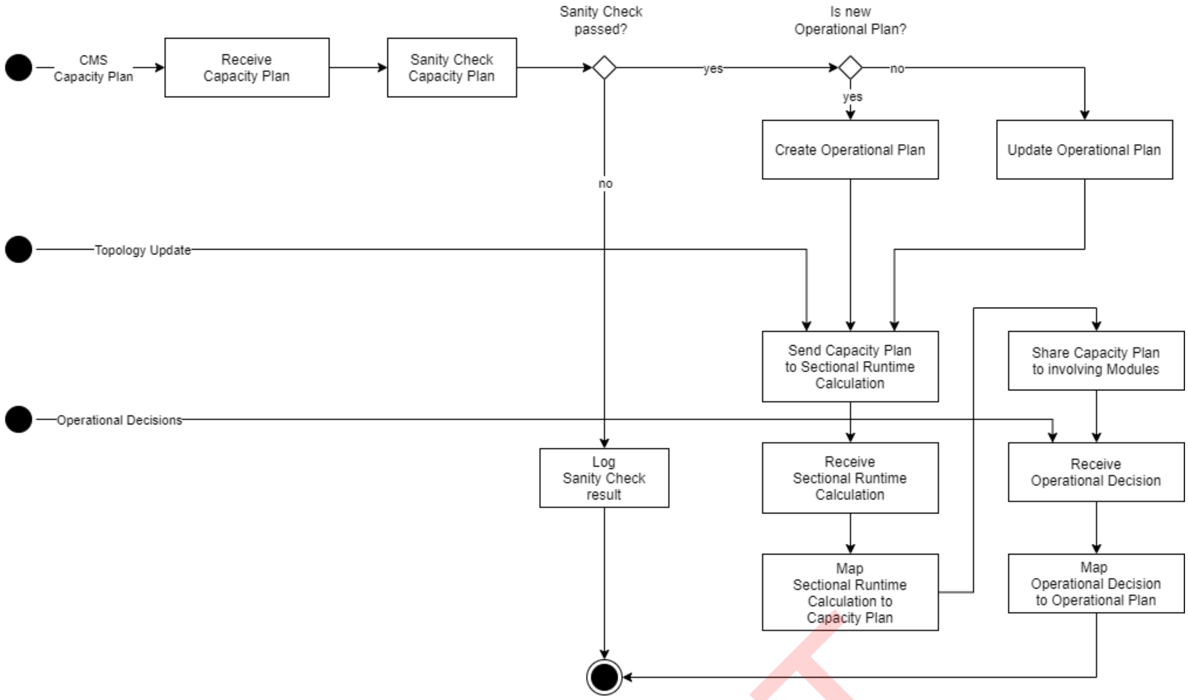
6.1.2.3.4 Workflow

The workflow consists of the following tasks:

- **Receive Capacity Plans:** from CMS.
- **Execute sanity checks:** verify and protect the system from poor data quality provided.
- **Manage Operational Plans:** Converting Capacity Plans to Operational Plans and managing synchronization.
- **Share Capacity Plans:** to component which require Capacity Plans.

[SPT3TMS-15348 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15325 **]**

6.1.2.3.4.1 Sanity Checks

The component is highly dependent on the quality of data and may be impacted by poor data quality from CMS. Therefore, the provision of Capacity Plans (via Capacity Plan Exchange item) is subject of sanity checks to mitigate the risk of poor data quality.
The following fields are considered by the sanity check:

- **Train identifier:** Check for conventions such as Train identifier and mandatory fields.
- **Locations:** Check for presence of topology data such as train paths, etc.

Irrespective, if a field may not pass the verification, the sanity check is considered to be failed. Consequently, respective trains are not considered by the TMS - the rejection will be logged.
[SPT3TMS-15347 **]**

6.1.2.3.4.2 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
CMS Capacity Plans	A Capacity Plan is ready for production. In certain situation, a new Capacity Plan which referes to an existing Operational Plan, is send to TMS. In this case, the existing Operational Plan will only be updated	Initiate the creation / update of an Operational Plan based on a Capacity Plan.
Topology Updates	The Topology was updated and all existing Operational Plans are outdated.	Check all Operational Plans for out-dated Topology data and update if applicable.
Operational Decisions	An Operational Decisions was published.	Updated respective Operational Plan to include Operational Decisions.

[SPT3TMS-15352 **]**

6.1.2.4 Deviation Detection

TYPE: SERVER ACTOR: TMS ARTIFACT: DEVIATION

The Deviation Detection component monitors deviations for all capacity productions which is the base for potential conflict detection by the Forecasting component. The component is capable to identify deviations by comparing all Capacity Plans with Operational States. The component is part of the overall workflow for conflict detection and conflict solution. To provide reliable near real-time information the component assesses deviations every second. The root cause for potential deviations are recorded and shared with the Forecasting component. [SPT3TMS-10554]

6.1.2.4.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
Control Command and Signaling (CCS)	INBOUND	FUNCTIONAL-EXCHANGE	Provision of Operational State via Operational State Exchange item + commence of workflow.
Capacity plan and decisions processing	INBOUND	COMPONENT-EXCHANGE	Provision of Capacity Plan via Capacity Plan Exchange item.
Forecasting	OUTBOUND	COMPONENT-EXCHANGE	Follow-up process via Deviation Exchange item.

* Besides the named actors above, the component will receive Topology information. Please refer to Topology Import Events. [SPT3TMS-15462]

6.1.2.4.2 Exchange items

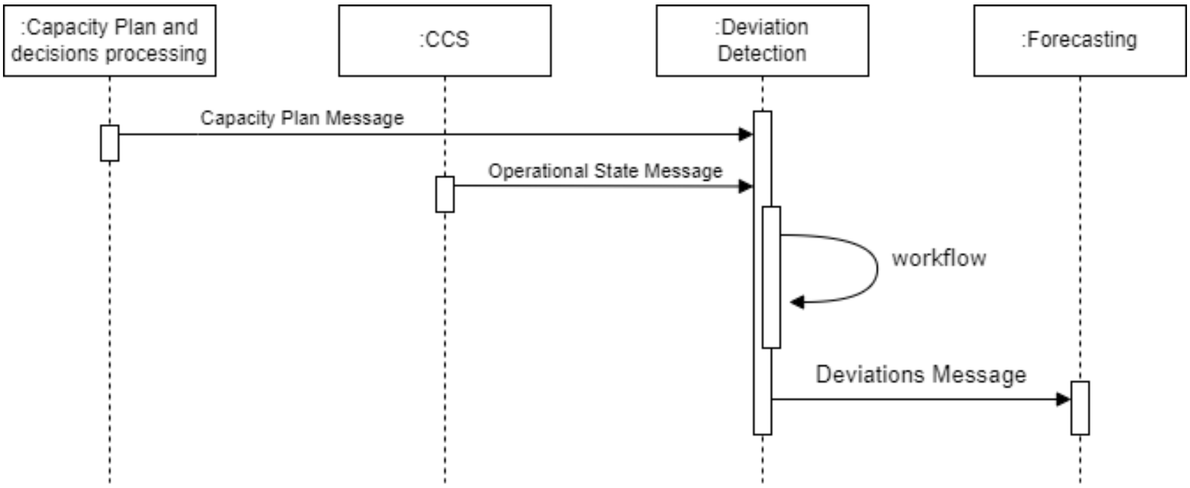
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Capacity Plan	INBOUND
Operational State	INBOUND
Deviation	OUTBOUND

[SPT3TMS-15466]

6.1.2.4.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-10469 **]**

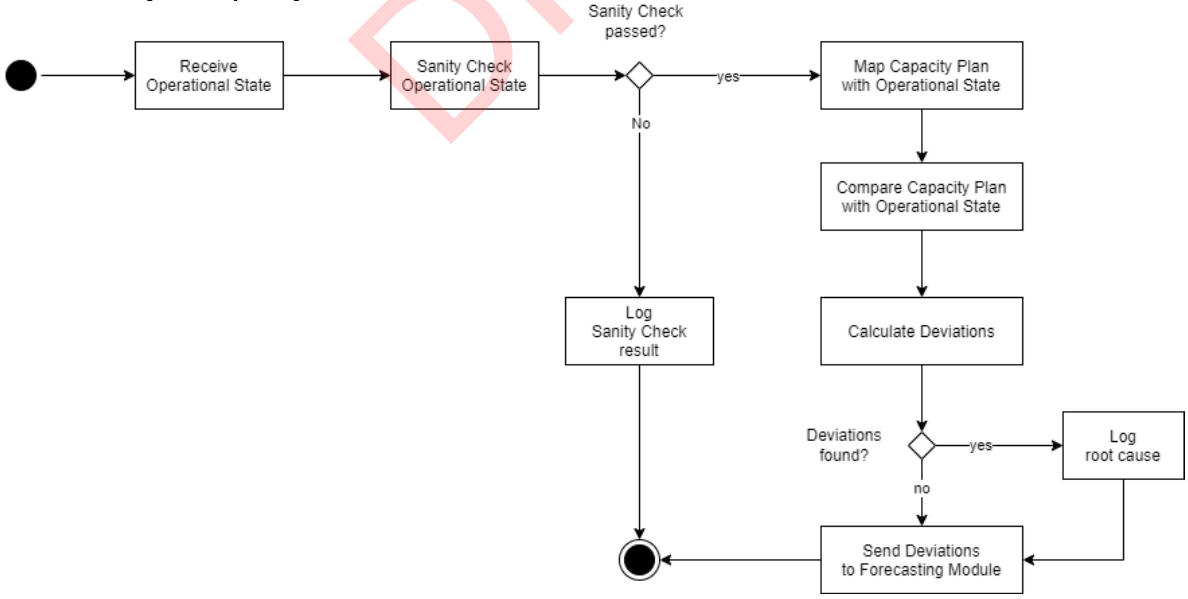
6.1.2.4.4 Workflow

The workflow consists of the following tasks:

- **Receive Operational State:** from Control Command and Signaling (CCS).
- **Execute sanity Checks:** verify and protect the system from poor data quality provided from external actors.
- **Execute Deviation & Matching Algorithm:** Match, compare and calculate deviations between Capacity Plans and Operational States.
- **Log:** Document decisions being made by the workflow.
- **Send deviations:** to Forecasting component.

[SPT3TMS-15351 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15324 **]**

6.1.2.4.4.1 Sanity Checks

The component is highly dependent on the quality of data of operational state and may be impacted by poor data quality from external actors. Therefore, the provision of Operational States (via Operational State Exchange item) is subject of sanity checks to mitigate the risk of poor data quality. **[**SPT3TMS-15349 **]**

The following fields are considered by the sanity check:

- **Train identifier:** Check for conventions such as Train identifier and mandatory fields.
- **Locations:** Check for presence of topology data such as train paths, etc.

If any field fails the verification, the integrity check is considered to have failed. Consequently, respective trains are not considered by the deviation detection and further not in the whole process - the rejection will be logged. **[**SPT3TMS-15350 **]**

6.1.2.4.4.2 Deviation and Matching Algorithm

The component is based on a deviation and matching algorithm which is responsible for the following:

- **Match:** Capacity Plans with Operational States based on the Train Unit Data Attribute and Track Allocation data.
- **Compare:** Capacity Plans with Operational States based on location (Train path) and time.
- **Calculate:** Deviation between Capacity Plans and Operational States based on location and time.

[SPT3TMS-15356 **]**

6.1.2.4.4.3 Logging Events

The following Log Events are available:

Event:	Description:
SANITY_CHECK_FAILED	One or more fields failed the sanity check. The Operational State is not further considered by the deviation detection and in the further process. It is suggested to improve the data quality to pass the sanity check.
DEVIATION_DETECTED	The deviation will be logged including the root cause.

[SPT3TMS-15355 **]**

6.1.2.4.4.4 Message Events

The following table describes the involved Message Events:

Event:	Direction:	Description:
TOPOLOGY-IMPORTED	INBOUND	Exchange / sync of Topology.
CAPACITY-PLAN-RECEIVED	INBOUND	Exchange / sync of Capacity Plans.
DEVIATION-DETECTION-COMPLETED	OUTBOUND	Indicates the completion of deviation detection.

[SPT3TMS-15354 **]**

6.1.2.4.4.5 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
Operational States	An Operational State from CCS was send to the component.	Initiate the calculation of deviations.

[SPT3TMS-15353 **]**

6.1.2.5 Forecasting

TYPE: SERVER ACTOR: TMS ARTIFACT: FORECAST

The main purpose of this component is to forecast all Capacity Plans to measure the level of conflicts correctly on time and content and forecasts the impact on the planned timetable and on the cross border's geographic regions/tracks. [SPT3TMS-10553]

6.1.2.5.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
Capacity plan and decisions processing	INBOUND	COMPONENT-EXCHANGE	Provision of Capacity Plan via Capacity Plan Exchange item.
Deviation Detection	INBOUND	COMPONENT-EXCHANGE	Provision of Deviations via Deviation Exchange item.
Real-Time Conflict Detection	OUTBOUND	COMPONENT-EXCHANGE	Follow-up process via Forecast Exchange item.

* Besides the named actors above, the component will receive Topology information. Please refer to Topology Import Events. [SPT3TMS-15461]

6.1.2.5.2 Exchange items

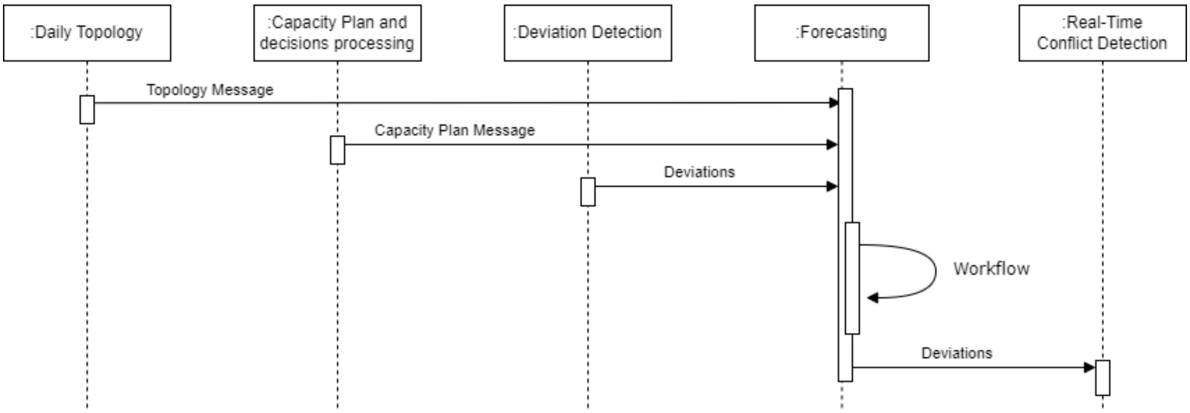
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Capacity Plan	INBOUND
Deviation	INBOUND
Forecast	OUTBOUND

[SPT3TMS-15289]

6.1.2.5.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-10911]

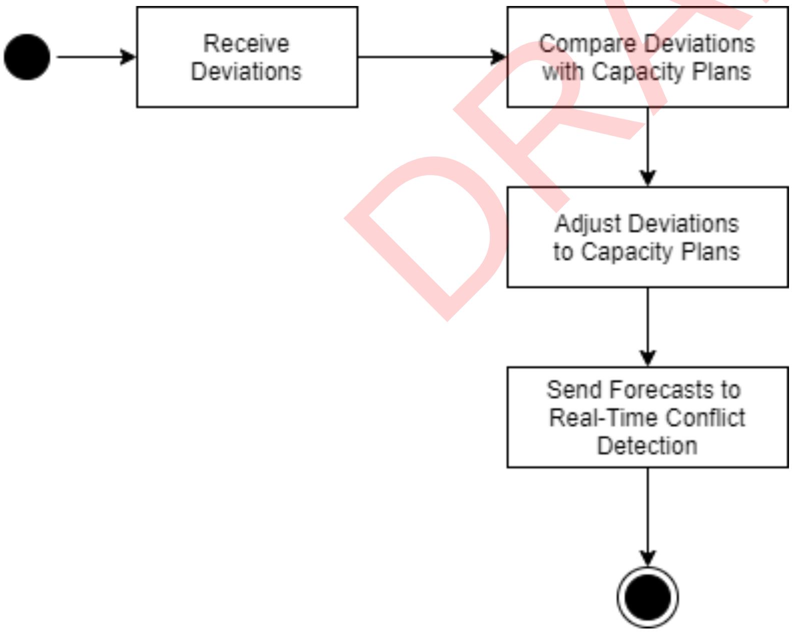
6.1.2.5.4 Workflow

The workflow consists of the following tasks:

- **Receive deviations:** from Deviation Detection component.
- **Execute forecasting algorithm:** Match, compare, Adjust, Detect and Resolve Operational Plans.
- **Send forecasts:** to Real-Time Conflict Detection component.

[SPT3TMS-15360]

The following activity diagram illustrates the workflow:



[SPT3TMS-15323]

6.1.2.5.4.1 Forecasting Algorithm

The component is based on a forecasting algorithm which is responsible for the following:

- **Match:** Capacity Plans with deviations (Only consider Deviations for the corresponding Capacity Plan based on Train nr.)

- **Compare:** Capacity Plans with deviations to measure further actions of adjustments. The Algorithm will also consider other Capacity Plans to resolve minor Conflicts by e.g. changing the order in the timetable.
- **Adjust:** Further timetable according to deviations.

[SPT3TMS-15359 **]**

6.1.2.5.4.2 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
Receive Deviations	A deviation from the Deviation Detection component was send to the component.	Initiate the calculation of forecasts.

[SPT3TMS-15358 **]**

6.1.2.6 Real-Time Conflict Detection

TYPE: SERVER ACTOR: TMS ARTIFACT: CONFLICT

The main purpose of this component is to detect conflicts on all Capacity Plans based on forecasts to measure the impact on the planned timetable and on the cross border's geographic regions/tracks. **[**SPT3TMS-10982 **]**

6.1.2.6.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
Capacity plan and decisions processing	INBOUND	COMPONENT-EXCHANGE	Provision of Capacity Plan via Capacity Plan Exchange item.
Forecasting	INBOUND	COMPONENT-EXCHANGE	Provision of Forecasts via Forecast Exchange item.
Automatic Conflict Solution	OUTBOUND	COMPONENT-EXCHANGE	Follow-up process via Conflict Exchange item.
HMI	OUTBOUND	COMPONENT-EXCHANGE	Visualization of via Conflict Exchange item.

* Besides the named actors above, the component will receive Topology information. Please refer to Topology Import Events. **[**SPT3TMS-15460 **]**

6.1.2.6.2 Exchange items

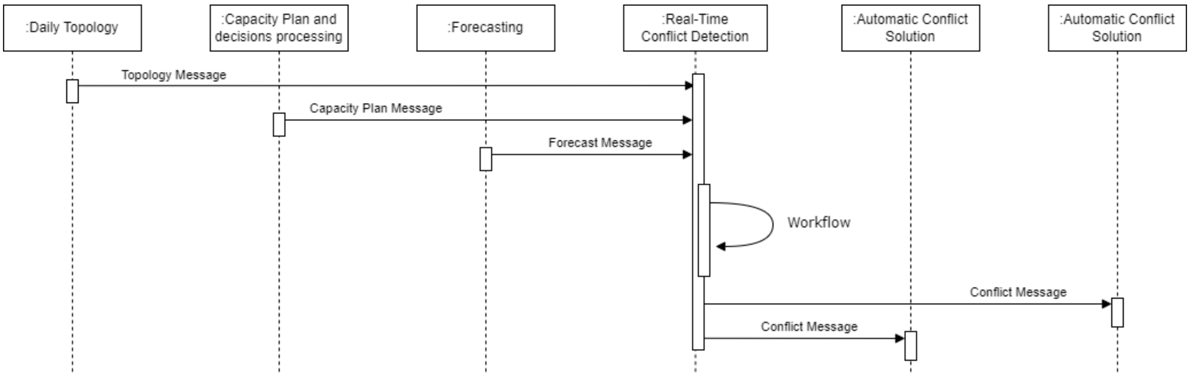
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Capacity Plan	INBOUND
Forecast	INBOUND
Conflict	OUTBOUND

[SPT3TMS-15294 **]**

6.1.2.6.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15322 **]**

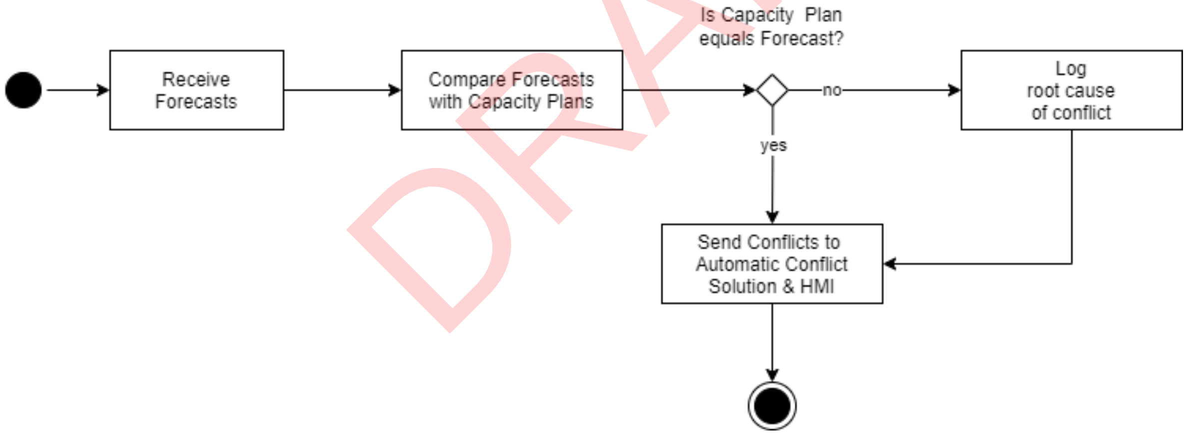
6.1.2.6.4 Workflow

The workflow consists of the following tasks:

- **Receive forecasts:** from Forecasting component.
- **Execute conflict detection algorithm:** compare Capacity Plans with forecasts
- **Send conflicts:** to Automatic Conflict Solution component and HMI.

[SPT3TMS-15364 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15329 **]**

6.1.2.6.4.1 Conflict Detection Algorithm

The component is based on a conflict detection algorithm which is responsible for the following:

- **Match:** Capacity Plans with forecasts (Only consider the Forecast for the respective Capacity Plan based on Train number)
- **Compare:** A conflict is by definition the difference between a Capacity Plan and a Forecast. The Algorithm will compare a Capacity Plan with the Forecast to identify differences.

[SPT3TMS-15363 **]**

6.1.2.6.4.2 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
Receive forecasts	A forecast from the Forecasting component was send to the component.	Initiate the calculation of conflict detection.

[SPT3TMS-15362 **]**

6.1.2.7 Human Machine Interface (HMI)

TYPE: CLIENT ACTOR: TMS

HMI is a visualizations tool for dispatcher to interact with near real-time train operations. The functional scope of HMI mainly consists of dispatching trains, resolving conflicts and improve train operations if necessary. Human intervention remains an important part of the overall architecture and may overrule any automatic set decision. **[**SPT3TMS-15361 **]**

6.1.2.7.1 View types

HMI consists of the following view types which can be used as a stand-alone view but also in combination:

- **Network plan:** Visualization of the railway network
- **Station plan:** Visualization of the train stations
- **Conflict view:** Visualization of conflicts
- **Traffic View:** Visualization of real-time traffic
- **Forecast View:** Visualization of forecasts

[SPT3TMS-15367 **]**

6.1.2.7.2 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
Real-Time Conflict Detection	INBOUND	COMPONENT-EXCHANGE	Provision of Conflict Exchange item.
Manual Conflict Solution	OUTBOUN D	COMPONENT-EXCHANGE	Provision of Operational Decisions Exchange item.

* Besides the named actors above, the component will receive Topology information. Please refer to Topology Import Events. **[**SPT3TMS-15453 **]**

6.1.2.7.3 Exchange items

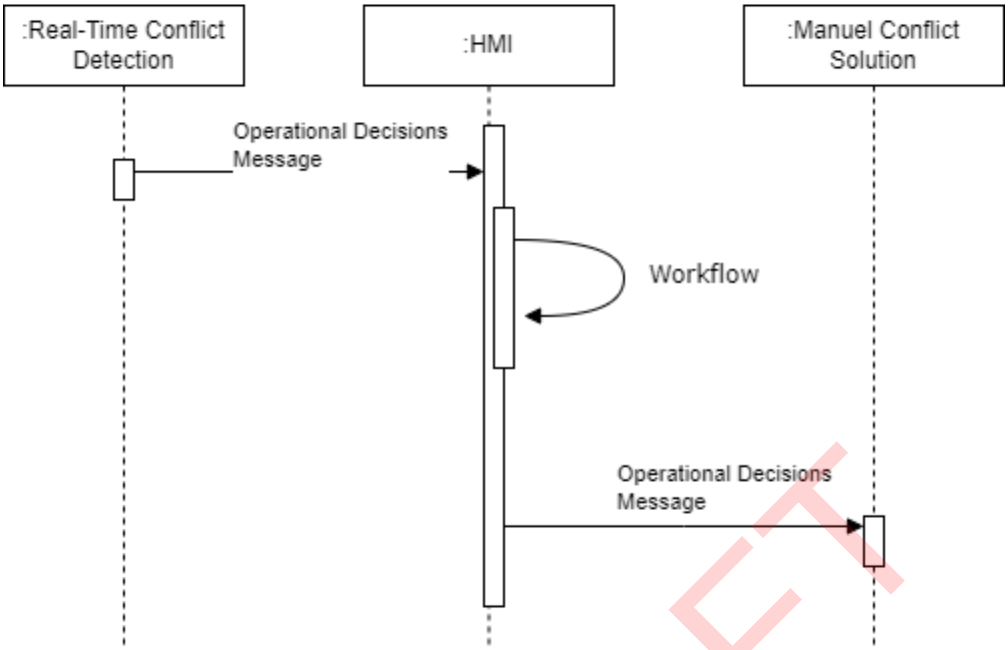
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Capacity Plan	INBOUND
Conflict	INBOUND
Operational Decision	OUTBOUND

[SPT3TMS-15293 **]**

6.1.2.7.4 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15328 **]**

6.1.2.7.5 Workflow

HMI is not based on a workflow. **[**SPT3TMS-15366 **]**

6.1.2.7.5.1 Workflow triggers

HMI is fully functional anytime - there is no trigger. **[**SPT3TMS-15365 **]**

6.1.2.8 Manual Conflict Solution

TYPE: SERVER ACTOR: TMS ARTIFACT: CONFLICT-SOLUTION

Manual Conflict Solution insists of user interactions by HMI; A dispatcher receives various options to resolve a conflict and shall then select the best feasible option. Once selected, the component further communicates the Operational Decisions to other components.
[SPT3TMS-15430 **]**

6.1.2.8.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
HMI	INBOUND	COMPONENT-EXCHANGE	Provision of Operational Decisions Exchange item.
Automatic Connection Management	OUTBOUND	COMPONENT-EXCHANGE	Follow-up process via Operational Decisions Exchange item.

* Besides the named actors above, the component will receive Topology information. Please refer to Topology Import Events. **[**SPT3TMS-15429 **]**

6.1.2.8.2 Exchange items

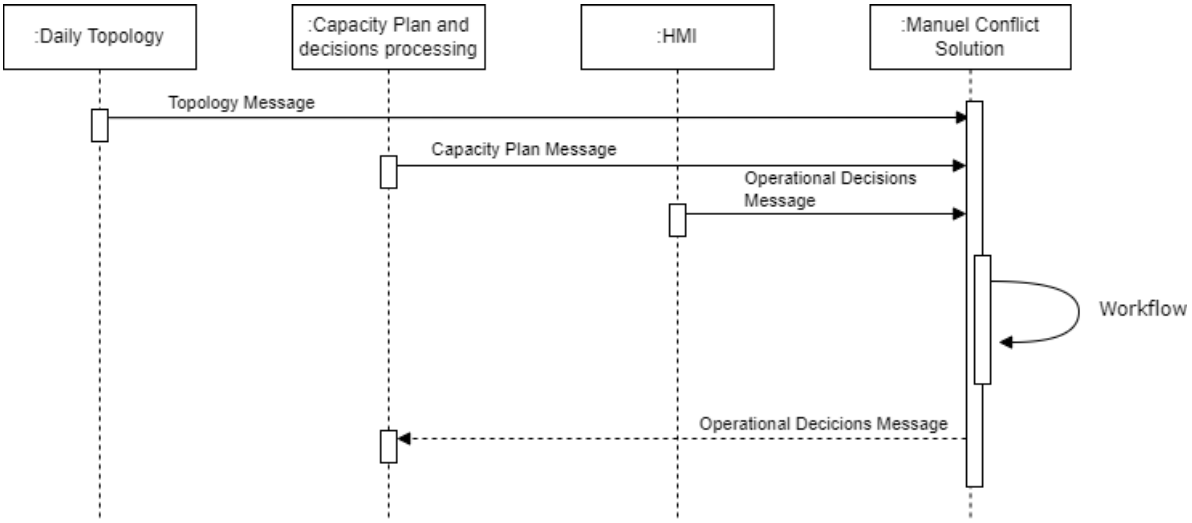
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Capacity Plan	INBOUND
Operational Decisions	INBOUND
Operational Decisions	OUTBOUND

[SPT3TMS-15295 **]**

6.1.2.8.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-10932 **]**

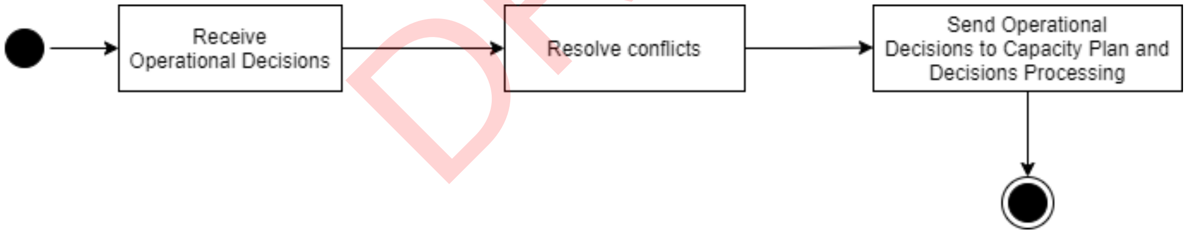
6.1.2.8.4 Workflow

The workflow consists of the following tasks:

- **Receive Operational Decisions:** from HMI.
- **Resolve Conflicts:** Based on conflict solution strategies, producing Operational Decisons.
- **Send operational decisions:** to Capacity Plan and Decisions Processing component.

[SPT3TMS-15428 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15327 **]**

6.1.2.8.4.1 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
HMI interaction	A dispatcher requests a conflict solution which must be provided to the dispatcher.	Resolve conflicts

[SPT3TMS-15427 **]**

6.1.2.9 Automatic Conflict Solution

TYPE: SERVER ACTOR: TMS ARTIFACT: CONFLICT-SOLUTION

Solves conflicts based on algorithms. May use optimization functions. **[**SPT3TMS-10551 **]**

6.1.2.9.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
Capacity plan and decisions processing	INBOUND	COMPONENT-EXCHANGE	Provision of Capacity Plan via Capacity Plan Exchange item.
Real-Time Conflict Detection	INBOUND	COMPONENT-EXCHANGE	Provision of Conflicts via Conflict Exchange item.
Automatic Connection Management	OUTBOUND	COMPONENT-EXCHANGE	Follow-up process via Operational Decisions Exchange item.

* Besides the named actors above, the component will receive Topology information. Please refer to Topology Import Events. **[**SPT3TMS-15434 **]**

6.1.2.9.2 Exchange items

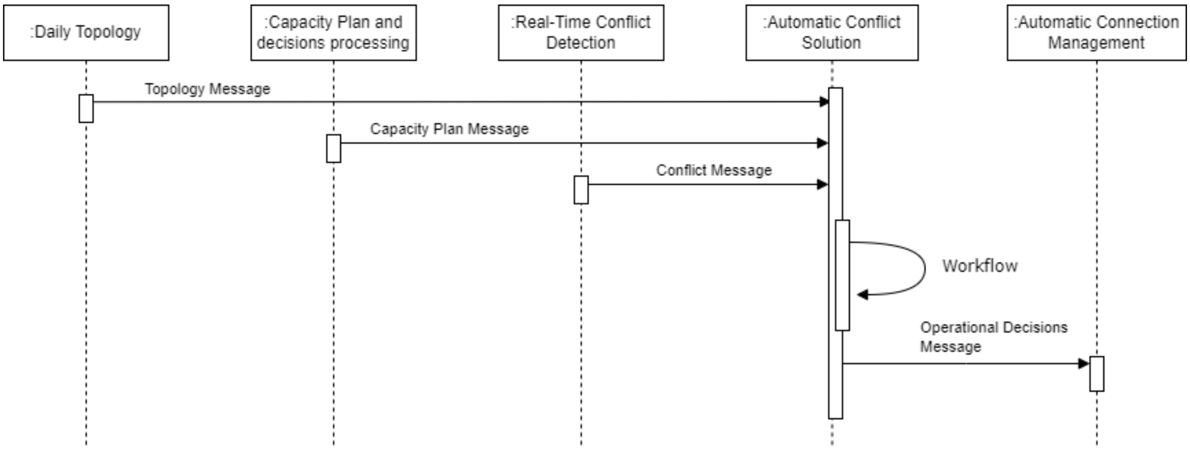
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Capacity Plan	INBOUND
Conflict	INBOUND
Operational Decisions	OUTBOUND

[SPT3TMS-15299 **]**

6.1.2.9.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-10474 **]**

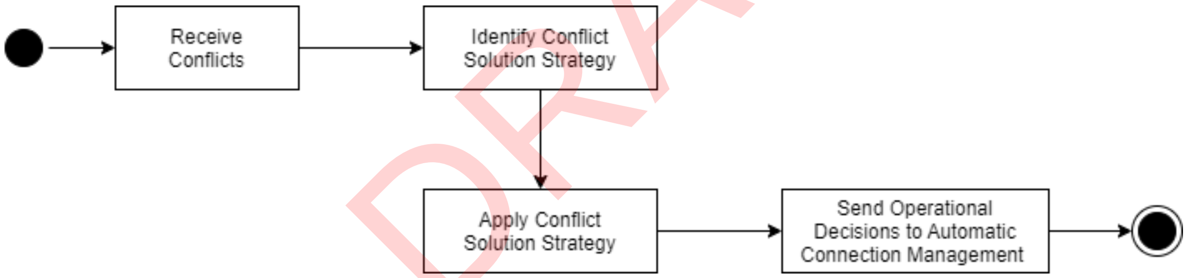
6.1.2.9.4 Workflow

The workflow consists of the following tasks:

- **Receive conflicts:** from Real-Time Conflict Detection component.
- **Execute conflict solution algorithm:** apply conflict solution strategies.
- **Send operational decisions:** to Automatic Connection Management component.

[SPT3TMS-15433 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15326 **]**

6.1.2.9.4.1 Conflict Solution Algorithm

The component is based on a conflict solution algorithm which is responsible for the following:

- **Evaluate:** Given by the nature of the conflict possible Solution Strategies are evaluated. Refer to Conflict Solution Strategies for more details.
- **Resolve:** Apply Solution Strategies to Capacity Plans and resolve conflict(s).

[SPT3TMS-15432 **]**

6.1.2.9.4.2 Conflict Solution Strategies

The following strategies are automatically applied by the Conflict Solution Algorithm based on the nature of the conflict and defined priorities:

- **Delay:** The train run will be delayed at a specific position
- **Cancellation:** The train run will be cancelled.

- **Change of platform:** The platform of the train run will be changed.
- **Change of sequence:** The sequence of the train run will be changed.
- **Change of Route:** The route of the train run will be changed.
- **Change of Track:** The track of the train run will be changed.

Some Strategies can also be applied in combination to resolve the conflict. **[**SPT3TMS-15431 **]**

6.1.2.9.4.3 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
Receive conflicts	A conflict from the Real-Time Conflict Detection component was send to the component.	Initiate the calculation of conflict solution.

[SPT3TMS-15436 **]**

6.1.2.10 Automatic Connection Management

TYPE: SERVER ACTOR: TMS ARTIFACT: OPERATIONAL-DECISION

The Automatic Connection Management component manages passenger traffic connections based on algorithms considering connecting times between train platforms and off-peak times. To identify connection hubs, the component constantly analysis traffic flows and applies operational decisions to Operational Plans mitigating delays and further interruptions. **[**SPT3TMS-10550 **]**

6.1.2.10.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
Automatic Conflict Solution	INBOUND	COMPONENT-EXCHANGE	Provision of Operational Decisions via Operational Decisions Exchange item.
Capacity plan and decisions processing	OUTBOUND	COMPONENT-EXCHANGE	Follow-up process via Operational Decisions Exchange item.

* Besides the named actors above, the component will receive:

- Topology information. Please refer to Topology Import Events.
- Capacity Plan information. Please refer to Capacity Plan processing Events

[SPT3TMS-15459 **]**

6.1.2.10.2 Exchange items

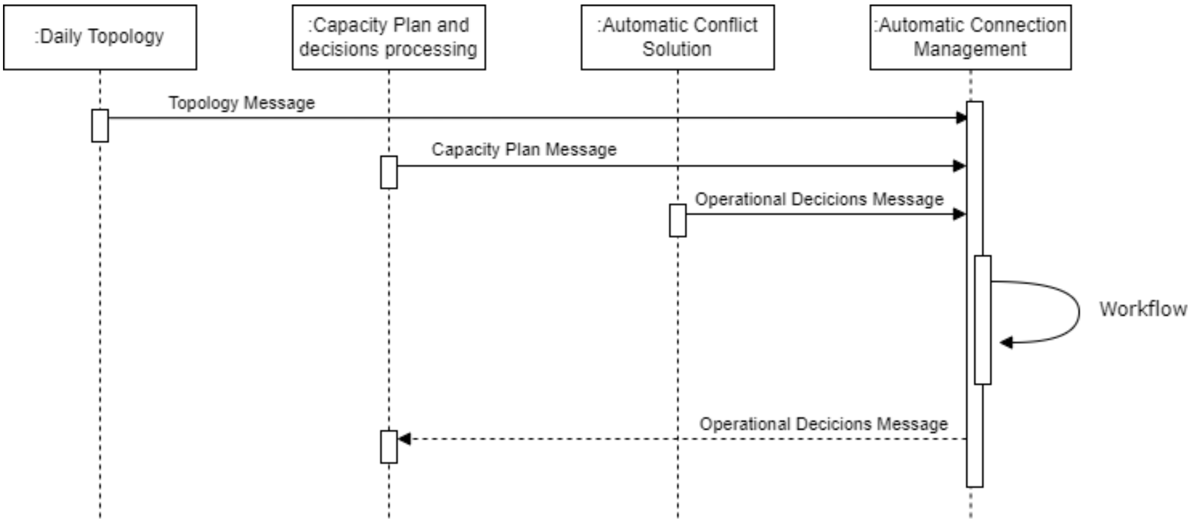
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Capacity Plan	INBOUND
Operational Decisions	INBOUND
Operational Decisions	OUTBOUND

[SPT3TMS-15298 **]**

6.1.2.10.3 interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15330 **]**

6.1.2.10.4 Workflow

The workflow highly depends on ROC and therefore may differentiate between the following:

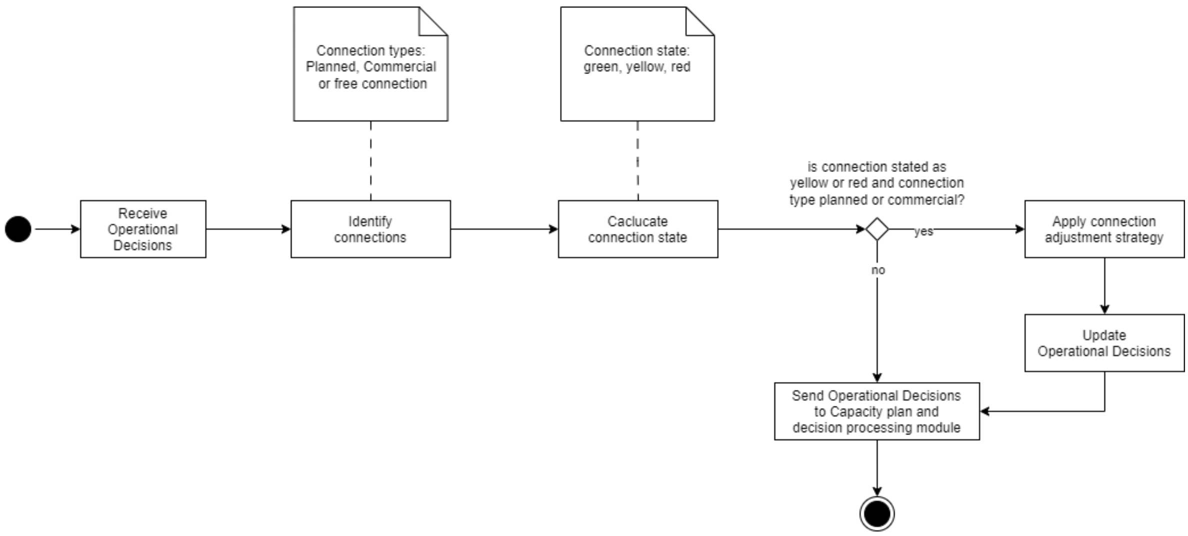
- **Provided connections:** ROC will provide respective connections - there is no further business logic to identify further connections beside the provided connections.
- **Calculated connections:** connections are calculated based on identification and adjustment algorithm. ROC may not interfere in the choice of connections.

The workflow for calculated connections consists of the following tasks:

- **Receive Operational Decisions:** from Automatic Connection Management component.
- **Execute connection identification and adjustment algorithm:** apply connection adjustment strategies.
- **Send operational decisions:** to Capacity Plan and Decision processing component.

[SPT3TMS-15441 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15336 **]**

6.1.2.10.4.1 Connection identification Algorithm

The component is based on a connection identification algorithm which is responsible for the following:

- **Compare:** Capacity Plans with common stations in region and time frames in near real time.
  A connection can be identified by the occurrence of Capacity Plans per region and time.
- **Typify:** Connections based on type (See Connection type) criteria.
  The Algorithm will consider rational connections, numbers of passenger transfers, etc.

[SPT3TMS-15440 **]**

6.1.2.10.4.2 Connection adjustment Algorithm

The component is based on a connection adjustment algorithm which is responsible for the following:

- **Rate:** Rate the state and relevance of the connections based on feasibility.
  The state and relevance is important to choose the correct adjustment strategies.
  (See Connection relevance and Connection state)
- **Adjust:** Capacity Plans based on adjustment strategies.

[SPT3TMS-15439 **]**

6.1.2.10.5 Connection type

The component distinguishes between the following connection type:

Type:	Description:	Strategy:
IMPLICITY	Includes local, high-frequent connection trains (e.g. commuter trains) which may not be relevant for other connections, as a missing connection can be replaced by the next near frequent train run.	Exclude from connection calculation and may only enable if desired.
EXPLICITY	Includes non-frequent connection trains (e.g. IC, ICE, EC) which are relevant for other connections and due the non-frequent characteristics the connection cant be replaced easily.	Include to connection calculation.

[SPT3TMS-15438 **]**

6.1.2.10.6 Connection relevance

The component distinguishes between the following connection relevance:

Type:	Description:	Strategy:
FREE	The connection is optional and there is no real interest of holding the connection.	Ignore connection
PLANNED	The connection is planned as statistically a significant number of travelers using the connection and a missing connection may impact further connections.	Hold connection if possible
COMMERCIAL	There are commercial interests in holding the connection even if the operating train may be late.	Hold connection in any case

[SPT3TMS-15445]

6.1.2.10.7 Connection state

The component rates the potential connection, based on the following states:

Rating	Description:	Intervention required:
GREEN	The connection will be accomplished– there is no interruption. Departure train will wait x minutes. Max connection time to be set.	no
YELLOW	There is an interruption, and the connection may not be able to be accomplish-able if the component will not apply ad hoc operational decisions. Departure time will not extend stop time. The actual Connection time is greater than the minimum connection time.	yes
RED	There is an interruption, and the connection is obviously not accomplish-able, even if the system will apply ad hoc operational decisions – The connection was missed. Arrival time of inbound train is greater than departure time of outbound train	yes

[SPT3TMS-15444]

6.1.2.10.7.1 Workflow triggers

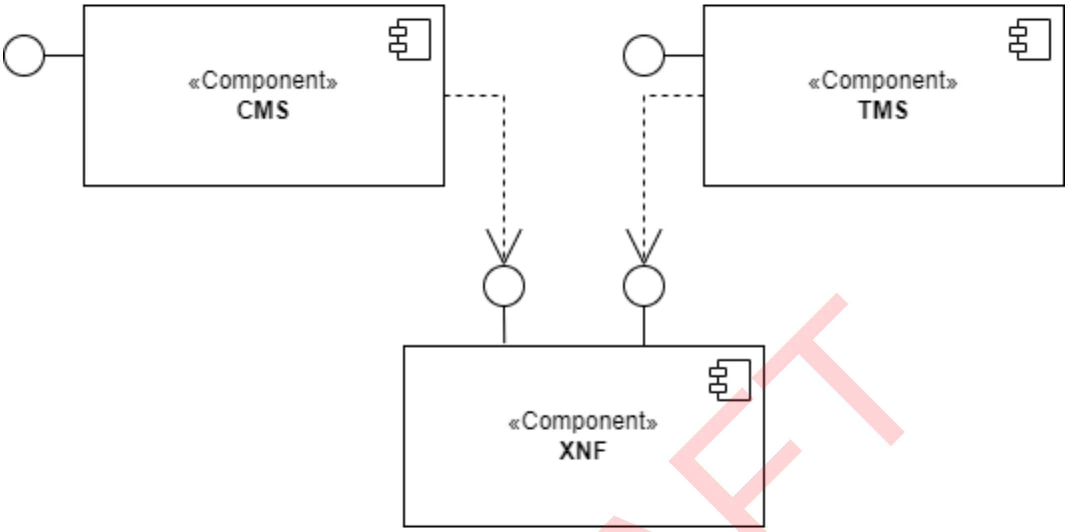
The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
Receive operational decisions	An operational decision from the Automatic Conflict Solution component was send to the component.	Initiate the calculation of connection management.

[SPT3TMS-15443]

6.1.3 Cross Functional (XFN)

The following chapters describe cross cutting functionality between CMS & TMS domain.
Both domains, CMS and TMS process data which have a common functional base. To stick to the defined design principles, the XFN domain will centralize the common functionality. Thereby redundant implementations can be prevented. In fact, the XFN domain is not a sub component. It represents the same behavior as the CMS and TMS domain. **[**SPT3TMS-15442 **]**



[SPT3TMS-15335 **]**

6.1.3.1 Topology Master Data

TYPE: SERVER ACTOR: TMS ARTIFACT: TOPOLOGY

This component builds a single source of truth for topology data for the CMS and TMS domain. The main task of this component is the acquisition and maintainability of topology data across the whole railway network including different versions for short term and long-term planning. **[**SPT3TMS-10562 **]**

6.1.3.1.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
Fixed Asset Management (FAM)	INBOUND	FUNCTIONAL-EXCHANGE	Provision of Topology via Topology Exchange item.
Topology Master Data Validation & Import (CMS)	OUTBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.
Topology Master Data Validation & Import (TMS)	OUTBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.

[SPT3TMS-15446 **]**

6.1.3.1.2 Exchange items

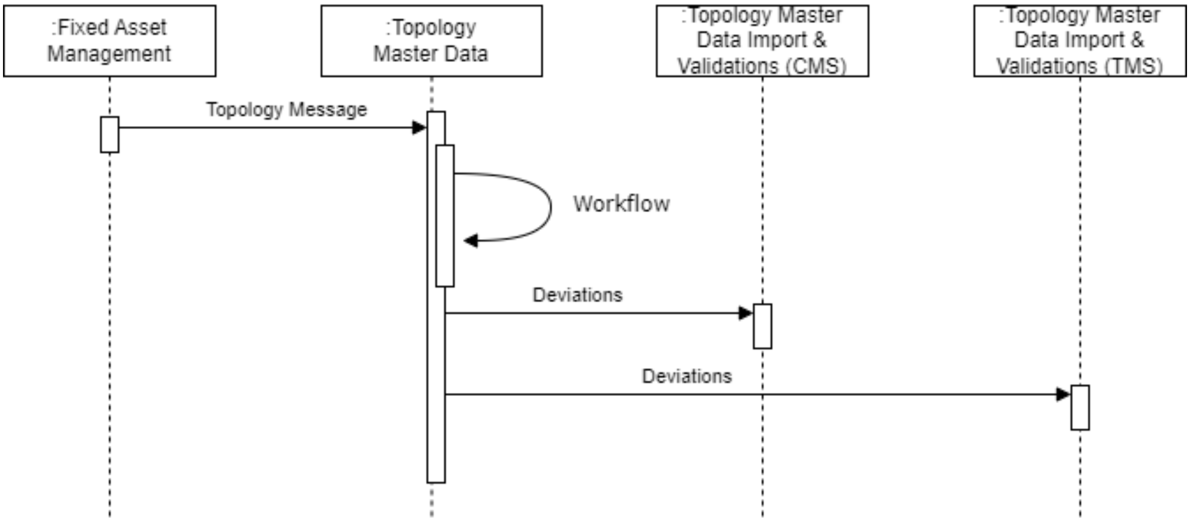
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Topology	INBOUND
Topology	OUTBOUND

[SPT3TMS-15297 **]**

6.1.3.1.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15338 **]**

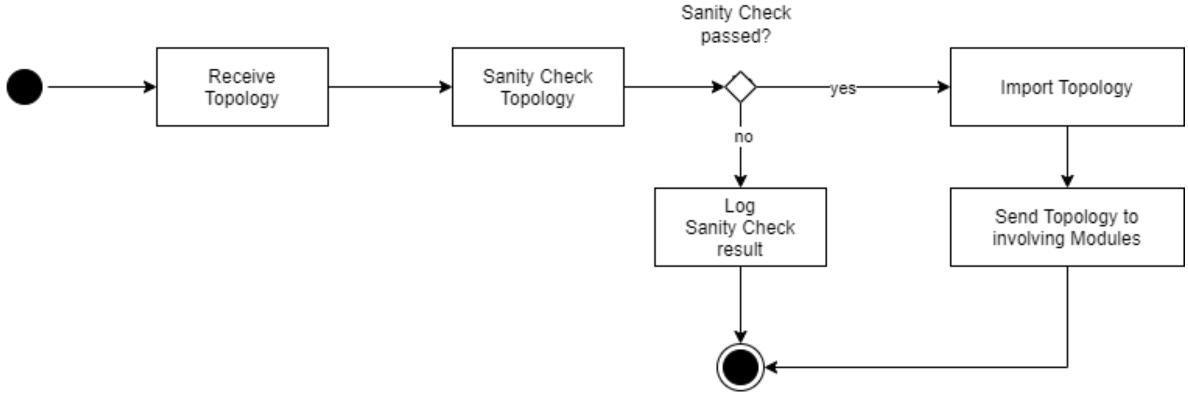
6.1.3.1.4 Workflow

The workflow consists of the following tasks:

- **Receive Topology:** from Fixed Asset Management (FAM)
- **Execute sanity Checks:** verify and protect the system from poor data quality provided from external actors.
- **Import Topology:** to reduce workload, the topology will be persisted in XFN and shared with CMS and TMS.
- **Share Topology:** Additions, modifications and deletions must be communicated to other components.

The import of topology data is triggered autonomously every night for TMS, and CMS depends on planning horizons (update on request). **[**SPT3TMS-15408 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15337 **]**

6.1.3.1.4.1 Sanity Checks

The component is highly dependent on the quality of data and may be impacted by poor data quality from external actors. Therefore, the provision of Topology (via Topology Exchange item) is subject of sanity checks to mitigate the risk of poor data quality.
The following aspects are considered by the sanity check:

- **Duplicates:** Avoid inconsistencies by identifying imported elements with different naming conventions (e.g., INTERSECTION-01 vs. INT1).
- **Mandatory fields:** considering locations, speed limits, track properties, etc

Irrespective, if an aspect may not pass the verification, the sanity check is considered to be failed. Consequently, respective topologies are not considered by the component **-** the rejection will be logged.
[SPT3TMS-15407 **]**

6.1.3.1.4.2 Topology import

Topology data are transferred in a bulk response containing the whole topology for a certain network including different versions and timespans. **[**SPT3TMS-15412 **]**

6.1.3.1.4.3 Workflow triggers

The following table describes the workflow triggers:

Trigger:	**Business-Event:**	**Action:**
On application start-up	The component started initially to sync topology.	Initiate the topology import.
Nightly	A nightly routing check may sync the existing topology.	Initiate the topology import

[SPT3TMS-15411 **]**

6.1.3.2 Sectional run time calculation

TYPE: SERVER ACTOR: XFN ARTIFACT: RUNTIME

Every train has a sectional minimal runtime depending on path, vehicle capabilities, rolling stock and driving strategy. Considering these inputs, a minimal runtime calculation for every single train can carry out. **[**SPT3TMS-10564 **]**

6.1.3.2.1 Actors

The following actors are directly involved in the components communication:

Actor:	Direction:	Scope:	Action:
Topology Master Data	INBOUND	COMPONENT-EXCHANGE	Provision of Topology via Topology Exchange item.
Path mgmt & processing (CMS)	OUTBOUND	COMPONENT-EXCHANGE	Provision of sectional runtime via Calculated Sectional Runtime Exchange item.
Capacity Plan and decision processing (TMS)	OUTBOUND	COMPONENT-EXCHANGE	Provision of sectional runtime via Calculated Sectional Runtime Exchange item.

[SPT3TMS-15410 **]**

6.1.3.2.2 Exchange items

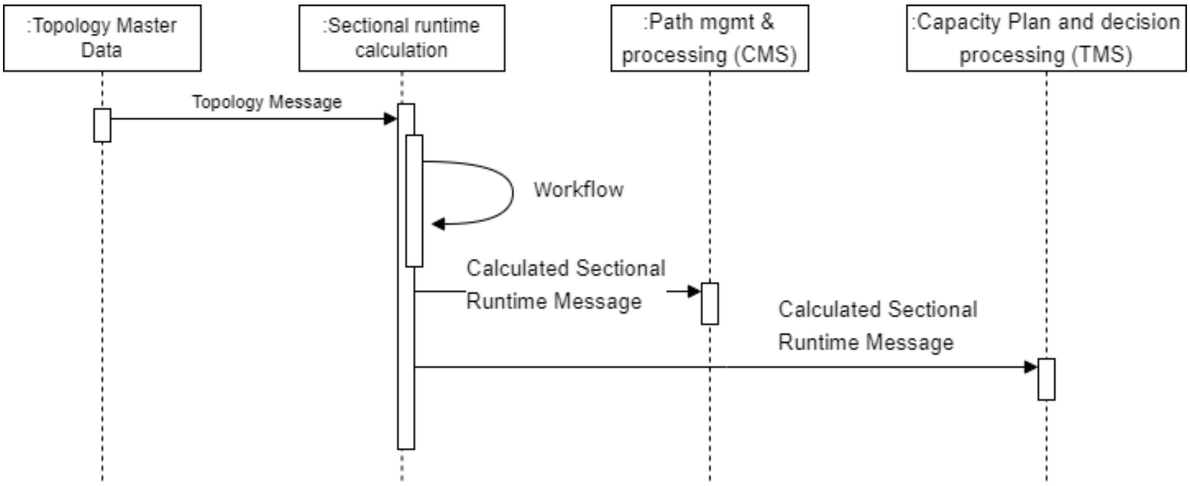
The following Exchange items are involved in the components process:

Exchange item:	Direction:
Topology	INBOUND
Calculated Sectional Runtime	OUTBOUND

[SPT3TMS-15296 **]**

6.1.3.2.3 Interactions

The following sequence diagram illustrates the respective interaction between all actors:



[SPT3TMS-15332 **]**

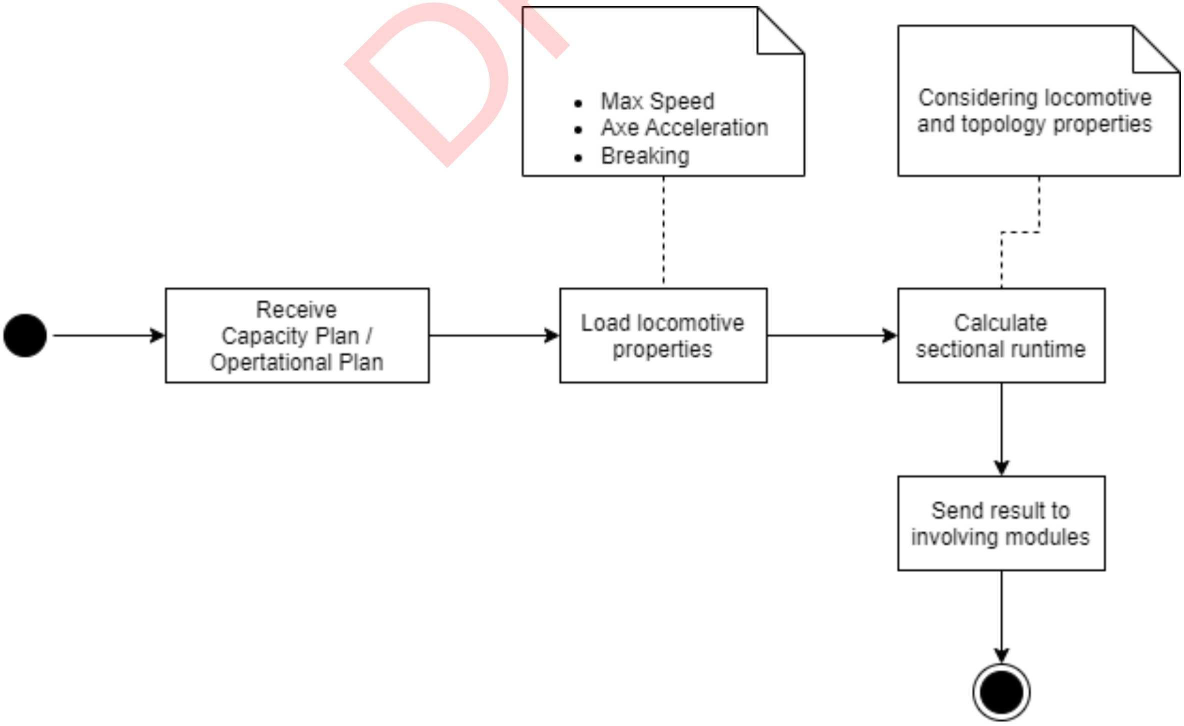
6.1.3.2.4 Workflow

The workflow consists of the following tasks:

- **Receive Capacity Plan / Operational Plan:** to consider the train run from point to point.
- **Load locomotive data:** to consider train properties such as speed and loading times.
- **Calculate sectional runtime:** using distance and speed profile, based on different time spans and topology data with various time horizons
- **Share sectional runtime calculation:** The result must be communicated to other components.

[SPT3TMS-15409 **]**

The following activity diagram illustrates the workflow:



[SPT3TMS-15331 **]**

6.1.3.2.4.1 Workflow triggers

The following table describes the workflow triggers:

Trigger:	Business-Event:	Action:
Creation and modifications of Capacity Plans	A Capacity Plan was created or modified, which requires a sectional runtime calculation.	Initiate the sectional runtime calculation for CMS.
Creation and modifications of Capacity Plans	A Operational Plan was created or modified, which requires a sectional runtime calculation.	Initiate the sectional runtime calculation for TMS.

[SPT3TMS-15416 **]**

6.2 Interface logical view

The following chapters describe the logical interaction and workflow outside of domain scope.
The Interface logical view is structured into the following sub chapters:

- **Interface Exchange item types:** Describes the concrete Exchange item types available for the interface.
- **Interface Interactions:** Describes the interactions between domains and interfaces.

[SPT3TMS-15415 **]**

6.2.1 ROC-TMS Interface

SCOPE: FUNCTIONAL-EXCHANGE ACTOR: ROC

The ROC-TMS interface (north bound) is the connection to the Rail Undertakings (RU) / Rail Operating Companies (ROC). It`s main purpose is a request of paths and sharing the real time status of the rail operation. **[**SPT3TMS-10563 **]**

6.2.1.1 Interface exchange item types

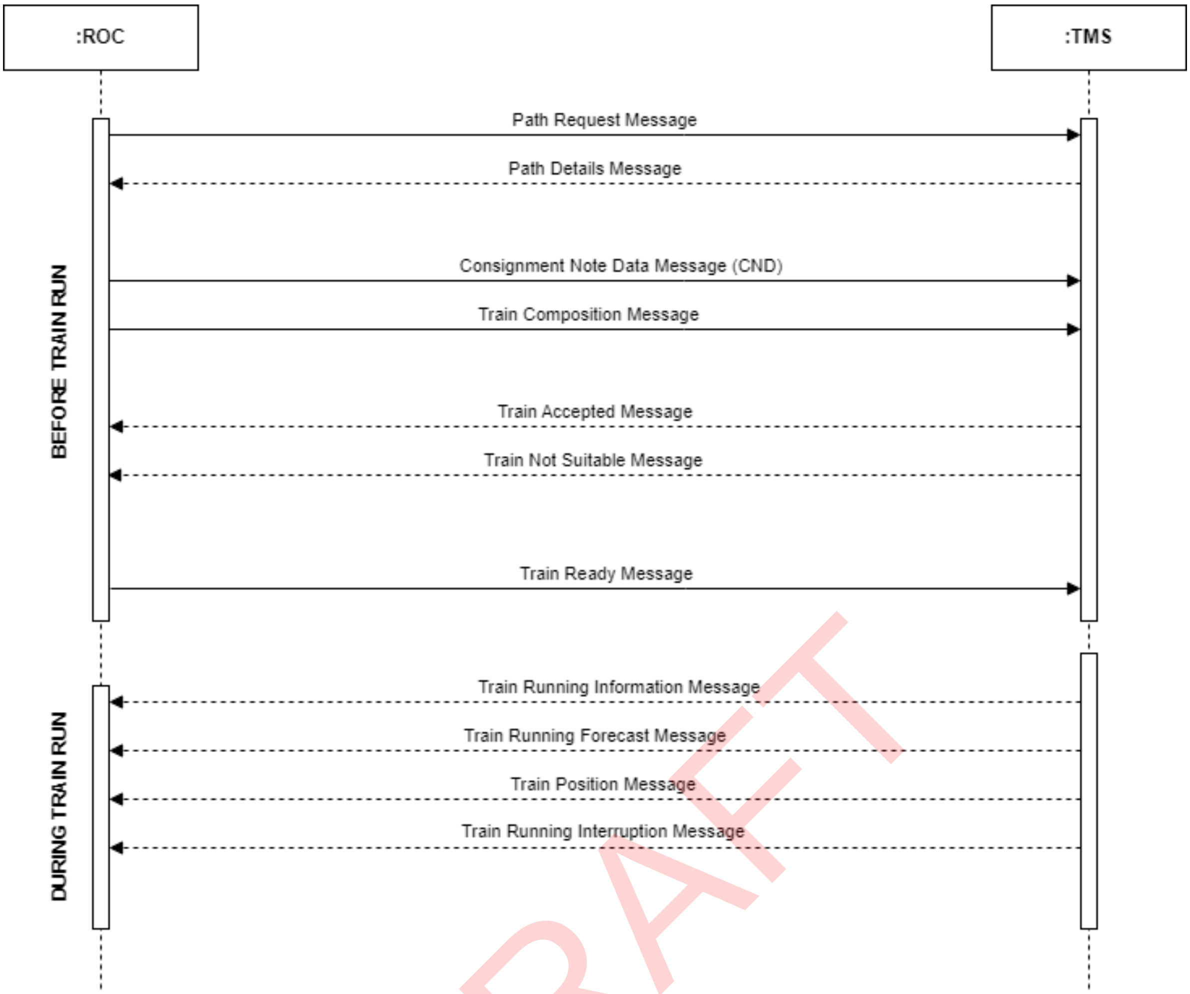
The interface complies to the specifications of TAF/TAP TSI and provides the following Exchange item types:

Exchange item type:	**Direction:**
Consignment Note Data Exchange item / Train Composition	ROC to TMS
Train Accepted Exchange item / Train Not Suitable	TMS to ROC
Train Ready	ROC to TMS
Train Running Information	TMS to ROC
Train Running Forecast	TMS to ROC
Train Position	TMS to ROC
Train Running Interruption	TMS to ROC
Train Request	ROC to TMS

[SPT3TMS-15414 **]**

6.2.1.2 Interface Interactions

The following figure illustrates the interaction between TMS and ROC:



[SPT3TMS-10479 **]**

6.2.2 CMS-TMS Interface

SCOPE: COMPONENT-EXCHANGE ACTORS: CMS , TMS

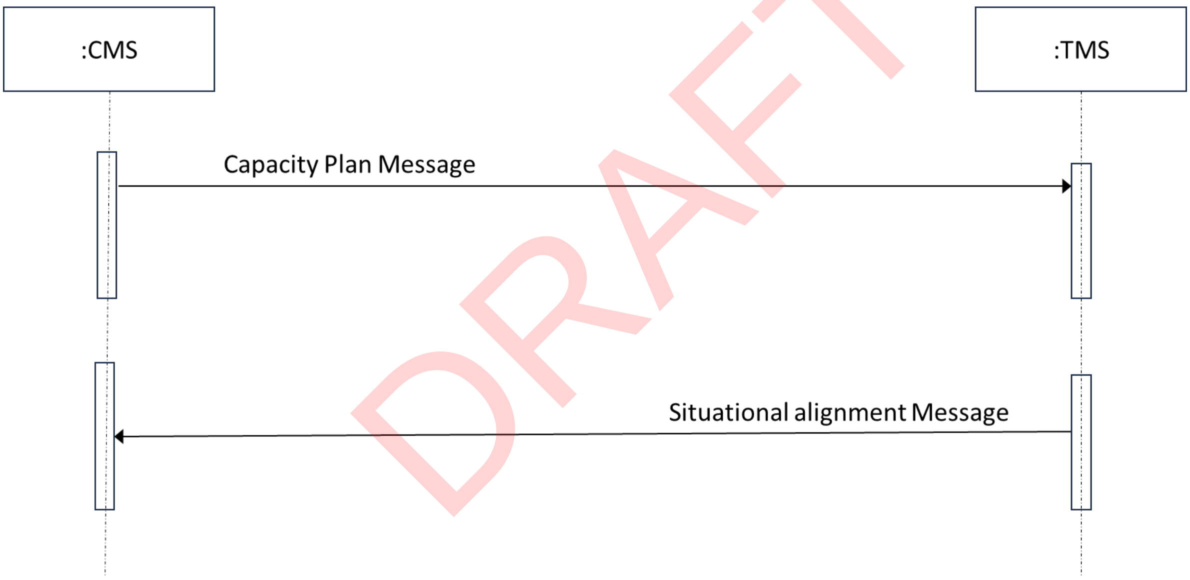
The CMS-TMS Interface implements the transition between Capacity planning and Capacity production, and the fedback of any change in the real-time situation which may have an impact on the CMS time horizion. [SPT3TMS-10555]

6.2.2.1 Interface exchange item types

The interface utilizes the Capacity Plan Message and the Situational alignment messages. [SPT3TMS-15413]

6.2.2.2 Interface Interactions

The following figure illustrates the interaction between CMS and TMS:



[SPT3TMS-16167]

6.2.3 TMS-CCS Interface

SCOPE: FUNCTIONAL-EXCHANGE ACTOR: CCS

The interface between TMS and CCS (south bound), referred to as “Standard Communication Interface Operational Plan (SCI-OP)” in the Reference CCS Architecture (RCA) is located at the system boundary of CCS and connects TMS with the CCS ATO Trackside (ATO-TS) and Plan Execution System (PES) capabilities. As such, it serves to ensure the flow of data between systems while encapsulating their internal logic. **[**SPT3TMS-10559 **]**

6.2.3.1 Interface exchange item types

The TMS-CCS interface covers three main fields:

Exchange item group:	**Description:**
Operational Plan Execution	The Operational Plan Execution is the logical representation of all Exchange items concerning the execution of Operational Plans in the Area of Control. This includes the Operational Plan Execution Request, the Operational Plan Execution Response, and the Operational Plan Execution Report.
Operating State	The Operating State is the logical real-time representation of the actual state of the physical railway system in the Area of Control. The knowledge about the Operating State enables TMS to keep itself current with the operational situation in the Area of Control and to recognise deviations from an Operational Plan during execution. Further, it allows for identifying upcoming or existing conflicts between Operational Plans and developing appropriate countermeasures.
Map Data Activation Time	The Map Data Activation Time defines a time window, defined by TMS and transmitted to Plan Execution and ATO Trackside (ATO-TS), in which the activation of the distributed Map Data for all CCS trackside systems in the Area of Control should take place. This time window is optimal for the activation of Map Data from an operational point of view

[SPT3TMS-15419 **]**

In respect to the Exchange item groups, following Exchange Item types are available:

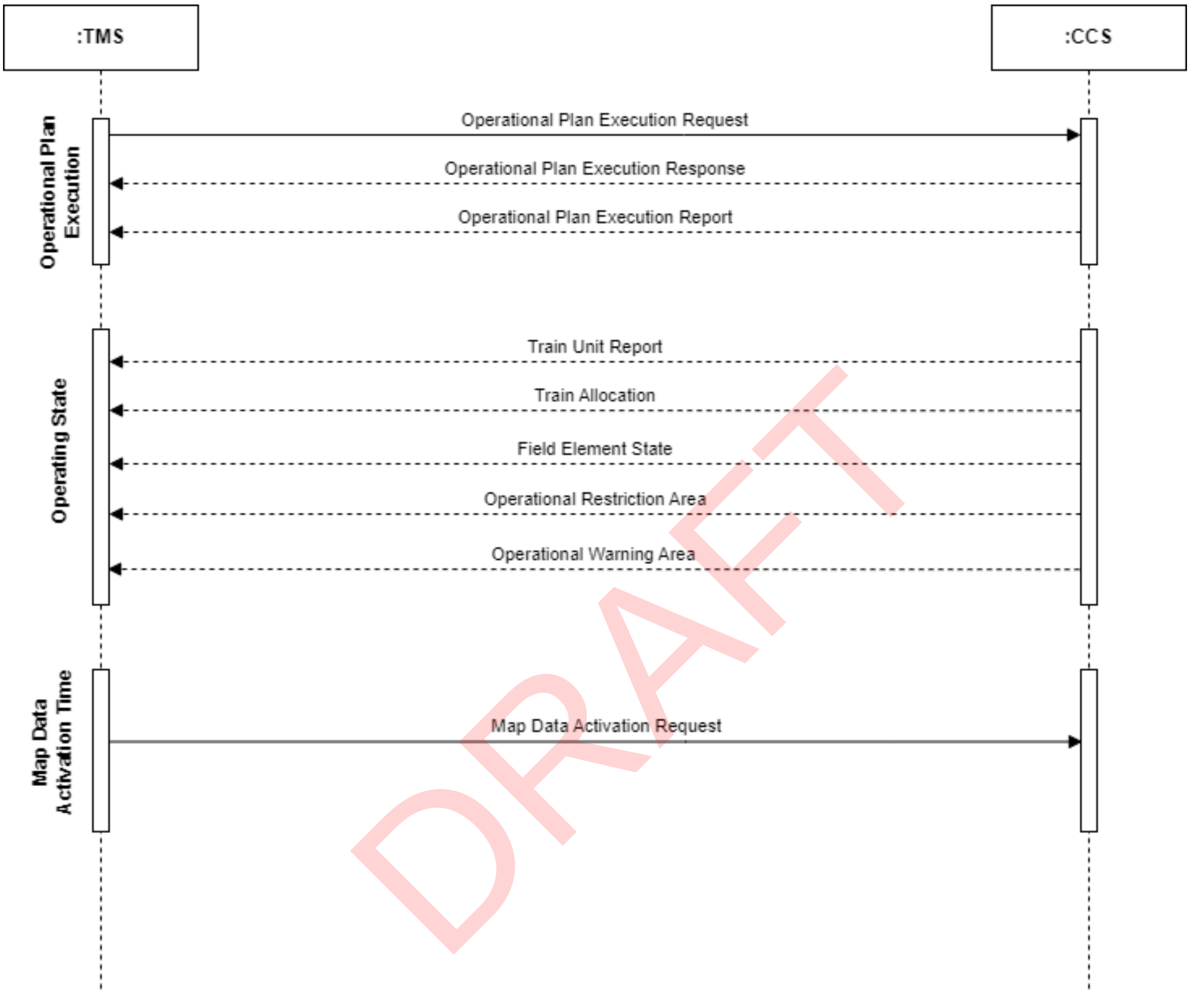
Exchange item group:	**Exchange item type:**
Operational Plan Execution	Operational Plan Execution Request
	Operational Plan Execution Response
	Operational Plan Execution Report
Operating State	Train Unit Report
	Track Allocation
	Field Element State
	Operational Restriction Area
	Operational Warning Area
Map Data Activation Time	Map Data Activation Request

[SPT3TMS-15418 **]**

6.2.3.2 Interface Interactions

The data flow between TMS and CCS is bidirectional and can be distinguished by downstream (TMS to CCS) and upstream (CCS to TMS) Exchange items. **[**SPT3TMS-10565 **]**

The following figure illustrates the interface interactions:



[SPT3TMS-10486 **]**

A detailed description of the respective Exchange Items can be found in the CMS-TMS interface concept: T3-ConceptInterfaceTMSCCS **[**SPT3TMS-16221 **]**

6.2.4 TMS-PIS Interface

SCOPE: FUNCTIONAL-EXCHANGE ACTOR: PIS

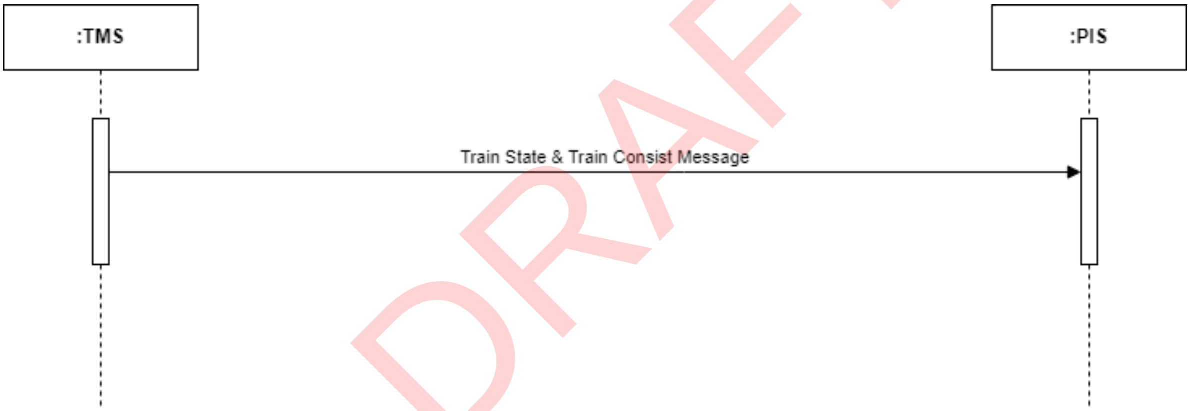
This Interface covers the request of operational information by the Traffic Information System (or Passenger Information System - PIS) to retrieve operational data. The PIS is responsible for informing the passengers of the long-term plan, the current status of the running trains and the compliance with the planning (delays, disruptions, etc.). PIS requests information from TMS. The frequency of these requests can be set automatically according to the type of request and the status of the previous information. TMS controls if the request is compliant with the PIS access rights (e.g., confidentiality filters). Currently no standardization is defined by TSI or RCA or EULYNX, or EU-RAIL. System pillar task 3 does not plan to do this specification. **[**SPT3TMS-10566 **]**

6.2.4.1 Interface exchange item types

The interface utilizes the Train State & Train Consist Message. **[**SPT3TMS-15300 **]**

6.2.4.2 Interface Interactions

The following figure illustrates the interface interactions:



[SPT3TMS-15333 **]**

6.2.5 Incident Impact Management

SCOPE: FUNCTIONAL-EXCHANGE ACTOR: IIM

The Incident Impact Management Interface is responsible to exchange TCRs (e.g., accident, technical malfunction, or weather) to the domain. **[**SPT3TMS-15417 **]**

6.2.5.1 Interface exchange item types

The interface Exchange item will be defined at a later stage. **[**SPT3TMS-15301 **]**

6.2.5.2 Interface Interactions

The interface Interactions will be defined at a later stage.
[SPT3TMS-15423 **]**

6.3 Scenario view

The following chapters describe concrete scenarios, involving multiple domains and interfaces. The Scenario view is structured into the following sub chapters:

- **Scenario Definition:** Describes the concrete scenario including all Pre-conditions.
- **Scenario Measurements:** Describes the required measurements to handle the scenario.
- ***Scenario Interactions:*** Describes concrete interactions between domains and interfaces.
- **Scenario Risks:** Addresses potential scenario risks.

[SPT3TMS-15422 **]**

6.3.1 Synchronized topology import with PE

In this concrete scenario, a topology modification will lead to an outdated Operational Plan resulting in a diverged state for external actors, consuming the outdated Operational Plan. The chapter will clearly explain, which measurements are considered to handle the scenario. **[**SPT3TMS-15421 **]**

6.3.1.1 Scenario definition

Operational Plans are created based on topology information towards a specific time frame. During the time, topology information are subject to changes which lead to outdated Operational Plans. The synchronization of such Operational Plans within CMS/TMS is described in chapter Topology Master Data. As Operational Plans are transferred to other actors (e.g PE) it may happen, that external Actors implemented such Topology changes earlier than TMS transferred an updated Operational Plan, resulting in a diverged state between Topology and Operational Plan. **[**SPT3TMS-15420 **]**

6.3.1.2 Scenario Measurements

A diverged state for external actors may lead to inaccurate actions. Therefore, a diverged state must be omitted at any time.
To omit the use of outdated Operational Plans for external actors, the following actions are mandatory:

- **Topology Version Control:** Comparison between local and Operational Plan Topology version and validity.
- **Topology Pre-Load and Activation Mechanism:** Distribution of Topology information across all involving actors.

[SPT3TMS-15426 **]**

6.3.1.2.1 Topology Version Control

The Version Control should be used to identify outdated Topology states and giving external actors the possibility to act accordingly. The concept mainly focuses on the following attributes to be implemented on every instance (incl. Operational Plans and external actors)

- **Version identifier:** A specific Topology version, will change once modified.
- **Expiration Date:** A specific date in the future, when the version will expire.

With given attributes the validity of Topology information is ensured, as external actors may compare the Topology version against the Topology version of the Operational Plan and further considering the expiration date. Attributes will be included on TrackEdgePoint and TrackEdgeSection level of the Operational Plan. **[**SPT3TMS-15425 **]**

The Expiration Date can also be "never"; when a new Topology is uploaded, the expiration date of the previous one shall be set to the day before the beginning of the validity of the new version. **[**SPT3TMS-15938 **]**

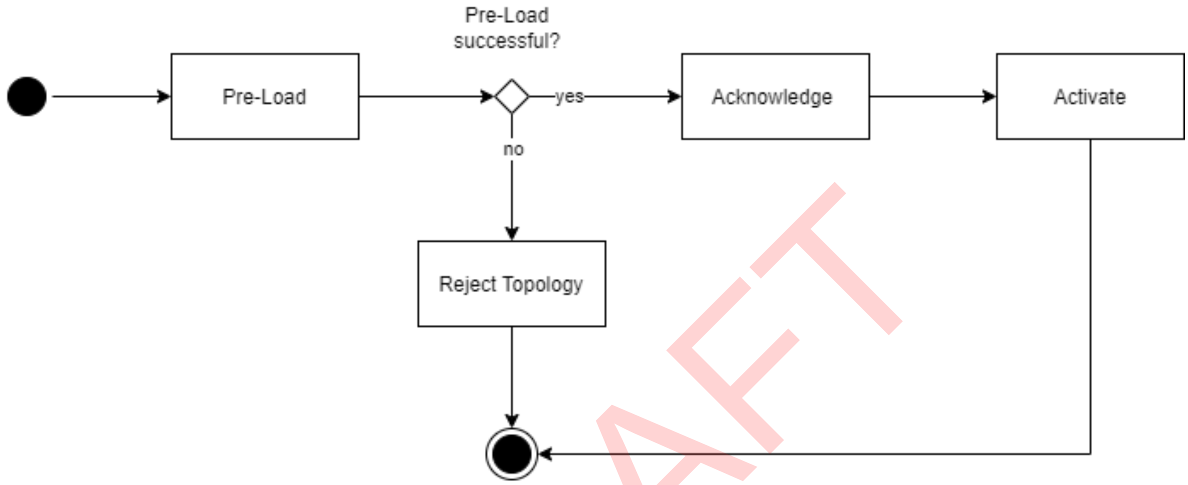
6.3.1.2.2 Topology Pre-Load and Activation Mechanism

The Pre-Load and Activation Mechanism is the process to securely distribute Topology information to all required actors. The process includes the following actions in given order:

- **Pre-Load:** Distribution of Topology information across all involving actors.
- **Acknowledge / Reject:** Confirmation or rejection of Pre-Load.
- **Activation:** Activation of Pre-Load Topology information.

[SPT3TMS-15424 **]**

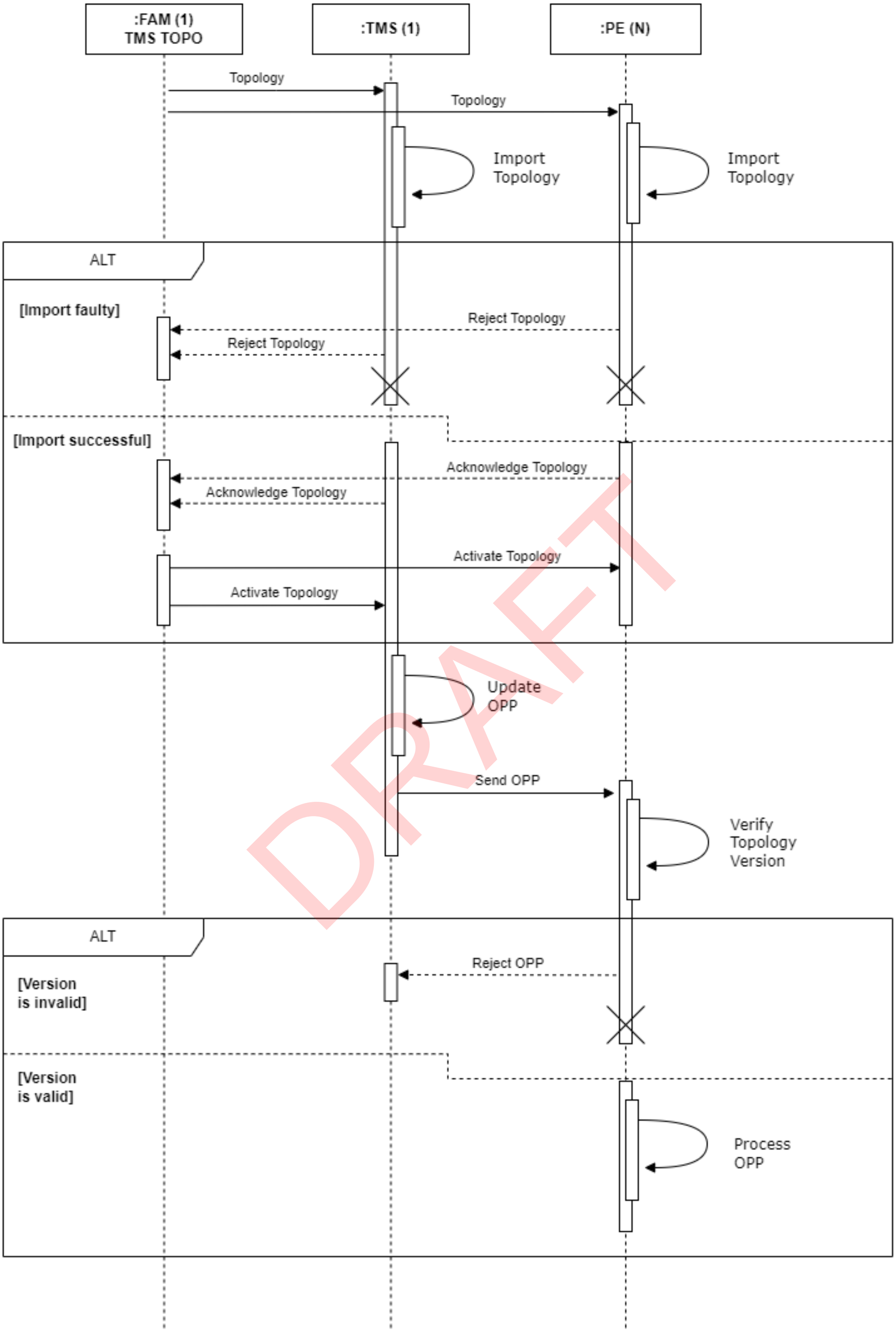
The following activity diagram illustrates the Pre-Load and Activation Mechanism:



[SPT3TMS-15339 **]**

6.3.1.3 Scenario interactions

The following sequence diagram illustrates the process of Topology Pre-Load and activation:



[SPT3TMS-15343 **]**

6.3.1.4 Scenario Risks

The following risks are not addressed to TMS, however should be addressed accordingly:

- No acknowledge of Pre-Load for single actors / instances
- Version of local and operational Topology differs

[SPT3TMS-15379 **]**

6.3.2 Operational Plans for cross border train runs

In this concrete scenario, an Operational Plan describes a cross border train run from Hamburg (D-AA) to Interlaken-Ost (CH-IO) based on the train run EC6. As designed, the train will pass a (national) border between Germany (DE) and Switzerland (CH). The scenario highlights the complexity of cross border train runs and addresses potential risks and measurements to overcome the complexity. **[**SPT3TMS-15378 **]**

The following table describes the train run from Hamburg to Interlaken in detail:

Station nr.:	**Station:**	**Country:**
1	Hamburg-Altona	Germany
2	Hamburg Dammtor	Germany
3	Hamburg HBf	Germany
4	Hamburg Harburg	Germany
5	Bremen HBf	Germany
6	Diepholz	Germany
7	Osnabrück Hbf	Germany
8	Münster (Westf) Hbf	Germany
9	Dortmund HBf	Germany
10	Bochum HBf	Germany
11	Essen HBf	Germany
12	Duisburg HBf	Germany
13	Düsseldorf HBf	Germany
14	Köln Hbf	Germany
15	Bonn HBf	Germany
16	Koblenz HBf	Germany
17	Mannheim HBf	Germany
18	Karlsruhe HBf	Germany
19	Freiburg (Breisgau)	Germany
20	Basel Bad Bf	Germany
21	Basel SBB	Switzerland
22	Liestal	Switzerland
23	Olten	Switzerland
24	Bern	Switzerland
25	Thun	Switzerland
26	Spiez	Switzerland

Station nr.:	Station:	Country:
27	Interlaken West	Switzerland
28	Interlaken Ost	Switzerland

[SPT3TMS-15377 **]**

6.3.2.1 Scenario definition

The depicted train run (as defined) will cross several borders, including a national border between Germany and Switzerland in Basel SBB (Station 21) and several regional borders within Germany. A border cross is connected to a change of responsibility leading to the use of a different TMS-instance. The scenario addresses the the risk and complexity of dealing with several TMS instances during the train run, dispatching the train from its origin to its final destination.
Its of highest relevance to streamline the process between all involving instances, otherwise multiple instances dispatch the same train run resulting in interfering behavior, inefficient decision and confusion. **[**SPT3TMS-15376 **]**

6.3.2.2 Scenario Measurements

The risk of interfering behavior, inefficient decision and confusion, caused by concurrent TMS instance interactions must be omitted at any time. The following measurements explain how to mitigate the risk:

- **Definition of clear responsibilities:** It must be clear who is responsible for the specific region of the train run.
  Further described in chapter: 6.3.2.3 - Area of Control (AoC)
- **Logical cut:** Operational Plans are cut into several parts along the border of its AoC
  Further described in chapter: 6.3.2.5 - Logical cut in Operational Plans

[SPT3TMS-15384 **]**

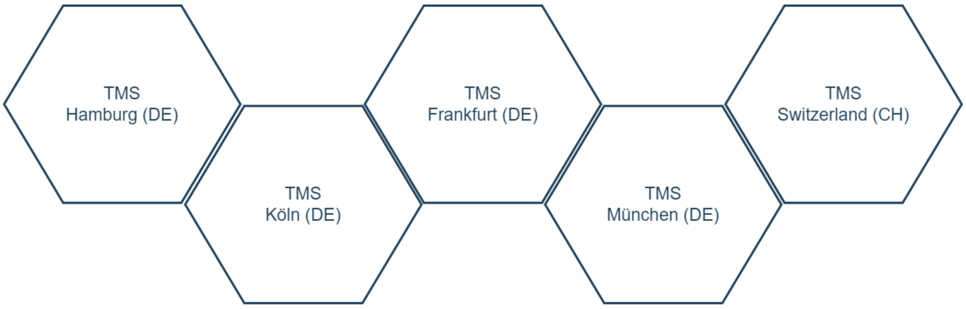
6.3.2.3 Area of Control (AoC)

An Area of Control (AoC) is a specific geographical region for a TMS in which the TMS will dispatch all train runs. The AoC clearly defines the responsibility-border for a TMS.
The geographical region of an AoC depends on the TMS type. This can be described as the following:

- **National TMS:** One TMS covering the entire nation (e.g. Switzerland)
- **Regional TMS:** One TMS covering a specific region - multiple TMS instances for an entire nation (e.g. Germany)

As next, trains crossing the border of a AoC must be handed over to the next TMS covering the current position of the train. The handover will be initiated minutes before the train will leave the AoC and requires a logical connection to the next responsible TMS. **[**SPT3TMS-15383 **]**

The following figure illustrates the different TMS types and its connection:

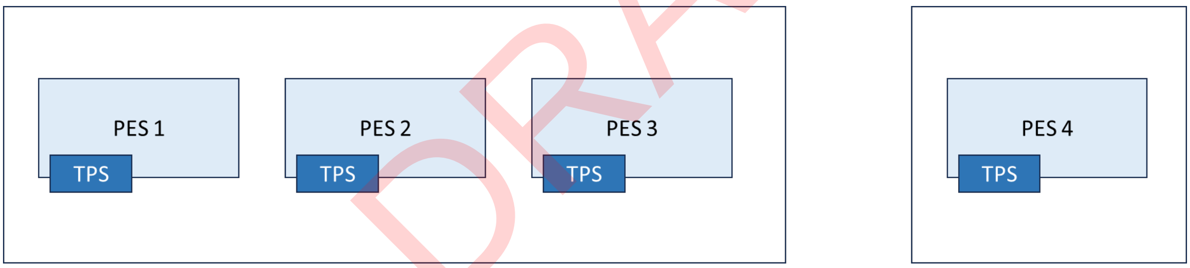


[SPT3TMS-15342]

6.3.2.4 Plan Execution System (PES) in Scope of TMS

Another important aspect relies in the process of Operational Plans in Plan Execution System (PES). The Operational Plan will be distributed to Plan Execution System (PES) using SCI-OP as described in chapter: 6.2.3 - TMS-CCS Interface. Plan Execution System (PES) will receive the Operational Plan and is directly connected to the physical asset (train) and Train Protection System (TPS).
Plan ExecutionSystem may cover the entire region of the AoC (1 TMS = N PES) - but can also be splitted into different smaller regions (1 TMS = 1 PES). In any case Plan Execution System will consume one Operational Plan from TMS and extracts the related part for Plan Execution System.This also implies to the Topology Version Control. [SPT3TMS-15382]

The following figure illustrates several instance of TMS, Plan Execution System and TPS:



[SPT3TMS-16168]

6.3.2.5 Logical cut in Operational Plans

A TMS is restricted by the geographical border of its own Topology within the AoC. As a consequence, TMS can only process Operational Plans within its own AoC - Topology information outside of the AoC doesn't exist. This requires the logical cut of Operational Plans in accordance to the number of passing AoC. Each part of the Operational Plan reflects the part of the train run within the AoC only. [SPT3TMS-15380]

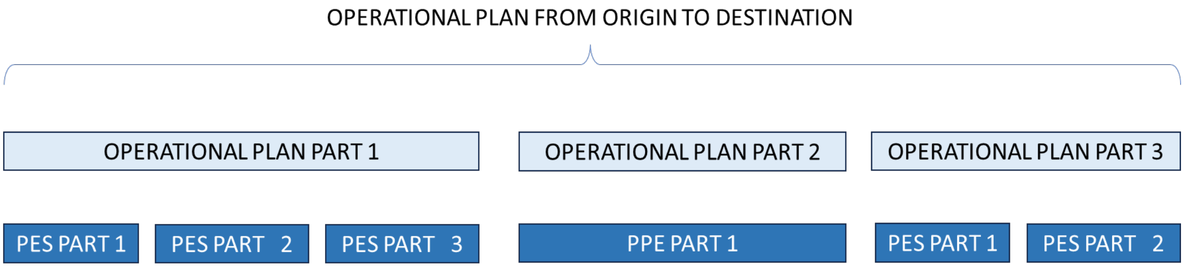
In respect to the given scenario, the train run will pass five different AoC's - resulting in five different Operational Plans:

- **(DE) Hannover:** Describes the train run from Hamburg to Köln
- **(DE) Münster:** Describes the train run from Köln to Frankfurt
- **(DE) Frankfurt:** Describes the train run from Frankfurt to München
- **(DE) München:** Describes the train run from München to Basel (CH)

- **(CH) Switzerland:** Describes the train run from Basel to Interlaken

[SPT3TMS-15390 **]**

The following figure illustrates the logical cut for an Operational Plan:



[SPT3TMS-16169 **]**

7 Deployment View

The Deployment View describes important technical infrastructure aspects and its relation to non functional requirements.
The choice or implementation of concrete solutions are not part of the document – However to avoid side effects its highly important to stick with the same solution across the whole system context.
The Deployment View is structured into the following sub-chapters:

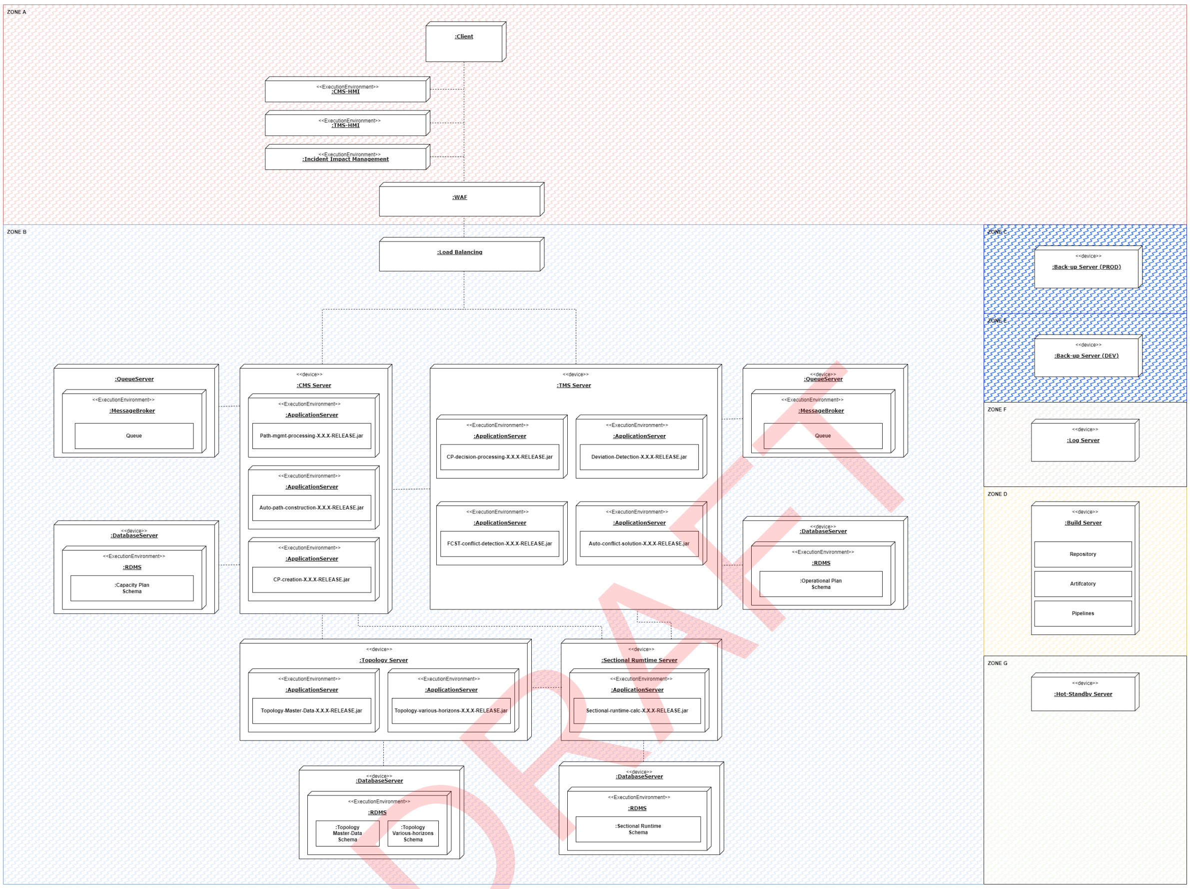
- **Infrastructure Components:** Overview of technical parts.
- **Infrastructure Scalability:** Scaling components based on load.
- **Availability Zones:** Isolation and separation of data located within specific geographical regions.
- **Infrastructure Platforms:** Overview of platforms and the implication of on-premise vs. Cloud-Computing.

[SPT3TMS-13918 **]**

7.1 Infrastructure Components

Infrastructure components have an important implication to the success of the overall as they correspond to Availability, Security and Functional requirements. Infrastructure components cover a wide range from data persistence, data transportation to access control and back up solution. **[**SPT3TMS-13815 **]**

The following deployment diagram illustrates the necessary technical infrastructure components to fulfill the described requirements and ensure the systems execution: **[**SPT3TMS-10567 **]**



[SPT3TMS-10485 **]**

The following table describes the infrastructure components:

Component:	**Multiplicity:**	**Rationale:**
Web Application Firewall (WAF)	1	Accept / Reject requests
Load Balancer	1+N	Distribute load
Application Server	1+N	Running applications, one Application Server per Microservice / Instance.
Database Sever	1+N	Storing data in a database, running in a cluster
Message Broker	1+N	Processing message queues
Backup-Server	2	Storing backups
Build Sever	1	

Component:	Multiplicity:	Rationale:
		Build Server for building and testing the system, reproducing bugs, etc
Log Server	1	Storing log information as centralized logging solution
Hot-Standby Server	1	Standby solution

[SPT3TMS-13919 **]**

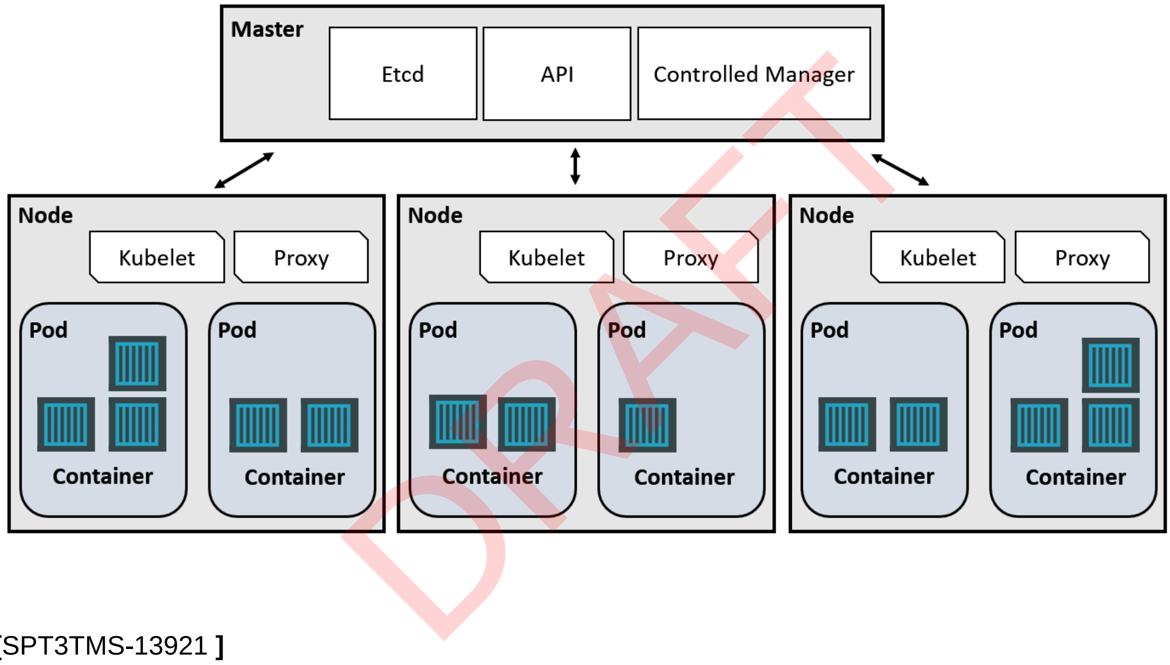
7.2 Infrastructure Scalability

Different demands during train operations are the reason for high peaks on infrastructure level. The infrastructure must be able to scale, in any situation. This includes up-scaling and down-scaling to align operational costs with efficient resource allocations.
The infrastructure must scale on the following components:

- **RAM/CPU allocation:** For vertical scaling services, the infrastructure must be able to allocate more/less server resources.
- **VM (de)activation:** For horizontal scaling services, the infrastructure must be able to manage VMs.

It is considered to manage the infrastructure in a cluster. This allows the scaling of resources based on workload. Each node represents a VM and consists of multiple pods. A pod consists of multiple containers which represents the running microservice. It is assumed, that a pod operates multiple instances of the same container type only.

The following figure illustrates an example of a cluster: **[**SPT3TMS-13922 **]**



[SPT3TMS-13921 **]**

7.3 Availability Zones

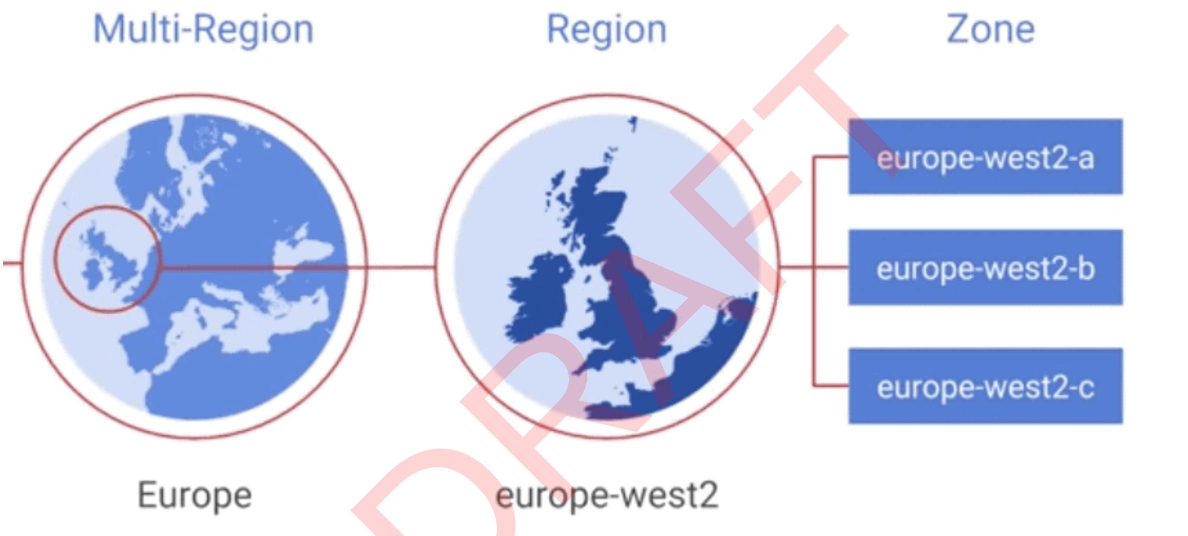
Availability Zones are a crucial part of the overall architecture.
With Availability Zones, the infrastructure can be operated from specific geographical region.

The following list describes the implications related to Availability Zones:

- **Logical implications:** Protect the system from total loss, if the geographical region collapses (e.g. earthquake, fire, etc).
- **Legal implications:** Comply to regulations and operate infrastructure only in approved geographical regions (e.g. EU, CH).

Availability Zones are restricted to the capability of the platform operator.
However, the infrastructure must be hosted within multiple geographical independent region to comply with the previously mentioned implications.
[SPT3TMS-13923 **]**

The following figure illustrates the structure of Availability Zones:



[SPT3TMS-13920 **]**

7.4 Infrastructure Platforms

Several factors have been identified to impact the decision of choosing the respective infrastructure platform. The infrastructure platform is the base for implementing all mentioned aspects of efficiency, performance, and reliability. **[**SPT3TMS-10568 **]**

The following table illustrates important considerations for choosing the infrastructure platform:

Consideration:	**Rationale:**	**Mitigation:**
Process of sensitive information	Legal/company/safety related restrictions may prohibit to store or process data outside of a specific domain.	Store sensitive data only in allocated protected zones and never outside the specific domain.
Dependencies to hardware / providers	The system will be affected if the dependence experience outages or failures.	Never rely on one specific hardware part or platform provider. Furthermore, rely on multi regions to compensate outages or failures.
Flexibility of (software) updates	The hardware / platform may be limited by the operating system (OS) to perform updates or may be forced to update to a specific version which is not foreseen.	Only consider hardware / platform with frequently update cycles and an active community.

[SPT3TMS-10571 **]**

8 Cross-cutting Concepts

This section describes overall, principal regulations and solution ideas that are relevant in all parts of the system.
The section is structured into the following parts: **[**SPT3TMS-10572 **]**

- **Availability Concepts:** Addresses important aspect to realize availability.
- **Reliability Concepts:** Addresses important aspect to build a reliable system.
- **Real time Concepts:** Addresses important aspect to realize real time behaviour.
- **Code Concepts:** Addresses code specific concepts (Clean Code, Design Principles, Testing)
- **Domain Concepts:** Addresses domain specific concepts (Concurrency, Caching).
- **Operational Concepts:** Addresses operational related requirements (Stages, Monitoring).
- **Automation Concepts:** Addresses automation possibilities (CI/CD).
- **Security Concepts:** Addresses security related concerns and how to mitigate harm and risk.

[SPT3TMS-10717 **]**

8.1 Availability Concepts

Legal regulations are the foundation for availability requirements of the system. Downtimes will have a significant impact to business goals and furthermore may lead to serious implications in economic and society. Any Downtime must be prevented at any time and any cost. **[**SPT3TMS-10575 **]**

The following table describes the availability goals of the system:

Domain:	Availability:	Rationale:
CMS	99,7%	Highly critical as a downtime would affect capacity production on short and long term. Compared to TMS the availability is minimally reduced.
TMS	99,98%	Highly critical as a downtime would affect capacity production immediately causing train delays and cancellation.

[SPT3TMS-10574 **]**

The following actions will make sure to ensure the availability goals of the system:

Actions:	Rationale:	Concept:
Scalability	The ability to increase/decrease the number of allocated workers to an estimated load.	8.1.1 - Scalability
Elasticity	The ability to increase/decrease the number of allocated workers to a dynamic load – flexibility for load peaks.	8.1.2 - Elasticity

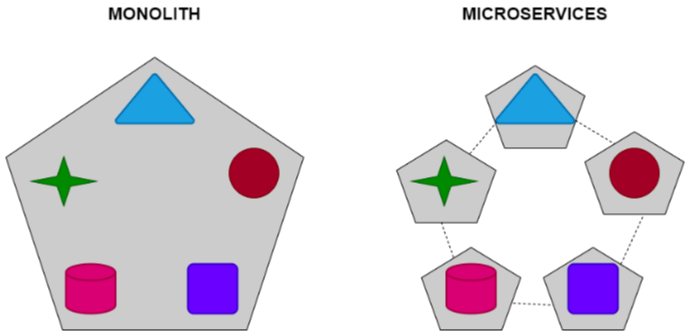
[SPT3TMS-10577 **]**

8.1.1 Scalability

Besides the targeted workload of every day, increased amount of work or demand affect the performance of the overall system. Peaks represent a serious risk to the system and its availability. The system must always act efficient, performant, and reliable.
In the following chapters will be explained how to achieve all aspects of scalability. **[**SPT3TMS-10576 **]**

8.1.1.1 Architectural Style

In respect to the technical requirements, architectural styles are important to consider. Monolithic and microservices are two styles commonly used in the field of software architecture.
The following figure illustrates both architectural styles: **[**SPT3TMS-10579 **]**



[SPT3TMS-10488 **]**

Monolithic architecture bundles the solution into one artifact, which simplifies the development and has a positive impact on the velocity and costs of the overall project. On the other hand, monolithic architecture is very limited in terms of scalability and represents a significant bottleneck in this regard. Scalability in monolithic architectures, can be achieved by instantiating the whole artifact repeatedly. Individual parts are not considered by this approach. Consequently, unnecessary functionality will also be scaled up. Scalability in monolithic architecture becomes complex and impacts the cost effectiveness of the overall. **[**SPT3TMS-10578 **]**

Microservice architecture is known specifically for the reason to be most supportive of scalability. The same software requirements are structured into small individual, isolated artifacts. By concept those individual parts of software can be scaled as desired. For the reason being, a microservice architecture should be chosen as an architectural style.
Scalability can be realized by increasing the number of allocated nodes (horizontal) or increasing the number of allocated server resources to the existing node (vertical). Both types are commonly used in the field of scalability. In general, horizontal scalability must be chosen whenever a microservice can be paralyzed, otherwise vertical scalability must be chosen. **[**SPT3TMS-13814 **]**

The following table illustrates the available microservices:

Microservice:	Type:	State:	Rationale:	Domain:
Topology Master Data Validation & Import	HORIZONTAL	STATELESS	The process is parallelizable.	CMS
Topology various time horizons	HORIZONTAL	STATELESS	The process is parallelizable.	
Path mgmt. & processing	HORIZONTAL	STATELESS	The process is parallelizable.	
Manual Path Construction	HORIZONTAL	STATELESS	The process is parallelizable.	
Manual Path Conflict Detection	HORIZONTAL	STATELESS	The process is parallelizable.	
Automatic Path Construction	HORIZONTAL	STATELESS	The process is parallelizable.	
Capacity Plan Export	HORIZONTAL	STATELESS	The process is parallelizable.	
Topology Master Data Validation & Import	HORIZONTAL	STATELESS	The process is parallelizable.	TMS
Daily Topology	HORIZONTAL	STATELESS	The process is parallelizable.	
Capacity plan and decision processing	HORIZONTAL	STATELESS	The process is parallelizable.	
Deviation Detection	VERTICAL	STATEFULL	It is expected to experience interferences with other instances when running horizontally. The process is not parallelizable as concurrent executions would decrease the calculation quality.	
Forecasting & RT-Conflict Detection	VERTICAL	STATEFULL	It is expected to experience interferences with other instances when running horizontally. The process is not parallelizable as concurrent executions would decrease the calculation quality.	
Automatic Conflict Solution	HORIZONTAL	STATELESS	The process is parallelizable.	
Topology Master Data	HORIZONTAL	STATELESS	The process is parallelizable.	XFN

Microservice:	Type:	State:	Rationale:	Domain:
Topology various horizons	HORIZONTAL	STATELESS	The process is parallelizable.	
Sectional Runtime calculation	HORIZONTAL	STATELESS	The process is parallelizable.	

[SPT3TMS-10580 **]**

8.1.1.2 Load Balancing

Load balancing is a further mechanism to distribute load within the system and a crucial step to realize high performance, real time, and high availability goals. Load balancing is the initial system between client/server communication and is used for the follow use cases: **[**SPT3TMS-10581 **]**

- **Load Distribution:** To minimize the load on one instance, load balancing will distribute the load on multiple instances.
- **SSL Termination:** Reject http requests and forward https requests only.
- **Network Zones Tunnelling:** The client is outside of the systems network zone and to communicate with the system (client / server communication), load balancing provides tunnelling.

[SPT3TMS-10718 **]**

Load balancing is based on algorithms to measure and decide how to distribute load between available instances. Instances are partitioned equally across the system **-** Therefore, the Least Connections algorithm should be considered. **[**SPT3TMS-10583 **]**

Please consider chapter 8.8.7 - Network Segmentation to understand the use of load balancing. **[**SPT3TMS-16222 **]**

8.1.2 Elasticity

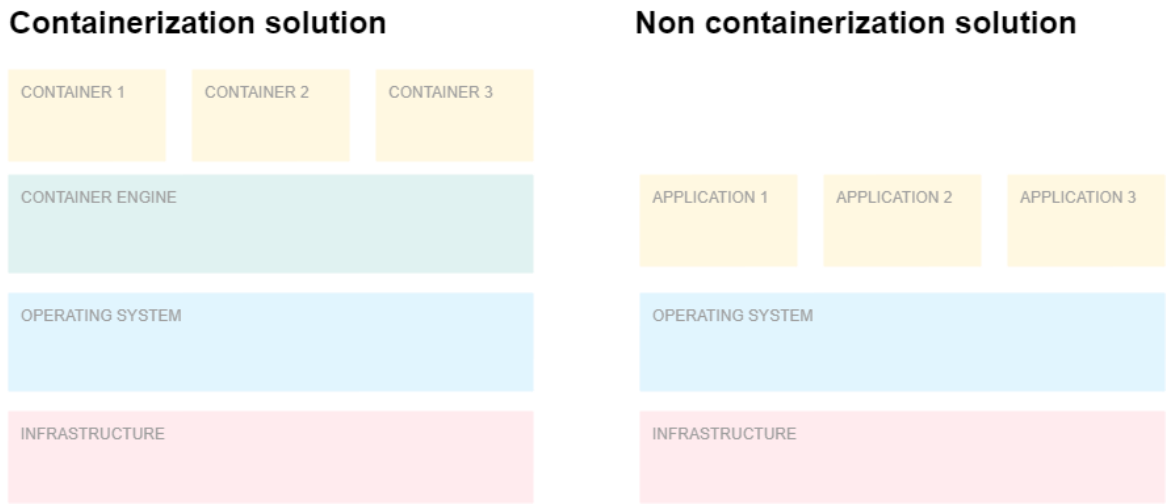
Elasticity refers to the ability to automatically expand or compress the infrastructural resources on a sudden up and down in the requirement so that the workload can be managed efficiently. This elasticity helps to minimize infrastructural costs.
The following components were identified to experience sudden workload fluctuations. Therefore, those components must be elastic: **[**SPT3TMS-10586 **]**

Component:	**Rationale:**	**Domain:**
Path mgmt. & processing	A high increase / decrease of capacity planning fluctuates the workload.	CMS
Capacity Plan Export		
Automatic Path Construction		
Manual Path Construction	Influenced by the quality and amount of available capacity plannings.	
Manual Path Conflict Detection		
Capacity plan and decisions processing	A high increase / decrease of capacity production fluctuates the workload.	TMS
Deviation Detection	Influenced by the quality and amount of available capacity productions.	
Forecasting & RT-Conflict Detection		
Automatic Conflict Solution		
Sectional Runtime calculation	A high increase / decrease of capacity planning / production fluctuates the workload.	XFN

[SPT3TMS-10719 **]**

8.1.2.1 Containerization

With containerization the system will be operated process isolated from the overall. Thereby, each component will be “containerized”: a bundle of the software artifact and all necessary dependencies. **[**SPT3TMS-10589 **]**

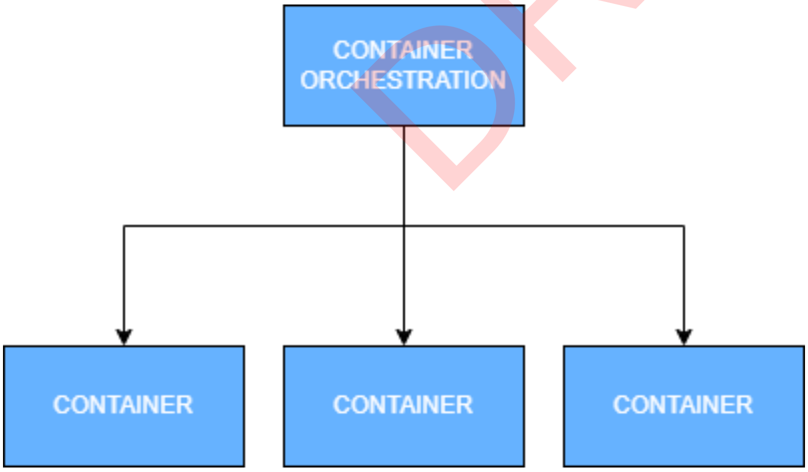


[SPT3TMS-15346]

With containerization the operation of the system becomes independent of the host operating system (if the containerization engine is installed). As a significant difference in the illustration, the containerization engine will decouple the system from the host. Host modifications will not immediately affect the container and therefore increases the stability and reliability of the system as disruptive interventions to the system are voided by design.
[SPT3TMS-15388]

8.1.2.2 Container Orchestration

Container orchestration is the key concept of realizing elasticity and an addition to the concept pf containerization. Container orchestration is the technology to manage multiple containers, their networking, and their resources in distributed environment. [SPT3TMS-10588]



[SPT3TMS-15345]

With container orchestration the system can be automatically scaled up and down dependent on the workload of the system.
Common examples of technology vendors providing container orchestration include Kubernetes, Docker Swarm, Hashicorp Nomad, Apache Mesos and Rancher. [SPT3TMS-10482]

8.2 Reliability Concepts

The reliability of the system is an important step. The following concepts will explain how the reliability of the system should be improved: **[**SPT3TMS-10585 **]**

	Actions:	Rationale:	Concept:
REMAIN-OPERATION	Code Analyse	Lacking code quality has a negative impact on the performance of the system and may lead to unexpected behaviour. By code analysis, lacking code can be identified and be fixed accordingly.	8.4.3 - Code Analysis
	Code Testing	System bugs prevent the system from being reliable. Drive by the complexity, code testing is the only approach to remain consistency and reduce the potential number of bugs.	8.4.4 - Code Testing
	Message Queues	With message queues, the system is decoupled from complex dependencies. Message queues distribute load and reduce workload on single notes.	8.5.1.2 - Message Queues
	Monitoring	Monitoring will highlight important metrics to identify potential improvements.	8.6.3 - Monitoring
	Security	Security concerns may lead to system faults or the need to shut down the system to reduce further harm. Only with a solid security concept, the potential risk can be mitigated.	8.8 - Security Concepts

	Actions:	Rationale:	Concept:
	Availability	With increasing workload, the system must be able to distribute load accordingly. Availability defines how to handle workload accordingly.	8.1 - Availability Concepts
	Disaster Recovery-Plan (DR)	Even if a certain situation will force the system to fault, a Disaster Recovery-Plan (DR) will ensure its operation.	8.6.4 - Disaster Recovery-Plan (DR)
RETURN-TO-OPERATION	Back-ups	Even if the system experience unforeseen exceptions (e.g., caused by hardware failures), back-ups will protect the system from data loss.	8.6.6 - Back-ups
	Versioning	Versioning will make the system capable to roll back to a stable application version.	8.6.5 - Versioning
	Monitoring	Monitoring will highlight important metrics to identify the issue.	8.6.3 - Monitoring
	Disaster Recovery-Plan (DR) / Failback Strategy	A total downtime of the system will require a Failback Strategy to mitigate the harm.	8.6.4.2 - Failback Strategy

[SPT3TMS-10584 **]**

8.3 Real time Concepts

The system must be able to process data on a real time basis. Real-time processing requires a continual input, constant processing, and steady output of data. **[**SPT3TMS-10591 **]**

The following table illustrates mandatory actions to accomplish a real-time processing:

Action:	**Rationale:**
Event-driven architecture (EDA)	Asynchronous processing of backend related data using a queue.
WebSockets	Asynchronous processing of frontend related data using WebSocket's.
Real-time flow	The real-time flow is designed to be fast using non-blocking frameworks and databases and an optimized queuing system to dispatch the requests as soon as possible. Additionally, asynchronous framework leverages to keep the flow non-blocking or delaying any process.
Offline flow	The Offline-flow is responsible to persist the data into different data-sources to ensure each transaction is persisted.

[SPT3TMS-10590 **]**

Furthermore, real-time processing depends on scalability concepts. Therefore, please read the following concept: 8.1.1 - Scalability **[**SPT3TMS-16217 **]**

8.4 Code Concepts

The following concepts will describe important code aspects. **[**SPT3TMS-10592 **]**

8.4.1 Code Standards

Clean code has an important implication of the project’s success as clean code is more efficient and reduces potential bugs.
The following code guidelines should be followed:

- **Readability:** Readable code is easy to follow and optimizes space and time
- **Conventions:** Follow conventions will structure the code in a better way and reduces unnecessary lines of code - less code is less complexity.
- **Indentation:** Proper indentation is very important to increase the readability.
- **Design Principles:** Resolves common problems in a very structured way.
- **Documentation:** The code should be properly commented for understanding easily.

Especially for microservices, the following guidelines should be followed:

- *Software components should be built as independent stateless services.*
- *All business logic in a service should be encapsulated with the data upon which it acts.*
- *There should be no direct access to a database from outside a service. Any and all access to a database should be accomplished by invoking a service specifically implemented to do so.*
- *Each* service *should publish an interface that enables access to its data and functionality by other services.*

Furthermore, the system follows the 12-factor-app methodology:

1. **Codebase:** One codebase tracked in revision control; many deploys.
2. **Dependencies:** Explicitly declare and isolate dependencies.
3. **Config:** Store config in the environment.
4. **Backing services:** Treat backing services as attached resources.
5. **Build, release, run:** Strictly separate build and run stages.
6. **Processes:** Execute the app as one or more stateless processes.
7. **Port binding:** Export services via port binding.
8. **Concurrency:** Scale out via the process model.
9. **Disposability:** Maximize robustness with fast startup and graceful shutdown.
10. **Dev/prod parity:** Keep development, staging, and production as similar as possible.
11. **Logs:** Treat logs as event streams.
12. **Admin processes:** Run admin/management tasks as one-off processes.

[SPT3TMS-10595 **]**

8.4.2 Design Principles

Design principles are one important aspect of the System. Design principles provide structural guidance in certain implementations. Overall design principles reduce risks and have a positive impact on maintainability and velocity.
Certain principles are common in the industry and highly recommended to follow: **[**SPT3TMS-10597 **]**

Name:	**Statement:**	**Rationale:**	**Implications:**
SOLID	Single Responsibility Open/Closed Liskov Substitution Interface Segregation Dependency Inversion	Guidance / Rules for object-oriented programming (OOP)	Provides structure, reduces complexity, effort, costs, and maintainability.
DRY	Don't Repeat Yourself	Do not use different solutions to do the same thing.	Reuse already found solutions / Use a single component (when possible) to handle multiple capabilities
KISS	Keep It Short & Simple	Keep the solution easy to maintain.	If it's too complex, it will become costly and complicated, hampering maintainability
YAGNI	You ain't gonna need it	Don't over engineer the system.	If the functionality is not required, reduce effort/ costs, and avoid possible bugs.

[SPT3TMS-10594 **]**

8.4.3 Code Analysis

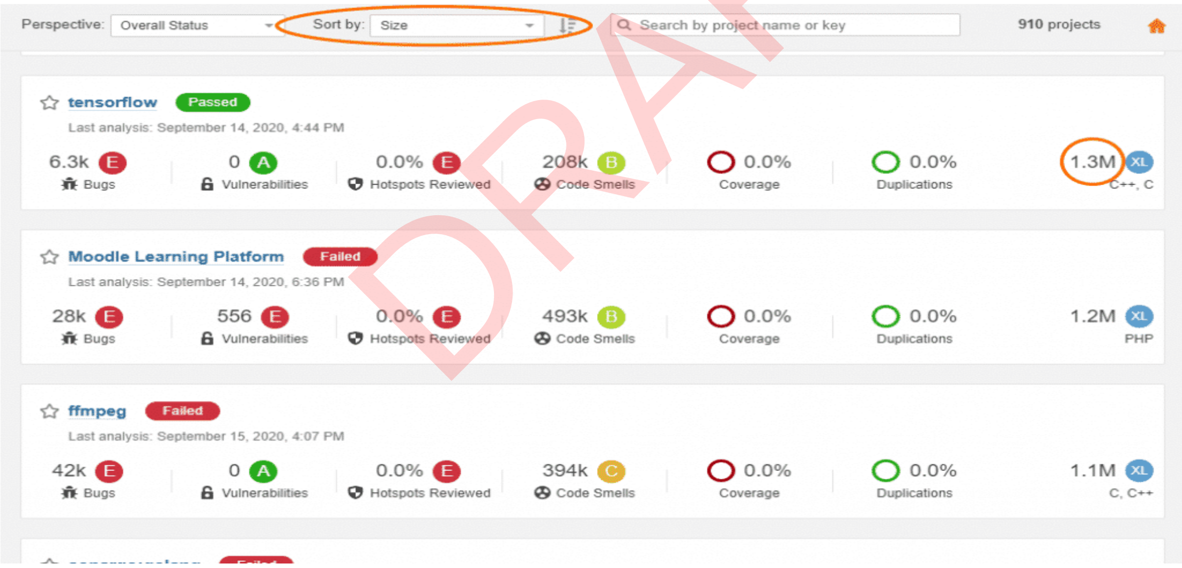
With code analysis multiple measurements ensure to meet defined quality goals of the system and reduce the potential risk of vulnerabilities and performance issues. With code analysis the following measurements can be measured: **[**SPT3TMS-10596 **]**

- **Bugs:** The functionality is buggy and is not working as intended. A refactoring of the respective code lines is mandatory.
- **Code smells:** The code quality is not sufficient and might impact the performance of the system. A refactoring of the respective code lines is mandatory.
- **Test coverage:** The percentage of code lines covered by unit tests. The test coverage must be at least 80%. A less value, increases the potential risk of bugs, code smells and vulnerabilities.
- **Code duplication:** The code quality is not sufficient and might impact the performance of the system. A refactoring of the respective code lines is mandatory.
- **Vulnerabilities:** The code lines are vulnerable and must be refactored to reduce the risk and harm to the system.

Common examples of technology vendors providing code analysis tools include SonarQube, Semmle, Semgrep, Coverity, DeepSource, Codacy, Checkmarx and SonarCloud.

The system must perform frequent code analysis before rolling out a specific release candidate and reject if the quality goal was missed.
[SPT3TMS-10593 **]**

The following figure illustrates an explanatory dashboard of a code analysis tool:



[SPT3TMS-10481 **]**

8.4.4 Code Testing

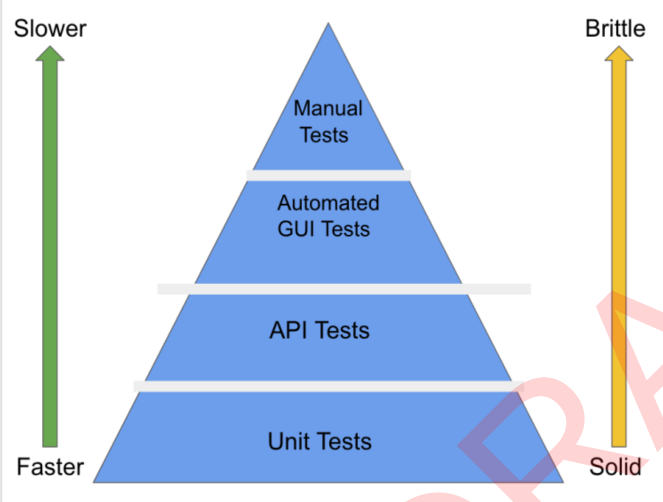
Due to the complexity of the system, testing is critical to avoid unnecessary bugs and downtimes which may affect the availability requirements. Cost and effort wise the system must follow the Test Pyramid model, which defines the following: **[**SPT3TMS-10600 **]**

- **Unit testing:** Maximum functionality tested by unit tests.
- **Less brittle test types:** More functionality tested by less brittle tests like unit and API tests.
- **Faster test types:** More functionality tested by faster-automated tests like unit and API tests.
- **Automation:** Most of the testing should be automated

[SPT3TMS-10599 **]**

With the Test Pyramid model, slower and brittle test types are restricted as they have a negative impact on the efficiency and cost of the overall. **[**SPT3TMS-10598 **]**

The following figure illustrates the Test Pyramid model:



[SPT3TMS-10484 **]**

Besides the Test Pyramid model, the system is subject of different test types which will focus on individual test scenarios. Quality goals are only reached in collaboration with all test types.
The following test types are subject of the system's test plan:

- **Unit tests:** Low-level code tests to verify individual methods and functions of the classes, components, or components. With increasing complexity, unit tests ensure the consistency of the feature and reduce unforeseen bugs.
- **Integration tests:** Mid-level code tests to verify the compatibility between services.
- **Functional tests:** Verifies the result of a specific functionality against business requirements.
- **End-to-End tests:** Verifies various user flows work as expected.
- **Acceptance tests:** Formal business requirements tests.
- **Performance tests:** Measures the performance of the system under normal load. These tests help to measure the reliability, speed, scalability, and responsiveness of the system.
- **Load tests:** Measures the performance of the system under high and low load. These tests help to determine how well the application can manage the expected loads.
- **Stress tests:** Measures the performance of the system under heavy load. These tests help to determine the application's breaking point.
- **Smoke tests:** Basic tests to verify if a certain build is stable to start further expensive tests.
- **Regression tests:** Verifies the stability and quality of the release.

[SPT3TMS-10601 **]**

8.4.5 Code Metrics

Will be added in later revision. **[**SPT3TMS-10607 **]**

8.4.6 Repository

A source code repository is central location where code and related resources for the application are stored and managed. Repositories play a crucial role in enabling collaboration, version control, and the management of source code.
A repository represents exactly one application. Merging between repositories is not planned and should be voided. **[**SPT3TMS-10606 **]**

8.4.7 Packaging

Packaging is the process of bundling a software application and all its associated files and components into a single, distributable package. This package includes the executable code, libraries, configuration files, and any other resources needed to install and run the application. **[**SPT3TMS-10603 **]**

Packages are used for the following purposes.

- **Distribution:** A single file to share.
- **Installation / Uninstallation:** The (un)installation is referred only to one file.
- **Dependency management:** All dependencies are managed at one place.
- **Version control:** The content of the application is identifiable.

The naming convention of packaging follows the following syntax:
<name>-<version>-<SNAPSHOT|RELEASE> **[**SPT3TMS-10608 **]**

The chapter 8.6.5 - Versioning will explain the concept of artifact versioning in detail. **[**SPT3TMS-10425 **]**

The following tables describes the expected packages:

<table>
<tr><th>Microservice:</th><th>Package:</th><th>Domain:</th></tr>
<tr><td>Topology Master Data Validation & Import</td><td>TOPOLOGY-MASTER-DATA-VALIDATION-IMPORT-X.X-SNAPSHOT|RELEASE</td><td rowspan="7">CMS</td></tr>
<tr><td>Topology various time horizons</td><td>TOPOLOGY-VARIOUS-TIME-HORIZONS-X.X-SNAPSHOT|RELEASE</td></tr>
<tr><td>Path mgmt. & processing</td><td>PATH-MGMT-PROCESSING-X.X- SNAPSHOT|RELEASE</td></tr>
<tr><td>Manual Path Construction</td><td>MANUAL-PATH-CONSTRUCTION-X.X- SNAPSHOT|RELEASE</td></tr>
<tr><td>Manual Path Conflict Detection</td><td>MANUAL-PATH-CONFLICT-X.X- SNAPSHOT|RELEASE</td></tr>
<tr><td>Automatic Path Construction</td><td>AUTOMATIC-PATH-CONSTRUCTION-X.X- SNAPSHOT|RELEASE</td></tr>
<tr><td>Capacity Plan Export</td><td>CAPACITY-PLAN-CREATION-X.X- SNAPSHOT|RELEASE</td></tr>
<tr><td>Topology Master Data Validation & Import</td><td>TOPOLOGY-MASTER-DATA-VALIDATION-IMPORT-X.X- SNAPSHOT|RELEASE</td><td rowspan="5">TMS</td></tr>
<tr><td>Daily Topology</td><td>DAILY-TOPOLOGY-X.X- SNAPSHOT|RELEASE</td></tr>
<tr><td>Capacity plan and decision processing</td><td>CAPACITY-PLAN-DECISION-PROCESSING-X.X-SNAPSHOT|RELEASE</td></tr>
<tr><td>Deviation Detection</td><td>DEVIATION-DETECTION-X.X- SNAPSHOT|RELEASE</td></tr>
<tr><td>Forecasting & RT-Conflict Detection</td><td>FOREVASTING-RT-CONFLICT-DETECTION-X.X-SNAPSHOT|RELEASE</td></tr>
</table>

Microservice:	Package:	Domain:
Automatic Conflict Solution	AUTOMATIC-CONFLICT-SOLUTION-X.X- SNAPSHOT| RELEASE	
Topology Master Data	TOPOLOGY-MASTER-DATA-X.X- SNAPSHOT| RELEASE	XFN
Topology various horizons	TOPOLOGY-VARIOUS-HORIZONS-X.X- SNAPSHOT| RELEASE	

Packaging should be automated using packaging tools such as Maven or Gradle. **[**SPT3TMS-10602 **]**

8.4.8 Artifactory

An Artifactory is a centralized place to store software artifacts more secure. With an Artifactory, the system can receive the exact same software artifact which was tested and approved. Furthermore, the system can access historical artifacts in the event of a rollback process. **[**SPT3TMS-10605 **]**

The chapter 8.4.7 - Packaging will explain the concept of packaging in detail. **[**SPT3TMS-10427 **]**

8.5 Domain Concepts

8.5.1 Data transfer

Data transfer is the foundation of any communication within (internal) and outside of the system (external) and is used to transfer data from one system to another. Therefore, the system relies on different communication methods as every use case is subject to specific communicational requirements. **[**SPT3TMS-10604 **]**

The following table illustrates the different data transfer methods:

Protocol:	**Type:**	**Rationale:**
HTTP	Synchronous	The communication to external actors is expected to be SOAP or REST based. (TAF/TAP TSI for example is based on SOAP).
Message Queue Protocol	Asynchronous	The communication between microservices within the respective domains is Message Queue based. The domain workflow is sequential – Sequential workflows are highly dependent on the previous and next tasks, covered by other microservices. In combination with Event-driven architecture and Message Queues, sequential workflow-tasks can be consequently managed, remain reliable and increase the efficiency at the same time.

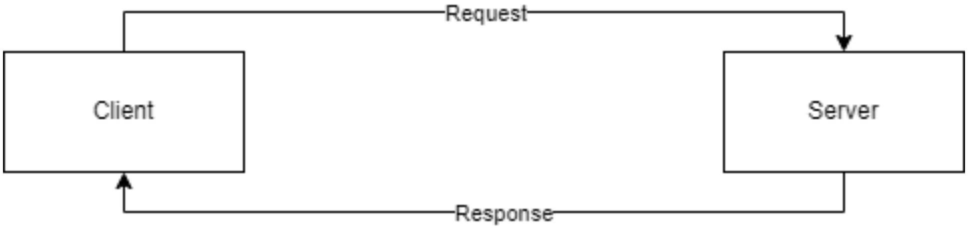
Data transfer must be secured to reduce any harm to the system. **[**SPT3TMS-10613 **]**

The chapter 8.8.3 - Encryption will introduce the topic of encryption in detail. **[**SPT3TMS-10426 **]**

8.5.1.1 REST / SOAP

REST and SOAP are based on HTTP and follow the request-response model. The model defines a communication pattern where a client sends a request to a server, and the server responds with a corresponding response. **[**SPT3TMS-10612 **]**

The following figure illustrates the request-response model of REST and SOAP:



[SPT3TMS-15344 **]**

8.5.1.2 Message Queues

Message queues are asynchronous service-to-service communications used in microservice architectures. In message queues, the sender and receiver of messages don't need to interact at the same time, messages are held in queue until the recipient retrieves them (Event-driven architecture). **[**SPT3TMS-10611 **]**

Message queues decouple heavyweight processes, to buffer or batch tasks, smooth spiky workloads and have a positive implication on the following:

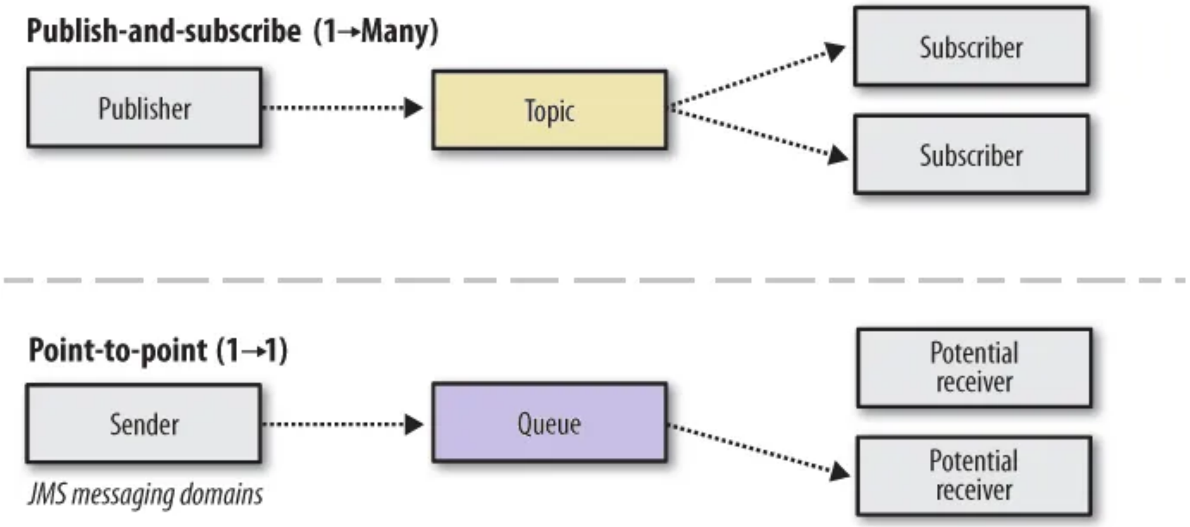
- **Better performance:** Message queues are Event-driven – there is no pending component waiting for another, which significantly optimize the data flow.
- **Increased reliability:** With messaging persistence, received messages are stored and protected against data loss. Even if the system experience downtime or unforeseen exceptions, the messages remain persistent. Message queues improve the fault tolerance.
- **Granular scalability:** With increasing workload, messages can be distributed by unlimited instances. Message queues itself can grow and shrink based on demand.
- **Simplified decoupling:** Message queues are an elegantly simple way to decouple distributed systems and every system can focus on its own domain logic.

Message queues provide the following types:

- **Publish/Subscribe:** A message is sent to multiple consumers (or subscribers) through a topic. The topic is the link between publisher and subscriber. The subscribers may or may not acknowledge the published message.
- **Point-to-Point:** Message(s) is sent from one application (producer/ sender) to another application (consumer/receiver) via a queue. There can be more than one consumer listening on a queue but only one of them will be able to receive the message. Hence, it is Point to Point or One to One.

[SPT3TMS-10610 **]**

The following figure illustrates the different message queue types:



[SPT3TMS-10489 **]**

8.5.1.2.1 Queue Events

8.5.1.2.1.1 Topology Import Events

This section explains the queue-based events of a Topology import. The Topology import is topic based to address multiple receivers. The whole process of Topology import consists of three sequential events. **[**SPT3TMS-10617 **]**

The following table describes the events, triggered doing a Topology import:

Order:	Event:	Type:	Rationale:
1	TOPOLOGY-RECEIVED	TOPIC	New topology data received and are now ready to be validated.
2	TOPOLOGY- VALIDATED	TOPIC	New topology data validated and are now ready to be imported
3	TOPOLOGY-IMPORTED	TOPIC	New topology data imported and are now ready to be processed.

[SPT3TMS-10616 **]**

The following table describes the Topology distribution:

Event	Publisher (Microservice):	Subscriber (Microservice):	Scope:
TOPOLOGY-RECEIVED	Topology Master Import	Topology Master Data Validation & Import	CMS
		Sectional Runtime Calucation	
	Topology Master Import	Topology Master Data Validation & Import	TMS
		Sectional Runtime Calucation	
TOPOLOGY-VALIDATED	Topology Master Data Validation & Import	Topology various time horizons	CMS
	Topology Master Data Validation & Import	Daily Topology	TMS
TOPOLOGY-IMPORTED	Topology various time horizons	Automatic Path Construction	CMS
		Manual Path Construction	
		Manual Path Conflict Detection	
		Path mgmt & processing	
	Daily Topology	Deviation Detection	TMS
		Forecasting	
		Real Time Conflict Detection	
		Automatic Conflict Solution	

Event	Publisher (Microservice):	Subscriber (Microservice):	Scope:
		Automatic Connection Management	
		Capacity Plan and Decisions Processing	

[SPT3TMS-13834 **]**

8.5.1.2.1.2 Capacity Plan processing Events

This section explains the queue-based events of a capacity plan processing. The capacity plan processing is topic and direct exchange based to address individual receivers. The whole process of capacity plan consists of three sequential events. **[**SPT3TMS-10614 **]**

The following table describes the capacity plan processing events:

Order:	Event:	Type:	Rationale:
1	CAPACITY-PLAN-RECEIVED	TOPIC	Capacity plan received and is now ready to be processed.
2	CAPACITY-PLAN-DECISIONS	DIRECT	Capacity plan processed and capacity decisions exists.
3	CAPACITY-PLAN-READY	DIRECT	Capacity plan is ready for production and must be transferred to TMS.

[SPT3TMS-10615 **]**

The following table describes the Capacity Plan distribution:

Event:	Publisher (Microservice):	Subscriber (Microservice):	Scope:
CAPACITY-PLAN-RECEIVED	Path mgmt & processing	HMI	CMS
		Automatic Path Construction	
CAPACITY-PLAN-DECISIONS	Manual Path Construction	Path mgmt & processing	
	Manual Path Conflict Detection		
	Automatic Path Construction		
CAPACITY-PLAN-READY	Path mgmt & processing	Capacity Plan Export	

[SPT3TMS-13835 **]**

8.5.1.2.1.3 Operational processing Events

The following table describes the capacity plan processing events: **[**SPT3TMS-10621 **]**

Order:	**Event:**	**Type:**	**Rationale:**
1	OPERATIONAL-PLAN-RECEIVED	DIRECT	Operational plan received and is now ready to be processed.
2	DEVIATION-DETECTED	DIRECT	Deviation detected and is now ready to detect conflicts.
3	CONFLICT-DETECTED	TOPIC	Conflicts detected and is now ready to apply solutions.
4	OPERATIONAL-PLAN-DECISION	DIRECT	Operational plan processed and operational decisions exists

[SPT3TMS-10622 **]**

The following table describes the Operational Plan distribution:

Event:	**Publisher (Microservice):**	**Subscriber (Microservice):**	**Scope:**
OPERATIONAL-PLAN-RECEIVED	Capacity plan and decisions processing	Deviation Detection	TMS
DEVIATION-DETECTED	Deviation Detection	Forecasting & RT Conflict Detection	
CONFLICT-DETECTED	Forecasting & RT Conflict Detection	HMI	
		Automatic Conflict solution	
OPERATIONAL-PLAN-DECISION	HMI	Capacity Plan and Decisions Processing	
	Automatic Conflict solution		

[SPT3TMS-13836 **]**

8.5.2 Persistence

Data persistence is another important aspect of the system. The system consists of permanent data which must be securely stored in a database to remain data consistency, especially in a multi scaled instance environment - a database remains the single point of truth. **[**SPT3TMS-10624 **]**

The following table described permanent data in the systems scope:

Data:	**Data type:**	**Rationale:**	**Scope:**
Capacity plans	MASTER-DATA	Capacity Plans are subject of different operational times and may be accessed several times on different timespans from different actions and events. To remain the data quality and consistency, a database for storing Capacity Plans is essentially important.	CMS
Operational plans	MASTER-DATA	Operational Plans may be accessed several times from different actions and events. To remain the data quality and consistency, a database for storing Operational Plans is essentially important.	TMS
Topology data	REFERENCE-DATA	Topology data are reference data synchronized once into the system. Whenever changes apply, the diff will only be synchronized to avoid massive data loading times. Topology data are stored for the whole life cycle and shared between many involving services. Storing topology data inside a database will ensure the integrity of such.	XFN

[SPT3TMS-10623 **]**

The system also considers different database types, as the complexity of master data and reference data vary.
Considering different database types implies in a more efficient process of storing and consuming data.
The following table describes the different database types:

Database Type:	**Rationale:**
Graph databases	Highly complex and dependent data structure are the foundation for Graph databases rather than a relational or NoSQL databases. Graph databases store information based on nodes, edges and properties and are most efficient to depict logical dependencies. The following functional requirements have implications with Graph databases: • **Topology:** highly complex and relates to many dependencies (to Topology segments, TrackEdges , etc). • **Capacity Plans:** Dependencies to other Capacity Plans. • **Operational Plans:** Dependencies to other Operational Plan.
Relational databases	Data replication (Database-Per-Service Pattern) in a microservice architecture is a common field for relational databases. The related data are mostly compromised and not complex anymore. Storing replicated and compromised data in a relational database will reduce the workload for transporting data from one microservice to another and may increase the availability and robustness of the system (loading local data vs a data

Database Type:	Rationale:
	cache).

[SPT3TMS-15357]

In respect to the above, the persistence concept applies different database types in combination with data replication to comply with separation of concern aspects.

The following table describes considered databases:

	Microservice:	Data:	Database type:	Scope:
Single point of truth	Path mgmt. & processing	Capacity plans	GRAPH	CMS
	Capacity plan and decision processing	Operational plans	GRAPH	TMS
	Topology Master Data	Topology data	GRAPH	XFN

[SPT3TMS-10619]

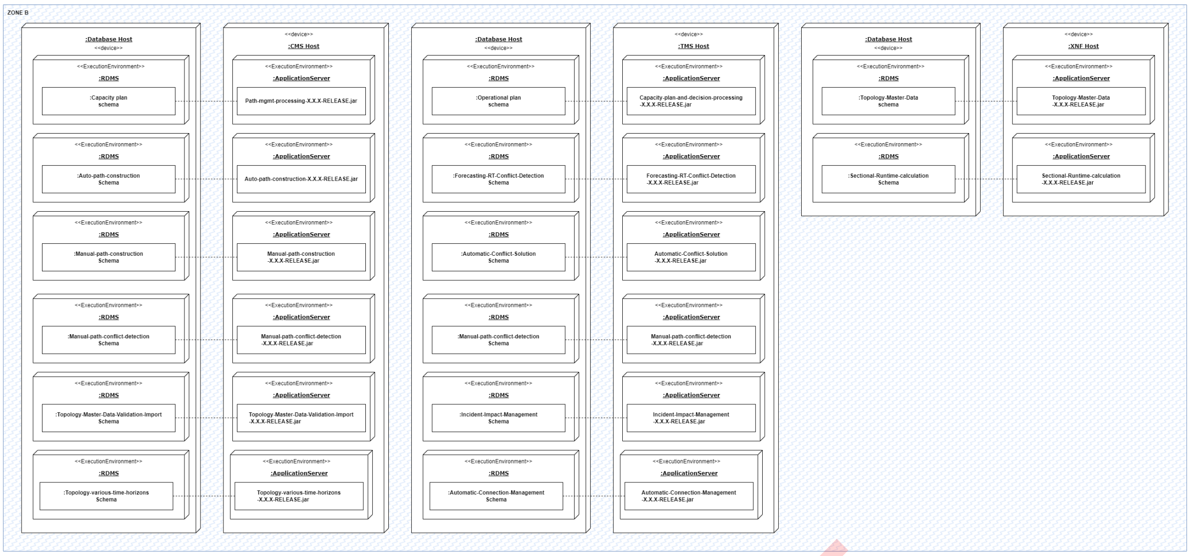
Furthermore, several services process master and reference data from above. For decoupling reasons, the following databases are considered as a performance buffer:

Attention: The single point of truth remains in the previous described table. The following databases are only considered as a working local copy at the dedicated service to comply with separation of concern aspects.

	Microservice:	Data:	Scope:
Replication	Topology Master Data Validation & Import	Topology data	CMS
	Topology various time horizons	Topology data	
	Manual Path Construction	Topology data	
	Manual Path Conflict Detection	Topology data	
	Automatic Path Construction	Topology data	
	Topology Master Data Validation & Import	Topology data	TMS
	Daily Topology	Topology data	
	Forecasting & RT-Conflict Detection	Topology data	
	Automatic Conflict Solution	Topology data	
	Automatic Connection Management	Topology data	
	Sectional Runtime calculation	Topology data	XFN

[SPT3TMS-10618]

The following figure illustrates the microservice to database assignment:



[SPT3TMS-10491 **]**

8.5.2.1 Database design

Especially in a microservice architecture, its highly important to consider the different approaches of database designs. The process of data in a multi-instance environment must be consequently streamlined to avoid side-effects and data inconsistencies. **[**SPT3TMS-10620 **]**

The following database designs exist:

- **Database per service:** Every microservice consists of its own database to comply with separation of concern aspects.
- **Shared database:** A single database shared between several microservices.

The shared database approach does not comply with separation of concern aspects and will not further be considered. **[**SPT3TMS-10630 **]**

8.5.2.2 Database clustering

Database clustering is a method to improve the performance, availability, and reliability of a database system. Clustering involves creating a cluster of database servers that work together to provide a more robust and scalable environment. **[**SPT3TMS-10629 **]**

The following aspects have been identified to have a positive impact to the system:

- **High availability:** By setting up multiple database servers in a cluster, if one server fails, the others can continue to handle requests and minimizing downtime.
- **Load Balancing:** To distribute workload evenly across multiple servers, preventing a single server from becoming a performance bottleneck.
- **Scalability:** As the system load increases, database clustering can add more servers to the cluster to meet performance requirements.
- **Data Replication:** Synchronisation between multiple servers enhance both data availability and fault tolerance.
- **Geo-Clustering:** Clusters can be geographically distributed, allowing for disaster recovery and improved performance in different regions.
- **Failover and Redundancy:** A failover mechanism automatically switches the workload to a healthy server in case of a server failure.

[SPT3TMS-10632 **]**

8.5.3 Concurrency

Concurrency is the process where multiple tasks or operations executing the exact same at the exact same time. If not considered, concurrency can lead to operational deficiencies.
The following sections will explain the possible actions to deal will concurrency. **[**SPT3TMS-10631 **]**

8.5.3.1 Concurrent tasks

This section will address the potential concurrent tasks within the system to better assess the behaviour and match the expectation. **[**SPT3TMS-10626 **]**

The following table will describe concurrent tasks:

Task:	**Behaviour:**	**Rationale:**
Two dispatchers assign a different path to the same train at the exact same time.	The database will sequentially process both tasks and the second operation will fail, due to the existing lock.	Concurrent tasks should be prevented by organizational or technical restrictions (technical access concepts).
Two heavy parallel tasks running at the exact same time. E.g., to analyse the conflict potential of the dispatching area.	There is no implication on the quality of data. Both tasks will finish, and the result will be the same.	Concurrent tasks with the same result should be avoided at any time, as it impacts the overall efficiency. This is usually as sign to scale services vertically instead of horizontally to avoid concurrent tasks.

[SPT3TMS-10625 **]**

8.5.3.2 Database Concurrency

Database concurrency is a unique characteristic enabling two or more applications to retrieve / update data from the database at the same time without affecting data integrity.
Database management systems are designed to apply different concurrency control techniques to overcome concurrency challenges, such as: **[**SPT3TMS-10628 **]**

- **Lock-based protocols:** Every transaction requires a lock before executing write operations to prevent data integrity issues. A lock merely denotes the type of operations permitted on a particular data object.
- **Timestamp-based protocols:** Concurrent transactions execute in an ordered manner based on the timestamps assigned to them. Older transactions are given priority while deciding the execution order of concurrent transactions in this method.
- **Multi-version concurrency control (MVCC):** MVCC is typically used with other concurrency control mechanisms for better results, such as multi-version timestamp ordering and multi-version two-phase locking.

[SPT3TMS-10627 **]**

Common examples of technology vendors providing concurrency control techniques include Oracle, Postgres and MSQL.

8.5.4 Caching

Caching enables the system to perform its tasks more efficiently. Frequently used data must be cached to reduce latencies and throughput. **[**SPT3TMS-10633 **]**

According to the different technical scopes, the system must apply the different cache types:

- **Frontend caching:** Resources such as stylesheets, images etc.
- **Backend caching:** Data cache within microservices **-** frequent database operations are also subject of backend caching.

[SPT3TMS-10637 **]**

8.5.5 Rights & Roles

Rights and Roles enable the system to limit the exposure of data and the execution of restricted actions to a certain user group with respective privileges. Rights are directly assigned to roles and describe the ability of the role, where roles are assigned to user to group list of allowed operations.
The system heavily relies on rights and roles otherwise the system will reject any operation. Any existing user in scope of the system is assigned to a specific role. If the user is not assigned to a role, the system treats the operation as misconfiguration and rejects the operation. Administrators are responsible to maintain user roles. **[**SPT3TMS-10636 **]**

The system is distributed in the following rights:

- **Read:** The assigned role is privileged to read the data only
- **Write:** The assigned role is privileged to execute operations which will add or modify data.
- **Delete:** The assigned role is privileged to delete data.

The system is distributed in the following roles:

- **Dispatcher:** An end user responsible for dispatching a certain sector or area. This role consists of read, write, and delete rights.
- **Observer:** An end user responsible for observing the current situation of a certain sector or area. This role consists of read rights only.

A user can only maintain one specific role.
Rights and roles are valid within the whole context of the system. Rights and roles are not subject of geographical segmentation.
[SPT3TMS-10635 **]**

8.6 Operational Concepts

The following concepts will describe important operational aspects. **[**SPT3TMS-10634 **]**

8.6.1 Stages

Stages are server environments serving as a quality gate to the system. Each stage serves a distinct purpose during the development and operation lifecycle. **[**SPT3TMS-10641 **]**

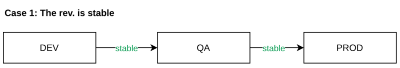
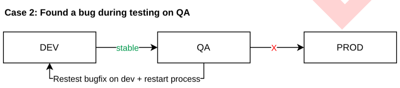
The following stages exist:

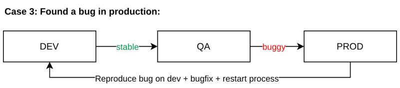
1. **DEV:** Working environment for tests during the development lifecycle. This stage might not represent the latest version of the system and serves only for test purposes.

1. **QA:** Test environment to concretize a specific version / release for the production stage. This stage should be close to production stage without real production data.
2. **PROD:** Real-Operation environment for the end user.

The deployment process to stages is sequential: From DEV to QA to PROD.
Stages should be designed identically as the PROD stage to avoid side effects due to different stage configurations. Stages can be downsized (in number of instances and allocated server resources, for e.g., RAM, CPU, etc) to reduce operational costs. **[**SPT3TMS-10640 **]**

The following table showcases potential use cases during a deployment to different stages:

<table>
<tr><th>Case:</th><th>Behaviour:</th><th>Action:</th></tr>
<tr><td rowspan="2"></td><td colspan="2">
</td></tr>
<tr><td>The rev. is stable and will pass the quality gate. The rev. was successfully deployed on PROD.</td><td>No actions required.</td></tr>
<tr><td rowspan="2">CASE 2</td><td colspan="2">
</td></tr>
<tr><td>The rev. is unstable as a bug was found on QA during testing.</td><td>The deployment to PROD will be omitted, as the quality of the rev. is not sufficient. The bug must be fixed and retested first. Once the bug is fixed, and the tests are successful on QA, a deployment to PROD will be initiated.</td></tr>
</table>

Case:	Behaviour:	Action:
		
	The rev. is unstable as a bug was found on PROD. The bug wasn't found during testing on QA.	Reproduction-work is being done on DEV. Once the bug was reproduced and the tests are successful on QA, a deployment to PROD will be initiated.

[SPT3TMS-10490 **]**

8.6.2 Logging

Logging is the process of recording events, activities, or data over time. With logging the actual behaviour of the system become transparent as logging can be used to analyse a concrete process for debugging purposes, performance tweaking or security related tasks. **[**SPT3TMS-10639 **]**

Log levels optimize the information flow in certain situations and serve as a filter and clustering of events of the same type.
The following log levels are supported:

- **TRACE**: Full visibility of every log message – there is no filtering at all.
- **DEBUG:** Log messages needed to diagnosis and troubleshooting.
- **INFO:** Log messages to document the normal process flow.
- **WARN:** Log messages indicating unexpected states and pointing to potential issues.
- **ERROR:** Log messages indicating an error, preventing the execution of a process.
- **FATAL:** Log messages indicating a fatal error of preventing critical functionality.

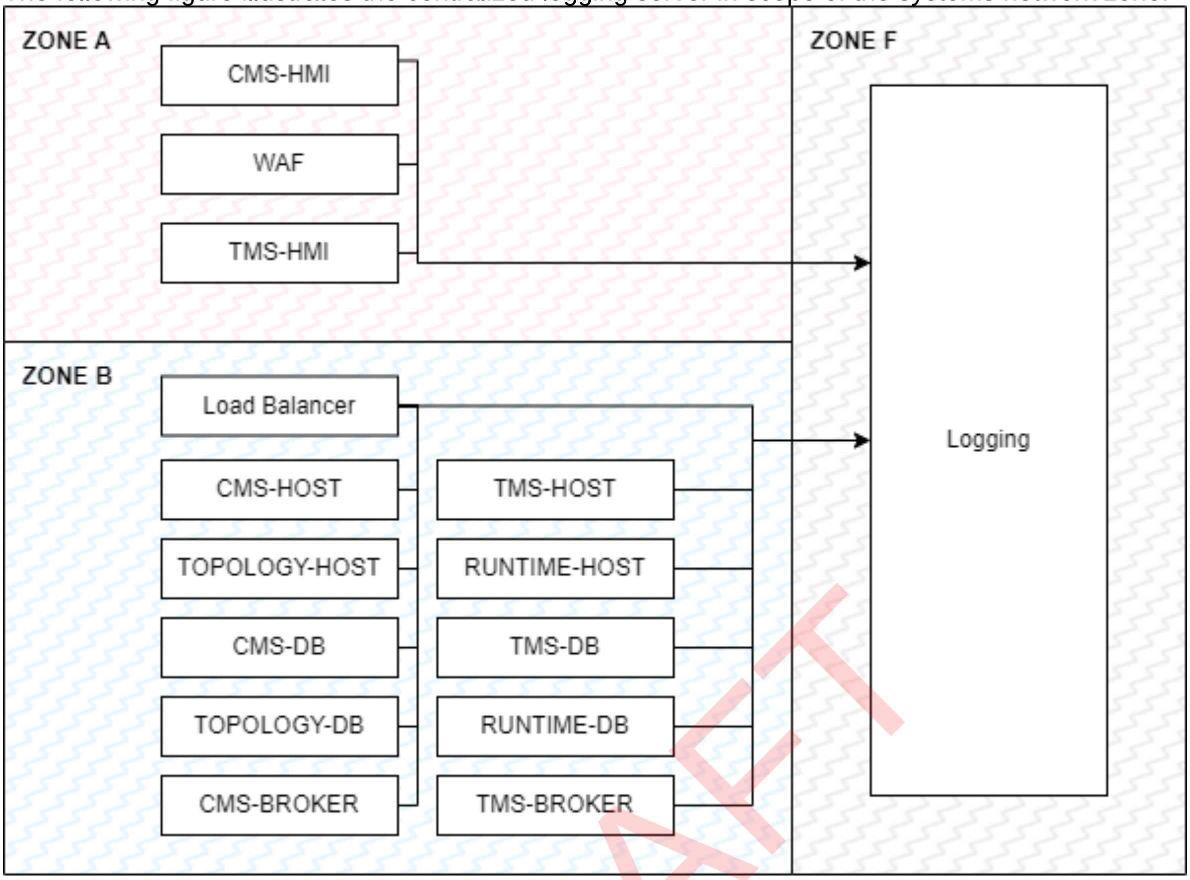
Logging is an integral part of every server, hosts, and instance and furthermore occurs on every system component and instance. The distribution of log messages across the systems component and instances becomes unmanageable – For the reason being, a centralized logging server will consolidate log messages and protects log messages from data loss. Consolidated log messages are visualized via a log monitoring tool, described in chapter 8.6.3 - Monitoring .
The logging server is located in network ZONE F, to mitigate risk and harm to log messages at the event of a security attack. Log messages are sensitive and requires extra security measures.
[SPT3TMS-10638 **]**

Access to log messages must be restricted to the following user groups:

User group:	Access level:	Rationale:
Server administrators	FULL	Monitor infrastructure issues
Security administrators	FULL	Monitor security related issues
System administrators	FULL	Monitor the overall systems health
Developers	PARTLY	Monitor diagnostic related messages

[SPT3TMS-13895 **]**

The following figure illustrates the centralized logging server in scope of the systems network zone:



[SPT3TMS-10493 **]**

The following table describes the different log types per component and zone:

Component:	**Type:**	**Zone:**
CMS-HMI	GUI-LOG	ZONE A
TMS-HMI	GUI-LOG	
WAF	ACCESS-LOG	
Load Balancer	LOAD LOG	ZONE B
CMS-HOST	RUNTIME-LOG	
TMS HOST	RUNTIME-LOG	
TOPOLOGY-HOST	RUNTIME-LOG	
RUNTIME-HOST	RUNTIME-LOG	
CMS-DB	DATABASE-LOG	
TMS-DB	DATABASE-LOG	
TOPOLOGY-DB	DATABASE-LOG	

Component:	Type:	Zone:
RUNTIME-DB	DATABASE-LOG	
CMS-BROKER	MESSAGE-QUEUE-LOG	
TMS-BROKER	MESSAGE-QUEUE-LOG	

[SPT3TMS-10647 **]**

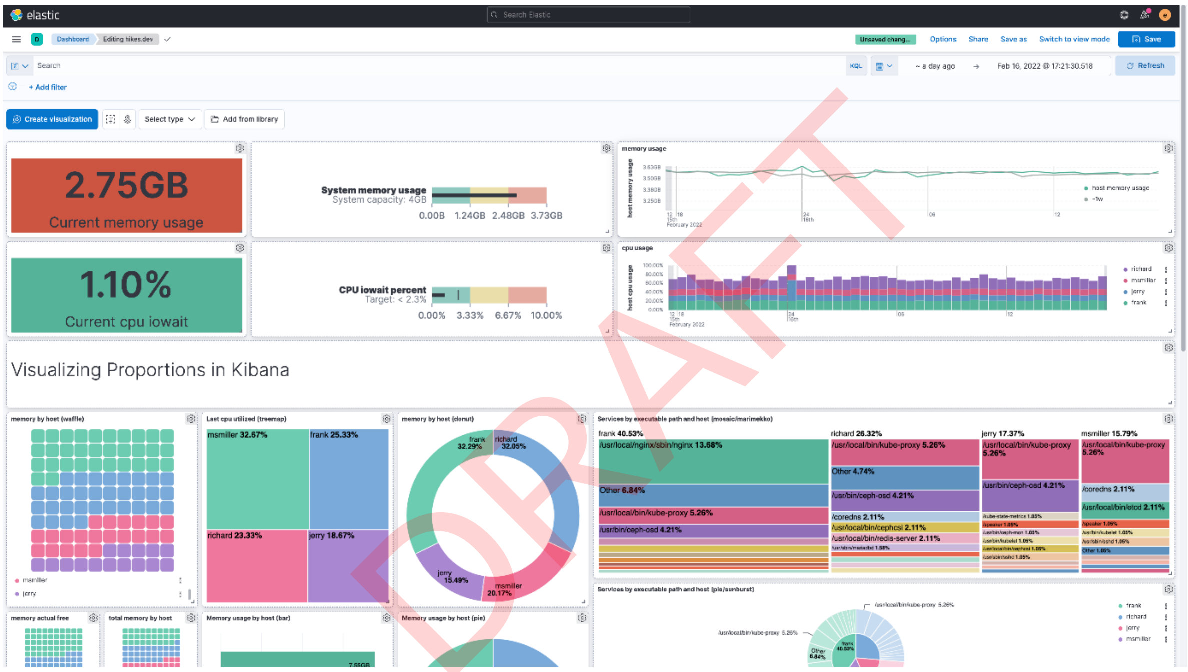
8.6.3 Monitoring

The system relies on a monitoring solution for data visualization and exploration to be used for log and time-series analytics, application monitoring, and operational intelligence use cases. The monitoring solution must offer the following features: **[**SPT3TMS-10646 **]**

- **Visualization:** Display of metrics commonly used as vertical bar chart, horizontal bar chart, pie chart, line graph, heat map etc.
- **Filter & Search:** reading log files for all available instances and filter and search for different log levels, search terms or any other conditions.
- **Reports:** Report function including an export functionality to compare different states.

Common examples of technology vendors providing monitoring solutions include Kibana, Grafana, Looker, Tableau, Knowi, Charted, Highcharts and OpenSearch Dashboards. **[**SPT3TMS-10649 **]**

The following figure illustrates an explanatory dashboard of a monitoring solution:

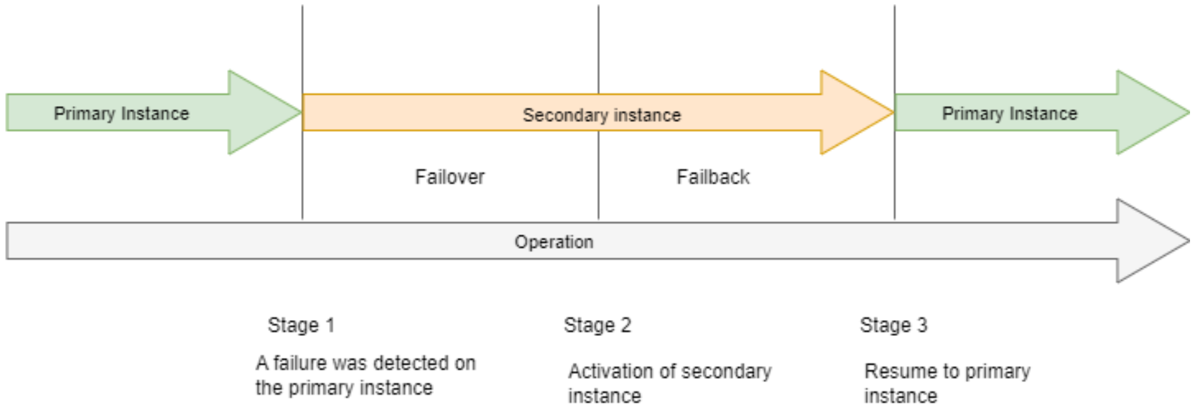


[SPT3TMS-10492 **]**

8.6.4 Disaster Recovery-Plan (DR)

The system consists of critical components; therefore, any kind of downtime must be prevented anytime. The system must ensure the operation in any failure scenario and must intervene fully automatic to maintain uptime. **[**SPT3TMS-10648 **]**

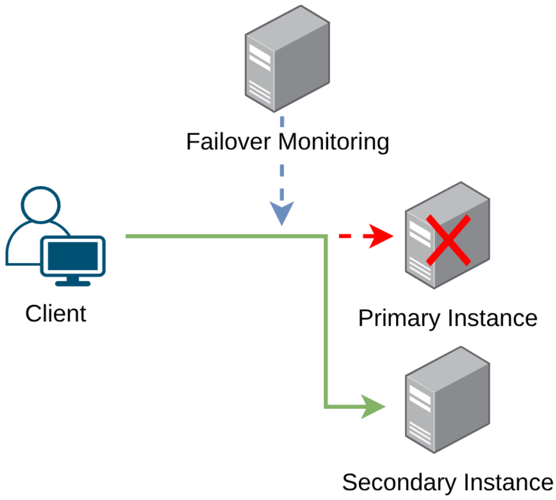
The following figure illustrates the disaster recovery timeline:



[SPT3TMS-10497 **]**

8.6.4.1 Failover Strategy

A failover strategy is the process of replacing a failing computing resource with a healthy resource. This must be done automatically to maintain the uptime of all services. Therefore, the system must provide a primary and secondary instance of the system (Hot standby). **[**SPT3TMS-10643 **]**



Whenever an unhealthy primary instance was detected, the failover strategy must intervene. Thereby, the primary unhealthy instance will be replaced by a healthy secondary instance.

Against the scalability concept the secondary instance is not designed to distribute workload. To switch between a primary and secondary instance, the application/data state must be up-to-date, even if a failure is not expected in this moment.

[SPT3TMS-10501 **]**

The failover strategy must be applied to all software / hardware / operational components, for criticality ranking: MAJOR and CRIITICAL. To mitigate resource failures the secondary instance must be hosted within a multi-region infrastructure. Even if a certain region of the infrastructure is not available, the failover strategy must be able to compensate the harm by consuming instances within another working region. **[**SPT3TMS-10642 **]**

The following table illustrates the criticality of the respective components of the system:

Component:	Criticality:	Rationale:	Scope:
Topology Master Data Validation & Import	MAJOR	The impact on those components is relative to the point in time of the failure.	CMS
Topology various time horizons	MAJOR		
Path mgmt. & processing	MAJOR		
HMI	MAJOR		
Manual Path Construction	MAJOR		
Manual Path Conflict Detection	MAJOR		
Automatic Path Construction	MAJOR		
Capacity Plan Export	MAJOR		
Topology Master Data Validation & Import	MAJOR	The impact on those components is relative to the point in time of the failure.	TMS
Daily Topology	MAJOR		
Capacity plan and decision processing	CRITICAL	Failures to those components have directly impact to railway operations.	
Deviation Detection	CRITICAL		
Forecasting & RT-Conflict Detection	CRITICAL		
HMI	CRITICAL		
Automatic Conflict Solution	CRITICAL		
Topology Master Data	MAJOR	The impact on those components is relative to the point in time of the failure.	XFN
Topology various horizons	MAJOR		
Sectional Runtime calculation	CRITICAL	Failures to those components have directly impact to railway operations.	

In a hot standby setup, data sharing is a critical aspect to ensure that the standby system remains synchronized with the primary system and can seamlessly take over in the event of a failure.
The following table illustrates critical aspect to realize a hot standby setup:

Aspect:	Rationale:
Database Replication	Replication enables data from one database server (known as a source) to be copied to one or more database servers (known as replicas). Replication is asynchronous by default.
Traffic Mirroring	Any kind of traffic are mirrored by a load balancer and distributed over all primary and secondary instances. Thereby the application state remains synchronized.

Aspect:	Rationale:
Security	Security measures protect data during replication, such as encrypting data in transit.
Test Failover Scenarios	Conduction of regular failover tests to verify an effectively take over when needed. This testing is essential to ensure minimal downtime and data integrity during actual failover events.
Regular Maintenance	The primary and secondary system must be patched and upgraded on a regular base to remain compatible and secure.

[SPT3TMS-10645 **]**

8.6.4.2 Failback Strategy

In case of an applied failover strategy, the system must make sure to return to its normal operation using the primary instances. Once the primary instance is operative again, the failback Strategy is the business continuity mechanism to synchronize the system to the latest operative state. **[**SPT3TMS-10644 **]**

The failback strategy differentiates between the following failure scenarios:

Failure scenario:	**Rationale:**
A failure of primary instance with a working and active secondary instance.	The system is operative by the secondary instance.
A total failure of all instances including primary and secondary.	The system is not operative and offline. There is no backup instance available anymore.

[SPT3TMS-10650 **]**

8.6.4.2.1 Failback from secondary instance

During this failure scenario the primary instance is offline, and the failover strategy activated the switch to the secondary instance - the system is fully operative. **[**SPT3TMS-10652 **]**

Order:	**Action:**	**Rationale:**
1	Assessment	The assessment should include hardware, software, and data integrity checks.
2	Data Synchronization	Any changes which were made during the failover are synchronized between the primary and secondary environment. This may involve replicating data, logs, or configurations.
3	Failback Execution	Reversing changes made during the failover (e.g., reconfiguring DNS, load balancers, or routing). Bringing the primary environment back online. Redirecting traffic and services to the primary environment. Monitoring and verifying the health and performance of the primary environment after failback.

[SPT3TMS-10654 **]**

8.6.4.2.2 Failback from total failure

During this failure scenario all instances (primary and secondary) are offline, and the failover strategy runs out of available instances - the system is not operative anymore. **[**SPT3TMS-10651 **]**

The implications of this failure scenario are the following:

- **Data loss:** The data transfer during the failure from external actors are not processed and therefore missing. This may include train states, capacity requests, assets, and topology data.
- **Data inaccuracy:** Caused by data loss, the system is not capable to include accurate data which results in inaccuracy data states (train position, forecasts, runtime calculation, etc).

For the reason being this scenario must be prevented at any time.
The following procedure determines the mandatory tasks to return to a normal operation:

Order:	**Action:**	**Rationale:**
1		The assessment should include hardware, software, and data integrity checks.

Order:	Action:	Rationale:
	Assessment	
2	Reactivation of secondary instance	Activating all system components (Microservice instances, queues, databases, etc). **[Attention]:** **The initiation requires multiple days.** **Due to the massive amount of data required to be loaded into the in-memory system cache. Furthermore, the system will lack of consistent data even the system is technically reactivated – This is related to train states, capacity plans, operational plans, transferred to the system only if the train fulfils a moving action. Otherwise, the system will not receive any up-to-date information and the system remains in an out-dated state.**
3	Reactivation of primary instance	Activating all system components (Microservice instances, queues, databases, etc).
4	Data Synchronization	Any changes which were made during the failover are synchronized between the primary and secondary environment. This may involve replicating data, logs, or configurations.
5	Failback Execution	Reversing changes made during the failover (e.g., reconfiguring DNS, load balancers, or routing). Bringing the primary environment back online. Redirecting traffic and services to the primary environment. Monitoring and verifying the health and performance of the primary environment after failback.

[SPT3TMS-10653 **]**

8.6.5 Versioning

8.6.5.1 Request Versioning

The system communicates with various external actor which may depend on a specific version of a certain message/interface. Therefore, the system must be able to support various versions of the same message / interface to support backward and forward compatibility. **[**SPT3TMS-10660 **]**

According to the various architectures of the system, versioning must be implemented as follows:

- **RESTful API's**: via URI versioning
- **SOAP API's**: via WSDL Schema / Namespace versioning
- **Message Queues**: via meta data versioning

[SPT3TMS-10659 **]**

8.6.5.2 Database Versioning

It is important to introduce versioning on database level to remain reliability. Only with database versioning, the system is capable to fulfil a rollback at the event of a primary data failure. **[**SPT3TMS-10662 **]**

Therefore, database versioning should follow these steps:

- **Migration script:** Modifications are introduced as migration scripts only.
- **Automation:** Tools to handle migration scripts to execute a rollback

Common examples of technology vendors providing database versioning include Flyway and Liquibase. **[**SPT3TMS-10661 **]**

8.6.5.3 Artifact Versioning

With Artifact versioning the system is capable to roll back to a specific snapshot / release of the software's artifact if mandatory. **[**SPT3TMS-10656 **]**

Artifacts should follow the following naming conventions:

1. **Name:** A unique name, describing the artifact.
2. **Index:** Unique identifier to express the current version of the artifact
3. **Suffix:** The artifact is suffixed as SNAPSHOT, if the artifact is under testing. If the artifact becomes concrete for production, the artefact should be suffixed as RELEASE.

Example:
`DEVIATION-DETECTION-1.0-SNAPSHOT` or `DEVIATION-DETECTION-1.0-RELEASE`
[SPT3TMS-10655 **]**

The chapter 8.4.8 - Artifactory will explain the concept of artifact storing in detail. **[**SPT3TMS-16218 **]**

8.6.6 Back-ups

The purpose of back-ups is to create a copy of data that can be recovered in the event of a primary data failure. Primary data failures can be the result of hardware or software failure, data corruption, or a human-caused event, such as a malicious attack (virus or malware), or accidental deletion of data.
Besides the positive impacts of back-ups, there are some important aspects to consider. Frequent back-ups increase the load of the system (back-up server) and implies in increasingly operation costs from time to time. As a trade-off between data security and efficiency, different approaches are in place to balance operation costs and data security. **[**SPT3TMS-10658 **]**

The following back-up types exist:

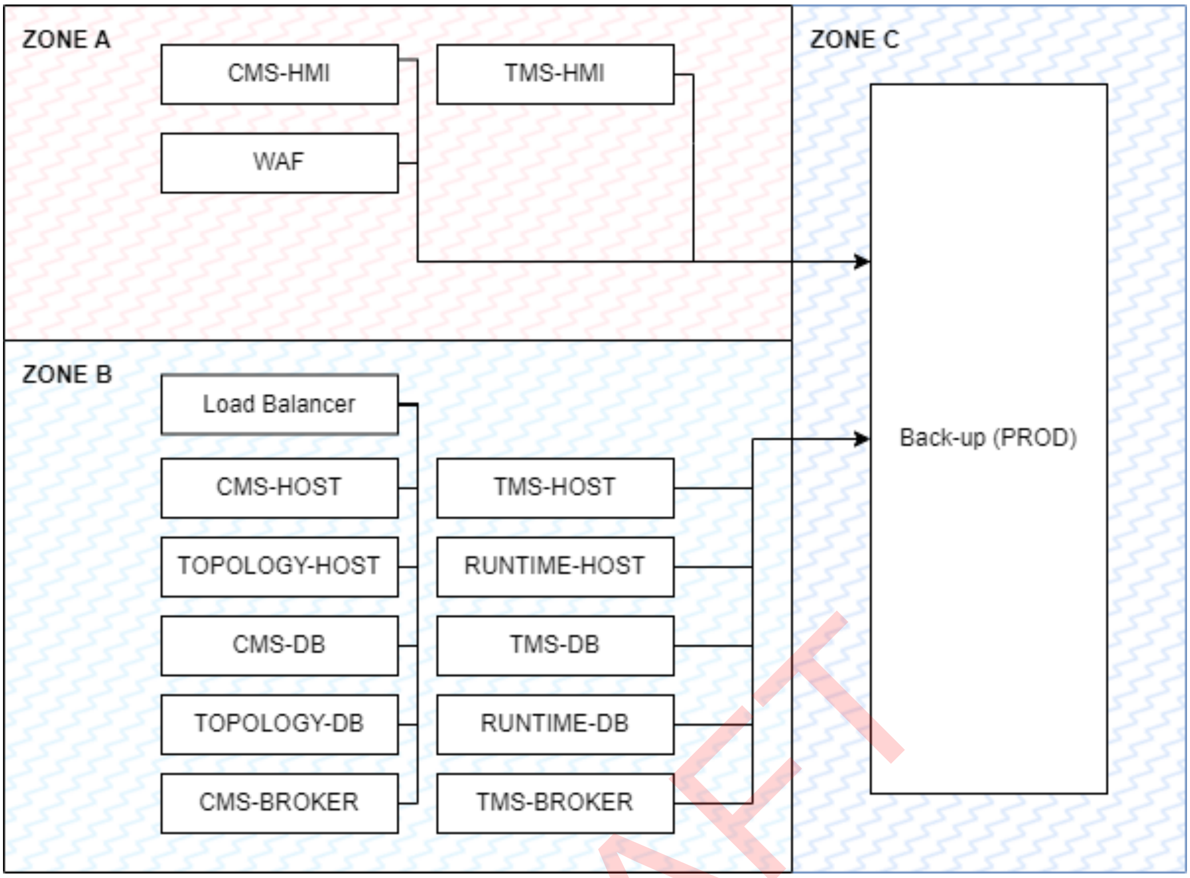
- **Complete back-ups:** Contains the whole content and does not rely on other back-ups.
- **Differential back-ups:** Contains the diff + all sub back-ups since the complete back-up.
- **Incremental back-ups:** Contains only the diff between the previous back-up based on the initial complete back-up.

The following table determines the level of back-ups incl. the back-up type:

System:	Occurrence:	Type	Rationale:
Database	Frequently	Complete back-up	The importance of data is driven by the reliability of the system. If the system must be reverted to a specific state and the data are lost, the system will experience a downtime for several days. A complete back-up strategy can speed up and reduce the complexity if the system must be reverted or a complete back-up must be applied.
Application & Server configurations	Frequently during development phase, moderate during operation phase	Incremental back-ups	A loss in application code or server configuration will have a serious implication to the operation of the system. A loss must be prevented at any time. As the frequency of changes is expected to be high only during development phase, incremental back-ups can reduce operational costs.
Temporary operational files	Frequently	Differential back-up	During the operation the system relies on temporary data such as caches, etc. A loss of temporary files implicates in a longer start-up time if the application would be restarted. A differential back-up would balance the efficiency of applying back-ups but saving resources at the same time.

Back-up storage locations are another important aspect to consider. Local back-ups on the respective environment are subject of data loss once the environment gets corrupted, shutdown, etc. For the reason being, the system follows a strict centralized back-up strategy, where any local back-up must be stored on a centralized back-up server. The back-up server must be protected against data-loss by RAID-5 and is located in ZONE C. **[**SPT3TMS-10657 **]**

The following figure illustrates the back-up server in the respective zone:



[SPT3TMS-10500 **]**

Every back-up must be protected to prevent data modification and data theft.

Data encryption will be covered by chapter 8.8.3.4 - Back-up encryption . **[**SPT3TMS-16216 **]**

8.7 Automation

Automation will simplify the process of reoccurring tasks and has a positive impact on the project efficiency and cost ratio. The project follows the approach, if reoccurring tasks can be automated with no significant effort, then the process shall be automated. **[**SPT3TMS-10663 **]**

The following reoccurring processes were identified and shall be automated:

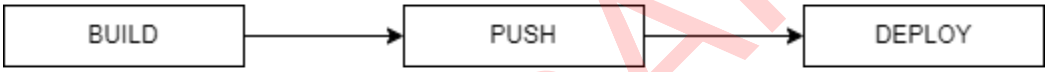
- **Unit tests, Integration tests, Functional tests:** Those test types shall be executed once a change to the system was populated. Driven by the high reoccurrence, automation can significantly reduce the need for manpower executing the tests.
- **Performance tests:** The duration of performance tests is usually expected to be more time consuming which can be parallelized (e.g., nightly execution) to receive the results the following day. Automation will significantly reduce the manpower for executing the tests.
- **CI/CD:** Software builds and deployments are frequent processes with an easy automation grade. However, a proper deployment concept must verify which stage shall be automated and which stage remains a manual process.

[SPT3TMS-10667 **]**

8.7.1 CI/CD

Continuous Integration (CI) and Continuous Delivery (CD) refers to a streamlined process of building, testing, and delivering the application. With CI/CD the development cycle can be fully automated (via pipelines) which has a positive impact on the velocity and reduces potential human errors.
[SPT3TMS-10666 **]**

The following figure illustrates the CI/CD process:



[SPT3TMS-10499 **]**

1. **BUILD:** Testing and packaging the application, the result is the application artifact.
2. **PUSH:** Uploading the artifact to the Artifactory.
3. **DEPLOY:** Deploying the artifact to the corresponding environment

[SPT3TMS-10665 **]**

8.7.2 Pipelines

Pipelines are used to automate occurring tasks within the system. Pipelines reduce the potential risk of human error. Pipeline can be manually based or fully automated without human intervention. However, the development of pipelines can be complex – before investing into new pipelines the impact must be assessed properly. **[**SPT3TMS-10664 **]**

The following table describes the available pipelines, considered to have a positive impact:

Pipeline:	Occurrence	Rationale:
BUILD	On every code commit / push.	Includes the execution of unit and integration test, software packaging and upload to an Artifactory.
DEPLOY	On merges to a specific branch (dev, release, master) Production environments are manual triggered only.	Includes the artifact download from an Artifactory, uninstalling the current running version, migrating introduced changes (e.g., database migration scripts) and installing the specified artifact.
TESTS	Nightly	Includes stress tests, load tests and performance tests.

[SPT3TMS-10670 **]**

8.8 Security Concepts

This section describes security concepts to mitigate the risk and harm to the system. **[**SPT3TMS-10669 **]**

All security concepts commit to the following specifications:

- Europes Rail System Pillar - Secure Communication Specification - DRAFT_01_2024
- Europes Rail System Pillar - Secure Component Specification - DRAFT_01_2024
- Europes Rail System Pillar - Security Program Requirements - DRAFT_01_2024
- Europes Rail System Pillar - Shared Security Services Specification - DRAFT_01_2024

[SPT3TMS-15373 **]**

8.8.1 Password Policy

The system must be designed to avoid the need for passwords and rely only on tokens. However, in certain situations (e.g., Keystores can additionally be secured via passwords) the use of passwords is mandatory. **[**SPT3TMS-10668 **]**

Therefore, the following password policy is recommended to be applied:

- It should consist of at least 8 characters.
- It should not contain any of project or personal information.
- It must be unique from the previously used passwords.
- It should not contain any word spelled completely.
- It should contain different types of characters, including uppercase and lowercase letters, numbers, and special characters.
- It should be valid for a certain time only and must be changed frequently.

Any password must be stored in a wallet (KeePass), a plain password within a config/profile file is strictly prohibited. **[**SPT3TMS-10672 **]**

8.8.2 Data states

Understanding the characteristics and differences between data states is important to handle sensitive information more securely. Data states consists of three possibilities: at rest, in motion and in use. The following measurements should make sure to prevent any data theft, data loss and data corruption. **[**SPT3TMS-10671 **]**

8.8.2.1 Data in Use

Data in Use refers to active information under constant changes stored physically on the server including databases, short term buckets, etc. To protect the data in use, controls must be put in place "before" accessing the content. This can be realized through: **[**SPT3TMS-10674 **]**

- **Identity management tools**: To check the authenticity (verifies that someone or something is who they say they are) and there has been no identity theft. In these cases, it is increasingly important to protect access to the data through a two-factor authentication.
- **Conditional Access or Role Based Access Control (RBAC) tools**: Allow access to data, based on the user's role or other parameters such as IP, location, etc.
- **Information Rights Management (IRM):** To limit the possibilities and actions (editing, printing, etc) once the data were delivered.
- **Data encryption**: Any kind of data must be encrypted and can only be decrypted if the above were met. In the unexpected event of a data theft, the data become useless.

[SPT3TMS-10673 **]**

8.8.2.2 Data at Rest

Data at Rest refers to any inactive information stored physically on the server including databases, long term buckets, external HDD's, etc. **[**SPT3TMS-10680 **]**

The highest priority of this state is to secure long-term data. This can be realized through:

- **Full disk encryption:** Protecting the system for undesirable access (for example by the provider) as encrypted disks makes the data useless if not decrypted. This will also prevent modifications on the host system (e.g., installing unintended software or manipulating backups, etc)
- **Database encryption:** Data manipulation / corruption can be limited if the data are encrypted. The system must be able to establish a encrypted database connection.

[SPT3TMS-10679 **]**

8.8.2.3 Data in Motion

Data in Motion refers to information that is being transferred from or to another system via interfaces, queues, etc. **[**SPT3TMS-10682 **]**

During this state of data, the highest priority is to ensure that data are not sniffed or modified during the transit. This can be realized through:

- **SSL encryption:** Any provided endpoint is secure via SSL.
- **Queue encryption:** Any provided message within the queue is encrypted.

[SPT3TMS-10681 **]**

The chapter 8.8.3 - Encryption will explain the concept of encryption in detail. **[**SPT3TMS-16195 **]**

8.8.3 Encryption

Encryption is used to protect data from being stolen, changed, or compromised and works by scrambling data into a secret code that can only be unlocked with a unique digital key. **[**SPT3TMS-10676 **]**

The following encryption types exists:

- **Symmetric Encryption:** [SECURE] Involves in using a single key to encrypt and decrypt data.
- **Asymmetric Encryption:** [SECURE] Involves in using two keys to encrypt and decrypt data.
- **Hashing:** [IN-SECURE] Involves in a fixed-length value based mathematical formular

Hashing is well known to be unsafe as the hash becomes guessable with brute-forcing. For the reason, the system should always use (a)symmetric encryption only.
Encryption is considerably important for databases, communication, and back-ups. The following chapters explain how to realize encryption at a detailed level. **[**SPT3TMS-10675 **]**

8.8.3.1 Database encryption

Database encryption is another important step to build a more reliable system. Unauthorized user with database access can be limited by reading, modifying, or deleting sensitive data stored in the database. **[**SPT3TMS-10678 **]**

Database encryption should be realized in two steps:

- **Data Encryption at Rest:** physical stored data are encrypted and reading becomes impossible. Data are decrypted only if the correct encryption key was provided. This step is usually done by the DBMS.
- **Data Encryption In Transit:** The connection is encrypted and can only be established if the matching encryption key was provided. This will unauthorized user from modifying and deleting sensitive data.

Database encryption relies on symmetric encryption and requires a single key file. **[**SPT3TMS-10677 **]**

8.8.3.2 SSL encryption

SSL encryption ensures a secure HTTP connection between two parties (client/server) and protects the system against man-in-the-middle attacks (MITM) and sniffing. A secure connection can be established by asymetric encyrption where a public and proviate key must match (also called SSL handshake). **[**SPT3TMS-10683 **]**

The following figure illustrates an unencrypted vs encrypted HTTP concetions:

[SPT3TMS-10498 **]**

8.8.3.3 Queue encryption

Queue encrypting is essential to protect the confidentiality and integrity of the data being transmitted through the message queue. **[**SPT3TMS-10689 **]**

Queue encryption should be realized in three steps:

- **TLS/SSL encryption:** A secure communication channel between producers and consumers.
- **Data Encryption at Rest:** Messages store in a database must be encrypted.
- **Message-Level Encryption:** The message itself must be encrypted.

[SPT3TMS-10688 **]**

8.8.3.4 Back-up encryption

Back-ups are further security measurement to protect data loss but represents a serious vulnerability as the back-up could have been modified, deleted, or used to obtain sensitive information. For the reason being every back-up must be encrypted to make it resistant against modifications, and data theft. Data back-ups relies on symmetric encryption and requires a single key file. **[**SPT3TMS-10691 **]**

8.8.4 Authentication

Authentication enables the system to keep the data secure by permitting only authenticated clients to gain access to protected resources. **[**SPT3TMS-10690 **]**

Authentication within the scope of the system is differentiated by the following:

- **End user authentication:** An end user (e.g., dispatcher) requires access to the HMI to dispatch the respective sector. The end user is only privileged to interact (read, write, create) within a specific sector and outside of the sector the privileges might differ.
- **External system authentication:** An external system will have to interact with the system and is only privileged to interact according to approved operations and resources.
- **Component authentication:** Components will have to interact with other components and such components are only privileged to interact according to approved operations and resources.

[SPT3TMS-10685 **]**

8.8.5 Integrity

Integrity is one important step to build a solid and secure system. The system relies on external dependencies which is potentially affected by "Dependency confusion and substitution attacks". Therefore, the system must always verify all external dependencies and reject any software packaging if the verification fails. **[**SPT3TMS-10684 **]**

The verification relies on a MD5 checksum check which comparison must match between the original vs. the downloaded checksum. If the checksum differs, the dependency was modified, and the build fails. **[**SPT3TMS-10687 **]**

8.8.6 Common Vulnerability Scoring System

The system is subject of potential vulnerabilities which may occur in the chosen programming language or dependencies / frameworks in all regards to the project phases (development phase, operation phase, etc.). Such vulnerabilities may affect the system in its operation and represents a serious risk to miss the defined availability requirements. **[**SPT3TMS-10686 **]**

To evaluate and mitigate the predominated risk of potential security concerns the system refers to the industry standards by following the Common Vulnerability Scoring System (CVSS). The CVSS captures the principal characteristics of a vulnerabilities and produces a numerical score reflecting its severity. The numerical score is a qualitative representation (such as low, medium, high, and critical) to properly assess and prioritize the vulnerabilities. **[**SPT3TMS-10695 **]**

The following table illiterates the rating of potential vulnerabilities:

Rating:	CVSS Score:	Rationale:
NONE	0.0	**[NO-ACTION]** There is no risk, no further actions are required
	0.1 – 3.9	**[SCHEDULED-ACTION]** There is small vulnerability, but it is not affecting the system in terms of availability. A fix can be planned in on time.
MEDIUM	4.0 – 6.9	**[SCHEDULED-ACTION]** There is vulnerability which should be checked and fixed within the next time. Considered to HIGH and CRITICAL rating, this can be planned for the next time frame.

Rating:	CVSS Score:	Rationale:
	7.0 – 8.9	**[IMMEDIATE-ACTION]** There is vulnerability which should be check asap and an action plan should be considered.
	9.0 – 10.0	**[IMMEDIATE-ACTION]** The system may be affected by a critical vulnerability which may affect the availability of the system. Immediate actions should follow to resolve the vulnerability.

[SPT3TMS-13816 **]**

8.8.7 Network Segmentation

Network Segmentation is another important step to make the system more secure and reliable. The main objectives of network segmentation are to protect the system, detect intruders, contain attacks, and prevent these interferences from reaching the internal system. **[**SPT3TMS-10694 **]**

The system distinguishes between the following network zones:

Zone:	**Level:**	**Description:**
ZONE A	UNPROTECED	Client Zone
ZONE B	PROTECED++	Logic Zone
ZONE C	PROTECED++	Back-up Zone (PROD)
ZONE D	PROTECED--	Development Zone
ZONE E	PROTECED--	Back-up Zone (DEV)
ZONE F	PROTECED++	Log Zone
ZONE G	PROTECED++	Hot-Standby Zone

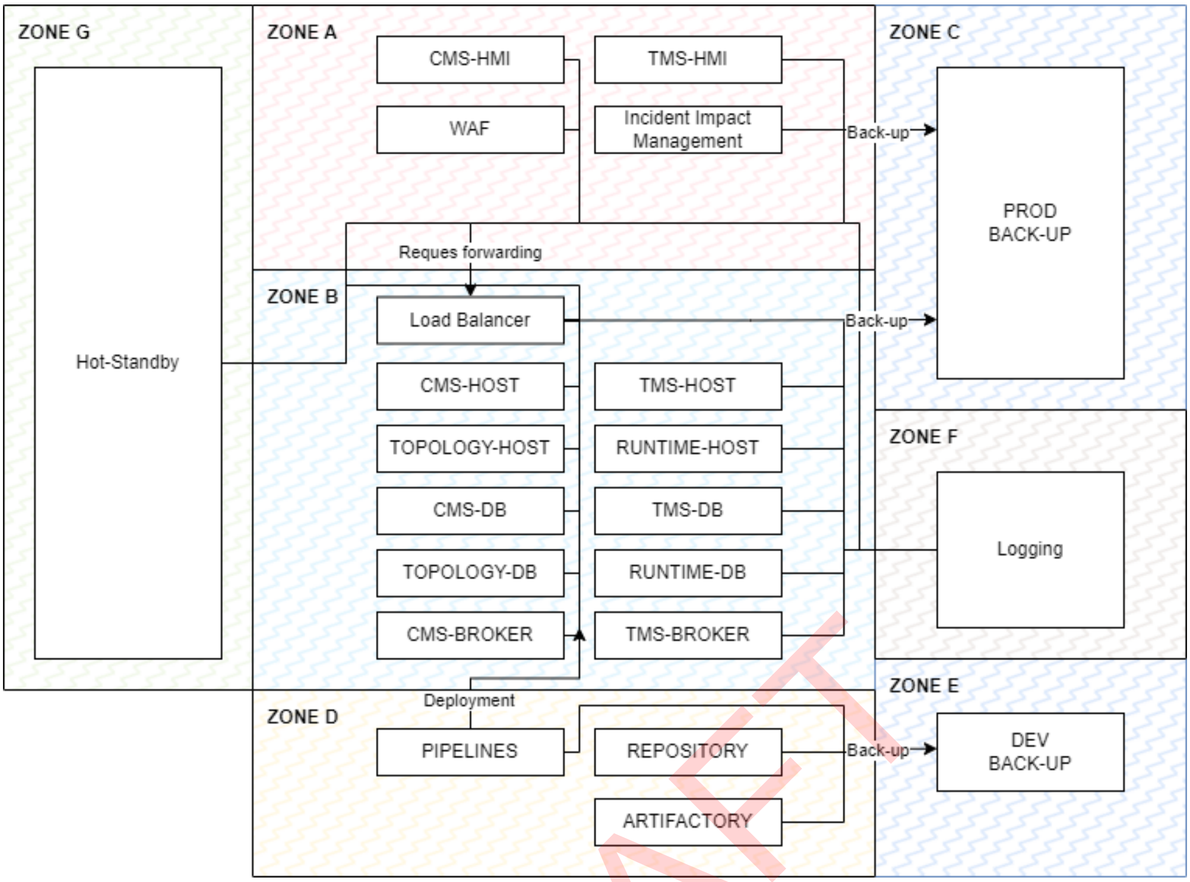
[SPT3TMS-10693 **]**

The following table illustrates the concrete components-to-zone assignment:

Component:	Rationale:	Zone:
CMS-HMI	Frontend clients are running locally on the client's workstation and the client might not be connected to a protected network zone.	ZONE A
TMS-HMI		
WAF		
Load Balancer	The components located in network ZONE B are crucial for the logical operation. The components must be protected more securely. Also, with network zone separation between ZONE A and ZONE B, direct access to other components is restricted. The access to all ZONE B components is only possible via WAF, located in network ZONE A.	ZONE B
CMS Queue Server		
TMS Queue Server		
CMS Database Server		
TMS Database Server		
Topology Database Server		
Sectional Runtime Host		
CMS Host		
TMS Host		
Topology Host		
Sectional Runtime Host		
Back-up Server (PROD)	PROD Back-ups must be securely store outside of its production environment to minimize data loss at the event of a server downtime.	ZONE C
Repository	This network zone consists of development critical components.	ZONE D
Artifactory		
Pipelines		
Back-up Server (DEV)	This is the network zone for development back-ups.	ZONE E
Log Server	Logs must be securely store outside of its production environment to minimize data loss at the event of a server downtime.	ZONE F
Hot-Standby	This network zone consists of Hot-Standby instances to ensure availability. Hot-Standby should be separated from network ZONE B, to minimize the risk and harm at the event of an attack.	ZONE G

[SPT3TMS-10692 **]**

The following figure illustrates the different network zones and its communication:



[SPT3TMS-10503 **]**

The following table describes the overlapping communication between different zones:

	ZONE A:	**ZONE B:**	**ZONE C:**	**ZONE D:**	**ZONE E:**	**ZONE F:**	**ZONE G:**
ZONE A:		X	X			X	X
ZONE B:	X		X	X		X	X
ZONE C:	X	X					
ZONE D:		X			X		
ZONE E:				X			
ZONE F:	X	X					
ZONE G:	X	X					

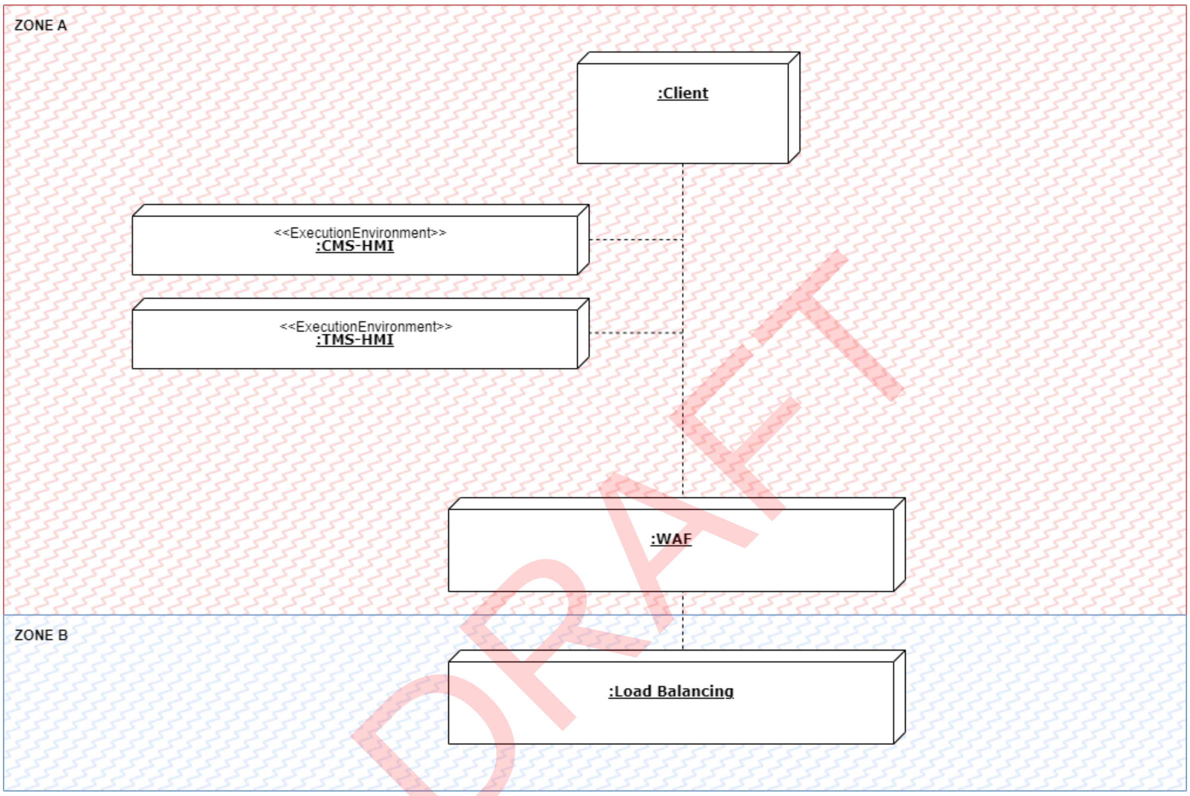
[SPT3TMS-10699 **]**

8.8.8 Web Application Firewall (WAF)

A WAF protects the system by filtering, monitoring, and blocking any malicious HTTP/S traffic traveling to the system and prevents any unauthorized data from leaving. By adhering to a set of policies WAF will determine malicious traffic types and will block the access. **[**SPT3TMS-10698 **]**

This is a crucial aspect of the system as WAF will mitigate the risk of denial-of-service (DDoS) attacks. A DDoS attack is a deliberate attempt by a hostile actor to disrupt operations of publicly exposed APIs, with the goal of degrading the experience of legitimate clients. **[**SPT3TMS-10697 **]**

The following figure illustrates the WAF in scope of system:



[SPT3TMS-10502 **]**

The following tables describes the communication between WAF and other components:

	WAF:	**Load Balancer:**	**CMS-HMI:**	**TMI-HMI:**
WAF:		X	X	X
Load Balancer:	X			
CMS-HMI:	X			
TMI-HMI:	X			

[SPT3TMS-10696 **]**

9 Risks & Mitigations

The system is subject to risks which endangers the system in business and operational goals. This chapter makes potential risks visible and explains how risk mitigation can counteract. **[**SPT3TMS-10432 **]**

The following table describes the systems risks:

Risk:	**Rationale:**	**Mitigation:**
External actors are not available.	The system heavily relies on communication with external system – The system will fail in its tasks if the system is not feasible to receive mandatory data.	Connect only external actors with high availability concepts and ensure availability on contract base.
External actors lack on data quality.	The system is dependent on the accuracy of data provided by external data. The system will also lack on data quality if the system receives poor data quality (GARBAGE IN, GARBAGE OUT PRINCIPLE).	Assumingly the system relies on multiple external actors for the same type of data, then the system can prevent lacking by switching to other data providers if those data are accurate. However, there is no mitigation if data lack on all available external actors.
External actors flood the system with too many requests.	The communication with external actors represents a serious bottleneck and if the number of requests exceeds the capacity of the system, then the system will start to throttle.	If the system will be flooded by unnecessary requests (e.g., DDOS attacks), filters can block certain types of requests. If the system will be flooded by valid requests, the system must increase workers to fulfil the current workload. Therefore, the system must verify if the request is valid. The increase of workers will have a significant impact on the operational costs and should only be considered if the increase is mandatory.
External actors implement changes which is not compatible with the system.	If external actors modify the Exchange items or interfaces, then the system won't be able to partly or full receive mandatory data. This will also lead to the same behaviour as "External actors are not available."	By contract, external actors are restricted to modify interfaces / Exchange items without further confirmation. If external actors plan to modify interfaces / Exchange item, then the system should be adjusted accordingly.

[SPT3TMS-10433 **]**

10 Glossary

-

ATO	Automatic Train Operation
AoC	Area of Control
CCS	Control Command and Signaling
CI/CD	Continuous Integration / Continuous Deployment
CMS	Capacity Management system
CPU	Central Processing Unit
CTC	Centralized traffic control
CVSS	Common Vulnerability Scoring System
DBMS	Database Management System
DDOS	Distributed Denial of Service
DEV	Development
DLP	Data loss prevention
DNS	Domain Name System
DR	Disaster Recovery-Plan
EDA	Event-driven architecture
ELK	Elasticsearch, Logstash, Kibana
FAM	Fixed asset mgmt
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
IRM	Information Rights Management
OS	Operating system
OR	Operational Requirements
NTMS	Neighborhood TMS
MD5	Message Digest Method 5
MITM	Man-in-the-middle attack
MS	Microservices

MVCC	Multi-version concurrency control
PES	Plan Execution System
PIS	Passenger Information System
POS	Path Order System
PROD	Production
QA	Quality Assurance
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RBAS	Role Based Access Control
REST	Representational State Transfer
RIM	Rail Infrastructure Manager
ROC	Rail Operating Company
RT	Real Time
RU	Rail Undertakings
SAD	System Architecture Description
SCI-OP	Standard Communications Interface – Operational Plan
SEMP	System Engineering Management Plan
SOAP	Simple Object Access Protocol
SoC	Separation of Concerns
SSL	Secure Sockets Layer
TAF	Telematics Applications for Freight
TAP	Telematics Applications for Passenger Services
TCR	Temporary Capacity Restrictions
TLS	Transport Layer Security
TMS	Traffic Management System
TPS	Train Protection System
TSI	Technical Specification for Interoperability
UML	Unified Modeling Language
WAF	Web Application Firewall
WSDL	Web Services Description Language
XFN	Cross functional

[SPT3TMS-10700 **]**