Rail to Digital automated up to autonomous train operation

D31.4 – Yard to station automatic movement system analysis

Due date of deliverable: 31/12/2024

Actual submission date: 13/05/2025

Leader/Responsible of this Deliverable: Matthieu THIBAULT (SNCF)

Reviewed: Y

Document status		
Revision	Date	Description
01	24/03/2025	1st draft version for WP31.1 review
02	13/05/2025	2nd draft version, WP31.1 approved, for TMT review
03	02/09/205	3rd draft version , consideration of TMT comments
04	19/09/2025	TMT comments consolidation
05	05/05/2026	JU comments consolidation

Project funded from the European Union's Horizon Europe research and innovation programme		
Dissemination Level		
PU	Public	

Start date: 01/12/2022

Duration: 42 months

ACKNOWLEDGEMENTS

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

REPORT CONTRIBUTORS

Name	Company	Details of Contribution
Matthieu THIBAULT	SNCF Voyageurs	Leader, Author Modelling process, ATO architecture, Architecture of model and driving simulator
Charles MOSTACCHI	SNCF Voyageurs	Author ATO architecture
Fabien BELLEMIN	SNCF Voyageurs	SNCF ATO projects
Thomas JOUSSELIN	SNCF Voyageurs	Author Speed profile optimization, Speed profile tracking
Maxence BOUHRIS	SNCF Voyageurs	Author Signal detection
Martin TEBOUL	SHERPA	Author Signal converter
Simon FUNKE	DB	Reviewer
Manuel KOLLY	SBB	Reviewer
Miguel Fernandez ELORRIAGA	CEDEX	Reviewer
Ignacio ALGUACIL VENTAS	INECO	Reviewer
Marc SANGO	SNCF	Reviewer

Disclaimer

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

EXECUTIVE SUMMARY

Throughout the history of engineering, working methods have continually evolved. These methods have had to adapt to changes in the field, addressing increasing complexity on one hand, and stronger demands for reduced timelines, costs, and higher quality standards on the other.

This document aims to present an approach suited to the design of modern, complex, digital, and multiphysics systems. This approach, known as Model-Based Design (MBD), seeks to establish a representation of the system being designed as early as possible in the development cycle.

Although this approach is widely used in the aerospace and automotive industries, it has been slow to gain widespread adoption in the railway sector. The intention of this document is to contribute to promoting this method within the railway industry.

The railway sector is actively working to improve its performance at the European level. The development of control and signalling systems (European Railway Traffic Management System) aims to enhance interoperability of train operations. New digital technologies have been identified as potential "game changers," enabling significant performance gains. Besides replacing the GSM-R communication standard with FRMCS (based on 5G), the introduction of geolocation for absolute train positioning, mobile block systems (ERTMS/ETCS L3), and train automation have been identified as technologies that can increase capacity and reduce energy consumption.

In this context, the MBD approach can be a key factor in facilitating and accelerating the introduction of these innovations. The main advantage lies in the early prototyping of systems in the development cycle. MBD relies on a representation language, standardized methodologies, and tools that ensure a rigorous formalization of the system's design characteristics. This approach speeds up the development cycle while ensuring reliable requirements (limiting the rate of changes and errors). It is part of a continuous and integrated design process that ensures consistency at every stage of development. During the architecture definition phase, MBD aims to define the system (structure, behaviour, data) within a model and represent it through specialized views. Simulatable models allow for early concept validation in these phases. During the definition phase, through automatic code generation, the model can directly serve as the source for the system's software. Models allow testing of the implementation as the system matures, particularly by modelling the interactors within the system’s environment. When deployed on test hardware, this environment modelling continues during the validation phase, ensuring continuous behavioural verification of the system.

This document aims to illustrate the design process using the MBD approach by presenting a case study of train automation. This use case includes the automation of technical movement operations between maintenance centers and stations (driverless GoA4) and the automation of commercial missions (under driver supervision GoA2). The scope is limited to the "Automatic Train Operation" (ATO) system. This application case is based on the standards defined in the Technical Specifications for Interoperability (TSI) for GoA2 and on published research results concerning GoA4.

This document outlines the Model-Based Design (MBD) approach, along with the associated processes and tools. It details the definition of an ATO system tailored to the use cases following the application of this approach. The document specifies a simulatable model that represents the system. The process described covers the phases from system definition and prototyping to software bench testing. The definition of the ATO system is the result of selected choices among several alternatives, with the modelling specifically tailored to the use case.

The next steps in WP31 will focus on the modelling and simulation of the finalized system, based on the specifications provided in this document.

After this project, future work should explore alternative ATO system architectures to identify potential variants and demonstrate the most relevant solution. GoA4 use case can also be extended to other technical movement (turnaround, inside yard). The GoA2 use case could be further enhanced by exploring additional performance gains through the implementation of artificial intelligence algorithms. Additionally, extending the modelling efforts by developing specialized tools and libraries would provide the industry with dedicated resources, accelerating the implementation process.

ABBREVIATIONS AND ACRONYMS

ATO	Automatic Train Operation
ATSM	Automatic Train Stopping Management
BIL	Basic Integrity level
CCS	Control Command Signalling
ERTMS/ETCS	European Rail Traffic Management System
ETCS	European Train Control System
GoA	Grade of Automation
MBD	Model Based Design
MPC	Model predictive controller
MBSE	Model Based System Engineering
OSP	Operational Speed Profile
R2DATO	Rail to Digital automated up to autonomous train operation
SE	System Engineering
SIL	Safety Integrity Level
SSEM	Supervised Speed Envelope Management
TCMS	Train Control and Management System
TSI	Technical Specification for Interoperability
TTSM	TimeTable Speed Management

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Driven by European initiatives, the deployment of ETCS, the European Train Control System, is progressing. The successive updates in ETCS standardization related to automatic train protection have led to the release of Baseline 4 R1 specifications, as outlined in the Technical Specifications for Interoperability (TSI) published in 2023. The specifications for the "Automatic Train Operation" (ATO) system were introduced in this version of the TSI for automation level GoA2, where the train operates autonomously but under the supervision of a driver in the cab.

These systems are a suitable response to the increasing rail traffic on certain already congested lines. ETCS Level 2, with continuous updates of signalling information, hybrid train detection, which reduces the distance between successive trains (mobile blocks), and ATO, which minimizes margins, all address the need for denser train operations.

Deployment is a critical issue in enabling networks to benefit from these performance improvements. While the deployment of ERTMS/ETCS is advancing across Europe, ATO deployment remains mostly in the project phase (such as the Stuttgart digital hub).

The complexity of ATO systems also needs to be addressed. These systems involve different use cases depending on the level of automation (GoA). In GoA2, driving is automated but supervised by a driver. In GoA4, driving is fully automated without a driver in the cab. Currently, only GoA2 is standardized. These systems are integrated within ERTMS/ETCS, featuring three interfaces (ETCS, Train Management Control System, ATO trackside). ATO is a system of systems, with both onboard and infrastructure components, involving multiple stakeholders (equipment manufacturers, operators, infrastructure managers, certifiers, laboratories). Safety constraints are critical for certain functionalities (manual/automatic switching, door management, etc.). In GoA2, human factors must be considered during automation transitions. In GoA4, the ATO system takes on responsibilities for train environment management, a task traditionally handled by the driver.

To accelerate the deployment of ATO systems, new methods focused on efficiency and speed must be applied. The goal of this document is to present a Model-Based Design (MBD) approach centered on early prototyping in the development cycle of complex systems.

For decades, the MBD approach has been extensively used in the aerospace and automotive industries. Some initiatives have led to its deployment in the mainline sector, mainly in use cases related to train control (traction and braking) and ETCS (openETCS). The aim is to promote a more systematic adoption of MBD in the development of signalling systems.

The present document constitutes the Deliverable D31.4 "Yard to Station Automatic Movement System Analysis" in the framework of IP/R2DATO/TE19 Modelling techniques/WP31 Automated Train Supervision and Control. This deliverable complements the work on a subtopic that addresses a use case for remote control and supervision of freight trains within the WP.

The originality of the WP31 approach lies in its application to the design of Automatic Train Operation system and its broad scope, covering a significant portion of the development cycle. In European projects, the MBD approach has typically been limited to the static description of architecture. WP31's work encompasses a wide range of modelling techniques, including simulation, integrated validation, co-simulation, code generation, and virtual reality.

The use case chosen to illustrate the method is also innovative. While GoA2 and GoA4 work generally involves train trials, GoA4 trials usually focus on movements within maintenance centers. In contrast, this study proposes to examine movements from the maintenance center to the station.

WP31 is part of the activities demonstrating the value of modelling in the development of digital technologies for ATO use cases, in addition to the ERTMS/ETCS L3 use case addressed by WP30. Within R2DATO, the WPs cover different levels of maturity in their demonstrations. By targeting a primarily software-based prototype (TRL5) in the system development cycle, the work presented in this document falls between the specification phase (WP5, 6, 7) and the train demonstration phase (WP42). Although evaluated in a less representative virtual environment, the approach enables testing a significantly larger combination of variants compared to field demonstrations, which are limited by operational constraints related to test organization.
By relying on rapid prototyping techniques, the developments are more flexible and adaptable than those in standard prototyping approaches that use compiled software (WP9, 10, 11).

The activities and the level of detail of deliverable will be limited . The objective to covers the development cycle (specification, test, deployment) is prioritized over achieving a high level of analysis detail. The aim is to give an overview of the potential and highlight the issues at stake. The model will implement functional behaviour. Lower layer of logical behaviour (especially network) will not be modelled. By simplification, hypothesis are made and exported constraint are taken in order to take into account maturity of technology (safety of perception for e.g) in order to consider a short term system deployment. There is no analysis of alternatives (cross-assessment and decision). Only one architecture will be studied.

The first part of the document provides a general overview of the Model-Based Design (MBD) approach. It introduces the underlying principles that enable accelerated development. Standardization ensures clear communication and mutual understanding among stakeholders. Formalization, rapid prototyping, and early testing contribute to reliability throughout the development cycle. Confidence in the development process remains steady, while the maturity of the design and the representativeness of support and testing environments increase over time. This section also explains how the MBD approach integrates into the development cycle, adapting to each stage and generating engineering artifacts suited to the process. Additionally, it presents the development environments and ecosystem that support the technical management of model development.

The second part describes the ATO system architecture as specified within the context of the use case and outlines the definition process followed. It details the results of the architectural modelling, which represents the system's structural, functional, and performance characteristics. Integrating the reference ATO frameworks established in the 2023 TSI, and inspired by X2RAIL and TAURO project, the innovative aspect of the approach lies in its tailored design for the identified use case.

The third part of the document describes the application of the Model-Based Design (MBD) approach to the use case, focusing on the development of a simulatable model. It details the development process, system architecture, operational use of the model, and simulation outputs. The originality of the approach lies in its broad scope, encompassing a complete system-level model rather than being limited to algorithmic aspects. The diverse technologies involved in such a system require a variety of modelling approaches. As a result, a single model integrates multiple components, including virtual reality elements (to simulate the perception system’s environment) and

various control/command algorithms—such as image recognition through machine learning, driving profile optimization, and traction/braking command regulation, among others. This section also highlights the specific contributions of modelling to the demonstration of system safety. The concrete implementation of the Model-Based Design approach, as applied to the specific ATO use case presented in this document, illustrates its adaptability to the development context of CCS systems.

This initial document (D31.4) is delivered at the conclusion of the specification phase for the ATO system dedicated to the specific use case, as well as for the model. It serves as an input document for the model implementation. A second document (D31.5) will be produced, summarizing the results of the simulations carried out.

2 SCOPE AND LIMITATIONS

In accordance with the Grant Adreement, this document is illustrating modelling techniques with an example for ATO GOA4 for a specific and national use case (movement from yard to station based on national lateral signalling) and ATO GoA2 for passenger missions. This is not to be considered as an architectural solution to be further elaborated on a European level.

The aim of this deliverable is to demonstrate the effectiveness of the proposed modelling techniques. The examples given are not meant to be a specification of an ATO GoA4 architecture, nor a validation of existing architectures.

This limitation implies that:

- The WP6 deliverables are outputs of the R2DATO project intended to serve as specification material. Consequently, the use cases and architectural views may differ.The architectures used for modelling are based on existing architectures definition available at this time (STI CCS, Shift2Rail, TAURO) and make assumptions and simplifications to reach the real target of this WP which is the use of modelling techniques and not defining or being coherent with the architecture under definition in R2DATO. The adaptations made on available architecture leverage the simplifications enabled by the constrained operational domain and reduced functional scope considered in the modelling activities.
- The modelling focuses on the ATO subsystem. The modelling of interfacing systems is limited to an environmental representation: reduced level of detail, compliance with interface standards where available, and consideration of an adapter perspective to support deployment.
- Safety-related considerations are intended to highlight critical issues, but they are not binding or definitive. The goal is not to perform a safety demonstration, but rather to describe the tools and methodologies that enable it.
- Only one architectural view is addressed. The objective is to cover a range of modelling activities across the development cycle, without assessing architectural alternatives or arbitrating efforts in relation to the allocated resources.

This deliverable's system analysis provides the foundational framework for the modelling activities carried out in this subtask, whose results will be detailed in Deliverable D31.5. Taken together, these

two deliverables demonstrate the value and outcomes of the modelling process, offering a comprehensive perspective from analysis to implementation.

3 MODELLING TECHNIQUES

This chapter aims to introduce the concept of a model, its implications, the expected benefits, and its practical implementation.

3.1 CONTEXT

Complex systems are characterized by many interfaces, strong interdependencies between their subsystems, and a diversity of interacting domains such as mechanics, fluid dynamics, and electronics.

They also exhibit multi-scale temporal behaviour, combining synchronous processes operating at different frequencies with asynchronous events. Additionally, these systems face stringent performance constraints, including safety requirements, response times, and data flow management.

They typically have long life cycles, requiring sustained operational readiness and maintenance over time.

Industrial sector faces increasing demands due to environmental evolution, such as sustainable development and cybersecurity requirements. They must integrate emerging technologies and continuously adapt to rapid technological advancements, such as 5/6G networks and artificial intelligence.

Complex systems encompass not only technical aspects but also organizational challenges. The rising number of stakeholders and interactions complicates collaboration, coordination, and effective sharing of requirements and information.

Designing complex systems requires adherence to structured processes involving specification, implementation, integration, verification, and validation. Dedicated methodologies are essential to effectively manage these increasingly sophisticated systems. Model-based approaches have been developed to address these challenges by providing structured frameworks and tools.

Mechanical engineering successfully reinvented itself by transitioning from traditional drafting tables to Computer-Aided Design (CAD). Extending this principle to complete systems enables the effective management of growing complexity.

Model-based design addresses complexity by decomposing intricate problems into simpler, more manageable components. Improved efficiency is achieved through enhanced collaboration among stakeholders and through systematic reuse of existing components and solutions. Rapid prototyping enabled by modelling facilitates early-stage verification, significantly reducing risks later in development.

This digital approach leverages advanced numerical technologies across all development phases, ensuring seamless continuity (Digital Thread) and full traceability. Each stage of development builds systematically on the previous one, maintaining end-to-end consistency and transparency.

Given its inherent complexity, the CCS (Control Command System) can be considered a complex system. Model-based approaches are particularly well-suited to support the needs and expectations of each stakeholder, while addressing the challenges and ongoing evolution of the railway sector.

Complexity factor	CCS characteristic	Model contribution
System of systems	Infrastructure: RBC, interlocking, PKI, Radio Network Rolling stock: ETCS, ATO, Radio	Centralized implementation and specific representation Responsibility sharing among stakeholder Collaborative work
Number of interfaces	European interface specifications [ref. ERJU-722] Internal: • ERTMS/ETCS: subset 026-7/8, 036, 037, 039, 101, 121, 130, 139 • ATO: subset 125, 126, 130, 139, 143, 147, 148 External: subset 027 ,147, 034, 119, 035, 126	Centralized implementation and specific representation Sequence representation for interaction Simulation for behaviour evaluation
Interdependencies	Movement authority implies subsystems feedback loop: traction/brake control command, position/speed evaluation, train detection	System simulations
Multiphysics	Pneumatics (brake model) Mechanics (braking curves) Electronics	Simulation based on specific physics libraries or co-simulation with physics specialized tools
Distance scale	from train scale to line scale	System simulation
Time scale	High rate (speed measurement, radio, balise) <ms Medium rate (network exchange, computing frequency) <1000ms Low rate (HMI, radio message) 1s<t<5s	System simulation with heterogeneous step time
Multi-technologies	Radio (telecommunication, balise) Network Real Time computing	Tailored level of abstraction for system simulation Dedicated simulation for technology evaluation
Lifecycle	life expectancy of a train is 40 years with fast evolving technology (telecommunication)	Support modularity identification and standardization of interface for upgradability
Human factor	Interaction between ETCS and Driver (attention management in automatic mode)	Ergonomic prototype Driving simulator
Level of safety	Heterogeneous: BIL to SIL4 (see EN50128 2020) High level of safety (SIL4) for protection functions	Support segregation of different level of safety

Complexity factor	CCS characteristic	Model contribution
Number of stakeholders	National Security Agency, ERA, Operator, Infrastructure manager, Entity in charge of maintenance, Transport Authority, Product designer, Rolling stock supplier, notified body, laboratory,...	Adaptation of representation to dedicated stakeholder Collaborative work

Table 1: CCS complexity and model contribution

Each stakeholder plays a specific role in the definition, evolution, production, and operational implementation of CCS systems. Modelling provides a tailored response to the specific challenges and concerns of each of them.

Stakeholder	Activities	Possible issues	Model contribution
Standardization bodies	Establish a standardized regulatory framework to promote interoperability and enhance system performance	Lack of clarity, inconsistency, lack of requirement	Specification validation System consistency guaranteed
European/National Security agency	Deliver authorization for ensuring system safety	Undetected flaw of safety design of system	Centralized and formal system definition, early and continuous validation ensures a high level of integrity and coverage of safety features.
Assessment bodies	System assessment with respect to the applicable set of regulatory requirements.	Document centric assessment approach based on non-standardized system description	Standardized system description
Operator, Infrastructure manager	Edit specification for tender	Misunderstanding with supplier (quality of requirements) Supplier dependency	Share model with tenderer Common language Clear and formal requirement formulation Interface definition for modularity and openness
Supplier	Design and manufacture product, train	Misunderstanding of customer needs Error avoidance through the development cycle numerous proof artefact production	Shared model with customer and subcontractor Model for specification validation Model for prototyping Automated verification, integration, deployment and proof artefact
Certification laboratory	Test interoperability component	Develop environment interface of product	Model environment interface and generate code to deploy on test bench (see also subset094 ref. ERJU-722)

Table 2: model contribution to stakeholder needs

3.2 MBSE AND MBD

In the project context, a model is a virtual representation of a system that captures its key characteristics with a tailored level of detail, including its behaviour, structure, and managed data. It is developed and used within the context of system design to support analysis, validation, and optimization.

Although the aim of MBSE according to INCOSE is “the formalized application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases”, we distinguish between two types of model-based approaches:

- **Model-Based Systems Engineering (MBSE):** Focuses on the architectural representation of a system, including its structure, behaviour, and data. It considers the system as a whole, considering internal interdependencies rather than treating each subsystem separately. MBSE supports core systems engineering activities and provides outputs for transversal activities, such as RAMS (Reliability, Availability, Maintainability, and Safety). human factor, risk management, configuration management, alternative evaluation and decision making.
- **Model-Based Design (MBD):** Focuses on simulation at different levels of system maturity, from conceptual representation to physical realization.

3.2.1 State of the art

The value of deploying MBSE (Model-Based Systems Engineering) and MBD (Model-Based Design) methods in industrial settings is well supported by both scientific publications and real-world feedback from industry implementations.

Numerous studies highlight the benefits of MBSE in the design of complex railway systems. For instance, it has been applied to the development of the Sydney rail network [ref. ERJU-705], as well as to railway monitoring systems [ref. ERJU-706].

Similarly, the MBD approach is addressed in several scientific papers, presenting structured methodologies for the design of railway systems. These include formal design approaches [ref. ERJU-697] and frameworks for the development of safety-critical software [ref. ERJU-698]. A specific use case involving the development of a model for the "Radio Block Center" component has been identified, led by SNCF Réseau, with the objective of obtaining an unambiguous specification that is independent of any particular solution [ref. ERJU-723]. This use case demonstrated the benefits of modelling, particularly in providing safety evidence.

Simulation software vendors also share customer feedback, illustrating practical applications of MBD in the railway domain. One such example involves the development of a rail propulsion system [ref. ERJU-700]. Additional case studies from the aerospace sector further support the relevance of the approach, such as its application in fuel management system design [ref. ERJU-704].

MBSE and MBD practices are already widely adopted in aerospace and automotive industries. Publications discuss their use in software validation [ref. ERJU-701] and the design of cyber-physical systems [ref. ERJU-702].

In the field of autonomous mobility, academic researchers have applied this approach to the development of autonomous shuttle systems [ref. ERJU-703].

Across all these domains, both literature and real-world experiences consistently highlight the effectiveness of MBSE and MBD in reducing development cycles and improving overall system quality.

3.2.2 Benefits

The Model-Based approach contributes to achieving projects objectives by optimizing several key factors:

- Cost Reduction: it minimizes project costs by streamlining the design effort and promoting efficiency in the development cycle.
- Effort Reduction: Effort in the design phase is decreased through MBD, which simplifies processes and enhances overall design effectiveness.
- Early Test Integration: it introduces early testing in the development cycle, expanding the test perimeter and thereby limiting issues that may arise in the field.
- Reduced Delays: it is instrumental in minimizing delays by shortening the design lifecycle and curbing the need for extensive corrections.
- Quality Assurance: it ensures high-quality outcomes by incorporating rigorous testing and validation procedures.
- Facilitates Change and Evolution: it provides a framework that eases the implementation of changes and supports the evolution of the project over time, ensuring adaptability and long-term sustainability.

3.2.2.1 Requirements

Benefits provided by the model approach on system requirements definition are given in Table 3.

Criteria	Model benefits
Specific/uniqueness	Each part of the system is represented by a model element (interface, component,…) Representation is limited and standardized Modelling avoids the risk to duplicate requirement
Measurable	Enables performance allocation to the system, like energy consumption or communication performance (time reaction, quantity of data exchange for bandwidth evaluation …).
Appropriate	SYSML standard is designed for system design
Realistic/feasible	Enables evaluation of complexity Modelling is a first stage of representation of the system which ease identification of part of the system that needs more development to reach the sufficient level of maturity
Testable/verifiable	Simulation enables tests with progressive introduction of hardware. (software, processor, hardware in the loop)
Clear	• Is based on common and understood standard (e.g. SYSML) shared with all stakeholders • Uses graphical representation • Enables purpose specific view based on a unique and centralized digital model. It is sufficiently flexible to adapt to different domain specialty (mechanic, electric, pneumatic, power,…), transversal specialty (human factor, RAMS, cybersecurity,…) and project management need (performance follow up, responsibility allocation and management,…)
Completeness	The implementation allows to identify lack of specification. By achieving a sufficient level to ensure simulation guarantee a good confidence in completeness. Derivative method through layer with traceability enable identification of element not covered from the top layer

Criteria	Model benefits
Upgradability	Enables identification of consequences of the modification to an element of the system connected to this evolution
Consistency	No conflicting or contrary requirement are ensured by a centralized representation of the system

Table 3: Model benefits

3.2.2.2 Return on investment

MBSE and MBD are not merely complementary tasks added to the development cycle — they represent a new way of approaching system development. By integrating validation activities earlier in the process, the initial effort is increased. New stakeholders are involved, and additional resources are mobilized, including specialized software and custom developments.

However, this increased upfront effort is balanced by a significant improvement in development quality, particularly through the reduction of undetected errors that would otherwise propagate further down the cycle. Indeed, the cost of errors left undetected early in the process grows exponentially as the development progresses.

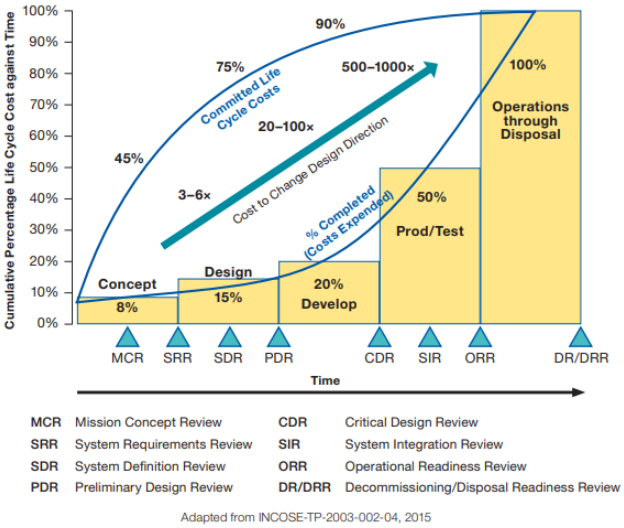


Figure 1: Source NASA engineering handbook- Life-Cycle Cost Impacts from Early Phase Decision-Making

	Software Cost Factors	Systems Cost Factors
Requirements	1X	1X
Design	5-7X	3X-8X
Build	10X-26X	7X-16X
Test	50X-177X	21X-78X
Operations	100X-1000X	29X-1615X

Figure 2: Source NASA Error cost escalation thrrough the project life cycle

3.3 MODEL BASED SYSTEM ENGINEERING (MBSE)

The MBSE approach is based on three key pillars:

1. A standard for system representation: SysML (Systems Modelling Language) provides a structured way to describe system architecture, behaviour, and interactions.
2. A step-by-step definition process: Frameworks like ARCADIA, CESAM, and MagicGrid guide system engineers through progressive modelling stages, ensuring consistency and completeness.
3. A modelling tool: Software such as Capella, MagicDraw, and others enable the practical application of MBSE by providing visualization, analysis, and documentation capabilities.

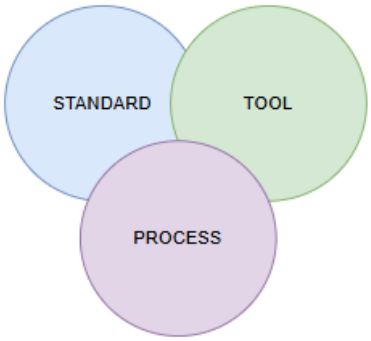


Figure 3: MBSE triptyque

3.3.1 Principles

In this project we will refer to the ARCADIA (Architectural Analysis and Design Integrated Approach) methodological which is the chosen framework for system design [ref. ERJU-724].

The method ARCADIA relies on successive analyses of the system, structured according to various layers or perspectives:

- operational analysis: identification of the actor's need in the domain of action of the system under consideration
- system analysis: identification of the services provided by the system under consideration
- logical analysis: functional analysis defining how the system will deliver its service.
- physical analysis: physical part which compounded the system.

At each level, various diagram types can be used to define system elements, such as stakeholders, functional interfaces, physical interfaces, components, and data. The system is thus progressively defined through abstract steps, with elements graphically represented according to their intended purpose and the target audience of the diagrams.

Through these diagrams, the major characteristics of the system will be defined:

- Structure
- Behaviour
- Data and information flows

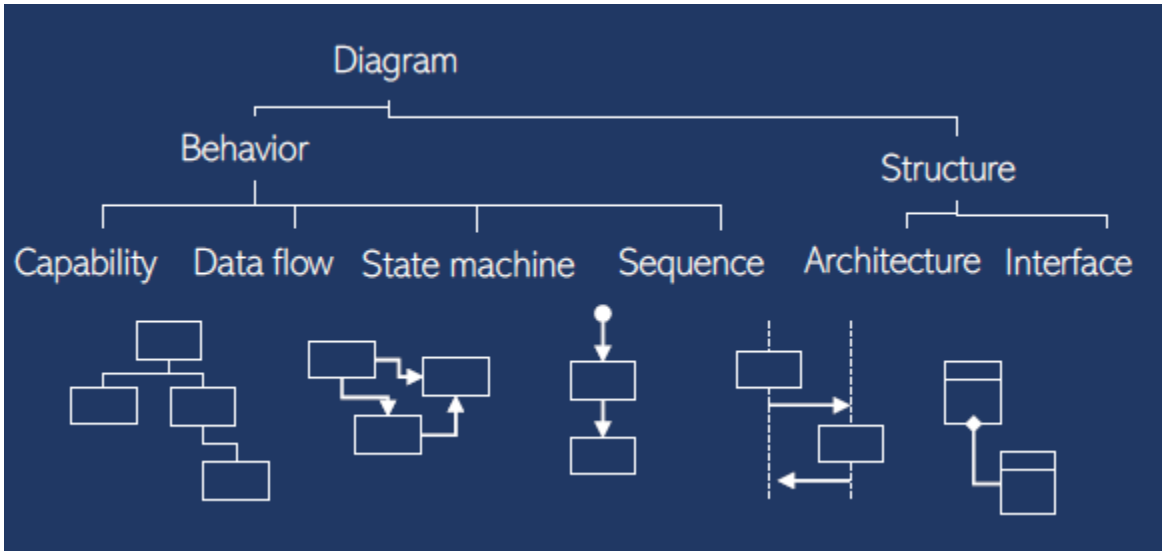


Figure 4: type of MBSE diagram

Not all layers are mandatory; however, System Analysis (SA) and Physical Architecture (PA) are essential steps within the ARCADIA method. For off-the-shelf product modelling, Physical Architecture (PA) alone may be sufficient.

Each modelling stage aligns with a corresponding design phase within the traditional V-cycle development process. The ARCADIA method is iterative; adjustments made at higher abstraction levels can be propagated downward to maintain coherence and consistency.

The successive steps enable the definition and refinement of system elements, which are consistently carried forward from one level to the next, ensuring continuity and coherence throughout the design process.

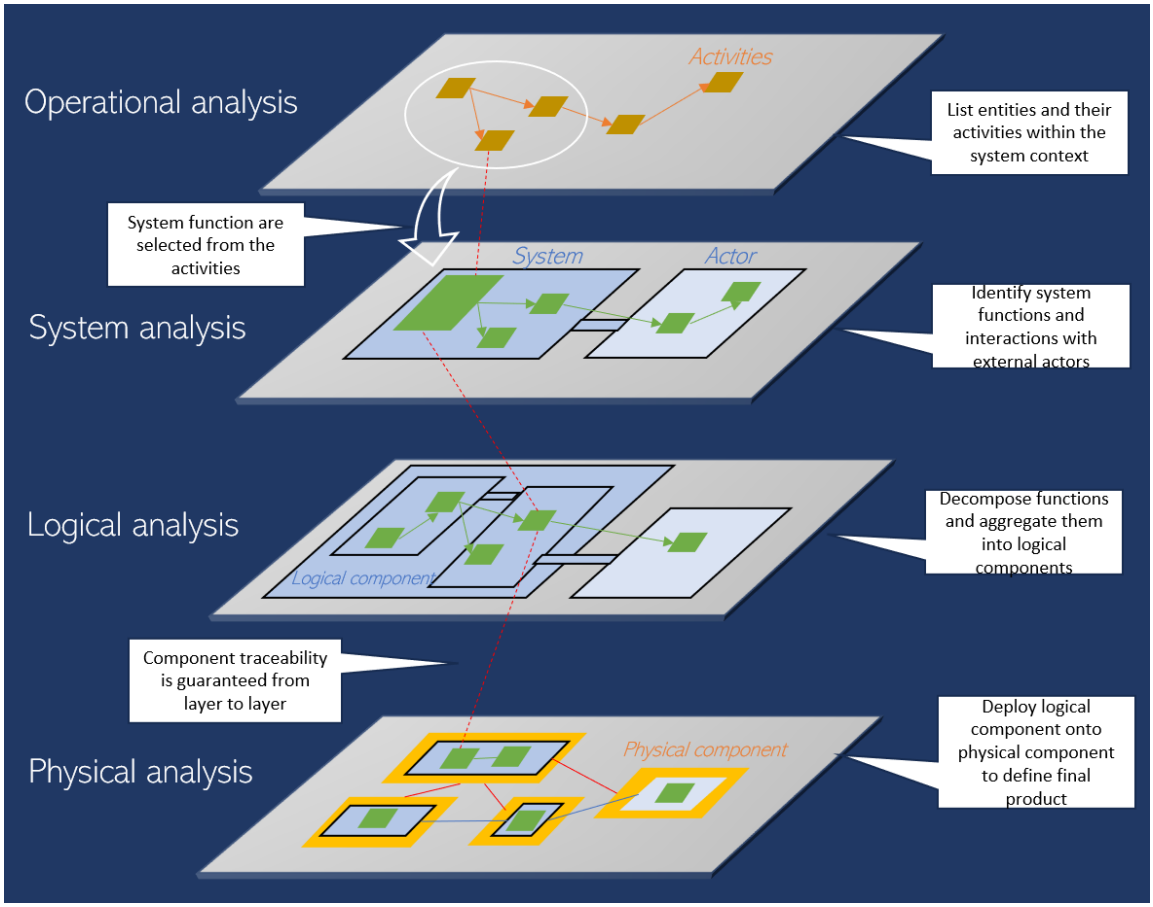


Figure 5: Arcadia analysis layers

Capella MBSE is a software tool implementing the ARCADIA methodology. It provides explicit semantics for defining systems at each engineering level. Capella also integrates verification functions to identify modelling errors, which may indicate potential design issues, such as integrity problems or inconsistencies between abstraction layers.

3.3.1.1 Diagram

Diagrams are used for architectural building but also in order to present information for a specific purpose:

- system specification
- interface specification
- test description
- input for transversal analysis: safety, cybersecurity,...

Different diagram types are used for different architectural perspectives:

- The capability diagram summarizes the services provided by the system to the actors. Based on the system's mission, which defines its purpose, a list of capabilities identifies the value provided by the system to the actors. Each capability is then associated with a sequence diagram, which explains how the value is delivered.
- Architecture diagram represents elements with their relation (interfaces)
- Sequence diagram represents in a timeline sequence of function and exchange between actor and logical/physical component
- Functional decomposition diagram represents decomposition of element in a hierarchical representation applicable to function, logical component and physical component
- Dataflow diagram represents directed functional relation
- Class diagram represents data and information managed by the system
- State machine diagram represents mode of the system and transition condition between these states

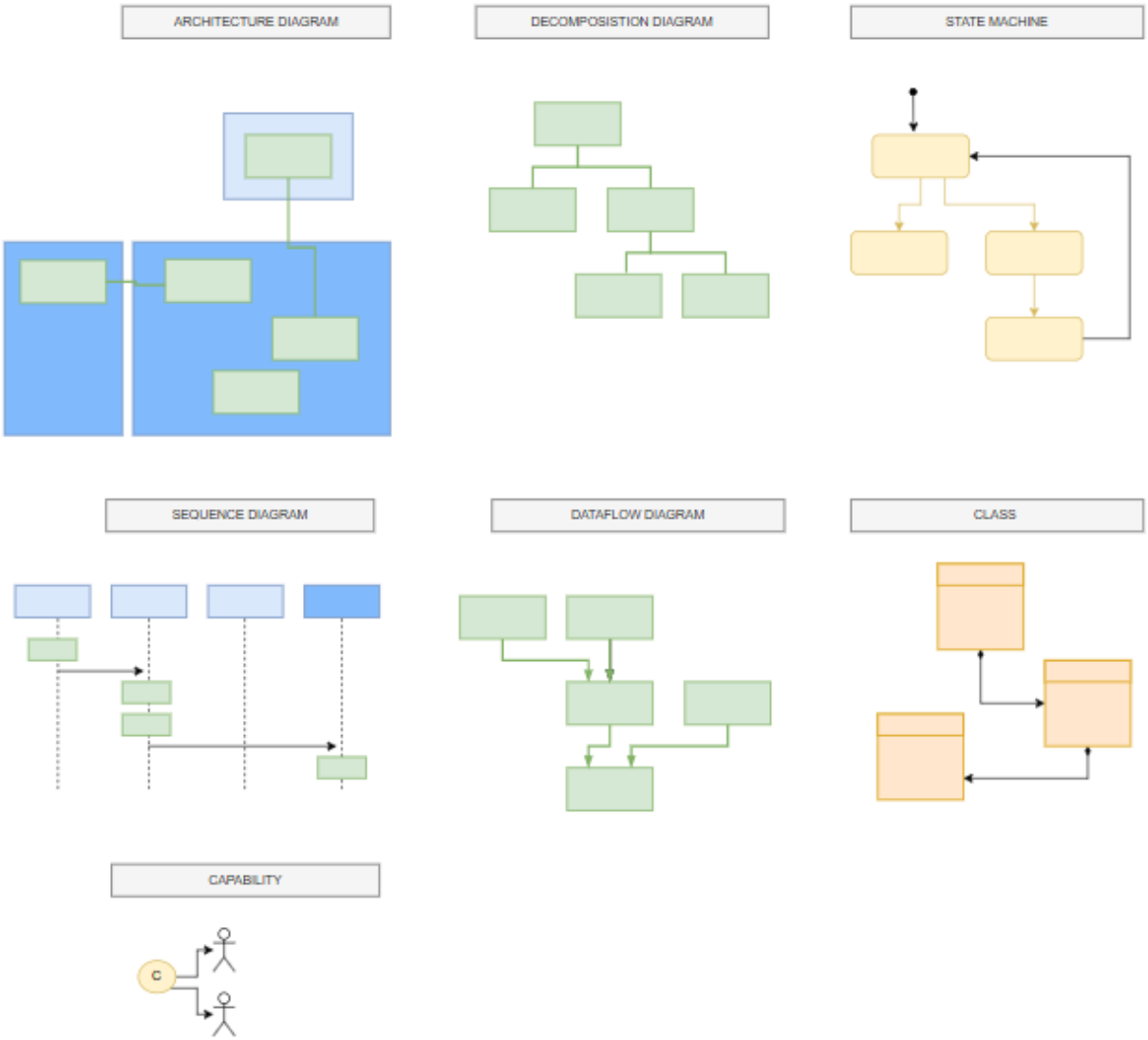


Figure 6: Type of MBSE diagram

3.3.1.2 Capella MBSE tool

CAPELLA MBSE Tool (https://mbse-capella.org/) is a tool that support the MBSE process based on Arcadia [ref. ERJU-725] . It offers different functionalities including representation of system component, especially:

- Actors represent distinct entities interacting with the system under consideration.
- Functions denote the transformations carried out by the system.
- A logical component is a thoughtful aggregation of functions based on specific criteria.
- A physical component embodies the physical implementation of a group of logical components.
- A physical interface links two physical components and represents a mechanical, pneumatic, electric, or power connection.

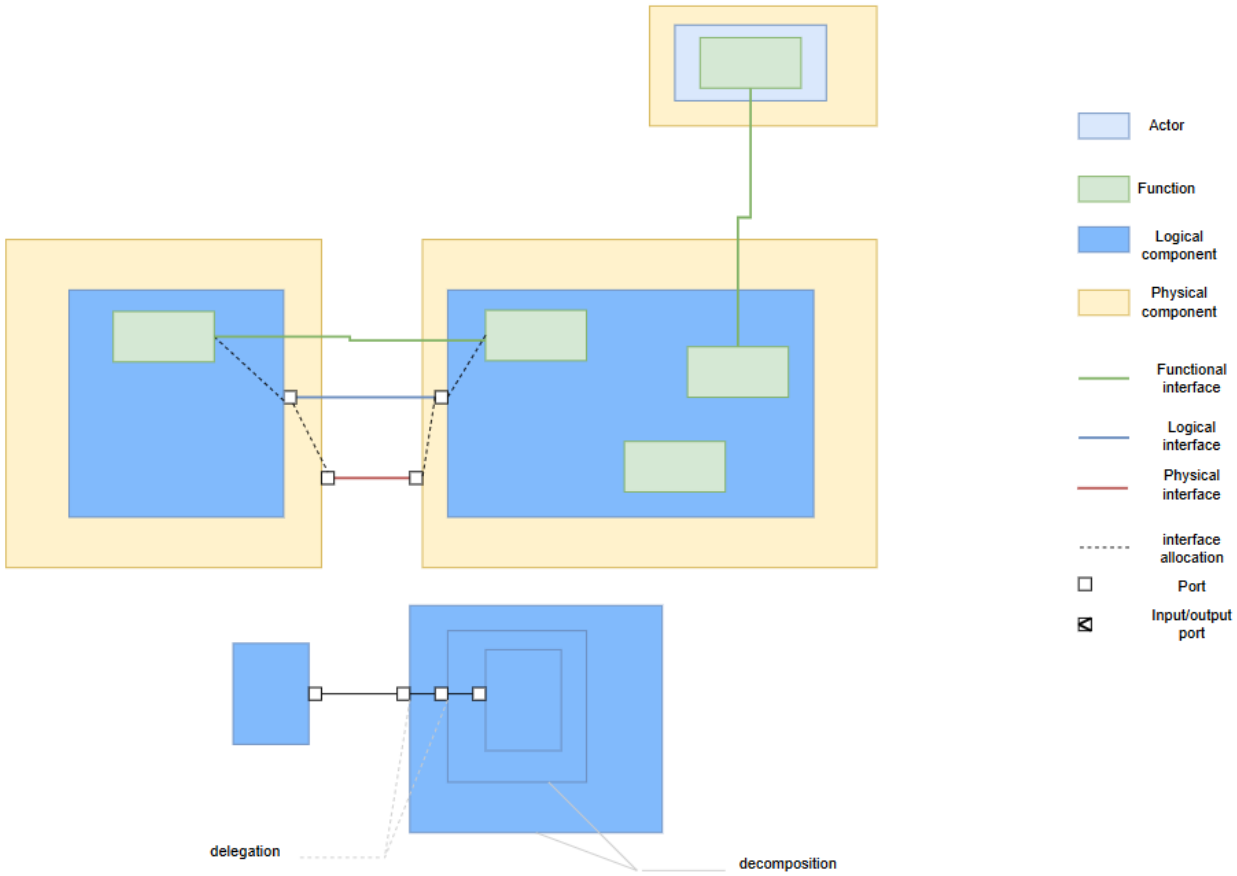


Figure 7: Capella component representation

The advantage of using a modelling software tool lies in its user interface, which provides standardized components. Creating diagrams is simple: components are either dragged and dropped from the palette onto the modelling canvas, or selected directly from the palette and then placed within the modelling area. The modelling explorer enables easy navigation through the model, organized either by diagram type or modelling layers. Hyperlinks embedded within diagrams help build a clear navigation structure between different views. Additionally, the semantic browser summarizes all relationships associated with a selected component. Various export options, such as image format (PNG) or HTML, allow users to preserve interactive links between diagrams.

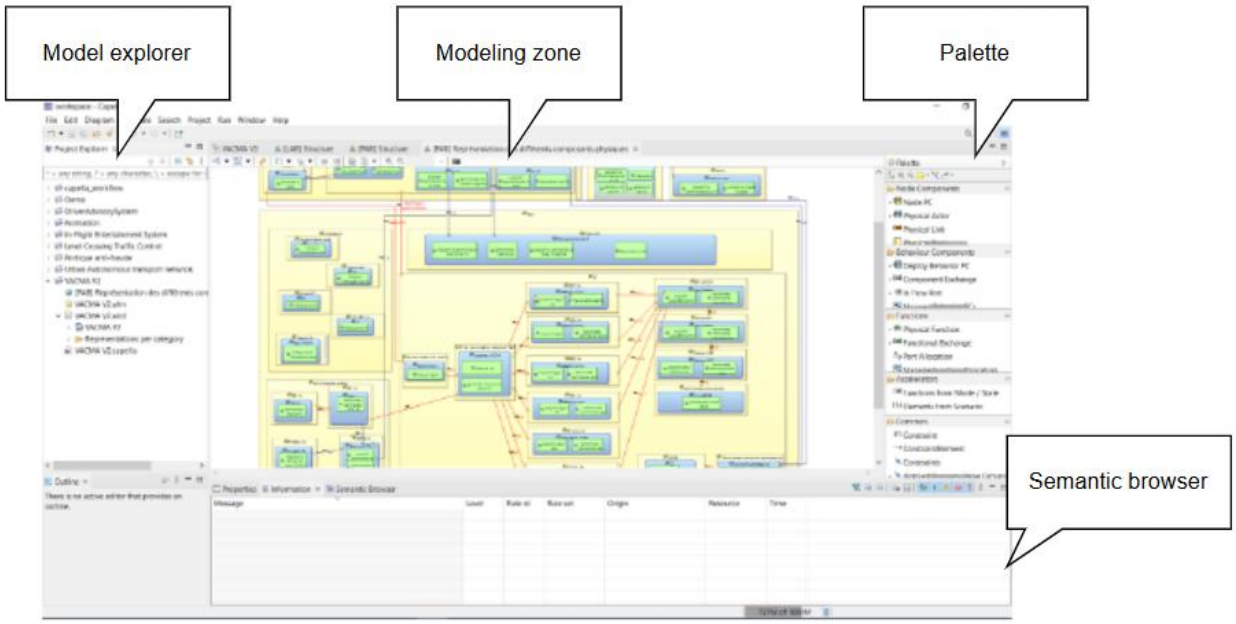


Figure 8: Capella development environment

Behind the graphical interface, the model is stored in a human-readable file encoded in a language similar to XML. Centralizing information ensures the model remains consistent. Specifically, a component is created only once and can be reused multiple times across different diagrams, with uniqueness ensured through a unique identifier. Consequently, any modification made to a component is automatically propagated throughout the entire model.

An integrated automatic analysis feature identifies inconsistencies within a single modelling layer or across multiple layers.

Furthermore, add-ons can extend the tool's capabilities. For example:

Add-on	Capabilities
Property value management tools	Enables allocation of properties to components
Python for capella	Enables python code embedded in the development Environment. Suitable for automatic analysis and export
M2Doc	Enables word export
TeamForCapella (commercial)	Enables collaborative works allowing simultaneous modification of a model
PublicationForCapella	Connects Capella to requirement management tool (Polarion, DOORS, ...)

Table 4: Capella add-ons

3.3.2 MBSE process

The four-step process follows the reference development approach (V-cycle), ensuring a progressive definition of the system by setting design assumptions at each stage and enabling exploration of various solutions.

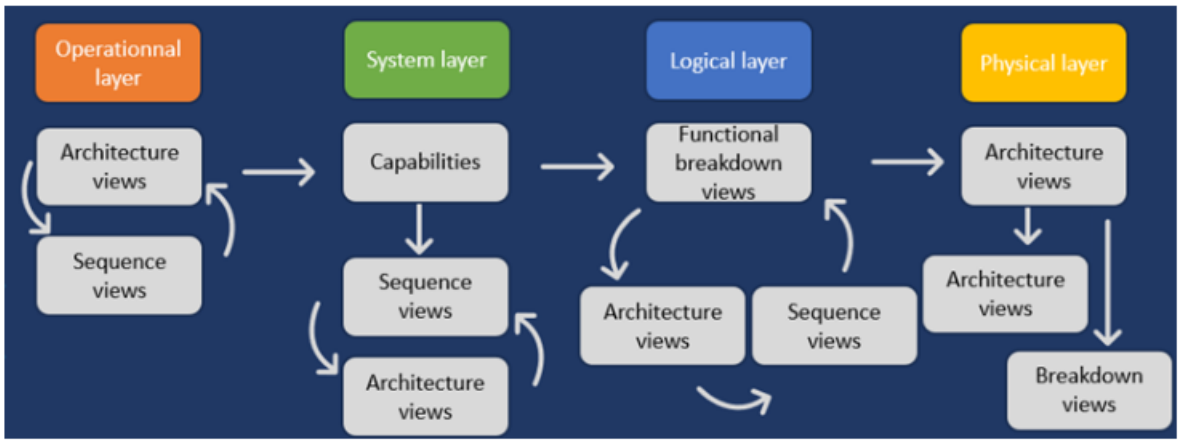


Figure 9: MBSE process

Layer	Design Phase (V-cycle)	Input	Output
Operational	Concept	Operation objectives provided by future user/customer directly expressed or indirectly expressed (market study/benchmark)	Sequence
System	System needs	Activities and actor from operational layer Mission objectives provided by program management Preliminary risk analysis (Safety and Security)	System capabilities System Interface Sequence
Logical	Architecture definition	Functions, actor, functional interface from System layer	Functional Breakdown Functional chains Sequence
Physical	Detailed definition	Logical component, functions, functional and logical interface from logical layer	Product breakdown Physical interface Data exchanged

Table 5: MBSE input/output

The MBSE modelling activities are integrated into the overall design process and interact iteratively with other systems engineering activities:

SE activities	
RAMS	MBSE provides functions, functional chains, capabilities MBSE requires assigning safety integrity levels to ensure the implementation of an architecture consistent with performance requirements (redundancy for e.g.)
Integration	MBSE provides physical interface
Validation	MBSE provides sequence views for validating behaviour and operational views for validating system objectives
Human factor	MBSE provides interaction sequence between human and system (mental load evaluation, co-activity, attention management, collaboration management), data model of information exchanged (formal exchange) and decision tree (risk of errors evaluation)
Performance evaluation and trade-off	MBSE provides performance capture and analysis (time reaction, parameter assessment,...)

Table 6: Table - SE activities

To stay aligned with the maturity levels of technologies and to ensure that system requirements and architectural decisions remain realistic and feasible in the industrial ecosystem, it is recommended to conduct a complementary bottom-up analysis. This involves assessing existing technological solutions, evaluating commercially available components, and identifying potential implementation constraints early in the design process.

3.3.3 Operational Analysis

The Operational Analysis aimed at identifying the context in which the system will be deployed and will operate.

The following key points are addressed:

1. Identification of **Main Actors:** The document identifies the primary stakeholders involved in the process and describes the role of each actor.
2. Actor **activities:** Series of specific tasks or actions performed by an actor to achieve a particular objective or to fulfil a part of their role within the system.
3. Actor **Interactions:** It specifies the interactions and exchanges between these actors to ensure efficient operation.

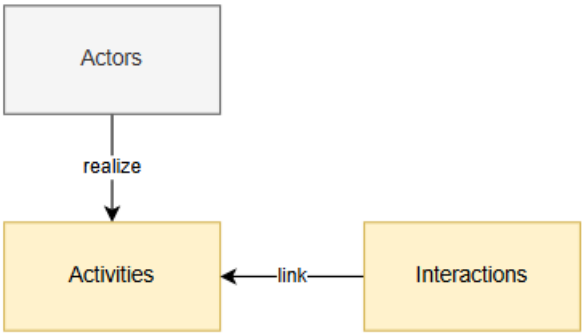


Figure 10: Operational analysis components

3.3.4 System Analysis

This System Analysis aims to define systems in terms of mission and capability, interfaces with external actors, system functions, sequence diagrams by phases, and overall performance.

The mission and capability define the purpose of the system and the services it provides to external actors. Interfaces with external actors detail the interaction points and communication protocols between the system and external entities. System functions describe the specific tasks and operations performed by the system. Sequence diagrams illustrate the temporal sequence of interactions between the system and external actors across different operational phases. Overall performance assesses the system's performance metrics and criteria to ensure it meets operational requirements.

The analysis is based on an operational analysis that identifies the actors involved in the operational context and their activities. Based on the set of identified activities, a portion of these activities is allocated to the system. The choice of system allocated activities is made according to the general functional objective carried out by the system.

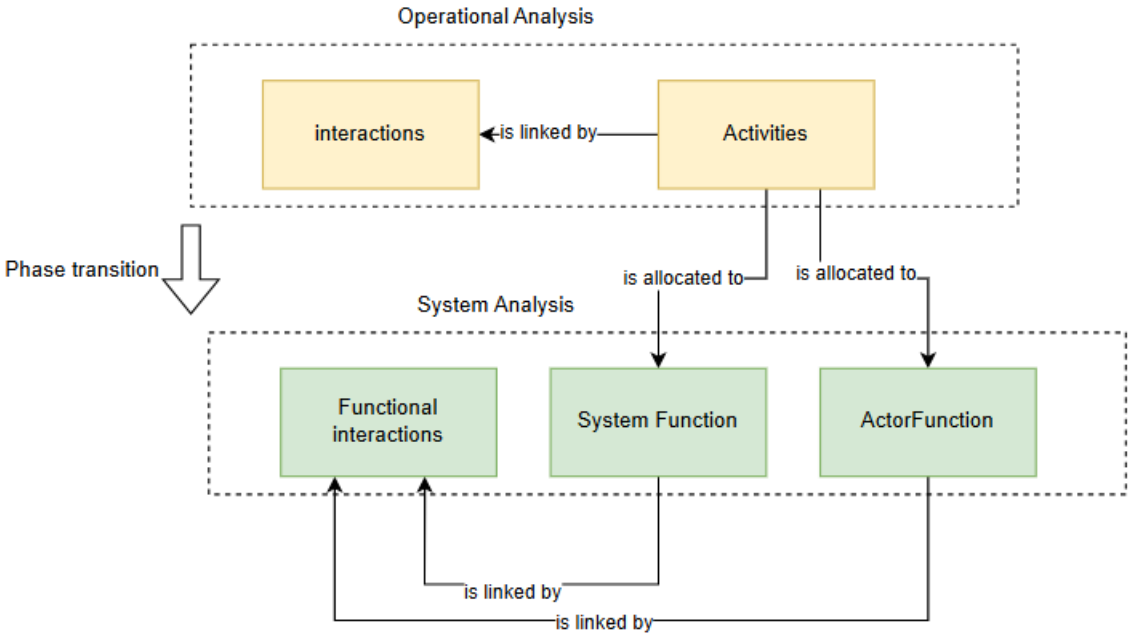


Figure 11: Operational analysis to system analysis transition

The system architecture is built iteratively by following these different steps:

- Decomposition of Operations into Operational Phases: Operations are decomposed into operational phases, each emphasizing a specific capability of the system. For each operational phase, a sequence diagram is produced which identifies interactions between the system and external actors.
- Production of a Functional Chain: A functional chain is produced to represent these interactions, linking the functions of the system and the actors.
- Description of Functions: The level of detail is sufficient to explicitly define all capabilities through sequence diagrams, linking actor functions to system functions.
- Aggregation into an Architecture View: Interactions and functions are gathered into an architecture view.
- Specification of Interfaces: Interactions with each actor are isolated to specify specific interfaces.

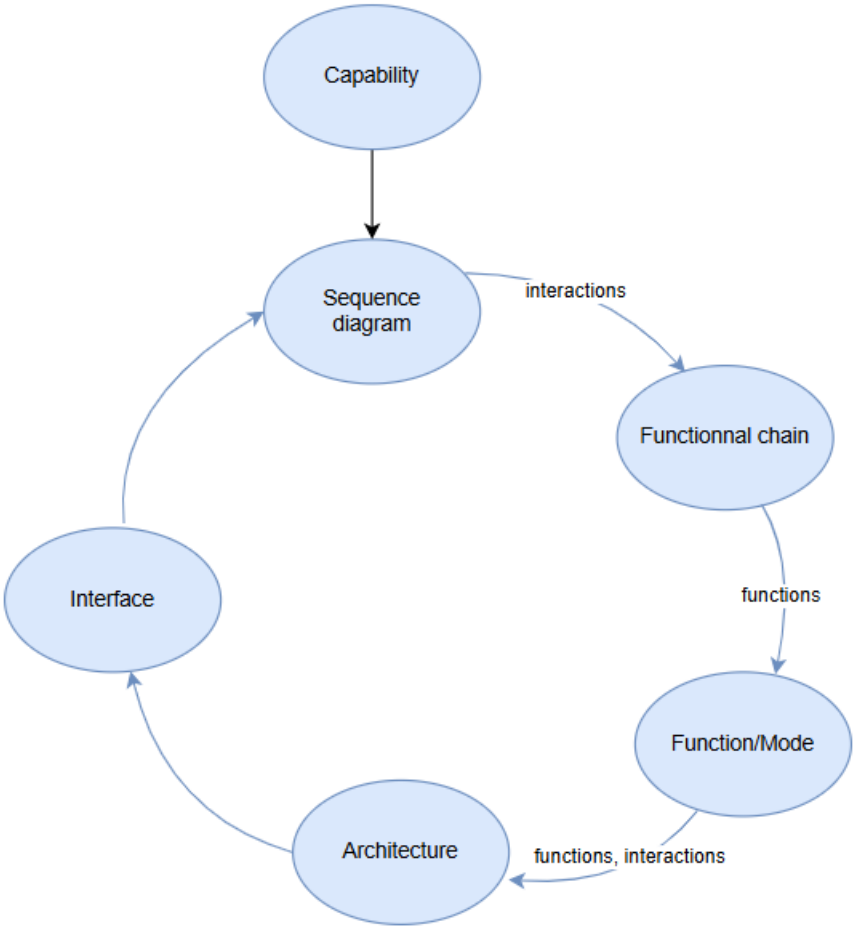


Figure 12: System analysis workflow

The system is defined through different perspectives:

1. **Mission**: The mission defines the reason for the system's existence. It describes the operational context in which the system operates and outlines the goals and objectives it is intended to achieve.
   The mission is established by the "requesting authority," which may be the customer or their representative (such as a product owner).
   Needs elicitation is a core activity of systems engineering, involving an iterative process between the requesting authority and the systems engineer. This collaboration ensures a clear and shared understanding of the needs at the start of the development cycle, both for the project team and for the customer stakeholders.
2. **Capability**: The services offered by the system to external actors to achieve its mission, detailing what the system is able to perform and deliver to meet user needs and requirements.
3. **Functions**: The specific actions or tasks performed by the system, breaking down the overall capabilities into detailed functional components.

4. **Interfaces**: Identifies the functional exchanges with external actors, supported by the logical exchange media. This includes the communication protocols, data formats, and interaction points between the system and other systems or users.
5. **Sequence**: Expresses the sequence of interactions between the system and external actors in a temporal view. It details the order and timing of events, ensuring that all interactions happen in a coherent and synchronized manner.
6. **Architecture:** Represents functions allocated to system or actors, their interactions and the logical interface between system and actors.

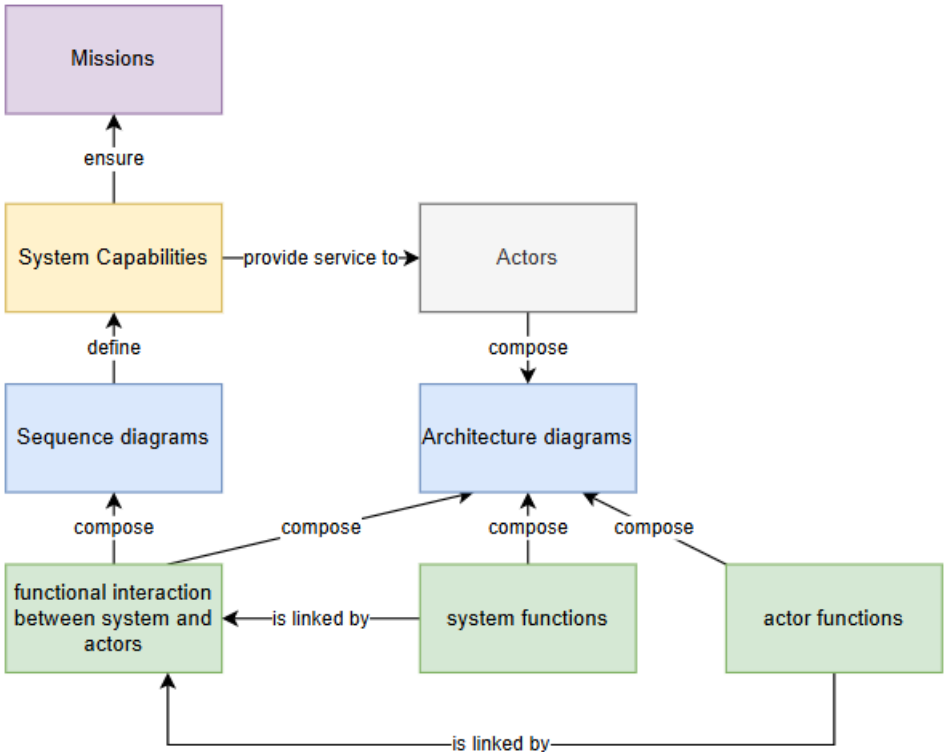


Figure 13: System analysis components relation

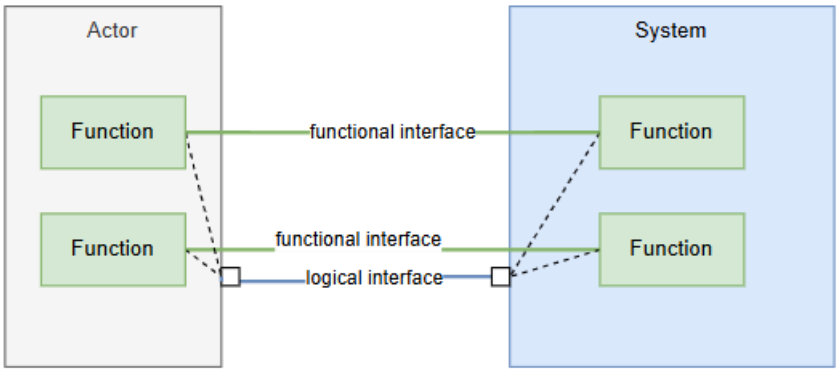


Figure 14: System analysis architecture representation

3.3.5 Logical Analysis

The following activities are conducted for defining this level:

- transition element from system analysis

- system function decomposition. The depth of the decomposition depends on the objective of the modelling. It will start at a high level from the operator's perspective (such as a railway undertaking or infrastructure manager) and progressively increase in detail for the integrator, and even more so for the product supplier, with a very detailed decomposition.

- regroup function into logical component. Logical component groups can be formed among the following criteria (not exhaustive):

  - homogeneous function according to service provided by the system, domain (pneumatics, electronics, mechanics,...) ,...
  - architecture objectives; modularity for e.g.
  - geographical proximity
  - level of safety
  - cybersecurity zone

- define functional chain: describe for a service provided by the system the list of function and interface involved

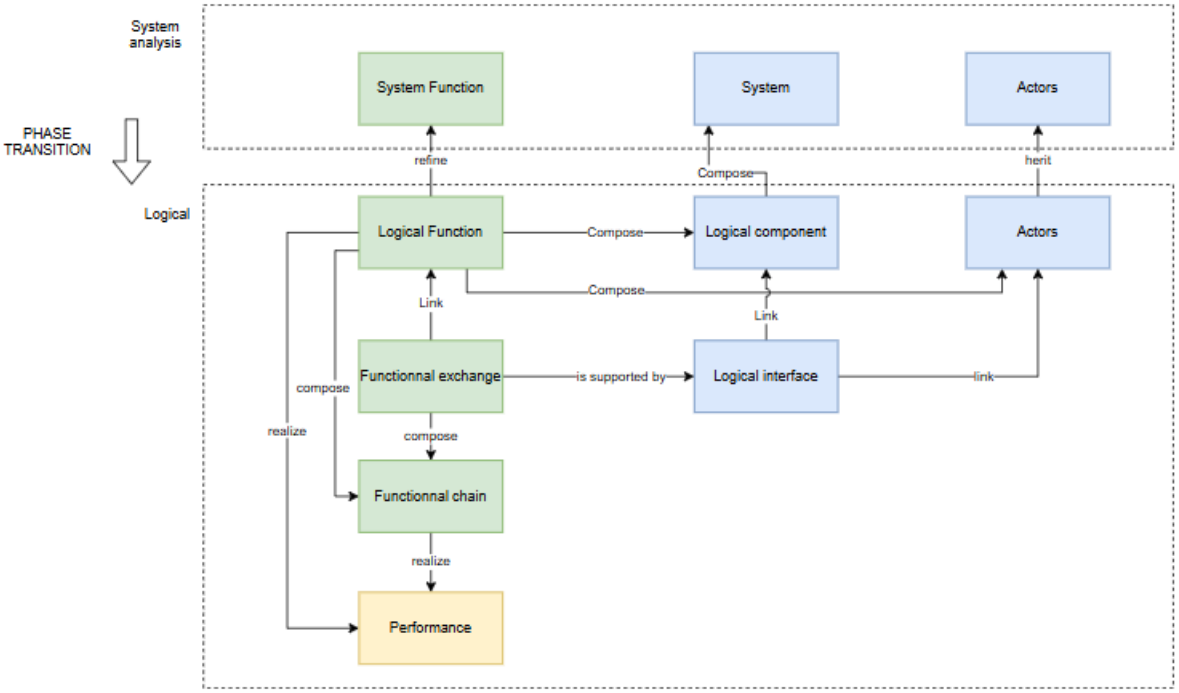


Figure 15: Logical analysis transition and components relation

3.3.6 Physical Analysis

The physical analysis aims to define the physical structure of the system. Physical components and interfaces are defined. They deploy logical components and logical interfaces inherited from the logical analysis. This analysis provides a physical view of the system through:

- Physical Architecture Definition
- Physical Breakdown Structure (PBS): a hierarchical decomposition identifying all hardware and physical components, clearly structured from the highest level down to individual elements.
- Physical Interface View: illustrates the interactions and connections between physical components, detailing interface characteristics (e.g., mechanical, electrical, communication protocols) and physical constraints.
- Allocation Matrix of logical functions onto physical components
- Physical Data Model identifying data exchanged through physical interfaces.

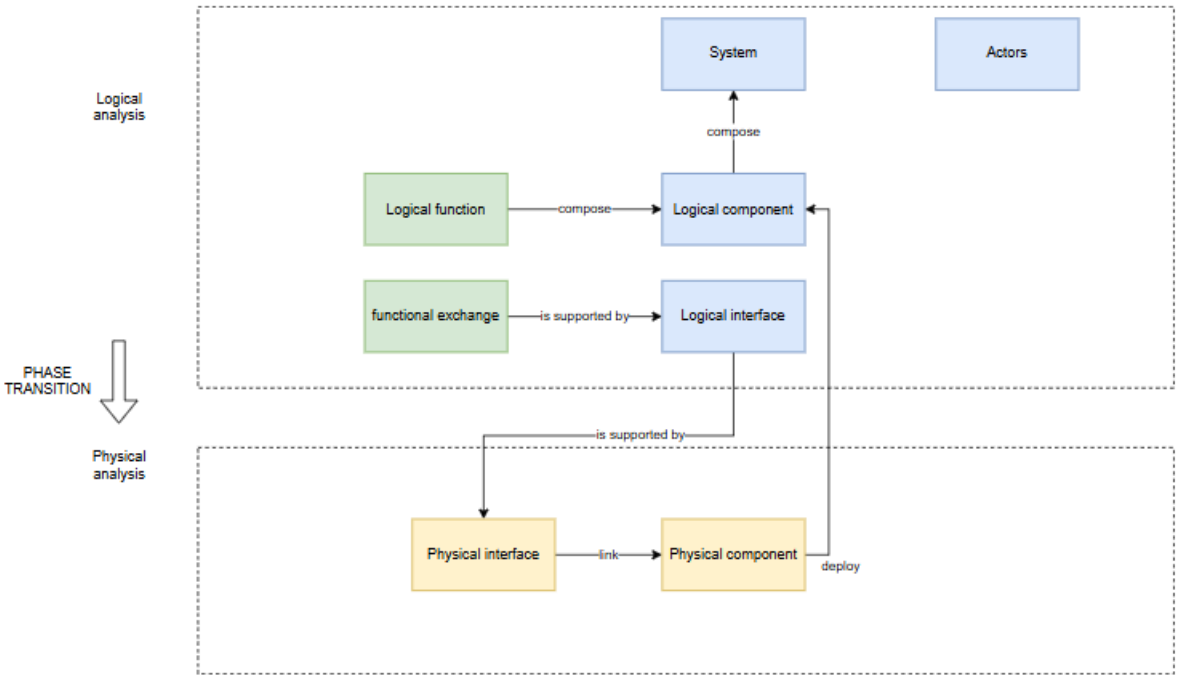


Figure 16: Logical anlysis to physical analysis transition and physical analysis components relation

3.3.7 Management

Integrating MBSE into a development cycle requires specific consideration within project organization. Therefore, dedicated technical processes must be included in the Systems Engineering Management Plan (SEMP), clearly defining modelling activities and interfaces with other systems engineering disciplines such as change management, configuration management, integration, validation, human factors assessment, risk analysis, and trade-off evaluations.

These processes supporting MBSE use should explicitly address the management of:

- Ontology: Establishing and maintaining a common vocabulary and structured representation of concepts and relationships.
- Tools and Specific Developments: Selecting, configuring, maintaining, and adapting MBSE tools tailored to project requirements.
- Ecosystem: Defining interactions, dependencies, and data exchanges within the broader engineering environment and among different stakeholders.
- Resources and Competencies: Identifying and securing necessary expertise, training teams, and ensuring ongoing skill development in MBSE.
- Collaboration Modes and Intellectual Property:
    - Clearly outlining collaborative frameworks, responsibilities and roles
    - Review and approbation process
    - Rules for managing intellectual property and information sharing
- Quality: requirement, style guide and control (completeness, consistency, clarity for e.g.)
- Evolution Management: Establishing robust processes for tracking, assessing, and managing evolutions in models, methods, tools, and project requirements throughout the project lifecycle.
- Implementation of traceability mechanisms linking MBSE artifacts to project requirements and validation criteria.
- Monitoring the effectiveness of MBSE integration through clearly defined metrics and continuous improvement practices.

First and foremost, processes must be tailored to project-specific needs. Therefore, a customized approach should be adopted, considering project complexity, scale, and specific challenges.

3.3.8 Ontology

Ontology is a formal, structured, and explicit representation of key concepts, relationships, properties, and rules specific to a technical or business domain. It provides a common framework ensuring consistency, interoperability, and shared understanding of models across different teams or tools involved in the systems engineering process.

SysML (Systems Modelling Language) is the standard modelling language for MBSE (Model-Based Systems Engineering). It provides a unified notation and semantics for specifying complex systems across multiple domains.

MBSE frameworks built upon SysML define structured processes for designing a system through successive abstraction layers—from high-level functional requirements down to detailed physical implementations.

As there is no single universally accepted reference process, project-specific methodologies are often established, clearly defining the modelling steps and artifacts required at each abstraction layer.

Libraries facilitate reuse of commonly-modelled elements or subsystems. Libraries help standardize and accelerate the modelling process, reducing redundancy and improving consistency across similar projects or systems.

Patterns provide systematic approaches for representing recurring architectural concepts and generic aspects of systems tailored to particular technological contexts or domain-specific requirements. These patterns allow engineers to leverage established best practices and enhance clarity, reliability, and efficiency in system modelling.

Among the classic design patterns applicable to railway signalling systems, the following can notably be identified:

- Digital Network Communication Layers:
  Establishing a communication architecture based on IEC 61375 segmented into functional layers, ensuring integrity (EN 50159), security (authentication, cryptography), and seamless data exchange between trackside/train-side equipment (in coherence with EIRENE FRS and SRS) or between onboard devices.
- Human-Machine Interface (HMI):
  Focusing on creating ergonomic user interfaces tailored to key railway system users, including train drivers (in coherence with ERA_ERTMS_015560), signalling supervisors, traffic regulators, and maintenance personnel.
- Control-Command Regulation Loops:
  These loops dynamically regulate critical operational parameters necessary for automated or semi-automated driving, such as precise traction and braking control. They are built around real-time regulation strategies that quickly and continuously adapt to varying conditions (rail-wheel adhesion, train load, track gradients, etc.) to maintain optimal performance, safety, and passenger comfort.

- Computer Vision:
  Based on automated analysis of images and video streams captured by onboard or trackside cameras, it enables advanced obstacle detection, automatic recognition of trackside signalling and signs, and detection of anomalies such as infrastructure deterioration or unauthorized access to tracks.
- Safety Programmable Logic Controllers (Safety PLCs):
  This pattern employs programmable devices specifically designed for reliably and safely executing critical railway signalling functions. Safety PLCs guarantee consistent execution of predefined scenarios, integrate fault detection mechanism.

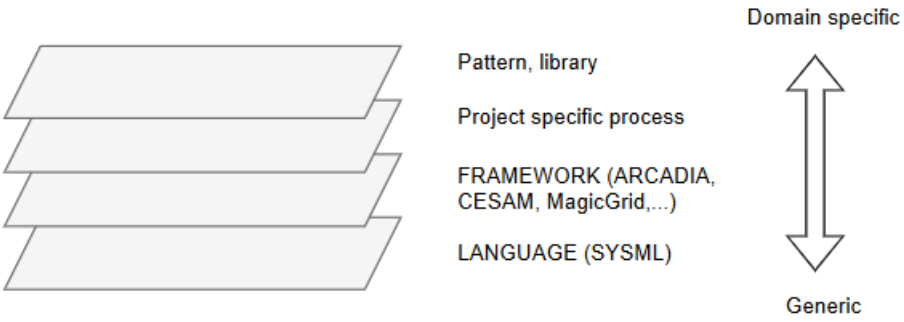


Figure 17: MBSE framework layers

The scope of MBSE is limited to the definition of systems or systems-of-systems. Systems-of-systems refer to interdependent sets of multiple systems delivering operational capabilities. For example, ERTMS/ETCS (European Rail Traffic Management System) is a system-of-systems that provides operational capabilities for safe management of train movements (including components such as RBC, ETCS, PKI, Eurobalise, FRMCS, ATO-OB) or traffic management (ATO-TS, TMS).

MBSE integrates within the systems engineering ecosystem and strongly interacts with external processes:

- Enterprise architecture: supported by specific framework (UAF, BPMN), MBSE is linked to strategies, stakeholder interactions with the system.
- Technical processes: requirement management, RAMS, V&V
- Project management:
    - Communication: MBSE provides high-level views to influential stakeholders or those with specific interests.
    - Planning: MBSE KPIs help assess project progress.
    - Cost: MBSE supports cost estimation and tracking of evolutions.
- Certification/Homologation: MBSE generates evidence incorporated into demonstration documentation.

MBSE relies on expert domain analyses and simulations from disciplines such as mechanics, hydraulics, electronics, etc.

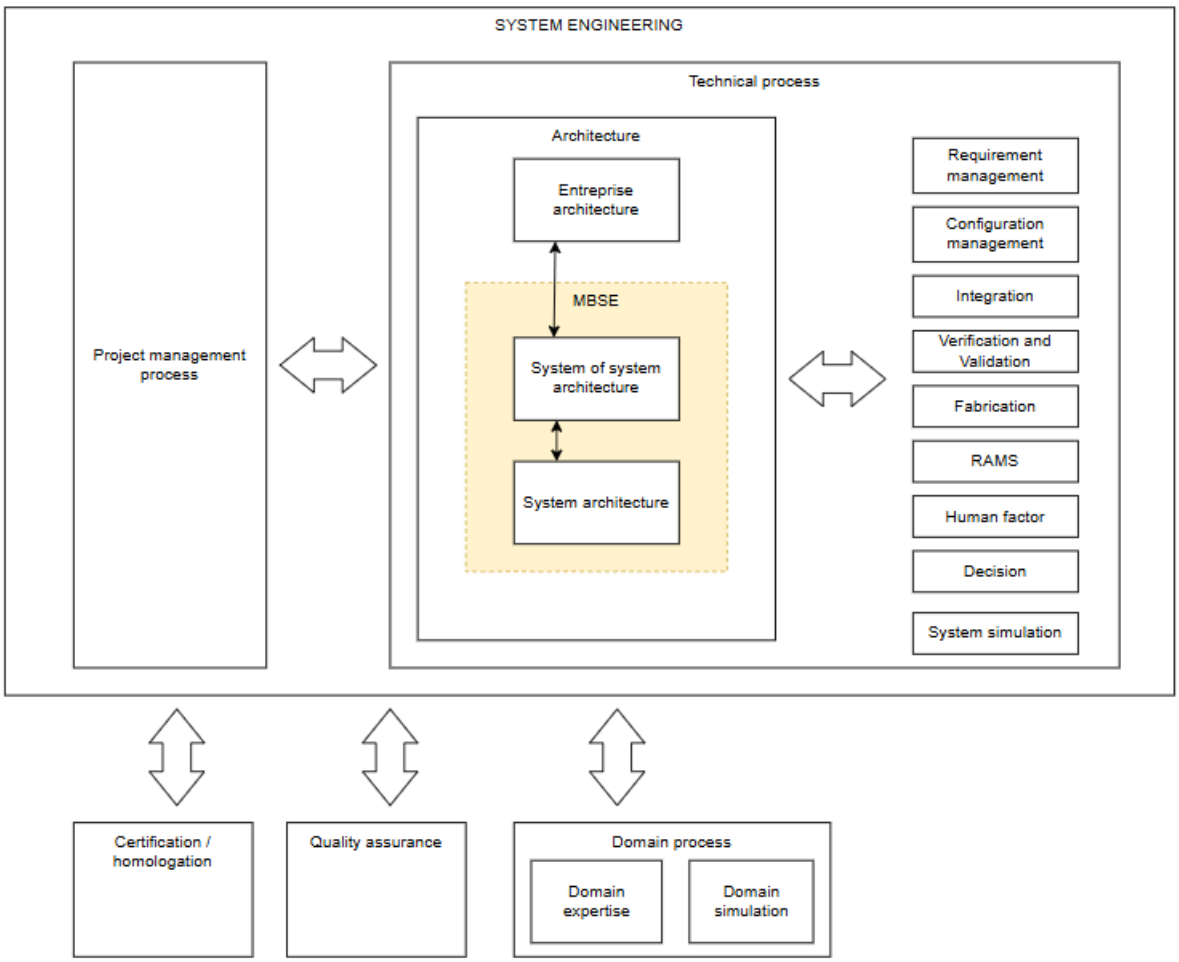


Figure 18: MBSE in SE ecosystem

A digital continuity can be ensured between these processes by relying either on integrated tools or on a decentralized approach with information exchange.
In particular, for the link with requirements management, an interface can be implemented between Capella and Polarion by leveraging model exports generated using the Python4Capella add-on. This interface also enables the export of model components to Excel.
Furthermore, the logical analysis can provide the subsystem block breakdown and the definition of interfaces forming the structure of a model that can be simulated in the Simulink environment.

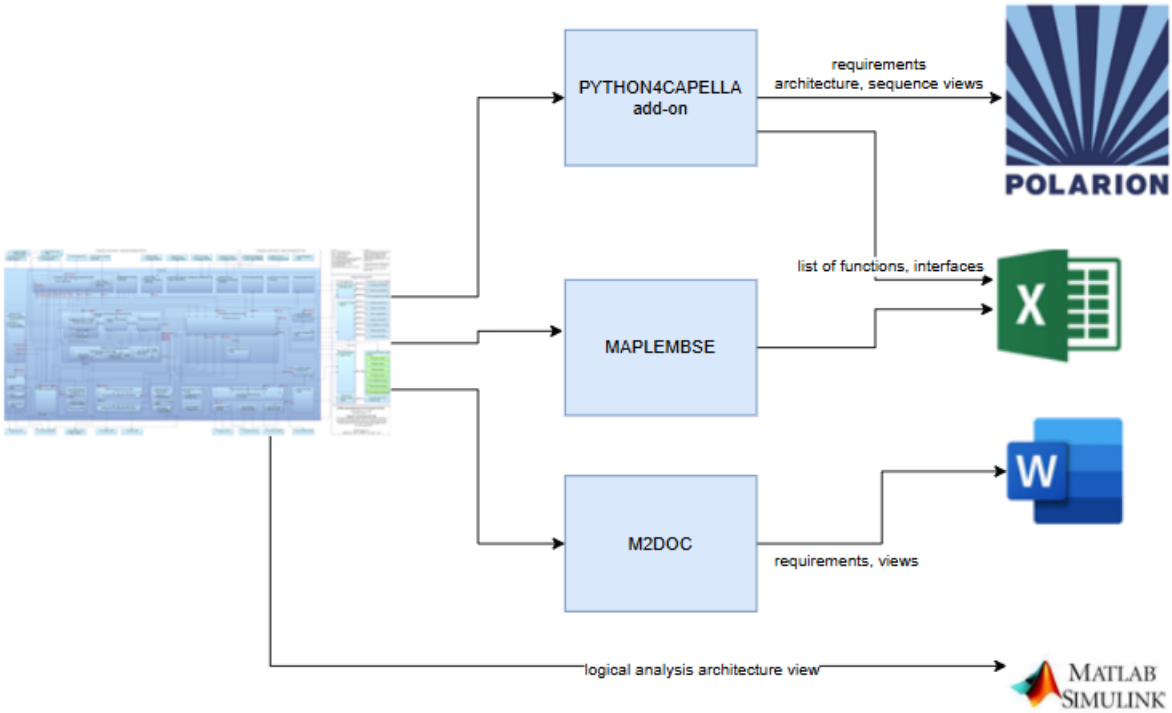


Figure 19: MBSE tools digital thread

3.3.9 Ressources

MBSE involves multiple roles and actors:

- System Engineer: Coordinates activities, ensures coherence of the system modelling process, and manages overall consistency.
- Modeler: A specialist in MBSE tools responsible for capturing architecture elements, modelling system structures, and generating relevant views tailored to stakeholders consuming model outputs.
- Subject Matter Expert (SME): Interacts with both the system engineer and modeler. While not directly interacting with the model, SMEs contribute analysing inputs, performing reviews, and using the outputs generated by the model.

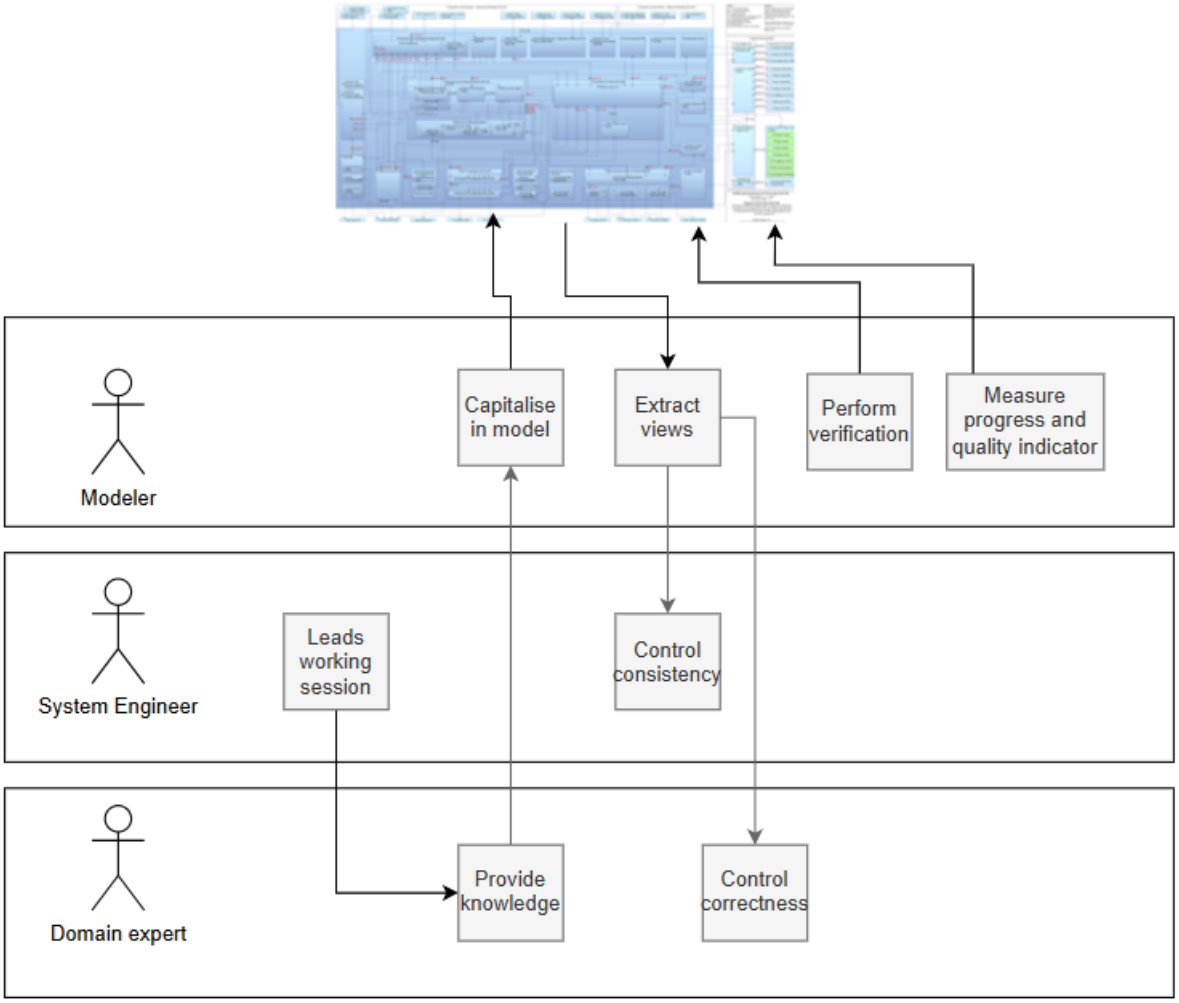


Figure 20: MBSE ressources

3.3.10 Collaboration

MBSE involves stakeholders with varying degrees of familiarity and sensitivity towards the methodology. Therefore, adopting MBSE should not become an obstacle but should instead be flexible enough to align with the audience's needs and expectations. To achieve this adaptability, tools and approaches must be carefully selected according to the participants involved and the maturity level of the system design.

At early design stages, when frequent changes occur and agility is essential, highly collaborative and flexible tools are most effective. Working initially on a physical or virtual whiteboard is beneficial, as it fosters flexibility and collaboration. Gradually, this preliminary work can transition to more structured diagramming tools such as PlantUML, Draw.io, or Visio. To streamline the later modelling process, it is recommended to agree early on a visual representation style closely aligned with the final modelling framework. Establishing this consistency upfront facilitates smoother integration and reduces effort when transitioning from informal discussions to formal system modelling.

More structured and specialized tools, such as Capella, become suitable once the initial scope and boundaries of the system have been clarified, and when modelling a stable initial version of the system seems feasible with a reduced likelihood of substantial changes.

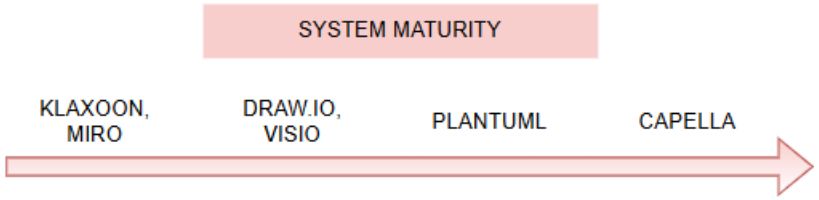


Figure 21: MBSE tools depending on maturity

3.3.11 Quality

The quality of the model is essential to ensure its usefulness and to guarantee the efficiency of the overall process.

- Completeness is achieved through the logical application of the method, which defines a step-by-step process, and through traceability that links components across all layers. This makes it possible to identify isolated components that may indicate gaps in the modelling.
- Consistency is achieved by the centralized definition of system components, ensuring that each component is defined only once. This approach guarantees the absence of duplicates and ensures a continuous and coherent description throughout the model.
- Understandability is achieved by defining a style guide, reviewing the model using checklists that include level-of-detail criteria, and adapting the views according to the target audience.

3.4 Model Based Design (MBD)

Model-Based Design (MBD) is a design methodology that employs executable virtual models to represent, explore, and verify a system's behaviour prior to its physical or software implementation.

We can distinguish between domain models, which address specific technical areas (e.g., mechanics, aerodynamics, fluid dynamics, etc.), and system models.

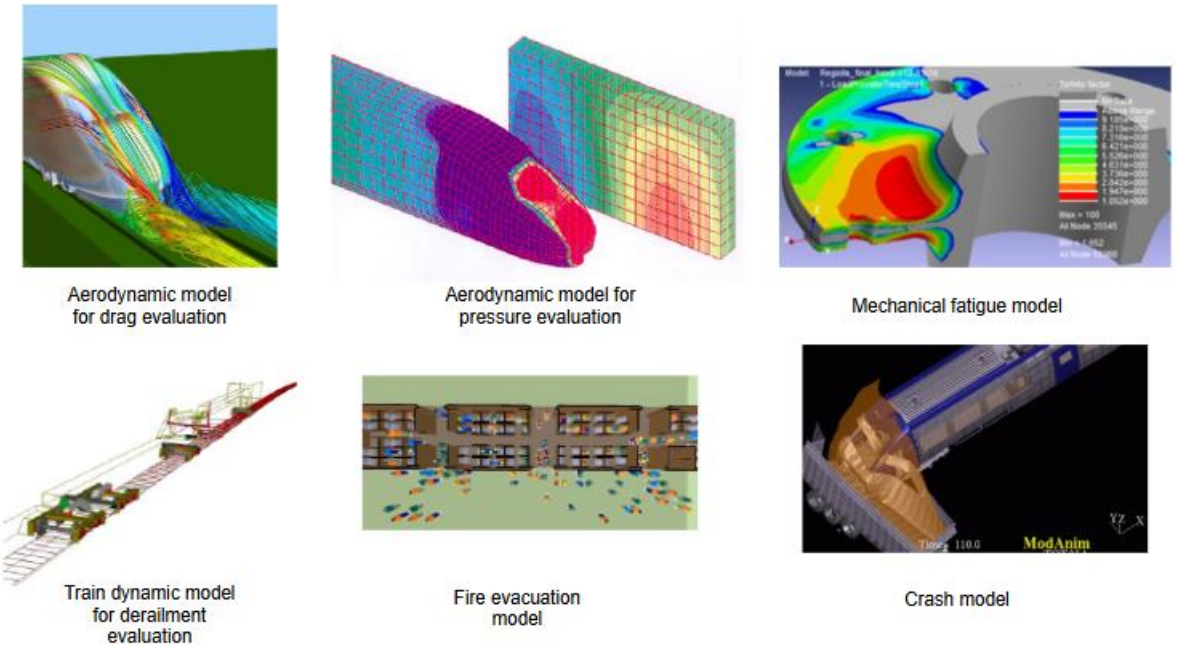


Figure 22: Domain models

System models aim to represent the behaviours and interactions within systems or systems-of-systems. Unlike domain-specific models, system models typically span multiple disciplines intrinsic to the systems themselves, as well as their operational environments. For example, signalling systems cover diverse technical fields such as electronics, computing platforms, software, dynamics, networking, and communications.

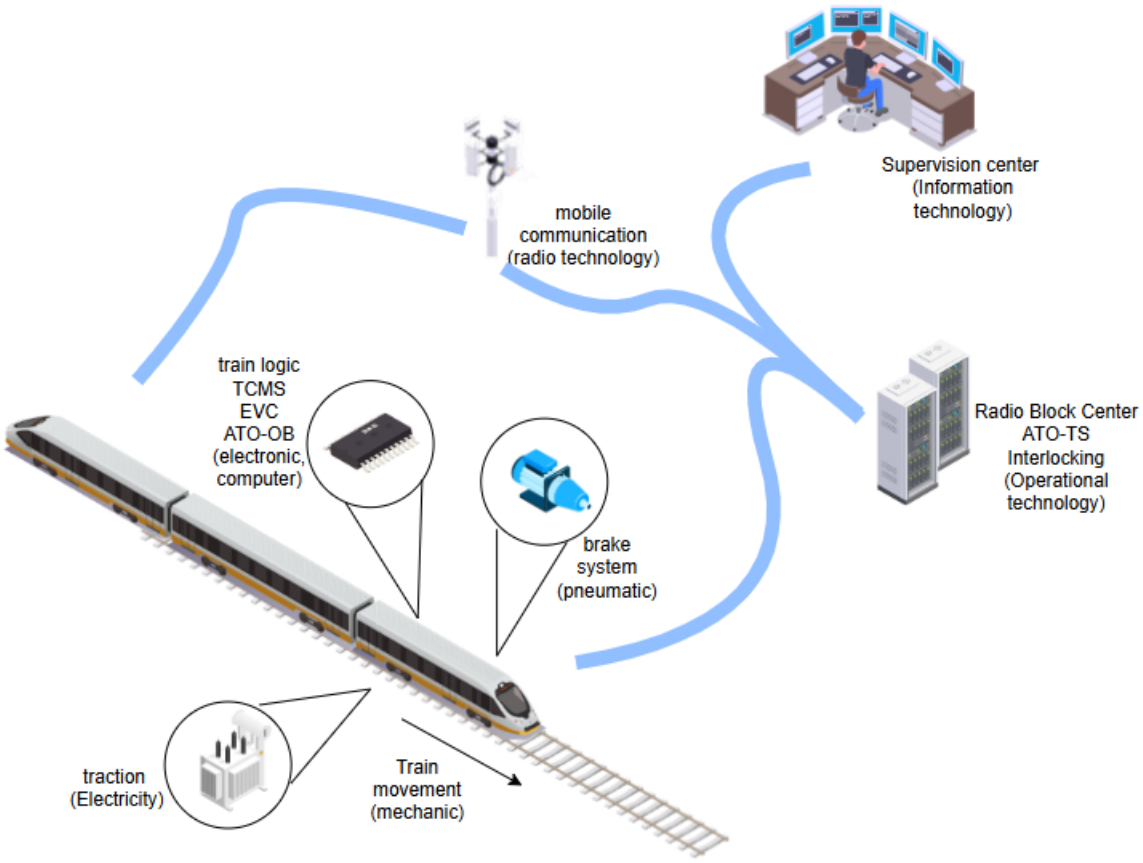


Figure 23: Systems of systems model scope for signalling

Modelling approaches are employed throughout the entire development lifecycle in various forms and at different levels of abstraction, enabling stakeholders to progressively refine, validate, and integrate their understanding of system behaviour and performance. From early conceptual design phases through detailed implementation and validation stages.

MBD is especially suitable for developing and testing "Software Defined" system and propose a progressive approach in the design cycle:

Dev. phase	**MBD**	
Concept / Architecture	Model in the loop - MIL	Controller logic and the system dynamics are represented as virtual simulation models.

Dev. phase	MBD	
Design	Software in the loop - SIL Processor in the loop - PIL	SIL: control algorithms are implemented as actual executable code, while the controlled system remains simulated PIL: involves executing the generated software on the actual target processor or a similar hardware platform, with the rest of the environment simulated. The objective is to validate and optimize software performance and compatibility directly on target processors
Verification and validation	Hardware in the loop - HIL	The actual hardware components (controllers or embedded systems) are tested against a real-time simulation of the physical system
Operation	Digital twin	Model is used as a reference to compare with the real system behaviour

Table 7: MBD in the development cycle

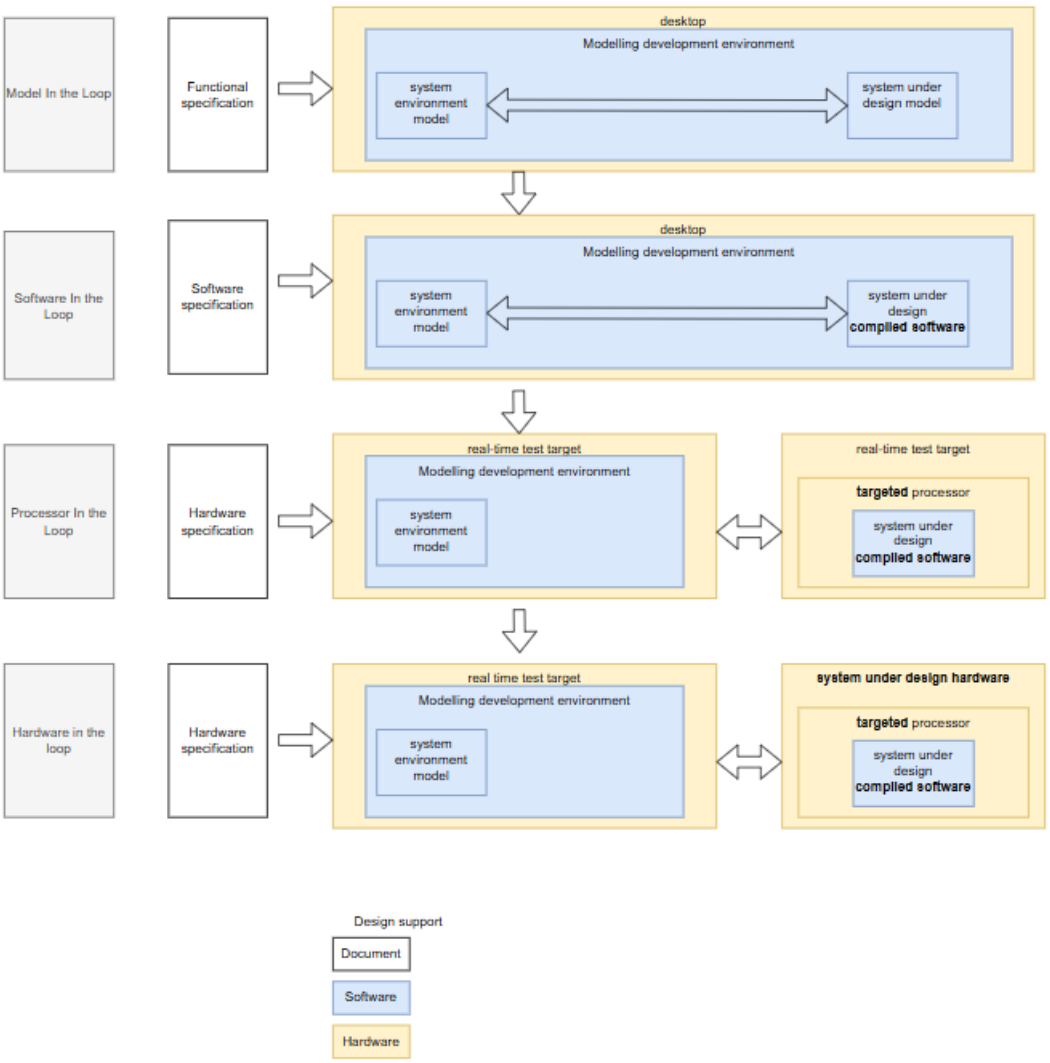


Figure 24: MBD process

In the development cycle, Model-Based Design (MBD) serves as a bridge between the initial requirements, documented system specifications, and field testing. MBD significantly reduces the likelihood of introducing errors throughout each step and accelerates development phases through rapid prototyping, especially by quickly evaluating a large number of configurations. Its implementation cost is notably lower compared to bench testing and substantially lower compared to on-track tests.

In 2016, SNCF Group launched its most ambitious R&D initiative: the "Autonomous Train Program," aiming to explore the autonomous train concept within the railway sector. Under this program, the DOS and TAS projects evaluated computer-assisted vision technologies for obstacle detection and signal recognition. These technologies were validated through field tests conducted using freight trains, while autonomous driving performance was assessed on passenger trains (Regio2N).

Another example is the DB AutomatedTrain program, with significant achievements in system modelling.

These projects involved consortiums comprising various railway companies, independently developing autonomous train concepts suitable for passenger and freight services. The objective

was to incrementally test, under real-world conditions, different autonomy levels from GOA2 to GOA4.

Both projects successfully demonstrated GOA4 operation within specific experimental contexts and generated significant findings integrated into previous European initiatives such as X2RAIL4 and the ongoing ERJU project.

Similar trials were conducted by Swiss Federal Railways (SBB-CFF) in 2021, evaluating GOA2 to prepare for introducing Automatic Train Operation (ATO) specifications into the 2023 TSIs:

- Passenger services: Lausanne–Villeneuve (30 km)
- Freight services: Sion–Sierre (17 km)

However, these trials remained limited due to significant operational constraints, including rolling stock modifications, restricted test routes, obstacle variability, operational speeds, traffic frequency, environmental conditions, and high associated costs (modification, immobilization, track access fees, etc.). While valid at the time of testing, these trials could not efficiently accommodate subsequent updates (such as error corrections or new functionalities).

The identified limitations were primarily due to system complexity and the vast number of test scenarios required to enhance system knowledge and maturity. To address this, simulation tools, modelling techniques, and virtual testing platforms provided critical support. These methods offer initial feedback about system behaviour in an entirely controlled environment managed by the testing team. This approach helps avoid complications inherent in real-world testing, such as rolling stock failures, environmental uncertainties, and other uncontrollable factors. Both the test environment and scenarios can thus be fully controlled, facilitating comprehensive data collection about system performance. Even though not all potential scenarios were covered during the program, this methodological approach clearly demonstrated its value in managing and understanding complex systems.

Figure 25: ATO Freight train SNCF project

Figure 26: ATO Freight train SNCF project – obstacle detection

Figure 27: ATO Passenger service SNCF project

Model-Based Design (MBD) both guides and enriches the development process but can also be enhanced by feedback from field tests and operational data. Indeed, today's complex systems generate vast amounts of data (computer vision, localization, etc.). Leveraging such real-world data facilitates the fine-tuning and training of machine learning-based systems. For instance, capturing video data of railway signals under various weather conditions not only helps qualify equipment but also provides valuable knowledge bases for training visual recognition algorithms.

Additionally, operational measurement data can serve as replay datasets for simulation, performance benchmarks to evaluate improvements introduced by new functionalities, and resources for training algorithms related to predictive maintenance, localization, and computer-assisted vision.

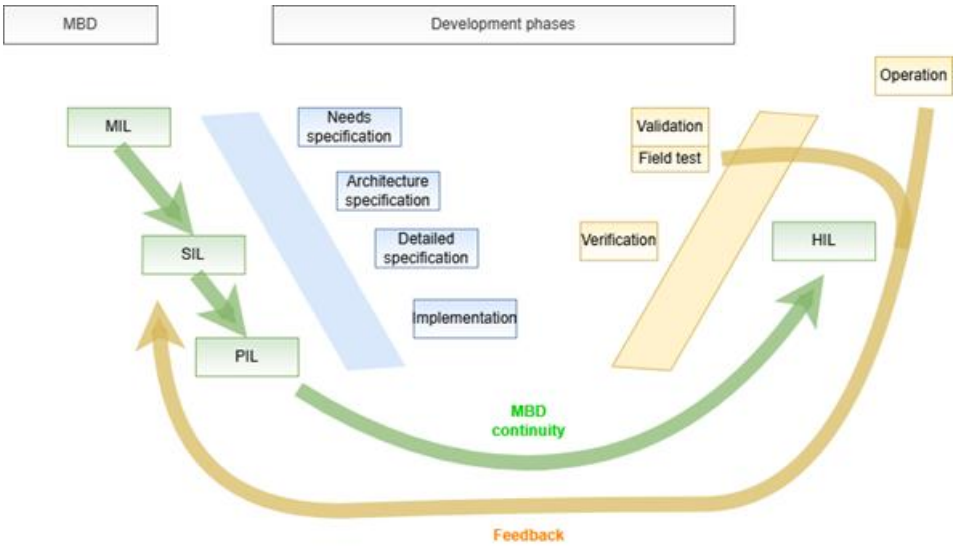


Figure 28: MBD loop

3.4.1 Software quality development

The EN 50716 standard defines project organization, activities, methods, and artifacts required to ensure software development quality, specifically addressing expected performance targets related to safety (software integrity level).

A Model-Based Design (MBD) approach is particularly suited for safety-critical software development, offering several key advantages:

- Traceability of requirements:
  It ensures explicit traceability from high-level requirements through detailed software design, significantly reducing ambiguity and improving compliance with safety standards.
- Progressive abstraction and complexity management:
  It facilitates incremental progression from conceptual modelling toward detailed implementation, allowing engineers to manage complexity effectively by starting at a high level of abstraction.
- Early activation of automated verification:
  The use of model-based techniques enables the early execution of automated verification and validation processes, identifying potential issues much earlier in the software development life cycle, thus minimizing risks and cost overruns.
- Modular approach:
  It supports modular software architecture, enhancing maintainability, reusability, and facilitating parallel development, integration, and testing of software components.
- Interfaces with requirements and change management tools:
  It integrates seamlessly with requirements management and change control systems, allowing efficient updates, clear communication of changes, and robust configuration management throughout the project.

- Continuity throughout development and integration phases:
  It establishes a consistent digital thread, enabling smooth transitions from design and development to integration, validation, and ultimately certification, enhancing overall coherence and reducing development time.

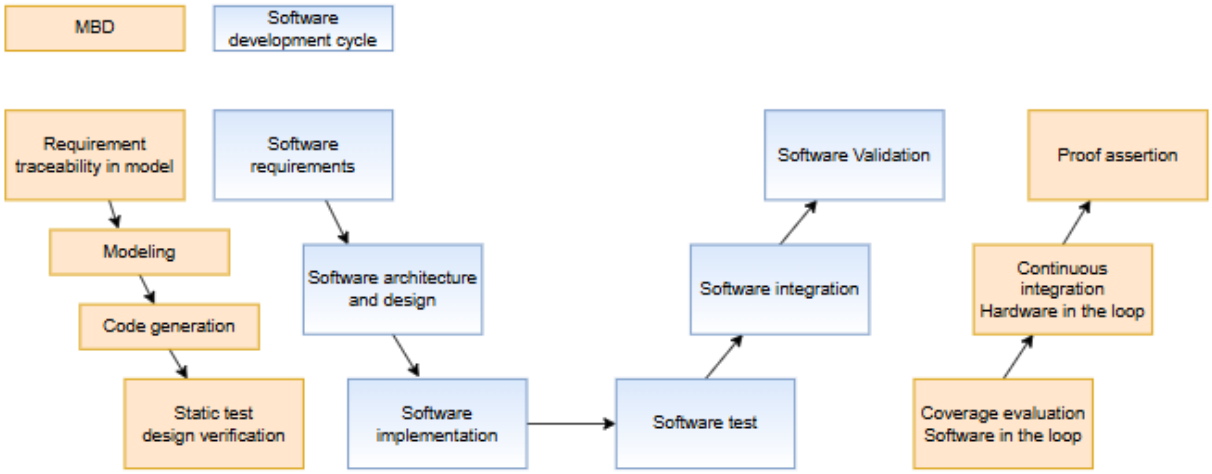


Figure 29: MBD and software development quality

3.4.2 Co-simulation

Co-simulation is a technique enabling simultaneous simulation of multiple models originating from different domains or tools, each representing a specific part of a complex system. This approach allows continuous interaction between these models during the execution of an overall simulation.

From the perspective of railway signalling systems, ETCS (European Train Control System) functionalities are closely correlated with train dynamics. Therefore, accurately simulating ETCS operations requires modelling these dynamic aspects.

In freight trains, mechanical clearances within couplings result in complex compression and tension phenomena. These mechanical behaviours significantly influence overall train dynamics. Additionally, braking initiation throughout a train is governed by pneumatic phenomena, specifically the propagation of pressure changes within the braking command circuit along the length of the train.

To achieve a representative simulation environment for ETCS systems, incorporating these mechanical and pneumatic phenomena is essential. This integration can be effectively managed by distributing the modelling across specialized, adapted simulation environments. For instance:

- Train and mechanical characteristics modelling can be executed using railway dynamics software, such as SIMPACK, capable of accurately capturing the complex coupling interactions and resulting train dynamics.
- Pneumatic phenomena modelling, which involves air pressure dynamics throughout the braking system, can be addressed using Multiphysics libraries within system modelling tools like Simscape integrated into Simulink.
- ETCS functional modelling, covering the signalling logic and safety protocols, can be developed using system modelling software platforms such as Simulink or SCADE.

A partial implementation of these components is detailed in Deliverable D31.5.

Interoperability among these distinct modelling environments is facilitated by adopting the Functional Mock-up Interface (FMI) standard (ERJU-726), ensuring seamless integration and co-simulation capabilities.

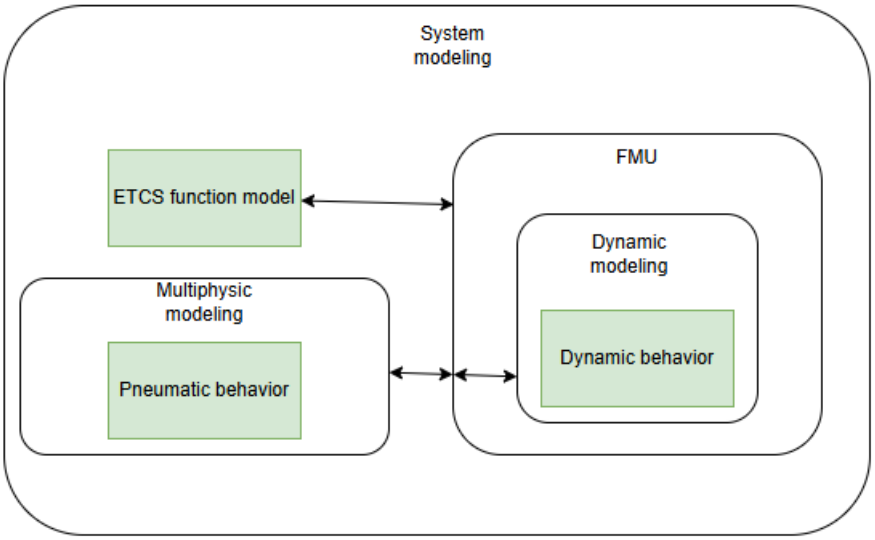


Figure 30: co-simulation

3.4.3 MBD and standardization

The MBD approach is also used in certification processes. Indeed, certain standards integrate this approach for system validation.

This is the case for pantographs according to the EN 50138 standard "Railway applications - Current collection systems - Validation of simulation of the dynamic interaction between pantograph and overhead contact line". Carrying out simulations helps reduce on-track testing.

The ERA braking curves tool is another example of the MBD approach for standardization. This tool is provided as a supplement to subset-026, "System Requirement Specification," allowing simulation of the supervision curves calculated by the ETCS based on configurable input data (characteristic deceleration, track profile, etc.). This model enables ETCS designers to compare their system's behaviour against a reference.

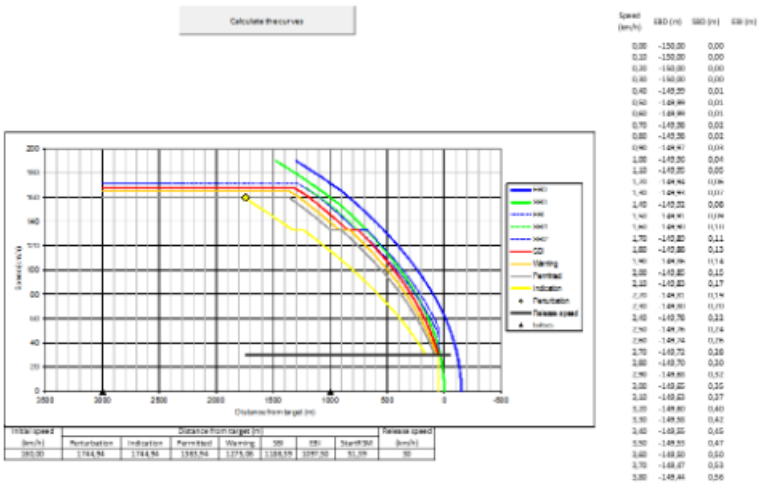


Figure 31: ERA braking curves tool

4 ATO SYSTEM FOR GOA2 AND GOA4 AUTOMATIC MOVEMENT USE CASES

To illustrate the modelling process, we propose to apply it to the following Automatic Train Operation use cases:

- Multiple unit train in GoA4 between station and depot

- Multiple unit passenger train GoA2

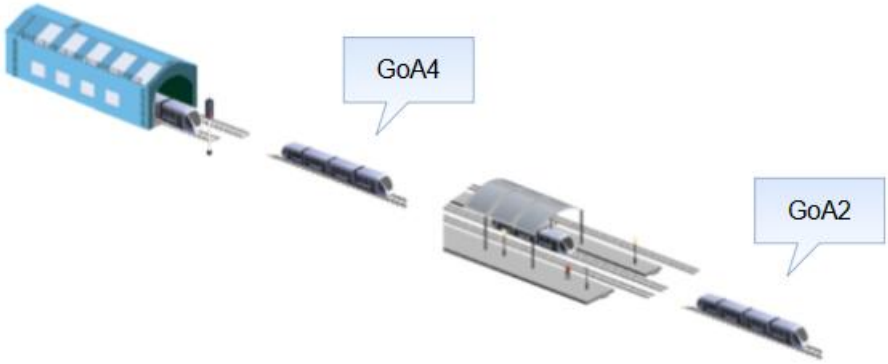


Figure 32: WP31.1 subtask depot to station GoA2/4 use case

Since this is only an example, the architectures presented should not be considered as a reference, as they include simplifications and assumptions made for the sake of clarity (cf. §2 limitations).

In addition, the scope of each of these use cases is limited to energy optimization (GoA2) and ATO on lineside signaling (GoA4).

4.1 CONTEXT

The ATO use case has been defined considering the state of the art in ATO system maturity, as well as anticipated future needs based on ongoing experiments and announced projects. It addresses challenges where barriers still exist—either from a commercial standpoint, where the return on investment has yet to be demonstrated, or from a technical perspective, where such barriers could be tackled through use cases with a limited operational scope.

4.1.1 Automatic Train Operation

In order to improve rail system efficiency, ATO system is introduced as a part of the signalling systems.

T

here are different levels of automation for rolling stock, called GoA (Grade of Automation):

- GoA1: The driving is manual and there is an Automatic Train Protection (ATP) system. Its role is to protect the train from the risk of collision with another train by automatically triggering emergency braking if necessary.
- GoA2: In addition to the presence of ATP, the traction/braking function of the train is automated thanks to ATO. The driver is still present to handle degraded situations and monitor the presence of obstacles on the track. He also has the responsibility to activate (engage) ATO or not.
- GoA3 and GoA4: The train is completely autonomous throughout the entire mission. The train analyses its environment and signalling data to make driving decisions.

GoA2 definition is part of ERTMS/ETCS introduced in the 2023 TSI.

ATO aims to deliver performance gains across several key areas.

- (H) Punctuality: In GoA2 operation, ATO enables precise adherence to the timetable and activates catch-up strategies when delays occur, ensuring accurate passage through key points and precise arrival times at stations.
- (E) Energy efficiency: By optimizing traction and braking phases, ATO minimizes energy consumption while still meeting operational constraints.
- (S) Stability: On high-frequency lines where trains operate with short headways, ATO helps ensure smooth and coordinated movements, contributing to overall network stability.

GoA4 ATO builds upon these GoA2 benefits while also addressing driving resource optimization and operation flexibility (R), by enabling autonomous operation in certain segments.

These benefits can be fully realized when ATO is deployed on specific, well-targeted operational scenarios.

Operational domain	**GoA2**	**GoA4**
Medium traffic line	(H) (E) *	
High traffic line	(H) (E) * (S)	(R)
Technical movement between depot and station		(R) *
Station turnaround		(R)

(*) Scope of WP31 use cases

Table 8: WP31 scope regarding operational domain

4.1.2 ERTMS/ETCS Deployment

In the ERTMS/ETCS architecture, ATO is closely linked to the ETCS (European Train Control System). This means that in order to benefit from ATO functionalities, the infrastructure must be equipped accordingly. According to the TSI (Technical Specifications for Interoperability), the deployment of ATO falls under the responsibility of the infrastructure manager. However, as of today, not all infrastructures are equipped with ERTMS/ETCS, and deployment will follow national implementation plans defined by each Member State.

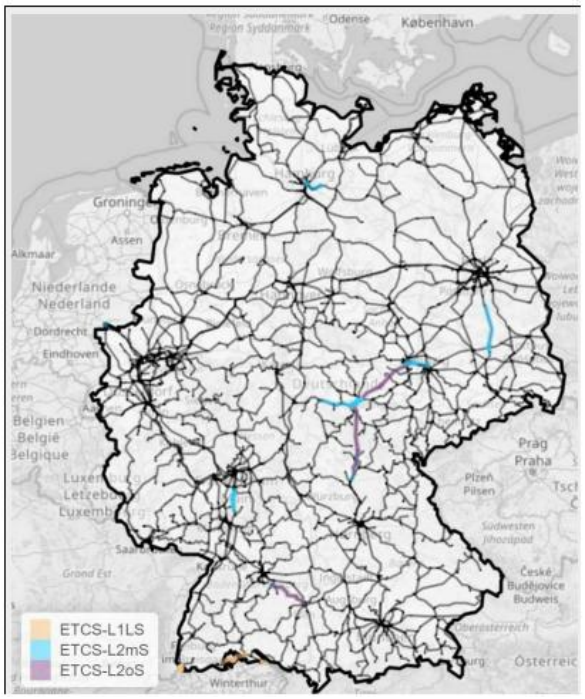


Figure 33: current deployment of ETCS in Germany [ref ERJU-684]

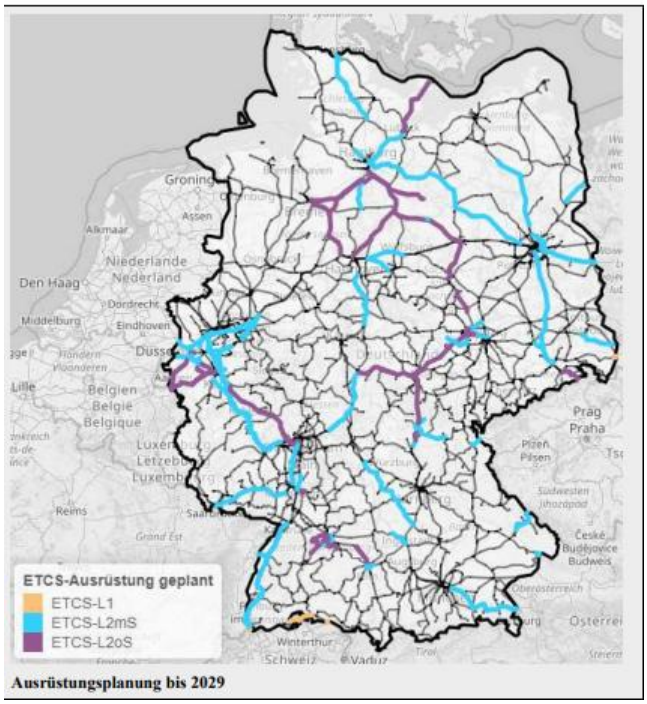


Figure 34: planned deployment of ETCS in 2029 (ref ERJU-684)

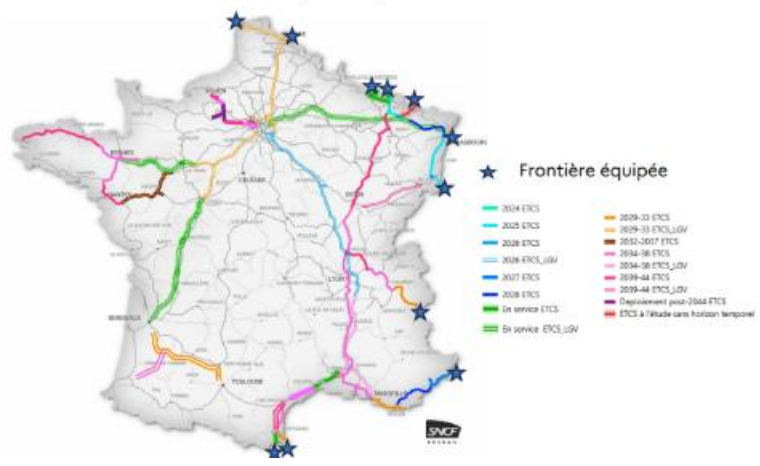


Figure 35: French National ETCS implementation plan [ref. ERJU-686]

The deployment of a system such as ERTMS/ETCS, particularly on existing lines, raises the challenge of asynchronous trackside and onboard equipment installation. In order to operate the line, a sufficient rate of train equipment must be achieved, which can take several years. Deployment strategies must therefore be developed to allow installation without interrupting train operations. Each country has defined its own approach. In Switzerland, ERTMS/ETCS equipment was initially deployed while still supporting national functionalities (such as ZUB). In France, the ERTMS/ETCS system is overlaid onto national systems, allowing both ETCS-equipped trains and non-equipped trains to operate simultaneously.

In line with the progressive deployment strategy for ATO, and leveraging existing lineside signalling, the GoA2/4 use cases investigated through simulation in WP31 focus on illustrating the ATO implementations compliant with the ERTMS/ATO European specification. The analysis also considers the integration of adaptor mechanisms designed to ensure interoperability and operational continuity throughout the transitional phase of ERTMS/ETCS deployment.

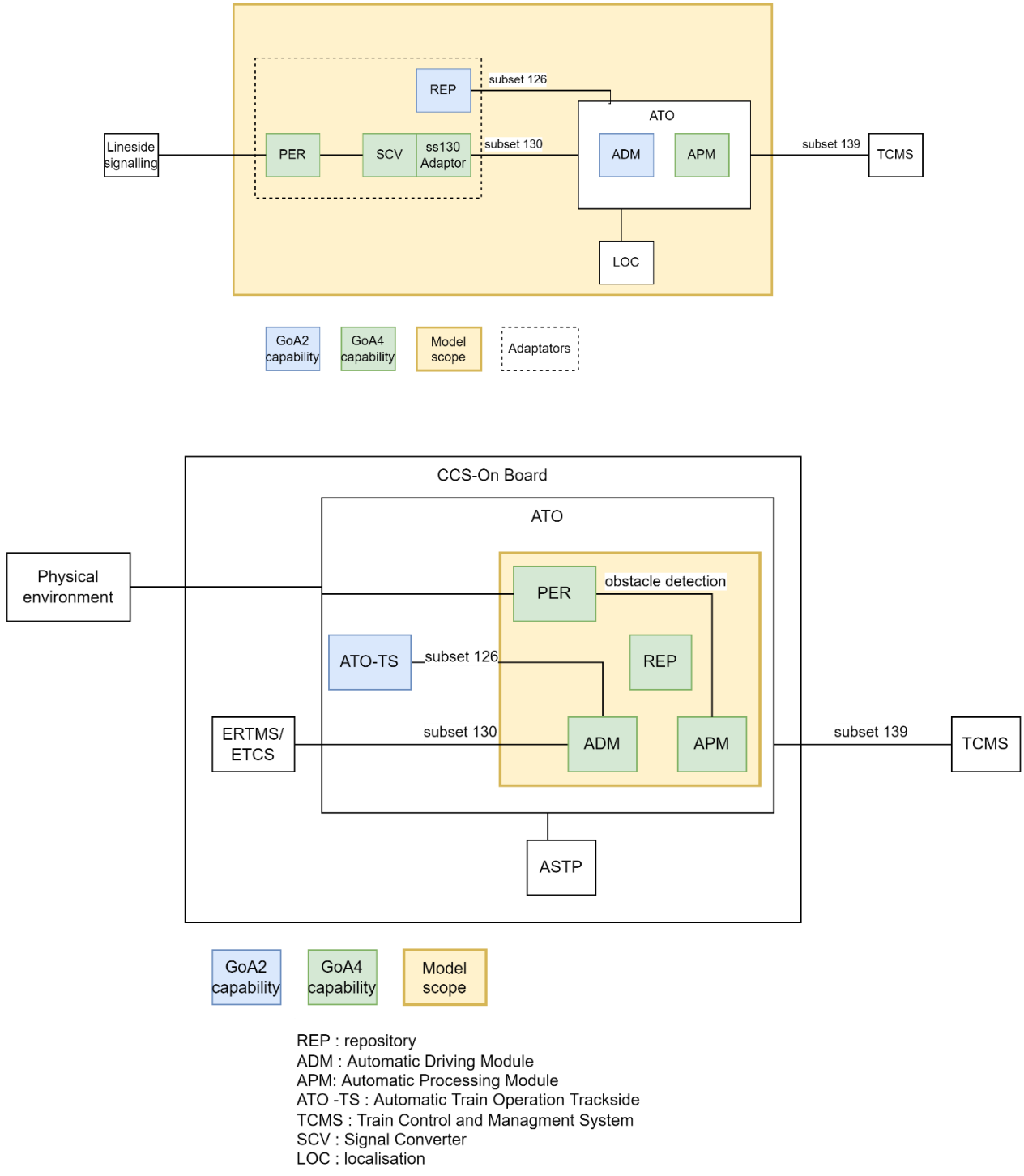


Figure 36 – ATO ready vs targeted architecture

Note related to figure 36:

- GoA4 capability extends GoA2 capability.
- Perception system is GoA4 capability in the context of this specific project.

4.1.3 Depot to station operation

Train operations require regular maintenance. For this reason, trains frequently return to maintenance facilities known as "depots," where inspections and necessary interventions can be carried out (such as brake shoe replacement, refilling of water, sand, fuel, etc.). These depots are generally located within a few kilometres from the stations. This proximity necessitates regular movements of trains between the depot and the stations, both before and after maintenance activities, in order to ensure continuity of commercial service.

These technical movements can be carried out either by specialized drivers or by mainline drivers. Specialized drivers are attached to the maintenance depot and are only authorized to operate trains between the depot and the station. In contrast, mainline drivers are generally authorized to drive trains beyond this segment, on one or more main lines, to perform commercial services. Involving mainline drivers allows for seamless continuity between the start of their shift and the beginning of the commercial mission.

However, the organizational process involved—starting the duty, transferring to the train's location, performing preparations, and moving the train to the station—requires driving time that cannot be dedicated to commercial operations.

The GoA4 use case presented here aims to address these operational scenarios by automating the non-commercial movements, thereby freeing up mainline drivers for commercial services.

This use case also responds to the need for greater operational flexibility, allowing movement demands to be fulfilled regardless of driver availability constraints.

The Japanese operator JR East has expressed its intention to deploy this use case for the operation of Shinkansen trains between Niigata Station and the depot. [ref. ERJU-679]

An experiment was conducted in 2021 in Hamburg to demonstrate train turnaround operations at a station. [ref. ERJU-681]. The GoA4 movement enables round trips to and from the stabling area, allowing the train to switch tracks.

In the FP2 project, the demonstrator developed in WP38 (Tasks 38.3 and 38.4) focuses on technical movements between the depot and the station. WP31, through its model-based and simulation-driven approach, serves as an intermediate level of representativeness between WP6 (specifications) and WP38 (demonstration). Indeed, modelling enables the specifications to be challenged through implementation, and allows for more flexible and extensive validation than what is achievable through field testing alone. While WP38 is constrained by the organizational context of on-site testing, it enables evaluation with physical systems in a representative environment.

4.2 OPERATIONAL ANALYSIS

The operational analysis aims to identify all stakeholders involved, along with their activities that contribute to fulfilling the technical movement mission. This analysis focuses on capturing the needs of each entity without prescribing any technical solutions. The purpose of the analysis was to capture actual practices, identify operational constraints, and determine the critical points that may influence the future design of technical solutions.

To achieve this, a structured approach was adopted. It began with interviews involving railway stakeholders: drivers, traffic controllers and maintenance staff. These discussions helped gather diverse experiences, highlight practices that are not always written in official documents, and provide insight into the day-to-day decisions made by field operators.

In addition, field visits were carried out to directly observe railway operations in real conditions. These visits offered a concrete view of drivers' working environments, the interaction between humans and technical systems, and the challenges encountered during operations. They also helped to identify gaps between prescribed procedures and actual practices, showing how flexibility is applied to maintain safety and punctuality.

Finally, the analysis included a systematic review of existing procedures, such as driving instructions, operating rules, and communication protocols between actors. This documentary review served as a reference point to confirm or refine the findings from interviews and field observations.

By combining these sources, a holistic view of operational functioning was built. This view provides a foundation for identifying future functional needs, defining potential automation scenarios, and anticipating the impacts on operational staff. It ensures that future technical decisions remain aligned with real-world operations while respecting the specific requirements of safety, performance, and reliability in the railway domain.

The ATO system architecture, taking into account this operational analysis, is addressed in section 4.5.

4.2.1 Activities and interactions

The technical movement mission is structured around five main phases:

1. Planning Phase
   During this phase, the fleet manager defines the maintenance schedule for the trainsets and requests the required entry and exit train paths. Based on this schedule, the depot manager assigns daily movement tasks.
2. Train Preparation Phase
   The driver collects their mission sheet, inspects the train, starts it up, and signals that it is ready for departure. The depot manager initiates the routing procedure from the depot to the station, in coordination with the traffic controller.
3. Departure Phase
   Once the signal is cleared, the driver sets the train in motion. They maintain the appropriate

speed while complying with light signals, trackside signage, and operational rules based on route knowledge. The train is brought to a stop at the designated platform marker in the station.

4. Station Arrival Phase
   Upon arrival at the station, the driver ensures that the train is safely parked and properly secured while awaiting the start of its commercial service.

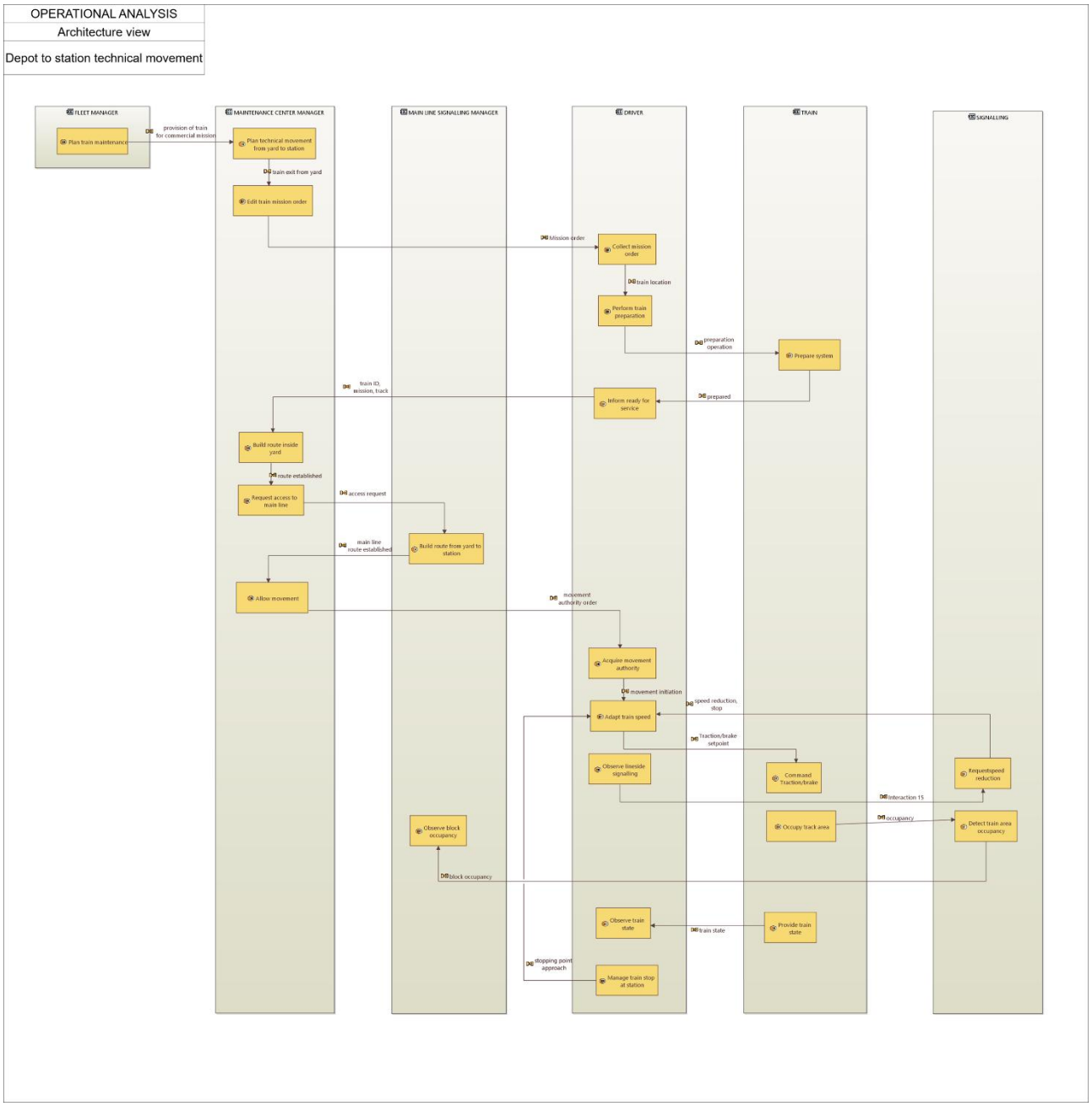


Figure 37: Operational analysis, yard to station technical movement

Allocating activities to the ATO technical system then enables the definition of the system's scope.

Initial actor	Activities	GoA2	GoA4
Driver	Observe lineside signalling		x
Driver	Collect mission order	x	x
Driver	Adapt train speed	x	x
Driver	Acquire movement authority		x
Driver	Observe train state		x

x : activities allocated to ATO on-board

Table 9: Activites allocation to ATO

4.3 ATO SYSTEM CONCEPT OF OPERATION

This chapter outlines the concept of operation considered in this project. It provides a comprehensive overview of the system's principles, behaviour, interactions, and performance. The chapter details the system's objectives, scope, and the roles and responsibilities of various stakeholders.

The concept presented in the project focuses on autonomous operation (GoA4) for the technical movement between the depot and the station. Autonomous movement is only initiated upon exiting the depot (movements within the depot are out of scope). Between the depot and the station, the system takes into account signalling information, and stops at a predefined point at the station platform. The concept also includes automatic operation in commercial service under GoA2. The concept is illustrated in Figure 38.

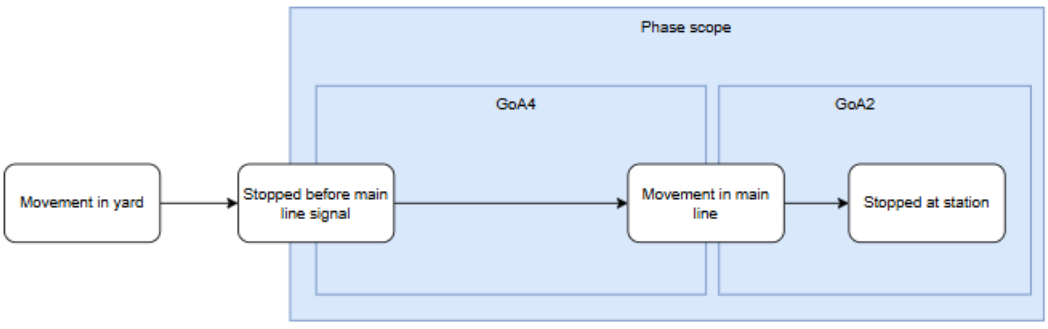


Figure 38: Use case scope

The scope was selected to ensure a reasonable modelling effort, considering both the system's complexity and its comprehensibility. The objectives are those defined in the R2DATO project. Especially objectives are linked to R2DATO key performance metric (see §4.3.3)

4.3.1 Exit of the yard

The activation of ATO upon exiting the depot is carried out by the depot driver. A technical process must ensure that the driver has left the train before initiating the autonomous movement.

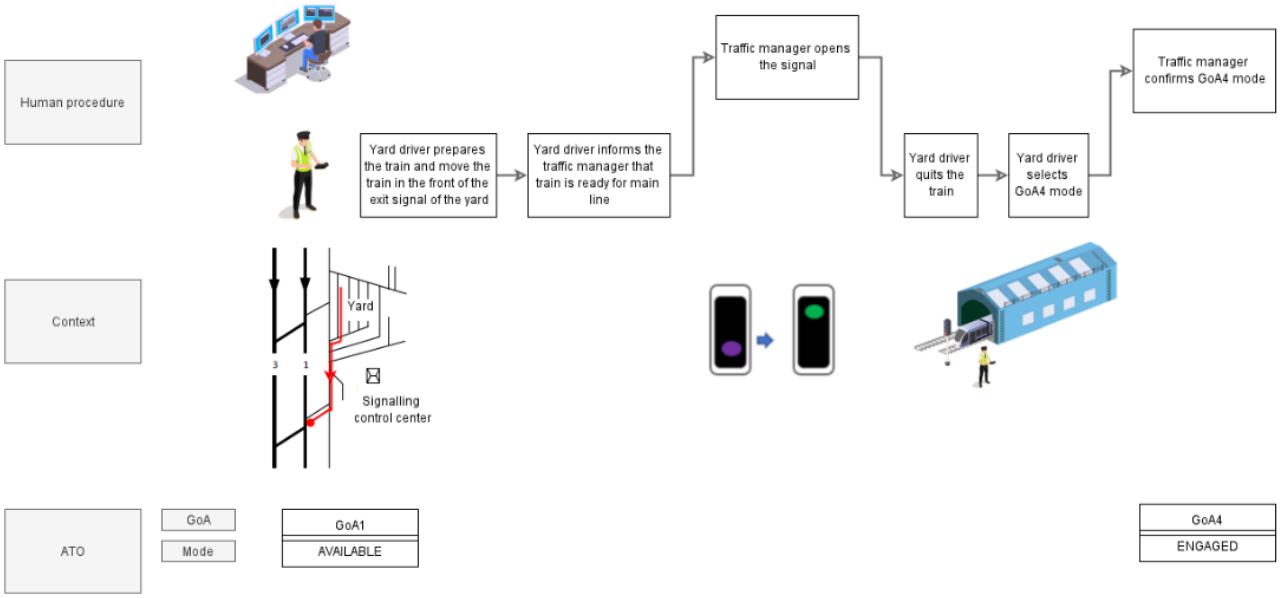


Figure 39: Use case scope

4.3.2 Depot to station movement

On the route between the depot and the station, the system must adapt to trackside signalling and adjust speed accordingly. In case the signalling requires a stop, the resumption of movement must be carried out safely. A traffic supervisor monitors the movements of the train fleet between the depot and the station, receives alert messages from the onboard ATO systems (such as train failures, stops in front of a red signal, signal state inconsistencies, etc.), and takes appropriate actions.

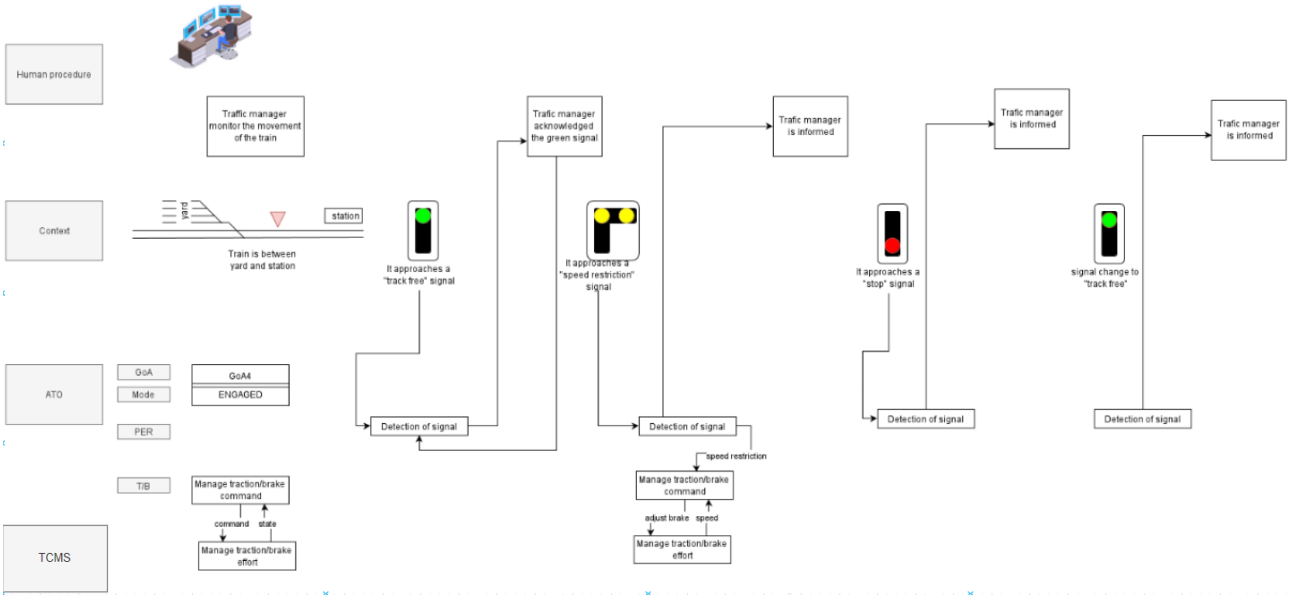


Figure 40: Depot to station concept of operation

4.3.3 GoA4 over lineside signalling

The principles for recognizing lineside signals follow the general principles TAURO project, while the GoA4 architectural principles follow those defined in the X2RAIL project, updated by WP6 (refers to task 6.5 and 6.3 for reference architecture view – cf §2 limitations).

The visual signals are detected and identified by the perception subsystem. The conversion module translates the perceived signals into movement authorizations. The ETCS 'braking curves' function generates control curves (EBD, SBD), which are then combined to form the SSEM. By integrating the SSEM with other envelopes such as ATSM and TTSM, the evaluated operational envelope enables the ATO to generate traction and braking commands.

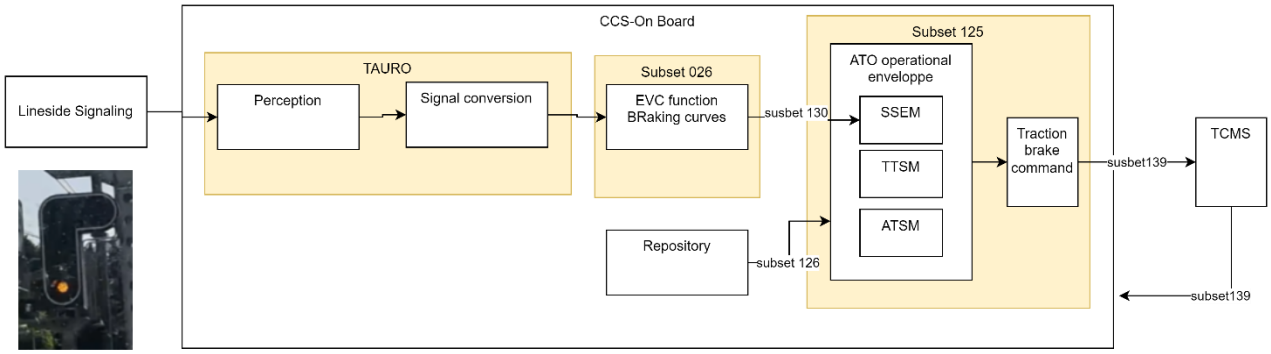


Figure 41: GoA4 use case scope

For the GoA2 use case, only the TTSM (Time Table Speed Management) and ATSM (Automatic Train Stopping Management) envelopes are implemented. The implementation is compatible with the ERTMS/ETCS architecture as defined by European standards (Subset-125, Subset-139), except for the ETCS interface (Subset-130), which is not in the scope of the model. Additionally, the update of Journey Profiles (JPs) from a ground system is not covered in the model implementation. This information is considered as static information stored in a database (repository component – REP)

From an energy efficiency perspective, ATO is considered in the general case where the train is running on time and the track ahead is clear representing the majority of operational scenarios (approx. 90% timetable reliability for French regional train ref. ERJU-719).

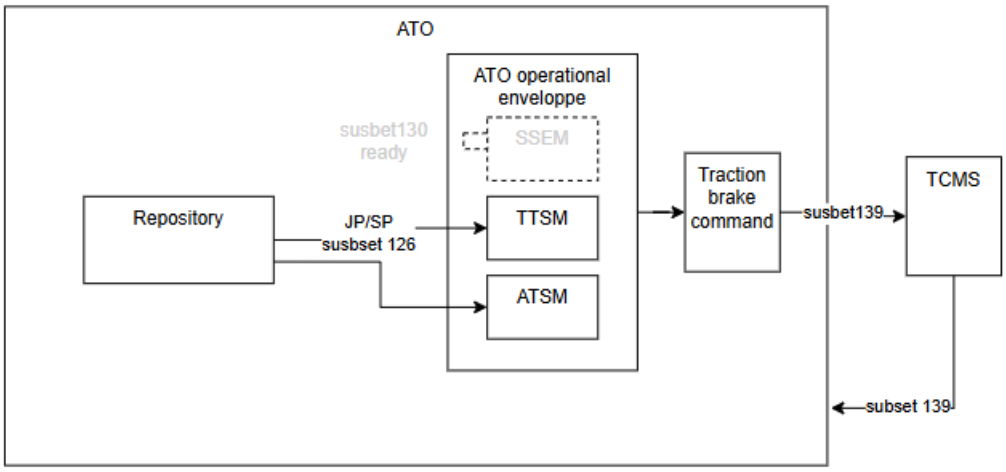


Figure 42 :GoA2 use case scope

	GoA4 depot to station	GoA2 passenger mission	remarks
ETCS braking curves function	Yes	No	GoA2: not implemented
ATO SSEM envelope	Yes	No	GoA2: not implemented
ATO ATSM envelope	Yes	Yes	
ATO TTSM envelope	Yes	Yes	
ATO subset 126 interface	Yes	Yes	conform at the application level but hypothesis of interface between REP and ATO-OB (absence of ATO-TS)
ATO subset 130 interface	Yes	No	GoA2: not activated due to the hypothesis of absence of ERTMS/ETCS infra
ATO subset 139 interface	Yes	Yes	

Table 10: Synthesis of assumptions for function and interface implemented and activated in model

4.3.4 KPI

The two use cases considered for GoA2 and GoA4 aim to contribute to the three R2DATO KPIs (see 101102001 FP2 R2DATO Grant Agreement Part B):

- Energy Efficiency
  The proposed GoA2 use case focuses on identifying an automatic driving mode that minimizes energy consumption while meeting schedule and stopping accuracy requirements.
- Punctuality
  For the GoA2 use case, the ATO must be able to recover from delays while adhering to the timetable. Although it can generate faster speed profiles, these will be constrained by operational limits such as passenger comfort, rolling stock performance, and the margin between nominal and maximum authorized speed profiles.
  In the GoA4 use case, automation of movements between the depot and the station provides greater flexibility, reducing the risk of late train positioning before commercial service.
- Staff Productivity
  In the GoA4 use case, driverless operation allows driving resources to be reallocated to depot tasks or commercial missions.

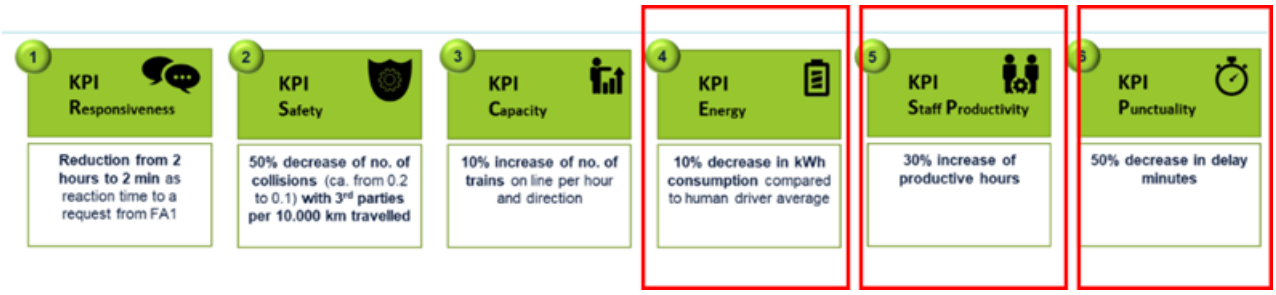


Figure 43: Use cases contribution to KPI

4.4 Operational Design Domain

The ODD can be defined as the set of conditions under which a system is designed and intended to operate effectively and safely. Part of the ODD, the "Operational Design Domain" delineates the operational boundaries within which the system functions optimally, ensuring its reliability and safety.

The Operational Design Domain (ODD) serves as a framework that directs the design, development, and implementation of intricate systems. It covers a multitude of factors including environmental conditions, operational constraints, and technological limitations. Through the delineation of these parameters, the ODD furnishes engineers and designers with an overall understanding of the system's operational environment. It specifies the constraints applicable to the system and outlines the design process, thereby providing a structured approach to system development and ensuring alignment with operational requirements.

Moreover, the ODD plays an important role in regulatory compliance and certification processes. Regulatory bodies often require thorough documentation of the system's operational capabilities and limitations, including its defined ODD, to assess its compliance with safety standards and regulations.

The concept of ODD covers the entire lifecycle of the system.

The subjects addressed by the ODD are:

- Reference regulatory
    - Certification process
- Design process: architecture (function, structure, performance, non-functional requirement), development, verification and validation, industrialization.
    - Safety: risk and analysis process
- Technical
    - Operational performance metrics
    - Environmental conditions and constraints
    - Technical environment
    - Operational procedure
    - Actor involved or concerned
    - Maintenance: principles, operator skills, process
- Technological limitations and capabilities
    - State of the art of autonomous train system reference architecture
    - Maturity
    - Evolution (obsolescence management)
- Human Factor

Table 11 provides an example of ODD on our use case. It is not intended to provide an exhaustive description of the modelled equipment. Instead, the table highlight the key interacting components and the most critical parameters for the model.

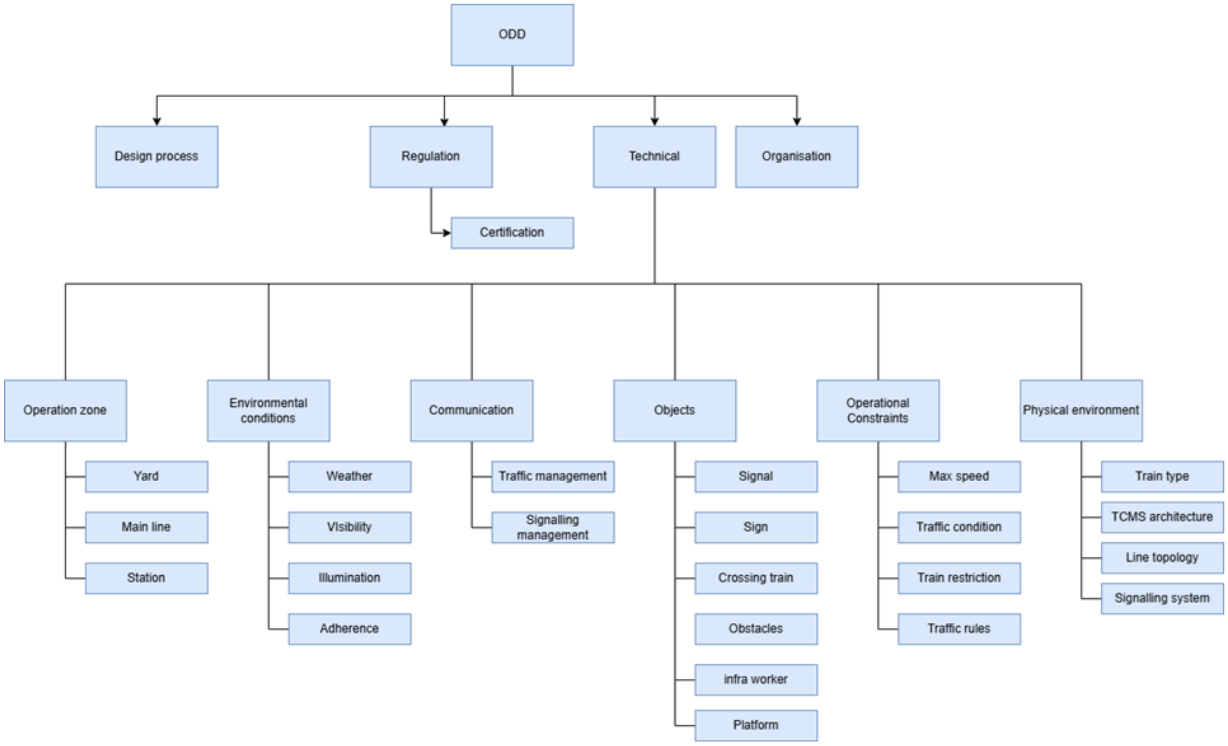


Figure 44: ODD classification

Subject	
Regulatory	Technical Specification for Interoperability: • Control Command and Signalling: ETCS baseline 4 • Operation and Traffic Management

Subject	
Operation scope	GoA4: Technical movement in main line From exiting from the yard to the stopping at station including several possible routes excluded: • movement inside yard • coupling/uncoupling • turnaround in station GoA2: Movement between station in conventional line
Environmental conditions	Light condition: • sun position • cloud • season Weather: • heavy rain (dynamic 1-50mm/h, particles 0,1 – 10mm) • thick fog (static, particles 1-10µm) • snow and hail (dynamic, particles max 15mm) (static hiding traverse and platform) Impact on tracks • standing water • degraded adhesion on wheel/rail contact due to rain, pollution, leaves Not included: • flood • platform subsidence • strong wind • mudslide

Subject	

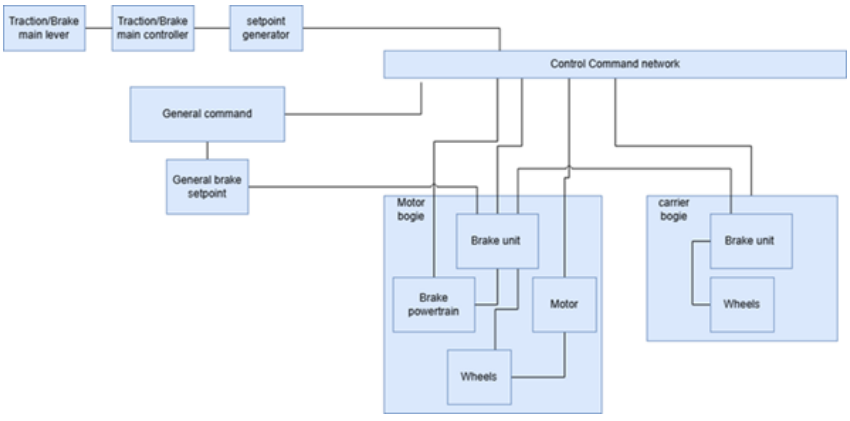
Train characteristic	• Length: 340m max
	• Maximum speed: 200km/h
	• Weight: 42t/ railcar
	• Acceleration: 1m/s²
	• Deceleration: 1,4m/s² (nominal), 0,4m/s² (safe)
	• Coupling: single, multiple x2, x3
	• Traction mode electric
	• TCMS Architecture:
	  ○ Traction component: motors, control unit
	  ○ brake components: pneumatic, electromagnetic, dynamic, control unit (anti-skid)
	  ○ Human machine interface: main controller
	  ○ Power: pneumatic power, electric power
	  ○ Communication:
	  ▪ control command network: IEC 61375 is considered (MVB and Ethernet)
	  ▪ pneumatic pipe
	 

Figure 45: TCMS architecture

Subject	

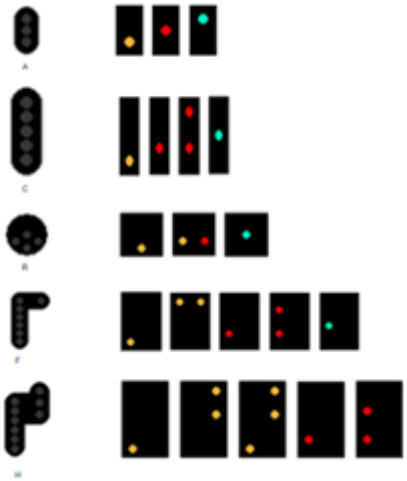
Signalling	GoA4:
	• Type of signal encountered
	  ○ Frame A, C, F H
	  ○ State: track free, warning, crossable stop, non-crossable stop, 30/60km/h speed reduction execution/announcement
	 

Figure 46: Type of signal considered

Subject	
Operational performance	GoA4: • Number of trains managed simultaneously: 5 • Frequency: 1train per 4min exit the yard • Max movement length: 10km • Maximum speed limit: 60km/h • Minimum time window between 2 trains for main line insertion: 10mn • Stopping accuracy: 10m • Time arrival accuracy: 20s GoA2: • Distance between stations: 10km max, 5km min • Number of static speed change: 100 max • Maximum slope: +/- 40o/oo • Stopping point accuracy: 10m • Time tolerance on stopping point: 5mn/100km • Tolerance on timing point: 2% of time between 2 TP

Applicable risk	Based on the established risk nomenclature [ref. ERJU-727]
	GoA4:
	• included
	  ○ collision with a vehicle: head-on, side wipe, rear end
	  ○ derailment
	  ○ overshooting
	  ○ exceeding speed limit
	• excluded
	  ○ collision with obstacle
	  ○ collision in level crossing
	  ○ disorder of infrastructure

RAMS	Process: EN 50716 Software development: EN 50128 Electronic system: EN 50129

Subject	
Cybersecurity	Risk analysis: EBIOS Risk management Threat model [ref. ERJU-728]

Table 11: Operational design domain

4.5 System Architecture

This chapter refers to the structured framework that defines the organization of the system. It includes the arrangement of components and their relationships, both internally and with external systems. The architecture outlines the system's hardware and software components, the data flows, the integration of various subsystems, and the communication networks.

This system architecture analysis serves as a basis for the model specification, whose results are presented in deliverable D31.5.

4.5.1 Capabilities

The capabilities aim to identify "services" offered by the system to external actors. It means that we can identify an external output of the system which interacts with an external actor.

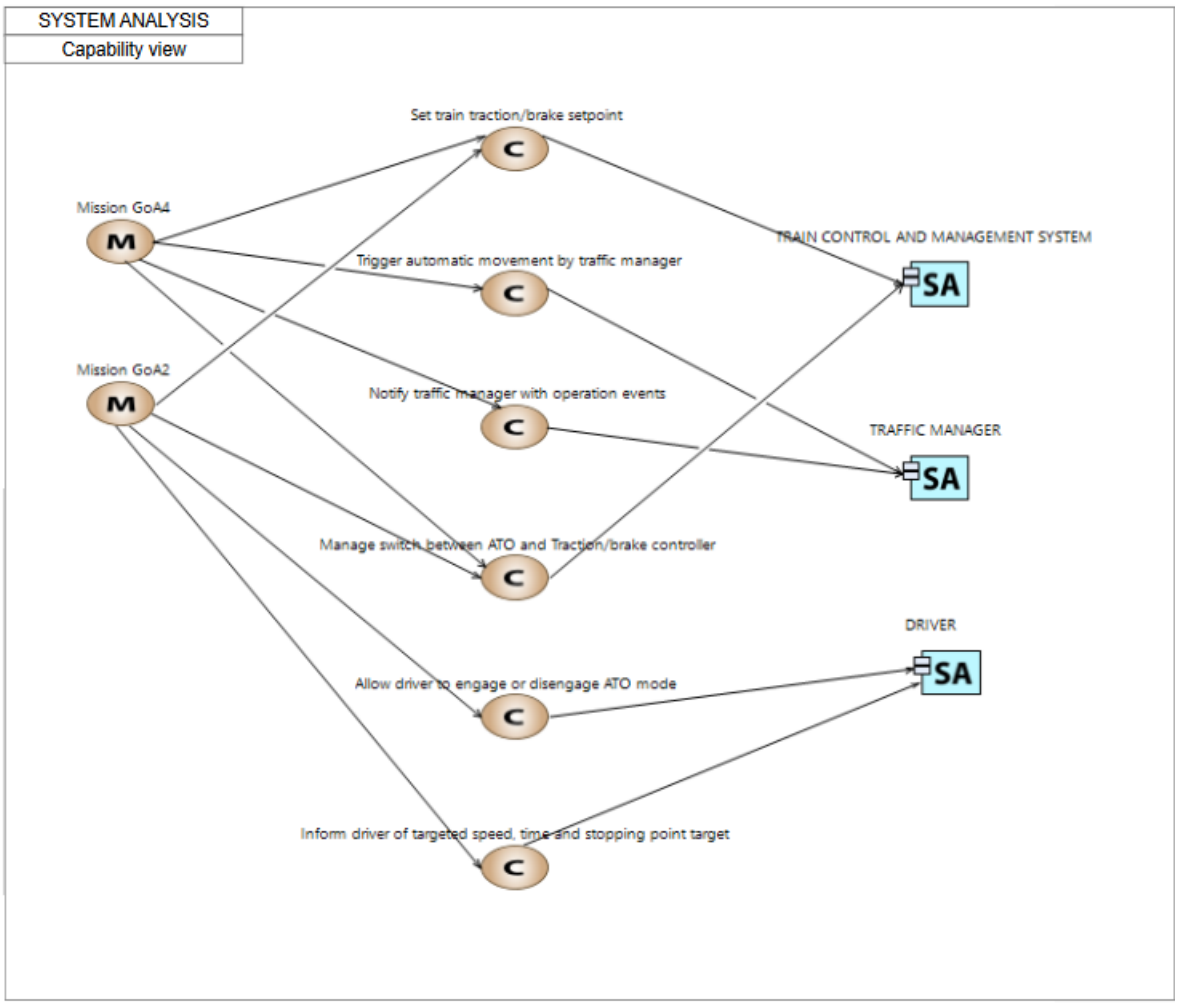


Figure 47: ATO system capability

4.5.2 Actors

Actor	Activities	use case
Signalling manager	+Manage switch to create route +Open route start signal +Communicate with traffic manager: warns of signalling problems and takes into account route prioritisation	GoA4
Yard driver	+Engage ATO GoA4 +Confirm engagement to traffic manager	GoA4
Mainline driver	+ Engage/Disengage ATO GoA2 + Observe signal state	GoA2

Actor	Activities	use case
Lineside equipment	+Provide signal state +Provide sign indication	GoA4
Train TCMS	+Manage traction/brake +Acquire driver action on vigilance actuator +Acquire critical train indication	GoA2 +GoA4
Train wired	+Manage Emergency brake	GoA2 +GoA4

Figure 48: List of actor and allocated activities

4.5.3 External interface

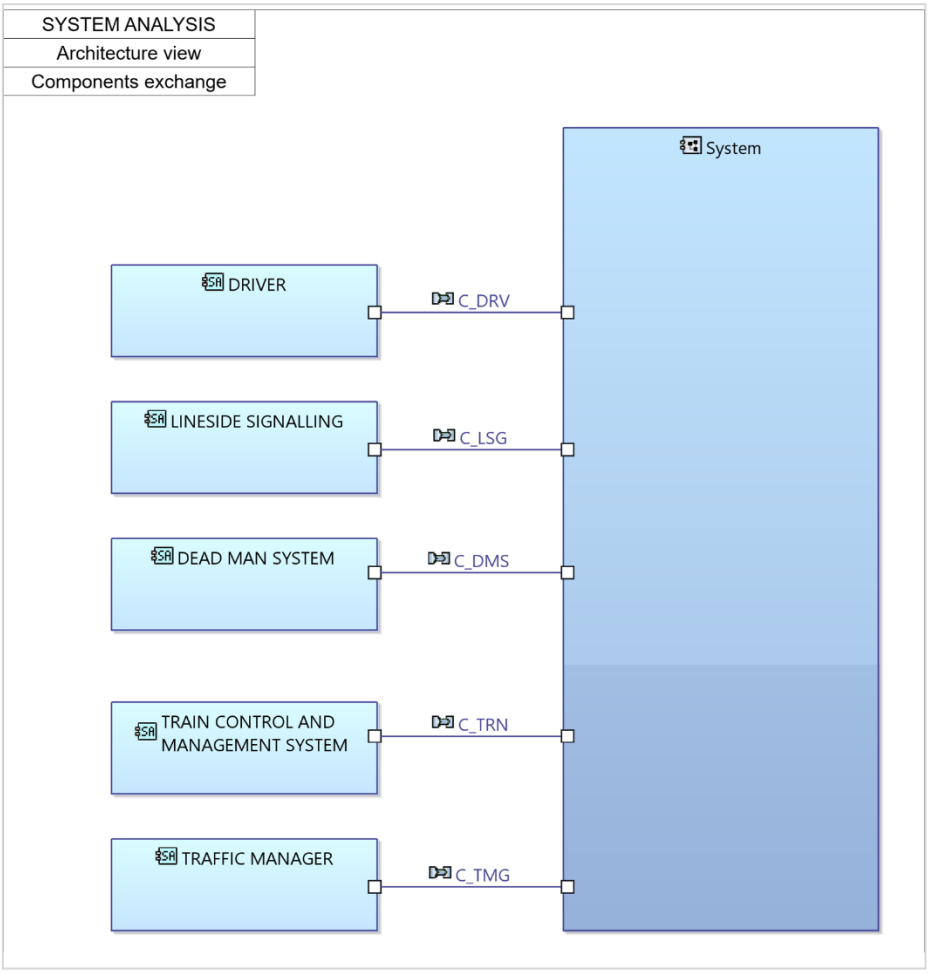


Figure 49: List of logical external system interface (for modelling purpose only – not a reference architecture cf §2 limitations)

Actor	Logical interface	information supported
Traffic manager	Communication channel	To system: ATO command, acknowledgement From system: ATO state, train position, warning
Yard driver	ATO IHM Communication channel	To system: ATO engagement, driverless selection From system: ATO state
Mainline driver	ATO IHM	To system: ATO engagement/disengagement From system: ATO state
Lineside equipment	Visual indication	To system: signal state
Train TCMS	TCMS network	To/From system: subset 139
Train wired	low voltage wire, relay, contact	To system: Emergency brake

Table 12: System ATO external interface

4.5.4 Systems functions

In the following scheme, the links between the actors and the system under development are explicitly highlighted. Using the information collected previously, a system analysis is performed where functions and interactions are identified for external actors and the system.

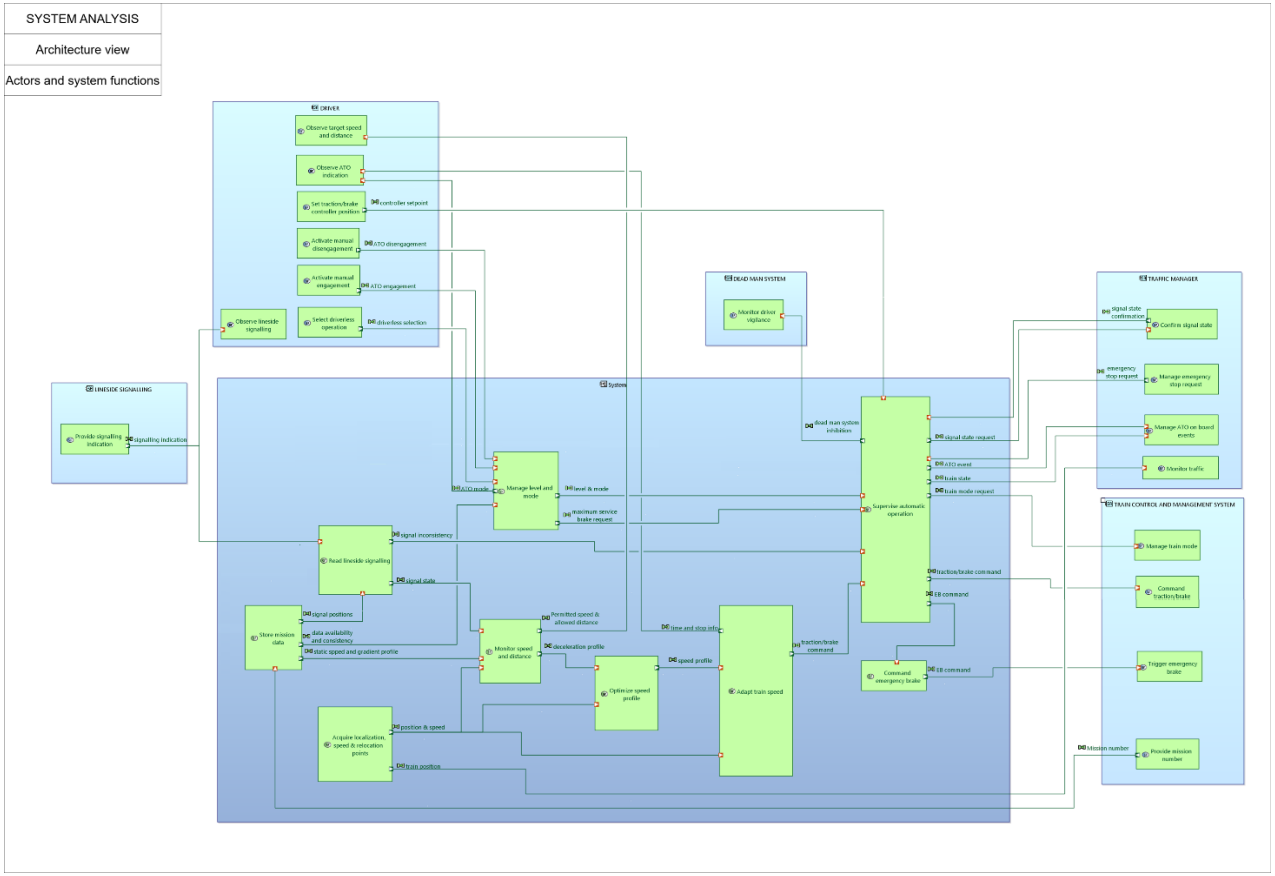


Figure 50: System functions (for modelling purpose only – not a reference architecture cf §2 limitations)

4.6 LOGICAL ARCHITECTURE

Logical Component are formed based on several design consistency and constraint criteria. These groupings facilitate the transition to logical and physical architectures while ensuring traceability of system functions. The main criteria are (not exhaustive):

- Functional Homogeneity
  Functions that share the same purpose or belong to the same functional family (e.g., data acquisition, signal processing, display management, etc.) are grouped together.
  Enhance readability, functional coherence, and specialization of components.
- Geographical Proximity
  Functions that must be performed in close physical proximity (e.g., within the same onboard subsystem, cabin, or physical area of an industrial site) are grouped.
  Reduce complex or costly physical interfaces (wiring, delays, synchronization).
- Performance or Safety Level
  Functions sharing similar performance requirements (e.g., real-time, robustness, latency) or safety levels (e.g., SIL, protection of human life) are grouped.
  Enable coherent justification with regard to safety and RAMS requirements.
- Integration or Technological Constraints
  Functions requiring the same kind of platform (e.g., real-time OS, FPGA, microcontroller, cloud) can be grouped to facilitate deployment.
  Optimize hardware/software implementation and reduce integration complexity.
- Organizational Ownership (Teams or Suppliers)
  Functions developed or maintained by the same team or supplier may be grouped accordingly.
  Clarify development responsibilities and ease contractual arrangements.

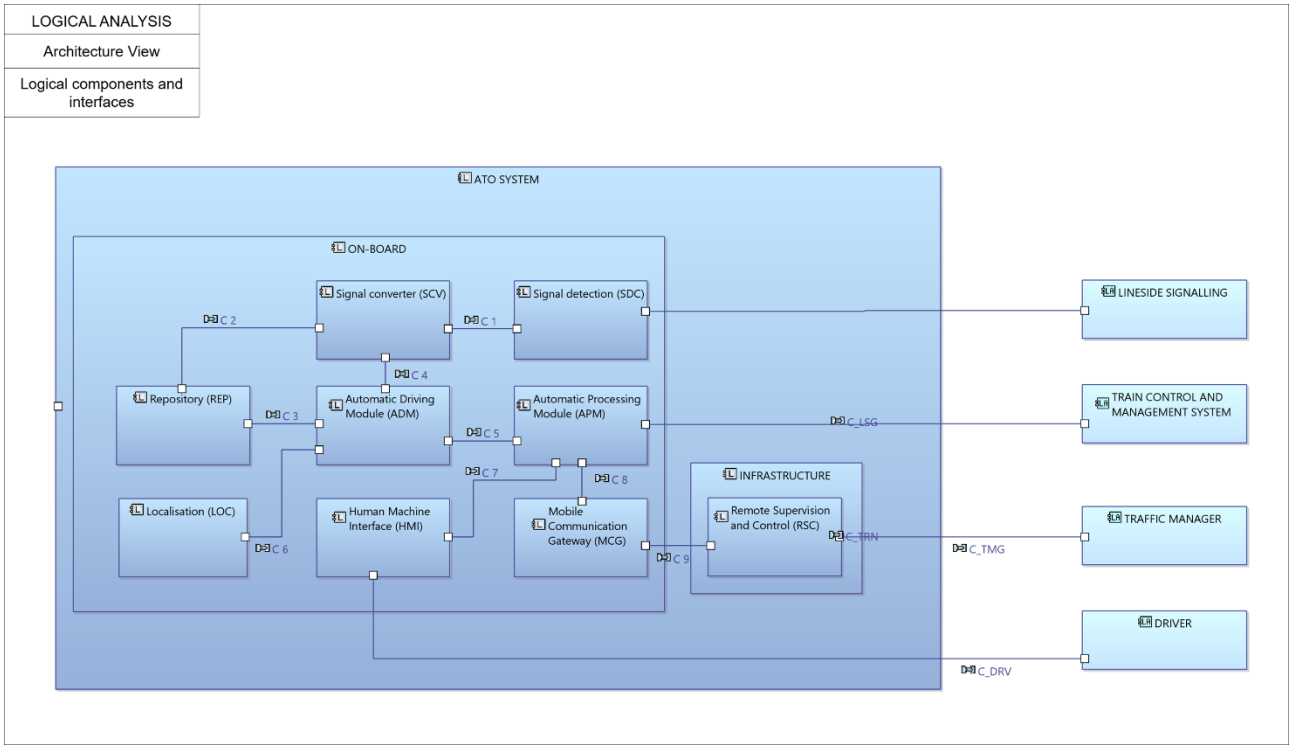


Figure 51: ATO logical components (for modelling purpose only – not a reference architecture cf §2 limitations)

ab.	Name	Description	Grouping rationale
REP	Repository	Gather storage function of information processed by ATO system	homogenous function of information storage. This information shall be saved even if the system is shut off
ADM	Automatic Driving Module	Gather function involved in setpoint and T/B command computation	Technology of control command
APM	Automatic Processing Module	Gather function of safety reaction	Level of safety State Machine logic
SDC	Signal Detection	Gather function of computer vision for signal state evaluation	Technology of machine learning for computer vision and image analysis

ab.	Name	Description	Grouping rationale
LOC	Localization	Gather function of speed, distance and localisation evaluation	Dynamic principles, signal filtering
RSC	Remote Supervision and Control	Gather function for traffic manager interface	Geographical concentration of function in infrastructure facility
MCG	Mobile Communication Gateway	Gather function of mobile communication	Mobile technology Level of cybersecurity
HMI	Human Machine Interface	Gather function of interface with the driver	Geographic: interface in the desk of the cabin Functional: human actuator and visual indication

Table 13: ATO logical components rationale

4.6.1 Alternatives

An architecture study should normally include an analysis of alternatives. In this case, a detailed assessment has not been carried out. However, possible alternatives are listed here for information purposes and to provide a perspective on the relevance of the selected architecture.

Function	retained	alternatives
Adapt train speed	ATO on board controlled This design choice mitigates real-time, integrity, and bandwidth constraints on ground-to-train communication. The train's onboard autonomy alleviates the need for time-critical intervention from the traffic management system.	Remote controlled
Journey profile	Stored on board Only a limited number of route combinations and few modifications to characteristics (e.g., TSRs in the mission profile) are considered.	Received from ATO trackside

Function	retained	alternatives
Detect signal	Lineside signal detection The use case targets lines not equipped with ERTMS/ETCS in order to promote onboard deployment.	ERTMS/ETCS L2 with GSM-R/FRMC communication network
Optimise speed profile	Deterministic algorithm This kind of algorithm is the most mature ready to market approach.	Data based or genetic algorithm

Table 14: Architecture alternatives

4.6.2 Logical functions

System functions are decomposed to a level of detail that allows the behaviour of logical components to be defined. The level of detail must be sufficient to support implementation and to enable unambiguous allocation to a component. Performance constraints (in particular safety) may require further refinement.

For clarity, the implementation choices made within WP31 are described in the section on logical function descriptions.

4.6.2.1 Read Lineside signalling (GoA4)

The function aims to detect and recognizes light signal. It translates it to command that enables train speed adaptation.

Detection consistency can be assessed through verification of the possible states recorded for the specific signal.

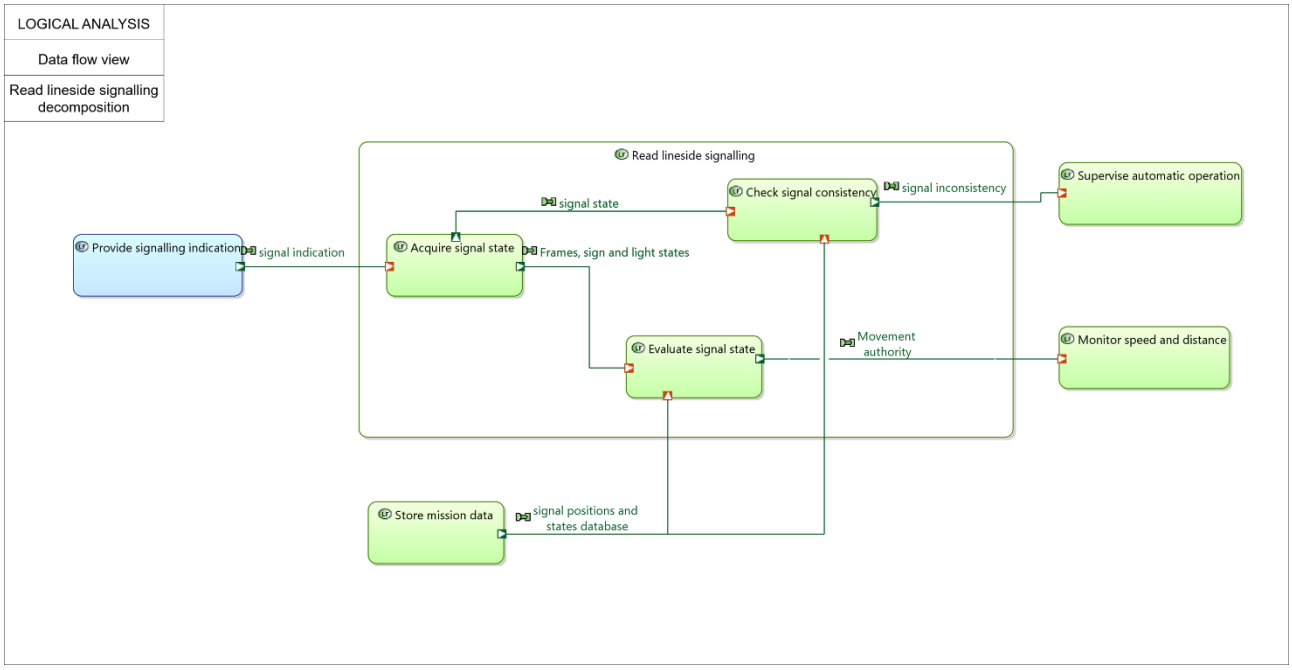


Figure 52: Read lineside signalling function (for modelling purpose only – not a reference architecture cf §2 limitations)

4.6.2.2 Supervise automatic operation (GoA4)

This function manages mode (cf § ATO modes)

A safety reaction leads to the emergency brake being triggered by specific events, such as:

- Traffic manager request for stopping point overshoot or undershoot.
- Inconsistency in signal detection

Recovery from a signal detection error is based on a process where a doubt removal is requested from the traffic manager.

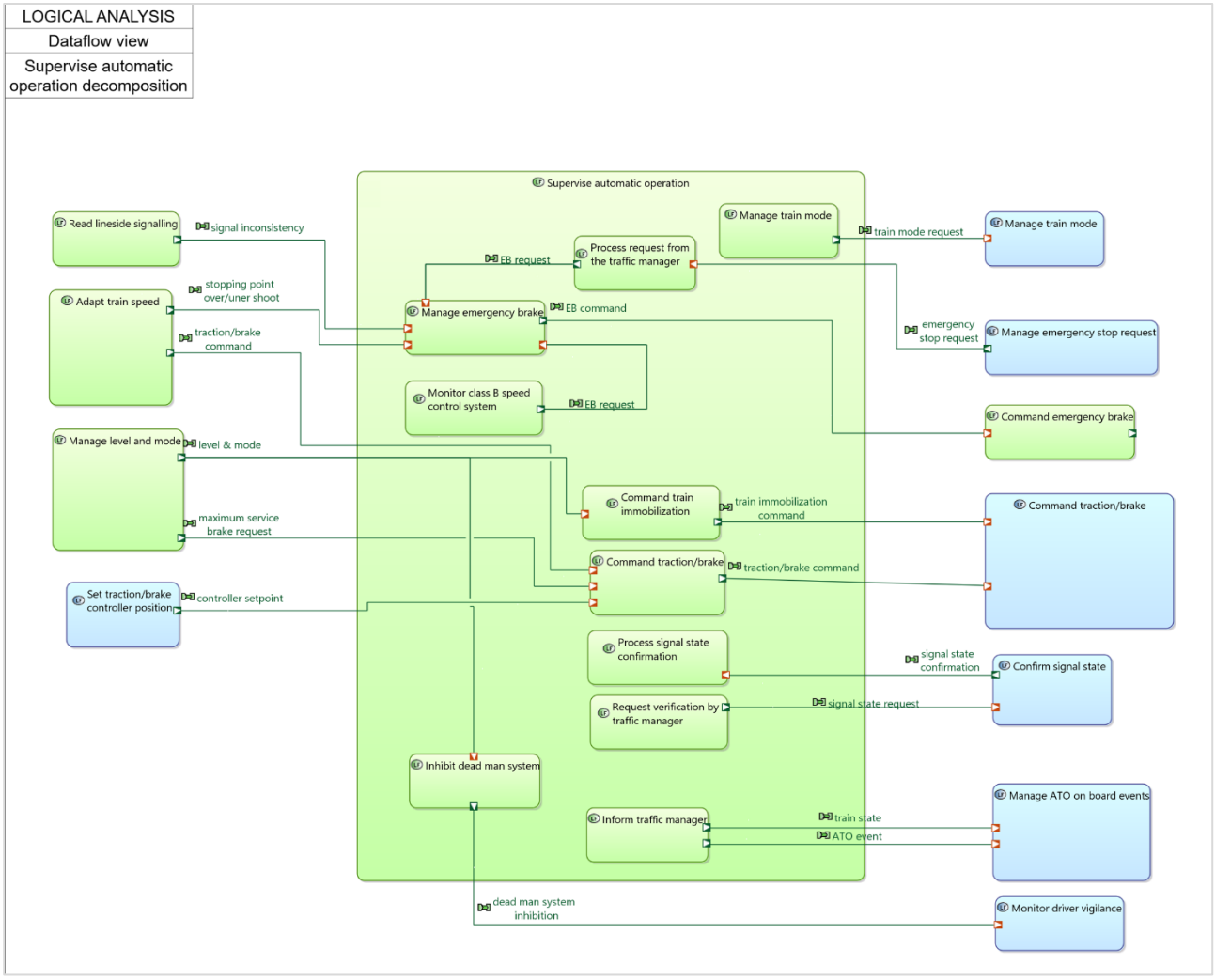


Figure 53: Supervise automatic operation (for modelling purpose only – not a reference architecture cf §2 limitations)

4.6.3 Scenario

The scenario description provides detailed narratives of how the system will operate under various conditions. These scenarios illustrate typical use cases, emergency situations, and maintenance procedures, offering insights into the system's functionality and robustness. Each scenario includes:

- Normal Operations: Descriptions of routine activities, such as train scheduling, passenger boarding, and on-time arrivals.
- Degraded Situations: Procedures and system responses to emergencies, such as train malfunctions, accidents, and adverse weather conditions. This includes protocols for ensuring safety and maintaining service continuity.

- Stakeholder Interactions: Roles and actions of various stakeholders (e.g., train operators, control centre staff, maintenance crews) in different scenarios, ensuring clarity in responsibilities and communication.
- System Responses and Adaptations: How the system adapts to varying operational demands and external conditions, maintaining performance and safety standards.

These scenarios help in understanding the practical applications of the system, identifying potential issues, and refining operational strategies to ensure efficient and reliable service.

4.6.3.1 Initialization GoA4

The initialization requires the mission number to be entered by the yard driver and the mission confirm by the traffic manager. If both match, the ATO is ready. The yard driver then switches to the driverless mode and exits the train. He confirms to the traffic manager that he has left the train. The traffic manager then requests activation. As the lineside signalling switches from stop to track clear, the perception system allows movement without speed restrictions.

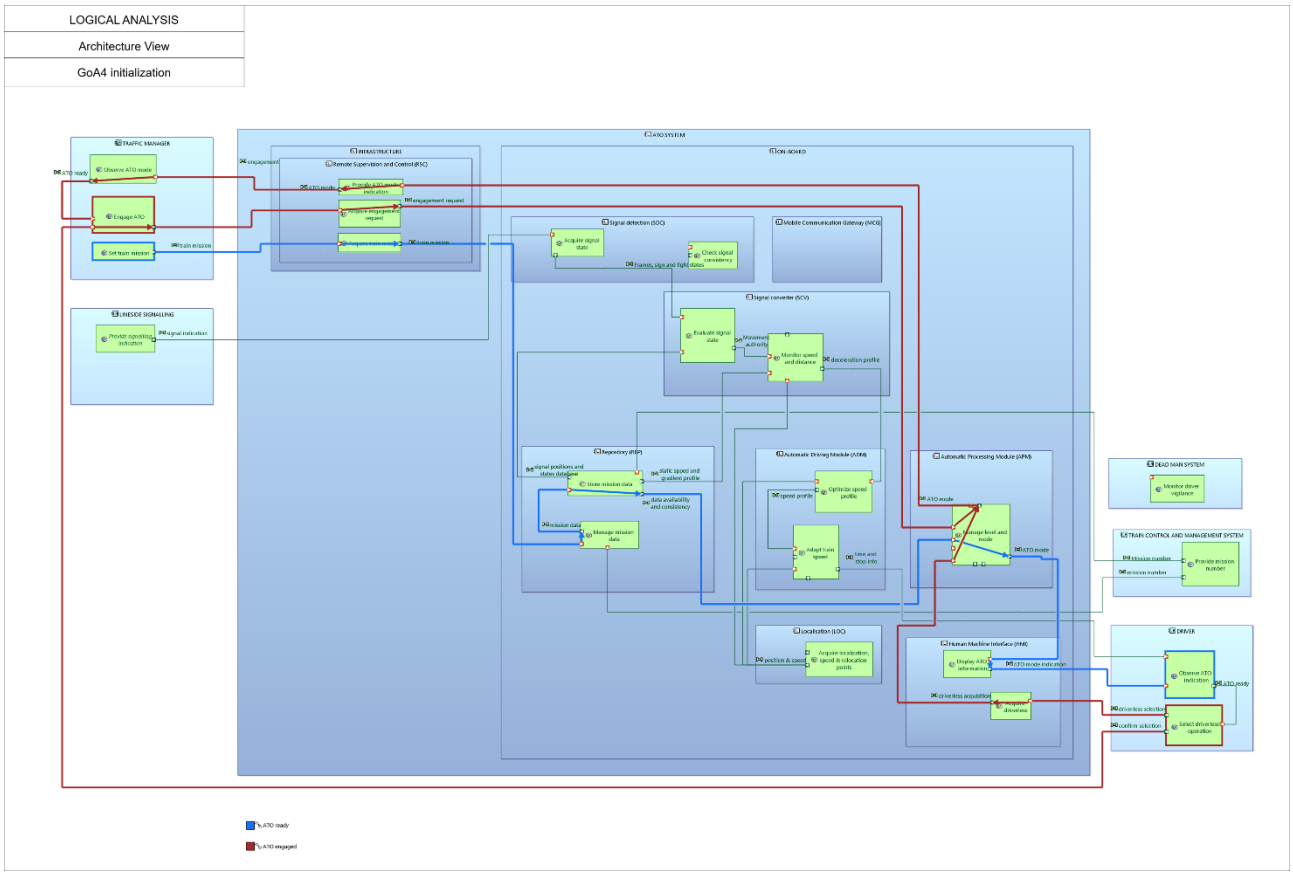


Figure 54: GoA4 initialization architecture (for modelling purpose only – not a reference architecture cf §2 limitations)

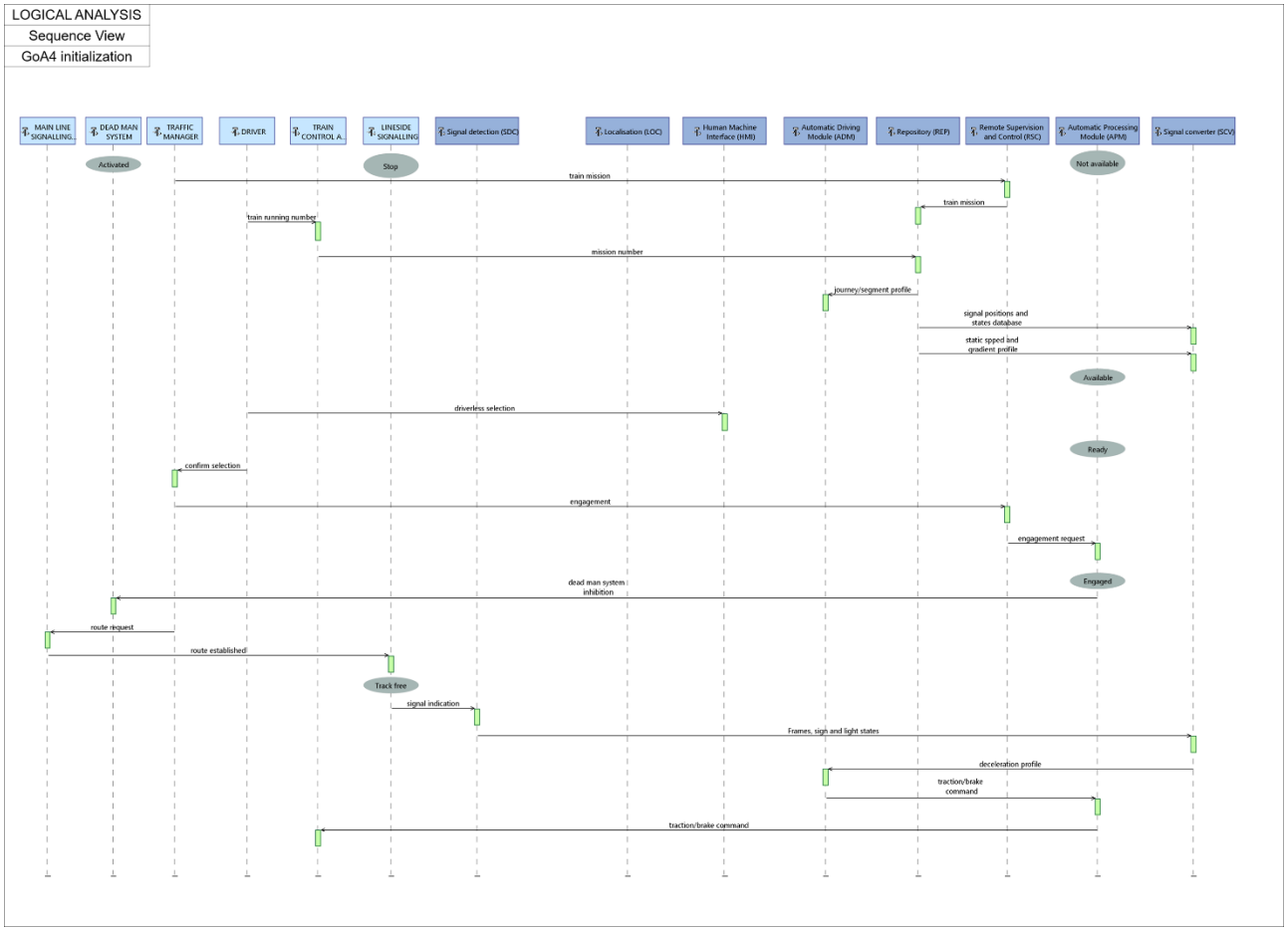


Figure 55: GoA4 initialization sequence (for modelling purpose only – not a reference architecture cf §2 limitations)

4.6.3.2 Mission GoA4

The perception system monitors the lineside signalling and requests speed limitations depending on the signal status. The ATO adapts its response to events: It requests confirmation from the traffic manager if a signal is ambiguous, stops the train until confirmation is received, and monitors the B speed control system. In the event of a malfunction, it stops the train.

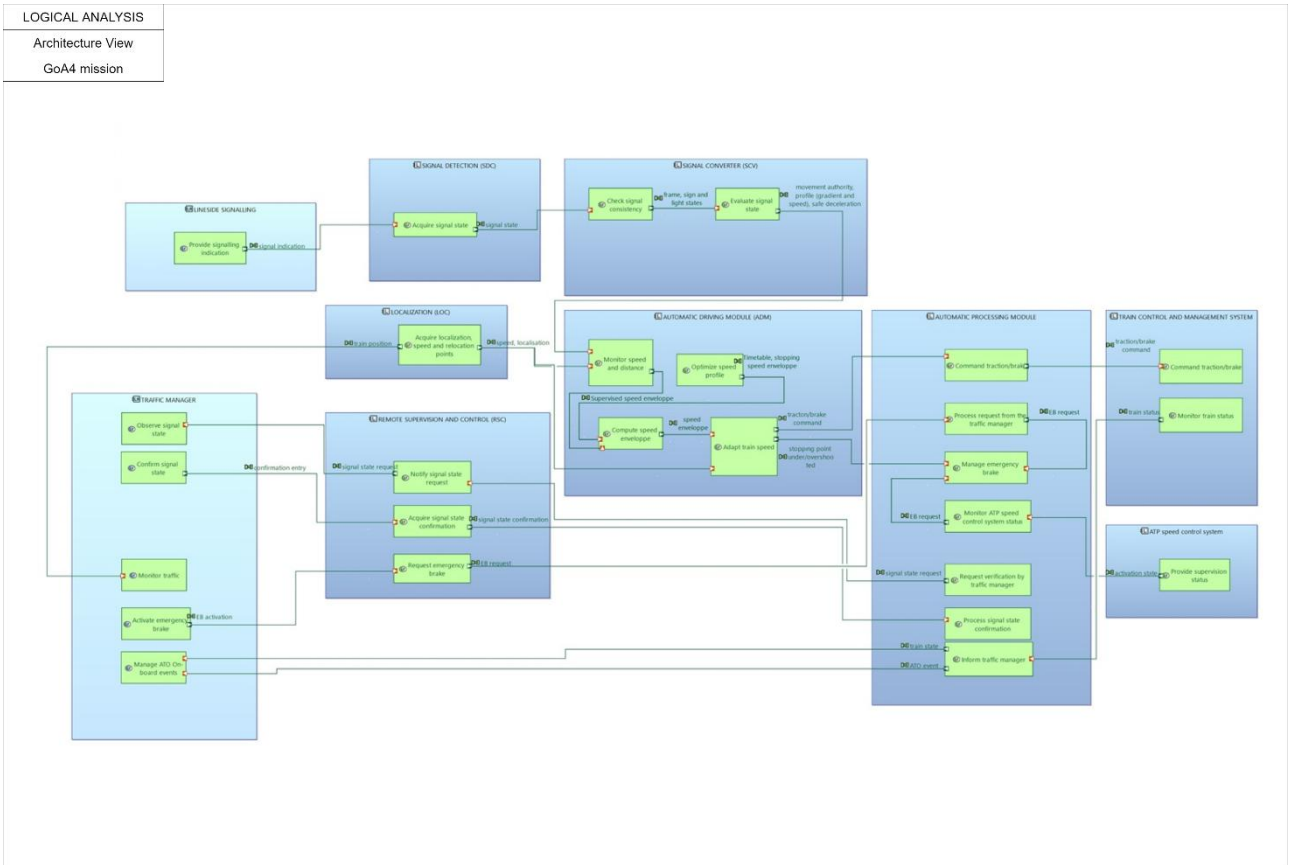


Figure 56: Mission GoA4 architecture (for modelling purpose only – not a reference architecture cf §2 limitations)

4.6.3.2.1 Signal crossing functional chain

Precondition:

- train is in movement
- a signal is present in front of the train at a perceptible distance

Sequence:

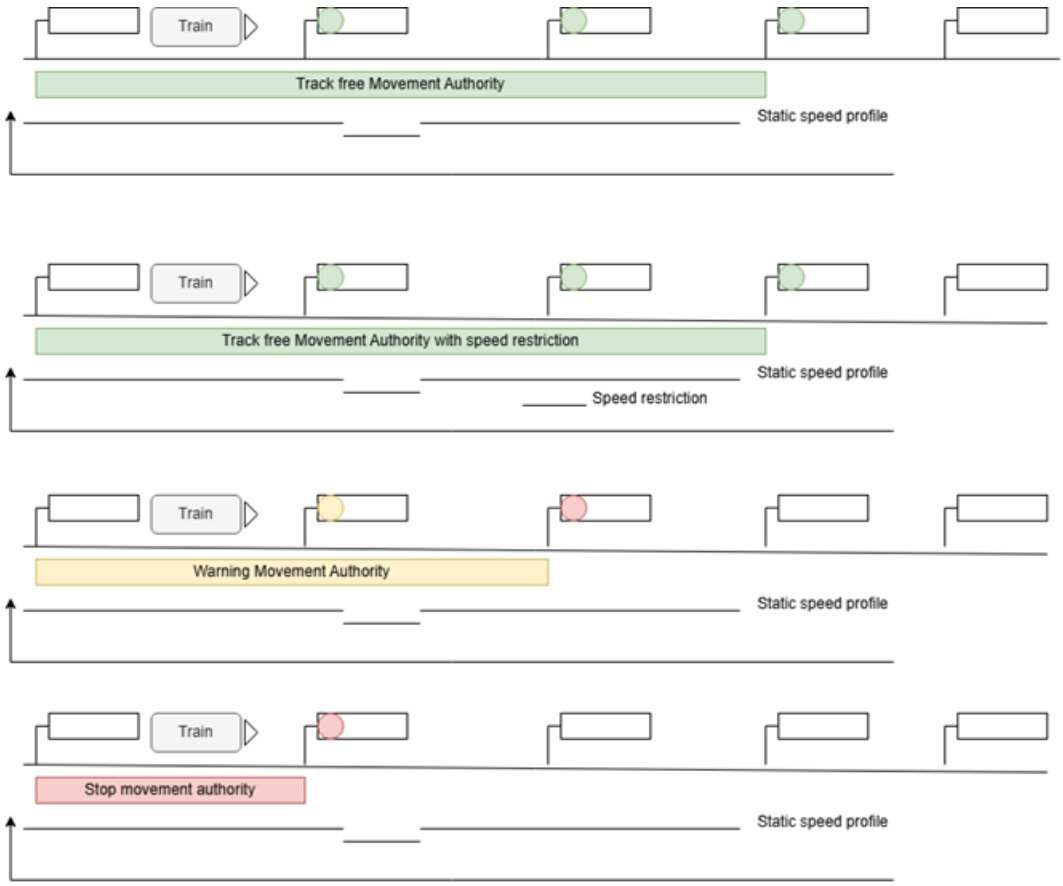


Figure 57: Movement authority depending on signal state

Postcondition:

1. train continue the mission
2. train decrease speed in order to be able to stop to the next signal or to cross track point at speed limit
3. train is stopped

4.6.3.2.2 Track free

When driving, the ATO regulates traction and braking according to a speed curve determined by the speed profile.

Precondition:

- ATO mode is engaged, level is GoA4

Sequence:

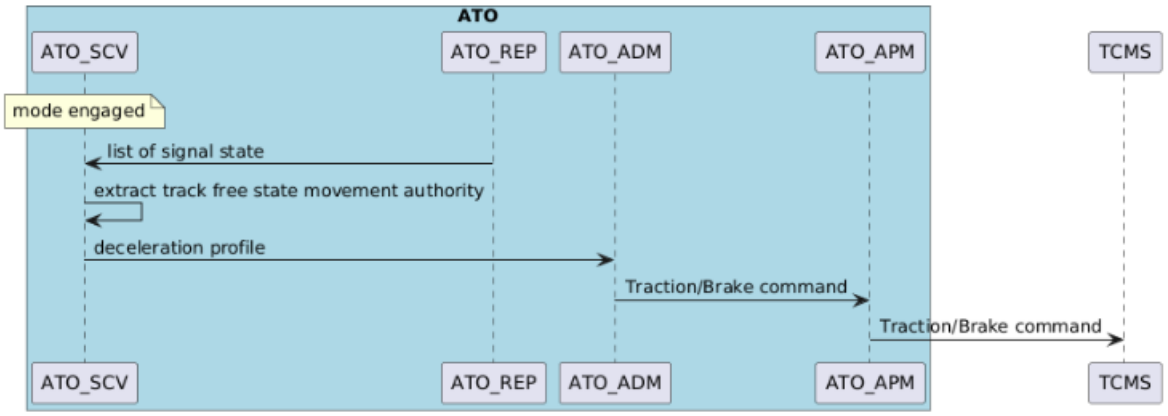


Figure 58: Track free signal crossing sequence (for modelling purpose only – not a reference architecture cf §2 limitations)

Postcondition:

- ATO command T/B

4.6.3.2.3 Warning reaction

When crossing a signal with a warning aspect, the ATO sets speed to approach the next signal at 15km/h. This speed is set in order to guarantee that the train will not engage danger point if signal is passed at danger.

Precondition:

- a warning signal is presented

Sequence:

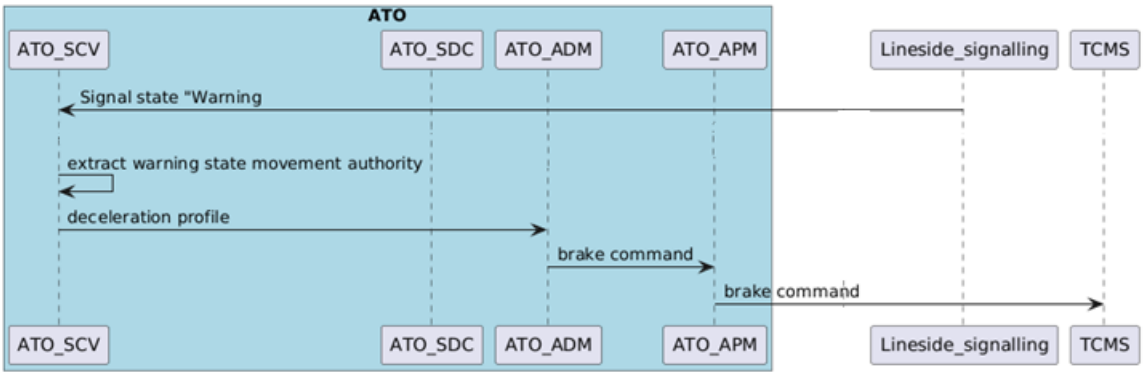


Figure 59: Warning signal crossing sequence (for modelling purpose only – not a reference architecture cf §2 limitations)

Postcondition:

- train decrease speed in order to be able to stop at the next signal

4.6.3.2.4 Stop reaction and recovery

When crossing a signal with a stop aspect, the ATO set speed in order to stop before the signal.

Once stopped before the signal, the ATO warns the traffic manager and monitors the signal state. When the signals changes from stop aspect to warning or track free aspect, the ATO resumes the movement according to the new signal state and informs the traffic manager.

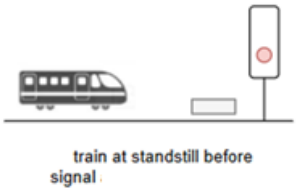


Figure 60: Stop signal crossing and recovery sequence

Precondition:

- a stop signal is presented

Sequence:

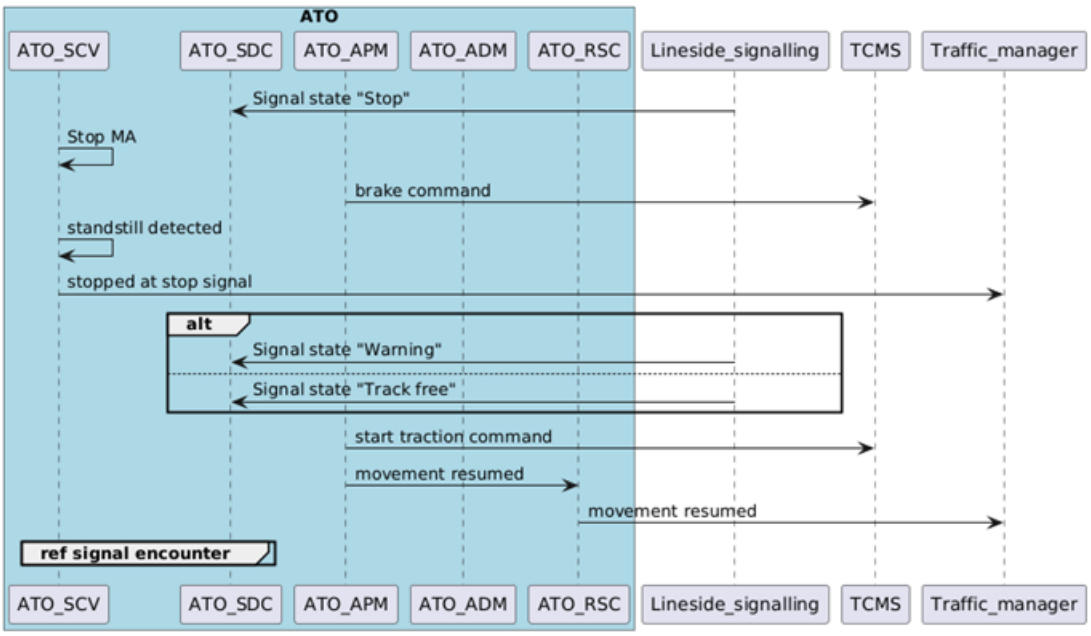


Figure 61: Stop signal crossing and recovery sequence (for modelling purpose only – not a reference architecture cf §2 limitations)

Postcondition:

- ATO movement according to new signal state

4.6.3.2.5 Inconsistency recovery reaction

In case of inconsistency detection on signal aspect, the ATO commands an emergency brake. This relates to warning and open signal aspect not detected. Inconsistency on stop signal implies the train to be tripped.

The traffic manager reviews the images from the perception system in order to instruct the ATO of the correct signal state and associated reaction.

The image review by the traffic manager is a safety related function, the system needs to ensure the displayed image and review instruction corresponds to the correct train and correct signal sequence. Alternatively, the traffic manager could also be given access to the signalling management interface to confirm the current signal state.

Precondition:

- An inconsistency has been detected by the ATO in signal recognition.

Sequence:

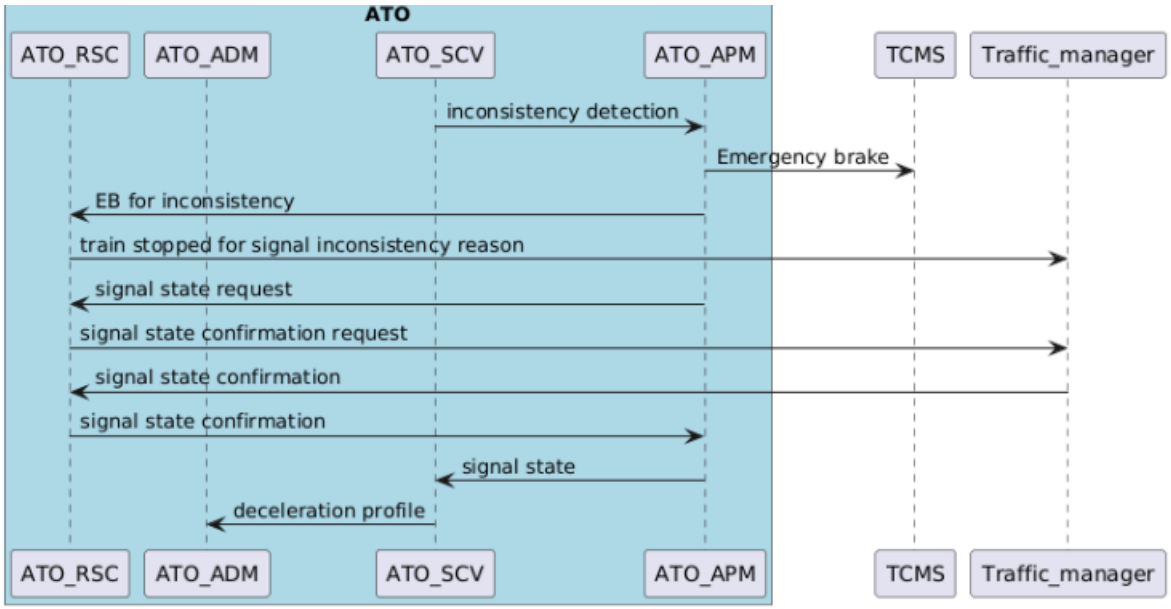


Figure 62: Signal inconsistency recovery (for modelling purpose only – not a reference architecture cf §2 limitations)

Postcondition:

- Train resumes mission following signal indication given by traffic manager.

4.6.3.3 Emergency brake requested by traffic manager in GoA4

For operational reasons, the traffic supervisor must be able to remotely command the train to stop.

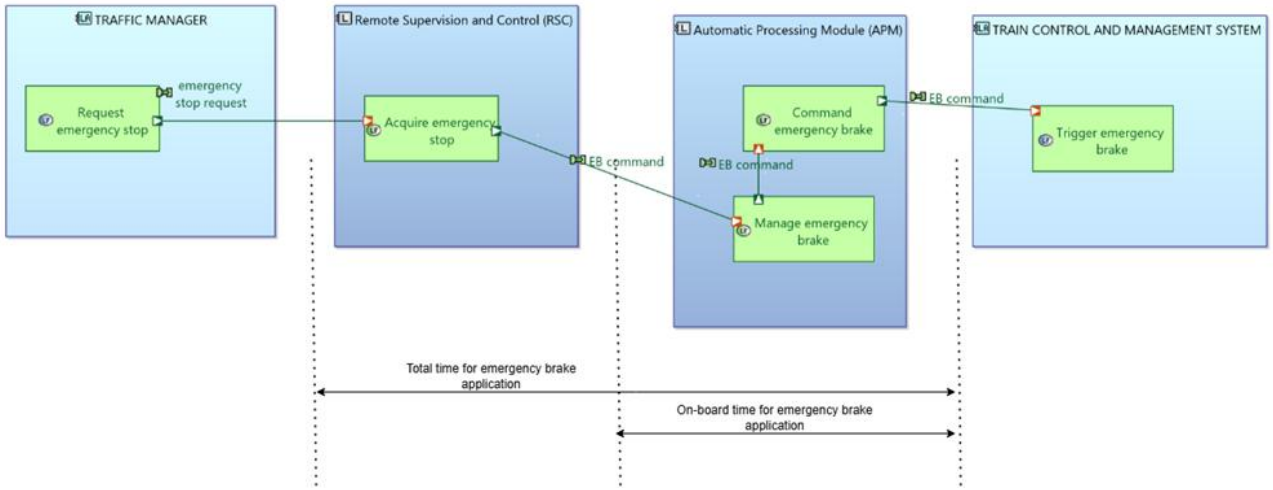


Figure 63: Emergency brake requested by traffic manager (for modelling purpose only – not a reference architecture cf §2 limitations)

4.6.3.4 ATO modes

The GoA mode transition state machine includes the specific transitions required for entering and exiting GoA4, as defined by the use case.

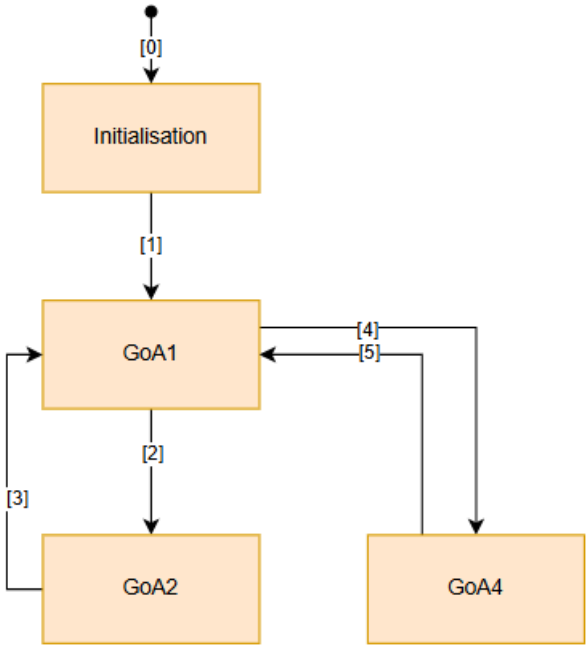


Figure 64: GoA mode transitions (for modelling purpose only – not a reference architecture cf §2 limitations)

Transition	Conditions
[0]	System is powered up
[1]	System is initialized
[2]	Cf. subset 125
[3]	Cf. subset 125
[4]	[Train is standstill] AND [Train is braked] AND [local driver engagement]
[5]	([local driver disengagement] OR [remote traffic manager disengagement]) AND [Train is standstill]

Table 15: GoA mode condition transitions

For both GoA mode, status mode are defined:

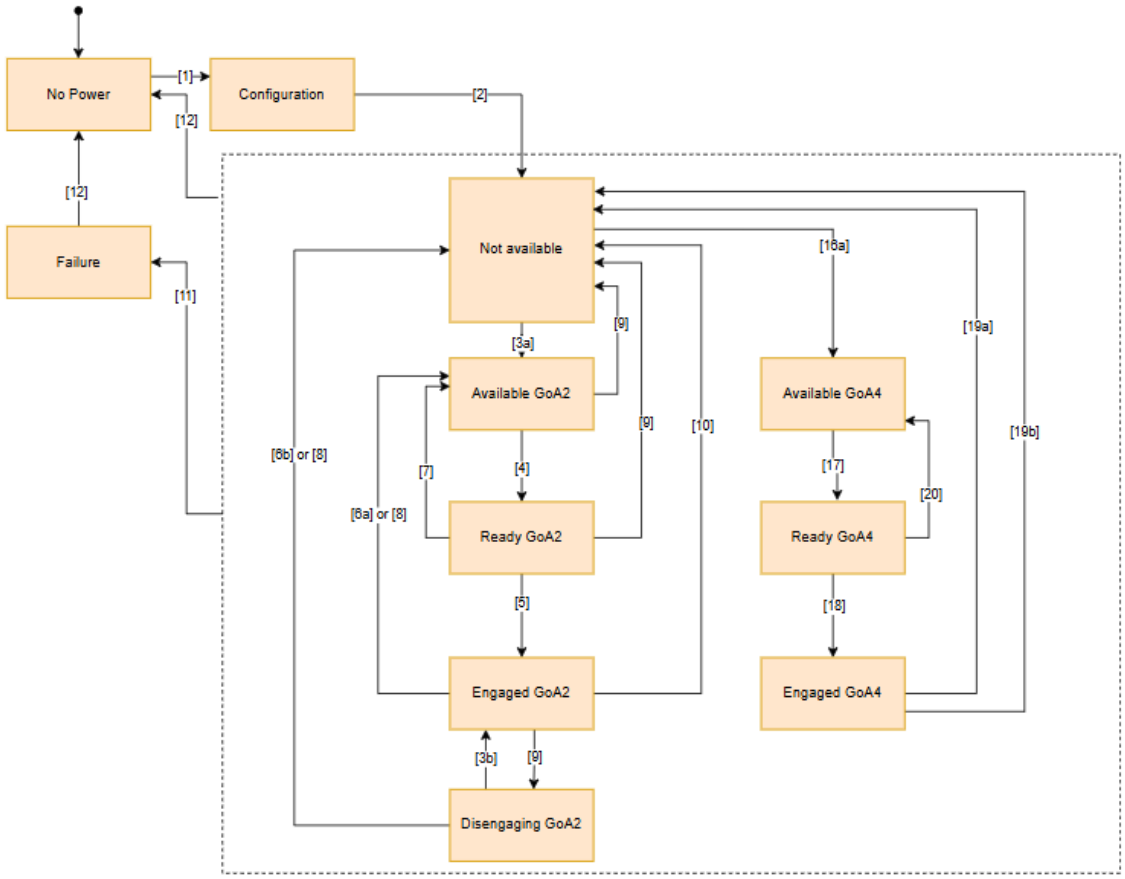


Figure 65: ATO mode transitions (for modelling purpose only – not a reference architecture cf §2 limitations)

Transition	Condition
[1]	(The ATO-OB is powered on)
[2]	(ATO is configured)
[3a]	(Operational Conditions are fulfilled) AND (mission number is GoA2 compatible)
[3b]	(ATO-OB recovers the ATO Operational Conditions within 5 seconds after changing to DE State)
[4]	(GoA2 engagement conditions are fulfilled)
[5]	(GoA level is 2) AND (Driver engage ATO)

Transition	Condition
[6a]	(The train stops at a Stopping Point considered as reached) AND (the train is stationary through at least the application of the Train Holding Brake)
[6b]	(The train stops) AND (the train is stationary through at least the application of the Train Holding Brake)
[7]	(The ATO Engagement Conditions are not fulfilled)
[8]	(The Driver manually brakes using the TBL)
[9]	(The ATO-OB loses any ATO Operational Condition)
[10]	empty
[11]	(The ATO-OB detects a fault which does not allow performing ATO functions)
[12]	(The ATO-OB is powered off)
[13]	empty
[14]	empty
[15]	empty
[16a]	(Operational Conditions are fulfilled) AND (mission number is GoA4 compatible)
[17]	(GoA4 engagement conditions are fulfilled)
[18]	(GoA level is 4) AND (engagement confirmed by traffic manager)
[19a]	(The ATO-OB loses any ATO Operational Condition)
[19b]	(The train stops) AND (the train is stationary through at least the application of the Train Holding Brake) AND (traffic manager acknowledge end of mission)
[20]	(The ATO Engagement Conditions are not fulfilled)

Table 16: ATO mode condition transitions

Operational conditions:

- a) Mission number is acquired. There is a compatibility between mission number and available ATO mode.
- b) No train emergency brake is applied

- c) Journey profile (JP) is present with a consistent segment profile (SP):
  - There is a timing point (TP) to be reached.
  - Train located in a SP included in the JP.
  - All referenced SPs up to the targeted TP are available on-board.
  - Data inconsistency affecting TP is not detected.

Engagement conditions:

- a) Direction controller is in forward position
- b) Train doors are closed and locked
- c) GoA2 specific conditions
  - Dwell time is elapsed.
  - When the ATO-OB is not engaged and the train is approaching a Stopping Point, the remaining distance is sufficient for the ATO-OB to stop the train (the minimum distance to stop the train is train specific).
  - The train brake lever (TBL) is not in brake position.
  - While the train is stopped the TBL is not in traction position.
- d) GoA4 specific conditions
  - The driver selects driverless mode.

4.6.3.5 Performances

The following performance parameters are considered as hypothesis for the modelling, for each use case. The values in the level column represent modelling assumptions. They have been selected based on an order of magnitude deemed relevant for the analysis. They should not be interpreted as performance requirements for a real ATO system.

Use case	Performance criterion	Expected level
GoA2	Safety	Basic integrity
	Energy consumption reduction regarding GoA1 operation mean	10%
	Timing point accuracy	+/- 10s
	Odometer precision	+/- 1m
GoA4	Safety, availability	Cf. § Safety consideration
	Minimal detection distance from signal in degraded environment (fog, rain)	20 m

	Time to emergency brake between traffic manager order and ATO output • Time between traffic manager order and on-board reception • Time between reception and ATO output	6 s • 5s • 1s
	Time between traffic manager engagement and ATO mode switch to engage	5s
	Stop accuracy at station Stop accuracy for buffer stop	+/-5m -10/0m
	Odometer precision	+/- 1m

4.7 PHYSICAL ARCHITECTURE

The proposed architecture is designed around key principles: modularity, segregation of heterogeneous safety levels, and scalability.

4.7.1 Physical components and interfaces

The Autonomous Protection Module (APM) is a safety-critical component, as it is responsible for reacting in hazardous situations and therefore operates with a high Safety Integrity Level (SIL). A digital I/O interface is provided to connect with the train's relay-based logic in order to control the emergency braking system.

Systems whose services may be shared with other onboard functions are deliberately separated from the ATO core. This applies, for example, to the Communication Module (MCG) and the Localization (LOC), which are decoupled from the ATO to preserve modularity and clearly delineate safety responsibilities.

Distributed remote subsystems—such as front-mounted cameras or roof antennas—are treated as standalone components to meet their specific integration constraints.

On the infrastructure side, a standardized information system architecture is deployed to support operational control centres and traffic management systems.

The LOC (Localization) module aligns closely with the concept of ASTP (Advanced Safe Train Positioning), providing high-integrity train localization services.

Finally, the architecture leverages the concept of a Common Control Network (CCN), which connects remote components through a shared Ethernet-based communication network.

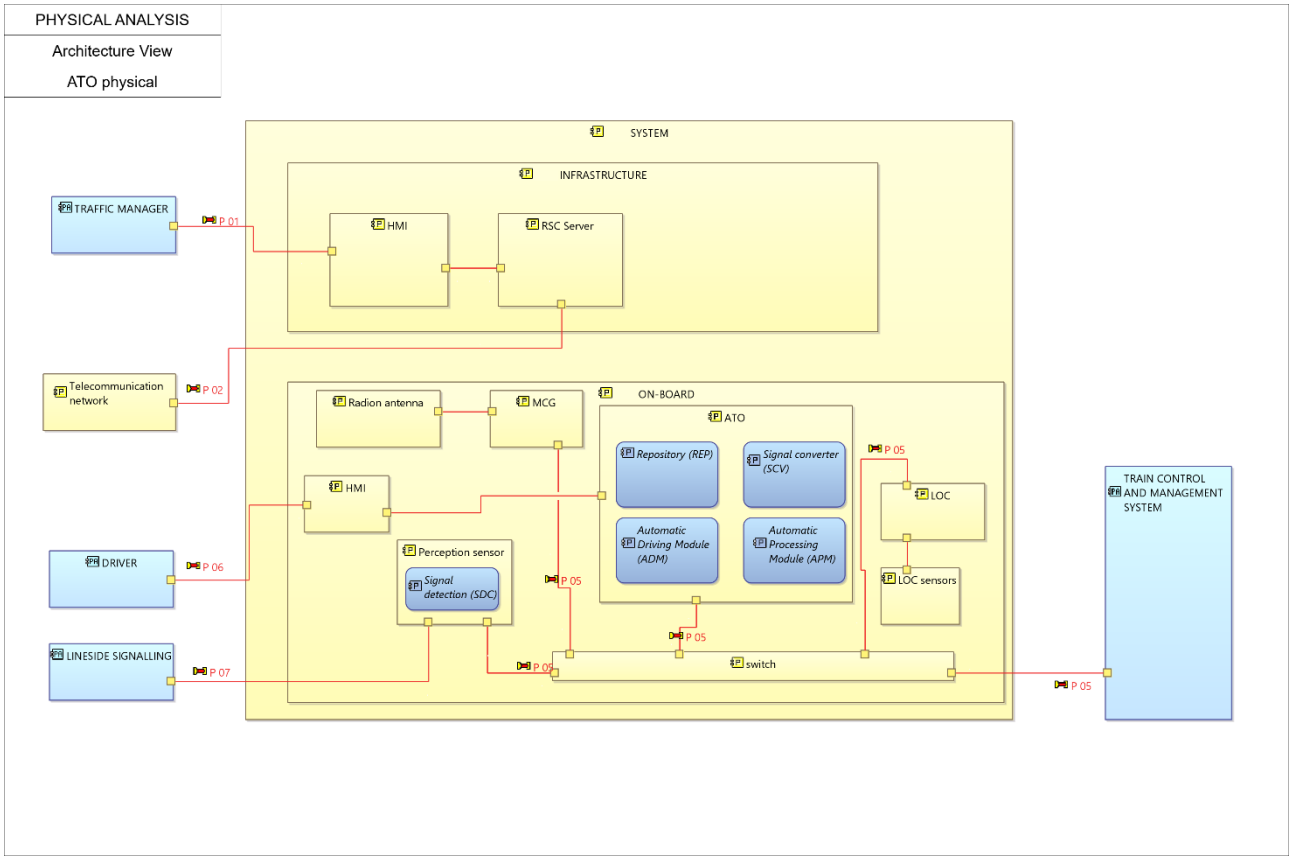


Figure 66: System physical architecture (for modelling purpose only – not a reference architecture cf §2 limitations)

Physical interface	Type
PhysInt 01	Human machine interface
PhysInt 02	Ethernet interface
PhysInt 03	Telecommunication interface 4G/5G
PhysInt 04	digital I/O interface
PhysInt 05	Ethernet interface
PhysInt 06	Human machine interface

Table 17: Physical interface

4.7.2 Product breakdown structure

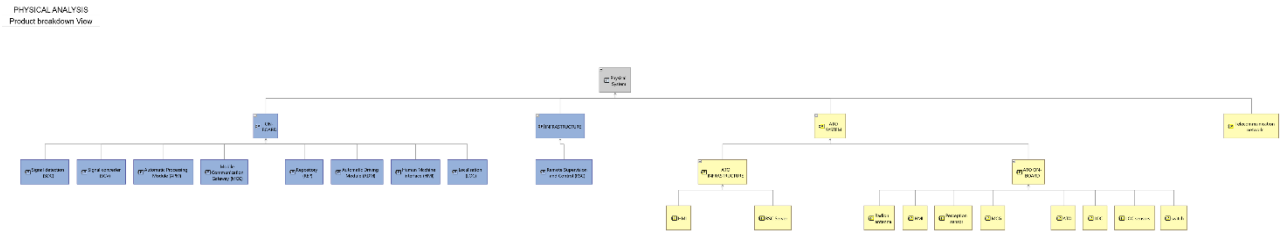


Figure 67: ATO product breakdown (for modelling purpose only – not a reference architecture cf §2 limitations)

4.7.3 Deployed logical component

The following algorithm definition is provided for modelling implementation.

4.7.3.1.1 Signal state acquisition

The signal state detection system implements computer vision algorithms. Specifically, a deep learning algorithm is used to detect rail lines, signal frames and active lights spots.

A neural network is a computational model inspired by the functioning of the human brain, structured in layers of artificial neurons. It analyses data by dynamically optimize its weights and biases, enabling it to learn hierarchical feature representations and predict complex patterns.

Figure 68: Frame and signal states

Video is provided in real-time, frame by frame, to the Signal Detection (SD) model. At the same time, this model receives a detection request from the SCV block, this signal may be empty if no detection is required. If a detection is requested, the SD model performs the detection on a sliding window and sends the detection response to the SCV model.

Figure 69: Context architecture

4.7.3.1.1.1 Signal detection

The 'Signal Detection' subsystem is responsible for detecting railway signalling. It takes as input a frame from the 3D environment, processed frame by frame, as well as the detection request, 'signalDetectionRequest'. The Signal Detection block receives the detection request, performs detection over multiple frames to create a sliding window, and obtains a stable and reliable result. The detection process consists of four main steps: rail detection, chassis detection of the signal light, detection of active light spots, and processing these detections to generate a stable and formatted detection response.

Traffic light state detection is decomposed into two steps: chassis type detection, followed by active light spot localization. This approach is best suited to the limited consistency and diversity of the FRSign dataset.

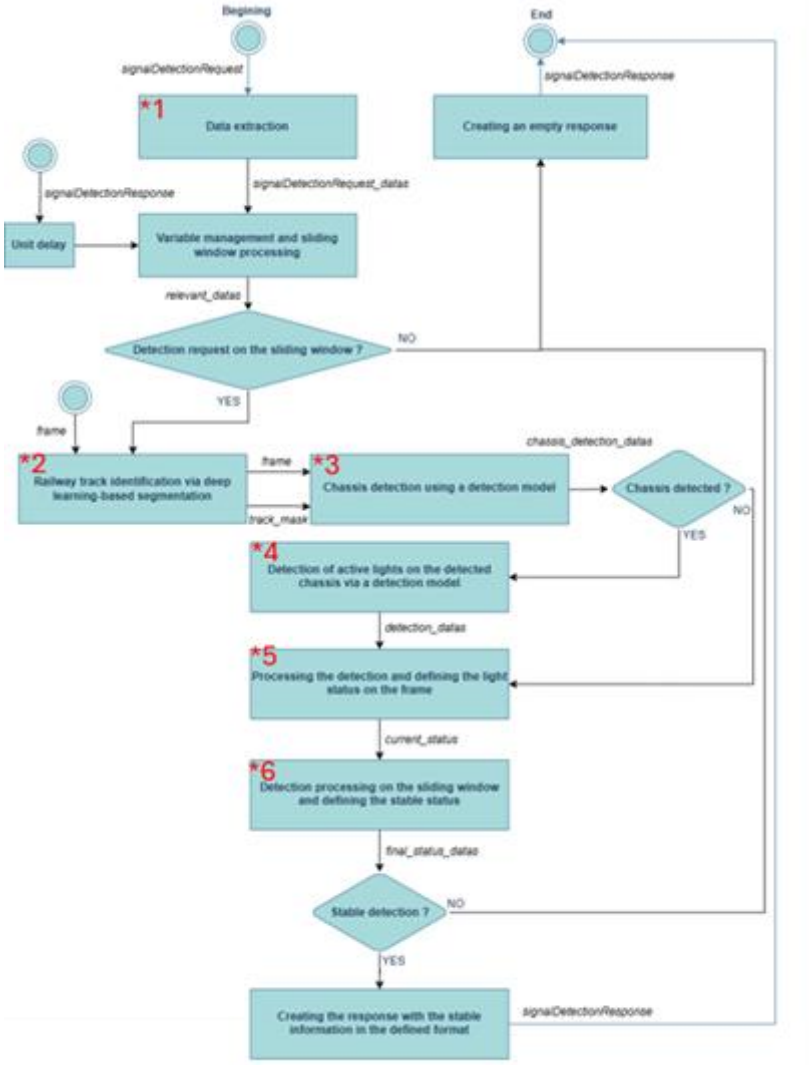


Figure 70: State machine representation of the signal detection algorithm

1- <u>Data extraction and variable management</u>

First, we extract the information contained in the detection request signal. If a request is received, we activate a variable that holds this request over a sliding window, meaning across several simulation steps after the request. We store the relevant information transmitted by the request to have access to it throughout the detection process. To monitor if the detection is ongoing within the sliding window, we track the output of the Signal Detection model with a unit delay. If a stable detection is made, the detection process is stopped. We update the various variables as each step.

<u>Input:</u>

- *<u>signalDetectionRequest</u>: signal detection request sends by SCV to SD*
- *<u>signalDetectionResponse</u>: signal detection response sends by SD to SCV*

<u>Output:</u>

- *<u>relevant datas</u>* – includes:

- *NID_C: Country identifier*
- *QSDRINIT: Detection request flag*
- *N_FRAME_ITER: Number of frame iterations*
- *N_FRAME_TYPE: Frame type*
- *QSDRINIT_STABLE: Detection request flag for the sliding window*
- *incrementalIndex: Counter indicating the number of detections performed*

Detection request?

- If the detection is not requested, we create an **empty** signalDetectionResponse.
- If the detection is requested, we proceed to the **step 2**

2- Railway track identification

The railway signal light detection process begins with rail segmentation using a U-Net deep learning model train with railsem19 dataset.

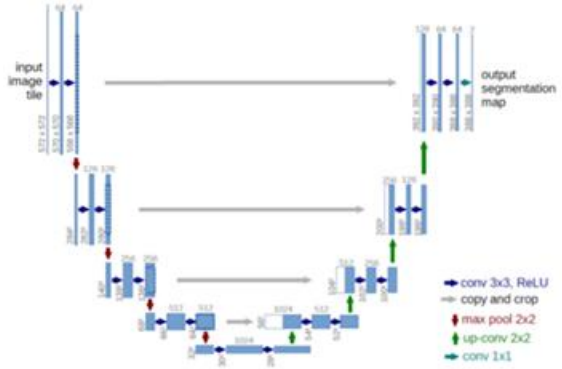


Figure 71: U-NET model architecture

The U-Net model was chosen for railway track segmentation due to its ability to deliver accurate results even with a limited number of annotated images. Its U-shaped architecture, consisting of an encoder that extracts visual features and a decoder that reconstructs a pixel-wise segmentation map, enables precise identification of the railway track in the image. The skip connections between the two parts help preserve fine details, which is crucial for accurate edge detection.

We first preprocess the image to match the model's input format. This includes grayscale conversion, resizing, normalization, and reshaping the tensor dimensions.

Figure 72: Raw image (inputImage)

Figure 73: Processed image

Next, the model performs image segmentation and outputs a prediction map indicating the probability of each pixel belonging to the RAIL class. This map is then processed to generate a binary mask of the rails by applying a threshold and resizing the image accordingly.

Figure 74: Raw binary mask

We then select the most relevant rail path from this binary mask. Morphological filtering techniques such as dilation and erosion are applied, followed by size-based filtering. The final selection is based on the intersection of potential rail paths with a region of interest (ROI) located at the bottom center of the image where the rail is expected to appear (the red square in Figure 65), given the fixed position of the camera mounted on the train. This results in a final mask containing only the relevant rail.

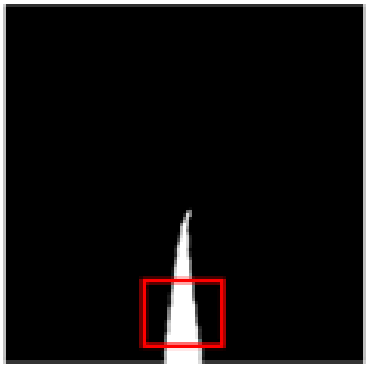


Figure 75: Processed binary mask (center_trak_mask)

Input:

- *inputImage*: The raw image to be processed.

Output:

- *center_track_mask*: The binary output mask

3- Railway track identification

First, we preprocess the image to match the model's input format. This includes resizing, normalization, and reshaping the tensor dimensions.
Next, we perform the prediction using a re-trained YOLOv5s detection model with the FRSign dataset. YOLOv5 was fine-tuned to detect traffic light chassis due to its high speed and accuracy in object detection tasks. Fine-tuning allows the model to adapt to specific features of chassis in our dataset, improving detection performance. YOLO's architecture is a single convolutional neural network with three main parts:

1. Backbone: extracts features from the input image.
2. Neck: combines and refines features at different scales.
3. Head: predicts bounding boxes, object scores, and class probabilities

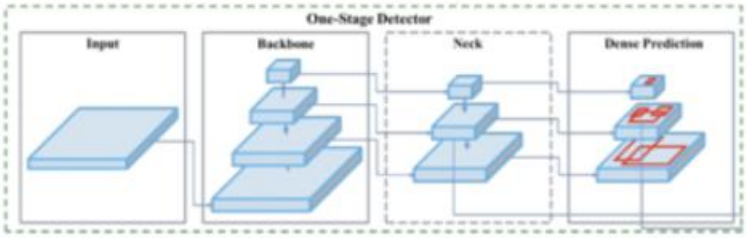


Figure 76: YOLO's architecture

YOLOv5 processes the entire image in one pass, quickly locating and classifying objects, making it well-suited for real-time detection of small and varied chassis shapes. We chose the small version of the model as it offers a good balance between performance and speed.

The model returns 25,200 detection proposals, each including:

- 4 coordinates representing the bounding box (x, y, w, h)
- A confidence score
- A probability corresponding to each class (chassis types: A, C, F, H, R)

We then filter these detections based on the confidence score, discarding detections with a confidence lower than 25% (defined according to the Precision-Recall curve). We further apply IoU and NMS algorithms to remove duplicates. This results in only relevant detections corresponding to the traffic light chassis present in the image.

If there are multiple chassis in an image, we select the one with the most relevant position relative to the lane. Specifically, we calculate the lane's centroid and select the chassis closest to it by comparing with the bottom-right corner of the bounding box, favouring those on the right side of the lane.

Finally, only one chassis remains, and we apply a final filter. In the 3D simulation, no type R chassis are present, so we ignore any detection of type R. We also discard any detection touching the edge of the image. The result is the most relevant chassis, along with its position and type.

Figure 77: Frame and track visualization on simulation video

Figure 78: Frame and track visualization on real video

*In Figures 68 and 69, the relevant frame is the green one.

Input:

- *inputImage*: The raw image to be processed.
- *center_track_mask*: The binary output mask.

Output:

- *frame_detection_data*: Data related to the bounding box, confidence score, and type of the selected frame.

Frame detected?

o If no frame is detected, we proceed to the **step 5.**

o If a frame is detected, we proceed to the **step 4.**

4- Detection of active lights

First, we preprocess the image to match the model's input format. This includes resizing the image to zoom in on the region of interest (the frame bounding box named ROI), normalization, and reshaping the tensor dimensions.

Next, we run inference using a re-trained YOLOv11s detection model, fine-tuned with relevant data from the FRSign and GERALD datasets.

YOLOv11 was fine-tuned to detect active lights spots due to its high speed and accuracy in object detection tasks. YOLOv11 processes the entire image in one pass, quickly locating and classifying objects, making it well-suited for real-time detection of small and varied frame shapes.

We chose the small version of the model as it offers a good balance between performance and speed.

The model returns 25,200 detection proposals, each containing:

- 4 bounding box coordinates (x, y, w, h)
- a confidence score

Figure 79: Visualization of detected light spots

We then filter the results to keep only the 10 most relevant detections, based on confidence scores, discarding any below 40% (defined according to the Precision-Recall curve). IoU and NMS algorithms are applied to eliminate duplicates, ensuring only the most relevant detections corresponding to active light frames within the ROI are retained.

Finally, any detection touching the image edges is discarded. The result is a clean set of the most relevant active light detections, along with their positions and scores.

Input:

- *inputImage*: The raw image to be processed.
- *chassis_detection_data*: Data related to the bounding box, confidence score, and type of the selected chassis.

Output:

- *outputLight*: List of filtered detections (up to maximum of 10)

5- Processing the detection and defining the light status on the frame

During this step, we work on the ROI image of the traffic light frame bounding box. We compare the positions of the detected active light spots with the expected light spot positions, based on the detected frame type.

First, depending on the frame type, we select the corresponding mask that defines the expected positions of the lights. This provides a set of expected spot positions. Around each of these, we define a margin a region of interest.

Next, we extract the centroids of the detected light spots. We then compare each centroid's position with the ROIs and retain only those detections whose centroids fall within these regions. Each detected light spot is thus associated with the corresponding region of interest it falls into.

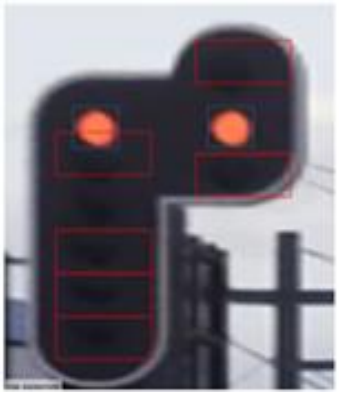


Figure 80: Visualization of the mask of an H-type chassis with detected light spots

Finally, we perform checks based on the number and positions of the matched detections. This allows us to determine the final signal state, since each possible state corresponds to a specific configuration of matched regions of interest.

Inputs:

- *imageROI*: The 3D environment image zoomed in on the region of interest.
- *frame_detection_data*: Data related to the bounding box, confidence score, and type of the selected frame.
- outputLight: Positions of the detected light spots

Output**:**

· *current_status:* A vector representing the bounding box of the traffic light frame region of interest, the traffic light class (indicates the frame type and positions of the active light spots), the confidence score of the frame detection, and a binary confidence flag set to 1 for a reliable detection in the current frame, or 0 otherwise.

6- Processing the detection and defining the stable status on the sliding window

Once the signal state is detected in the current frame, the information is integrated into a sliding window of 70 iterations, corresponding to 7 seconds with a Simulink step size of 0.1, this duration is sufficient to detect the blinking state of the light. This sliding window helps filter out temporary false positives and false negatives and enables handling of blinking traffic lights. 70 iterations were chosen as they offer a good balance between scene duration long enough to capture blinking lights and reduce false positives without detecting too early, which would mean identifying the signal from too far away, resulting in lower image quality.

At each simulation step, the new detection is added, validated, and erroneous detections are discarded. We then determine the final state using probabilistic methods. Among the reliable detections, we identify the most frequent state and assess its stability.

Majority state stability?

o If the majority state exceeds a predefined stability threshold, the detection is considered stable, and a detailed signalDetectionResponse is generated in **Step 7.**

o If the majority state is below the stability threshold but still significantly represented, we check for a possible blinking pattern by evaluating the second most frequent state.

o If both states fall within a defined threshold interval, we consider it a stable blinking state between the two detected states, and a detailed signalDetectionResponse is generated in **Step 7**.

o If neither condition is met, we check whether the sliding window is full.

o If the 70 simulation steps are completed without a stable detection, an empty signalDetectionResponse is generated.

o If the window is not yet full, we proceed to the next simulation step.

Input

- *current_status*: A vector representing the bounding box of the traffic light frame region of interest, the traffic light class (frame type and active spot positions), the frame detection

confidence score, and a binary confidence flag set to 1 for a reliable detection in the current image, or 0 otherwise.

Outputs

- *finalState*: The stable detected state within the sliding window.

M_STABLE: A boolean flag indicating detection stability (1 if stable, 0 otherwise).

7- Creating the final response: *SignalDetectionResponse*

When detection is stable within the sliding window, or the window has been fully processed, a signalDetectionResponse is sent to the SCV block. This response includes identification data, context, and processing results.

First, we create the FRAME_ASPECT signal, which describes the result of the detection. Then, a vector is constructed containing all relevant information in the format expected by the SCV block.

Finally, this vector is converted into a bus using Simulink blocks such as Demux and Bus Creator, ensuring strict compliance with the expected SCV format.

Input:

- *relevant_datas* – includes:
  - *NID_C: Country identifier*
  - *QSDRINIT: Detection request flag*
  - *N_FRAME_ITER: Number of frame iterations*
  - *N_FRAME_TYPE: Frame type*
  - *QSDRINIT_STABLE: Detection request flag for the sliding window*
  - *incrementalIndex: Counter indicating the number of detections performed*
- *finalState*: The stable detected state within the sliding window.
- *M_STABLE*: Boolean flag indicating whether the detection is stable (1 if stable, 0 otherwise).

Output:

- Bus – *signalDetectionResponse*, containing:
  - *relevant_datas*, including:
    - *incrementalIndex: Counter indicating the number of detections performed.*
    - *NID_C: Country identifier*
    - *N_FRAME_ITER: Number of frame iterations*
    - *N_FRAME_TYPE: Frame type*
  - *NID_VIP: Unique identifier number of traffic light signal*
  - *FRAME_ASPECT: Detected frame aspect*
  - *M_DETECTION_CONFIDENCE: Confidence level in the detected frame aspect*
  - *M_STABLE: Stability confidence over the sliding window*

4.7.3.1.2 Evaluate signal state

The aim of this function is to determine the current speed limitations and access constraints related to the track ahead of the vehicle. Perception of reality is provided by the Perception subsystem, which translates light signals and speed signs into digital data. The signalling converted to numeric data can then be used by various programs functions afterwards.

Once track data has been acquired, perception has its limits and can from time to time provide misleading information. Train decisions cannot rely on random data to determine its speed setpoint and movement instructions. Therefore, an additional step is required in order to verify signalling state acquisition.

Another source of information available is provided by the Repository subsystem. For each signal present on the track, the Repository contains a list of all the possible states the signal of interest can display. For every signal, it also provides all the SSP (Static Speed Profile) related to each possible state, to build the proper speed profile as soon as signalling has been acquired by Perception subsystem.

When a train approaches a kilometric point associated to a registered signal, a request is sent to Perception subsystem. Simultaneously, a second request is sent to Repository subsystem in order to acquire all possible states that signal of interest can take. Perception and Repository subsystems run their own process then return acquired data and database information respectively. The next step is to check that signal state returned by Perception is listed among all the possible states registered for that specific signal. This is the coherence check. The aim is to check information validity and prevent any incorrect detection.

Once validity has been confirmed, data about speed and access restrictions must be converted into a movement authorisation.
A movement authorisation is composed of three main information:
- length of the end section ahead
- speed limitation applied to this block to come
- gradient changes describing the slopes of block ahead

About signal approach, it is ruled by Virtual Information Points (VIP). These points of interest are virtual beacons attached to specific kilometric points. They are placed 10 meters upstream of each signal and represent the end of the previous Movement Authorisation and the beginning of the next one.
For a given signal, in order to make sure Perception subsystem gets triggered in time for signalling detection and acquisition, a distance interval upstream of the VIP virtual beacon is defined, called visibility area. This interval defines the distance offered to Perception subsystem to run its

acquisition and get data from signalling. The main goal is to guarantee synchronisation between acquisition phase from Perception subsystem and real-world location of signalling.

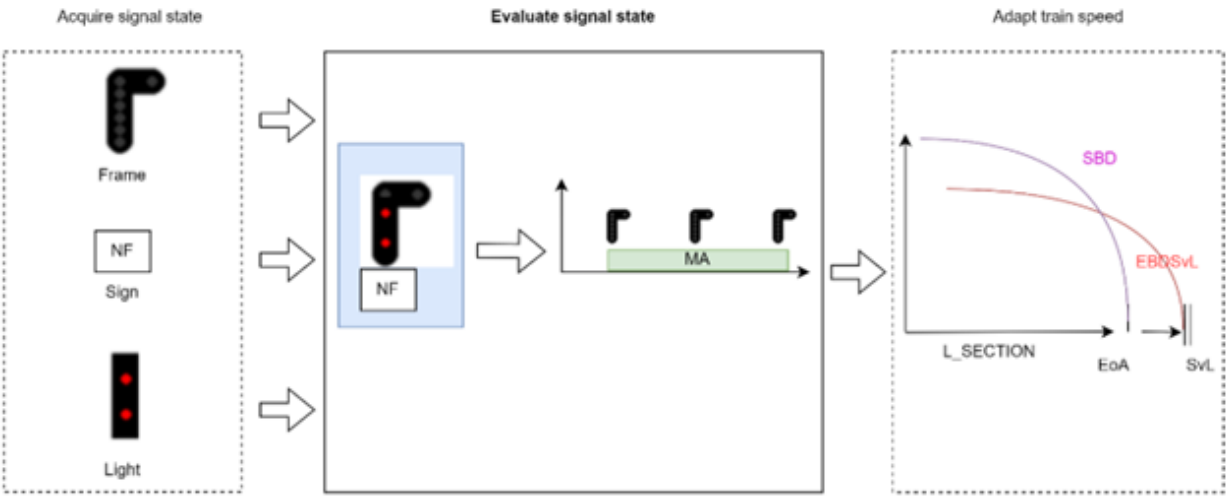


Figure 81: Evaluate signal state principles

Even though calculations and data exchanges are heavily specified in TAURO documentation, there are still some grey areas regarding implementation. One example is the format difference in response set from Perception and Repository subsystems. Repository response lacks some important data available in Perception response. This lack of information makes comparison step quite difficult to complete, in order to validate signalling attributes acquired by Perception subsystem. This imbalance could be fixed by enhancing signal information stored in Repository database.

4.7.3.2 Optimise speed profile

The determination of the optimized speed profile follows the requirements set out in the Subset 125 ("speed computation").

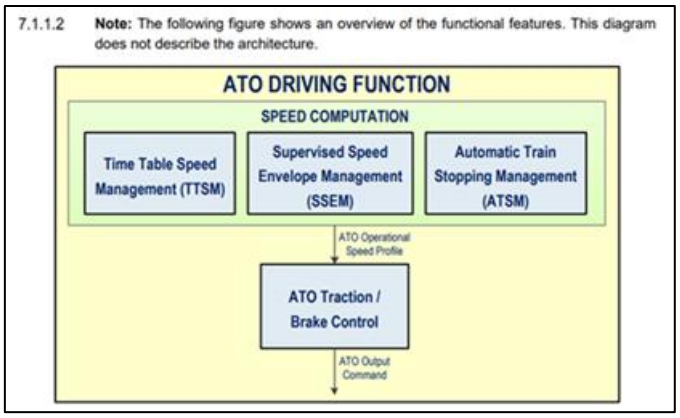


Figure 82: Susbset 125 ATO driving function

Three speed profiles are constructed:

- **Timetable Speed Management (TTSM):** a profile aimed at optimizing energy consumption while respecting timetable constraints (mission), track and signalling conditions, and the train's performance capabilities.
- **Supervised Speed Envelope Management (SSEM):** a profile that constrains the train's speed to prevent intervention by the ETCS, taking into account Temporary Speed Restrictions (TSRs) and the Movement Authority (MA) through the calculation of braking curves, as performed by the ETCS.
- **Automatic Train Stopping Management (ATSM):** a profile designed to ensure precise stopping at designated stopping points (stations).

Based on these three profiles, the **Operational Speed Profile (OSP)** is generated, which serves as the reference speed profile for the train control system.
We choose to construct the OSP by taking the minimum value among the three profiles: at any given position x, the OSP speed corresponds to the lowest speed value among the profiles defined at x. An example is illustrated in the following figure.

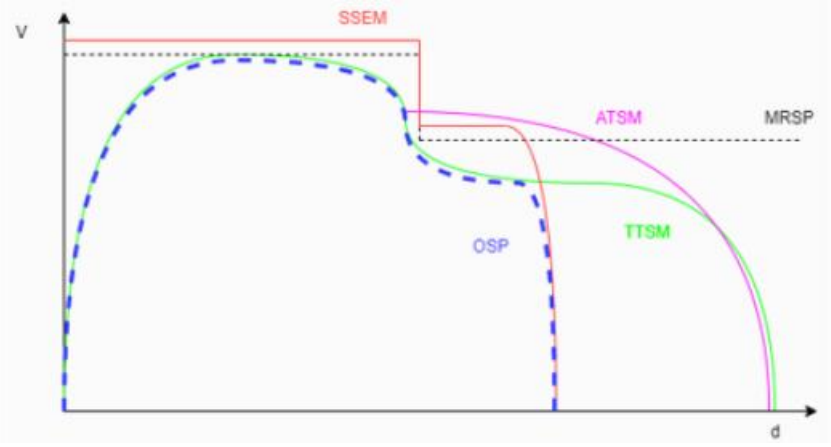


Figure 83: ATSM, TTSM, SSEM envelopes

4.7.3.2.1 TTSM and ATSM

The objective is to identify the profile that minimizes energy consumption among those that respect timetable and speed limits constraints.

The review of the scientific literature (see Appendix) highlights several categories of approaches: data-driven methods, deterministic algorithms, and machine learning-based algorithms. Most studies model train dynamics at a high level of abstraction, using fundamental principles of dynamics that incorporate traction force, braking force, gravitational force, and resistance to motion. The majority employ multi-objective optimization strategies, simultaneously addressing energy efficiency, timetable adherence, and stopping precision. Occasionally, additional optimization parameters such as wheel slip or mass variations are also considered. Recent developments also include the integration of onboard energy storage systems [ref. ERJU-777]. Metro and tram applications are predominant, whereas long-distance train scenarios remain relatively scarce. Moreover, existing publications have a strong algorithmic focus, emphasizing the novelty of an integrated system approach as proposed here.

Among the various approaches proposed in the literature, a numerical method based on graph construction and search [ref. ERJU-693] was selected, as it incorporates passing point constraints, i.e. locations where the train is required to pass within a specific time window, without stopping.

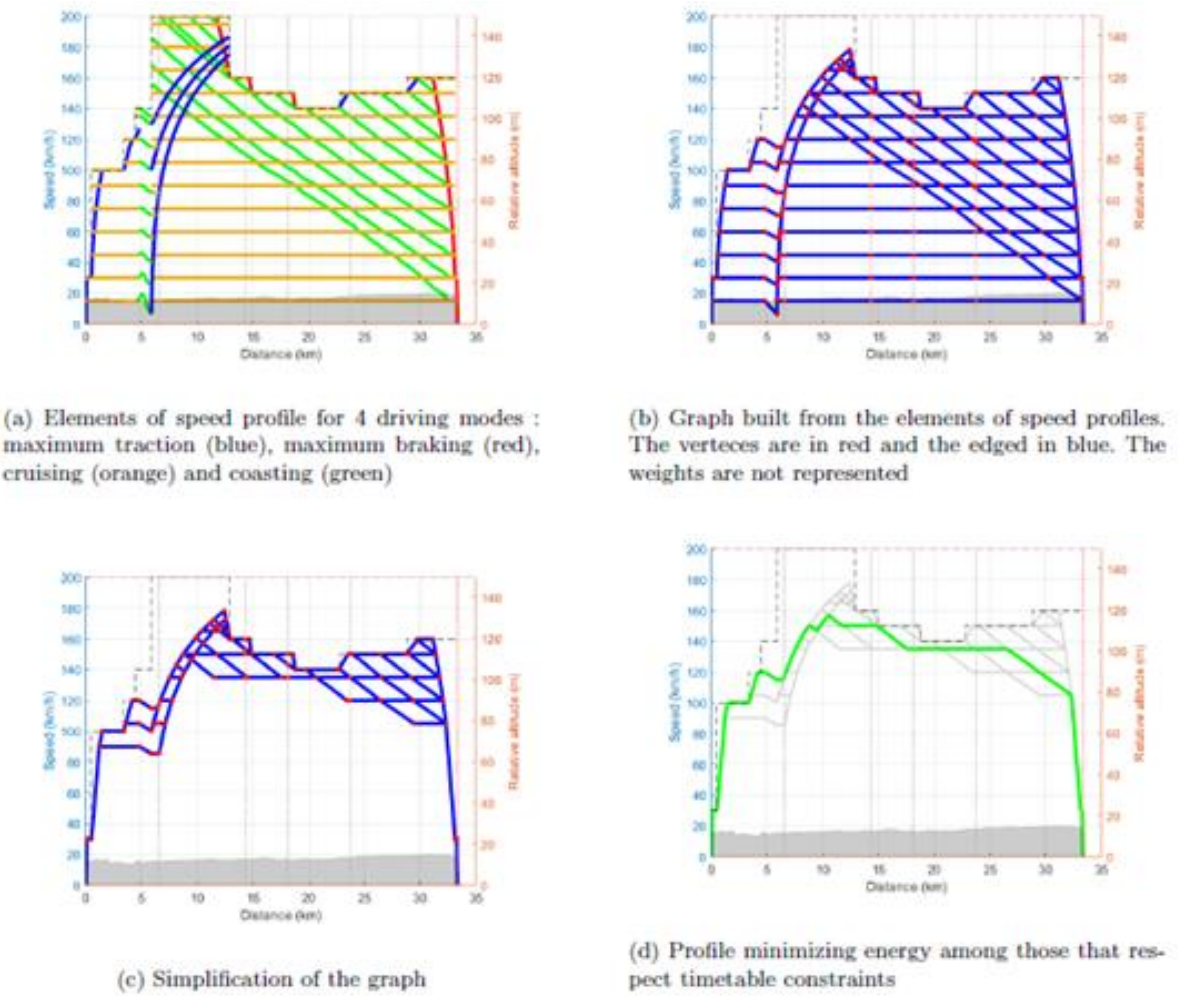
By applying Pontryagin's Maximum Principle [ref. ERJU-691], it can be shown that the optimal solution to the optimization problem is composed of four distinct driving modes: maximum traction, maximum braking, cruising (adjusting control to maintain constant speed) and coasting (applying no traction or braking effort). [ref. ERJU-689] Speed profile elements corresponding to these four modes are traced between the departure and arrival stations at strategically chosen points, respecting the speed limitations (see Figure 76a):

- maximum traction: at starting point, after traction cut-off section, after speed reduction area
- cruising: at fixed speed intervals. The width of the interval affects the calculation time.
- coasting: at fixed speed intervals beginning from the stopping curve.
- maximum braking: for the stopping curve, optimizing the stop accuracy and the passenger comfort.

Using these curves, a weighted directed graph is constructed, with vertices at the intersections of the curves and edges connecting these vertices along the curves. The edges are assigned weights based on the time and energy consumed during their traversal. (see Figure 76b) The graph is then simplified to reduce the computational effort of the search. Edges that would violate the timetable constraints if traversed are removed. (see Figure 76c) Sectioning zones were also taken into account in this strategy since no traction is possible in those sections.

Finally, a search is made in the simplified graph to find the path from the departure station to the arrival station which minimizes energy among those adhering to timetable constraints. (see Figure 76d) This type of problem belongs to Shortest Path Problem with Resources Constraints (SPPRC) problem category [ref. ERJU-692]. The solution is based on a dynamic programming approach, which enables the early elimination of suboptimal paths and of paths that do not satisfy the timetable constraints. This algorithm not only computes the optimal speed profile, but also provides a set of Pareto-optimal profiles, i.e. profiles for which there is no profile that consumes both less time and less energy (see figure 77). This graph search can become time-consuming when the graph is large,

so choosing the right algorithm parameters is essential to strike a good balance between computational cost and the optimality of the solution.



(a) Elements of speed profile for 4 driving modes : maximum traction (blue), maximum braking (red), cruising (orange) and coasting (green)

(b) Graph built from the elements of speed profiles. The verteces are in red and the edged in blue. The weights are not represented

(c) Simplification of the graph

(d) Profile minimizing energy among those that respect timetable constraints

Figure 84: Example of the phases of the proposed speed profile optimization heuristic. The grey zone is the elevation profile along the track

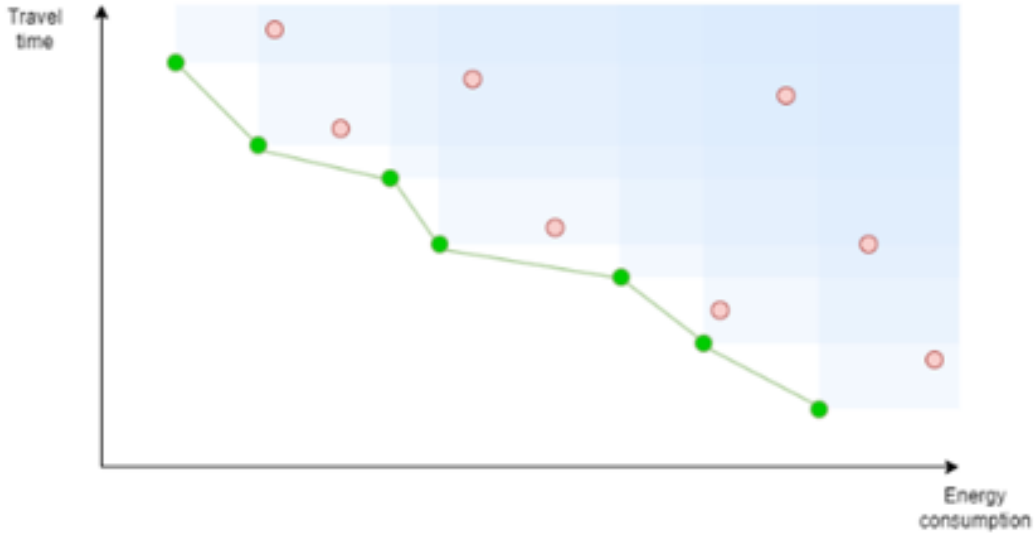


Figure 85: Pareto front. The green points represent Pareto-optimal solutions, i.e.solutions for which no other solution exists that consumes both less energy and less time

The optimal speed profile is the Time Table Speed Management (TTSM) profile. This speed profile is merged with the Automatic Train Stopping Management (ATSM) to stop precisely at the stopping points and the Supervised Speed Envelope Management (SSEM) to take into account the Movement Authority information transmitted by the ETCS.

In the GoA2 use case, integration within the OSP is limited to the ATSM and TTSM components. In the GoA4 use case, the SSEM component is additionally integrated into the OSP.

4.7.3.2.2 SSEM building and integration in OSP (GoA4)

The SSEM profile is a profile whose monitoring is intended to prevent intervention by the ETCS. To achieve this, the ETCS sends information to the ATO-OB, enabling it to generate the Emergency Brake Deceleration (EBD) curves, then the Emergency Brake Intervention (EBI) curves, and finally the SSEM profile (according to Subset 125, section 7.1.3). In our case, this information is provided by the SCV block.

EBD plotting

The plotting of the EBD curves is based on the deceleration grid A_BRAKE_SAFE (a grid of speed versus distance), the decelerations due to the gradient A_GRADIENT (negative for downhill slopes), and the maximum decelerations due to areas of reduced adhesion or inhibition of certain braking systems A_MAXREDADH (Subset 125, section 10.2.7.6). Based on these three types of deceleration, an A_SAFE deceleration grid is constructed according to the expression defined in Subset 026, section 3.13.6.2.1.3.

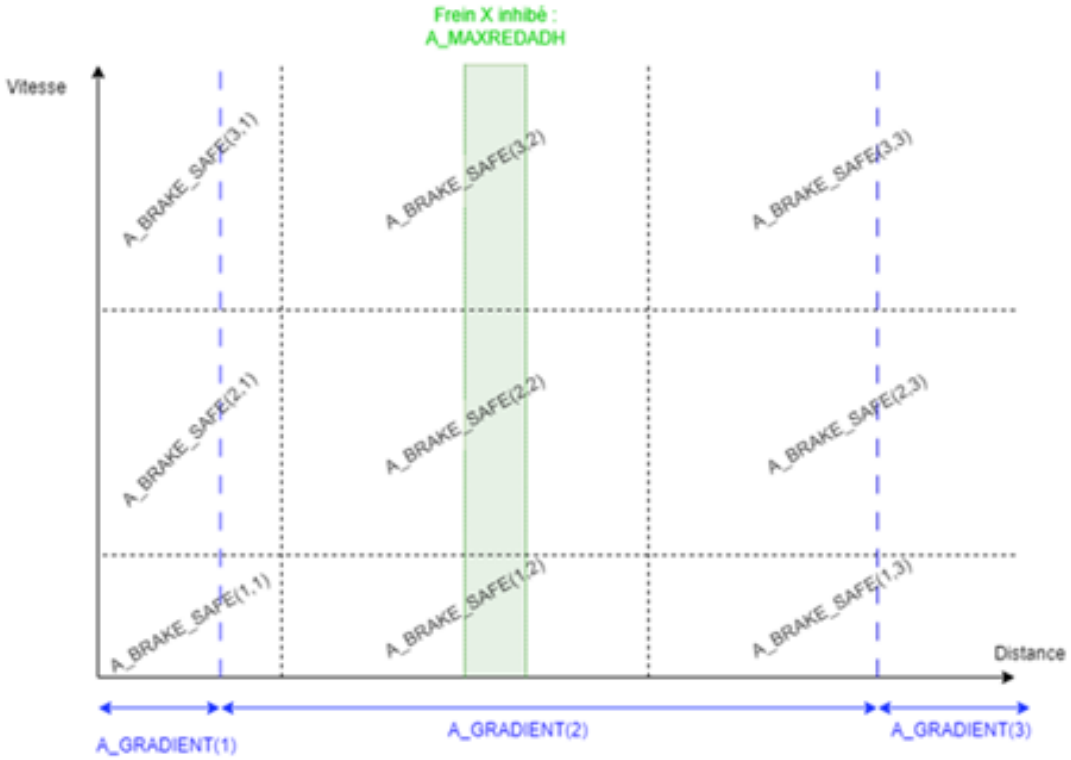


Figure 86: Input for EBD plotting

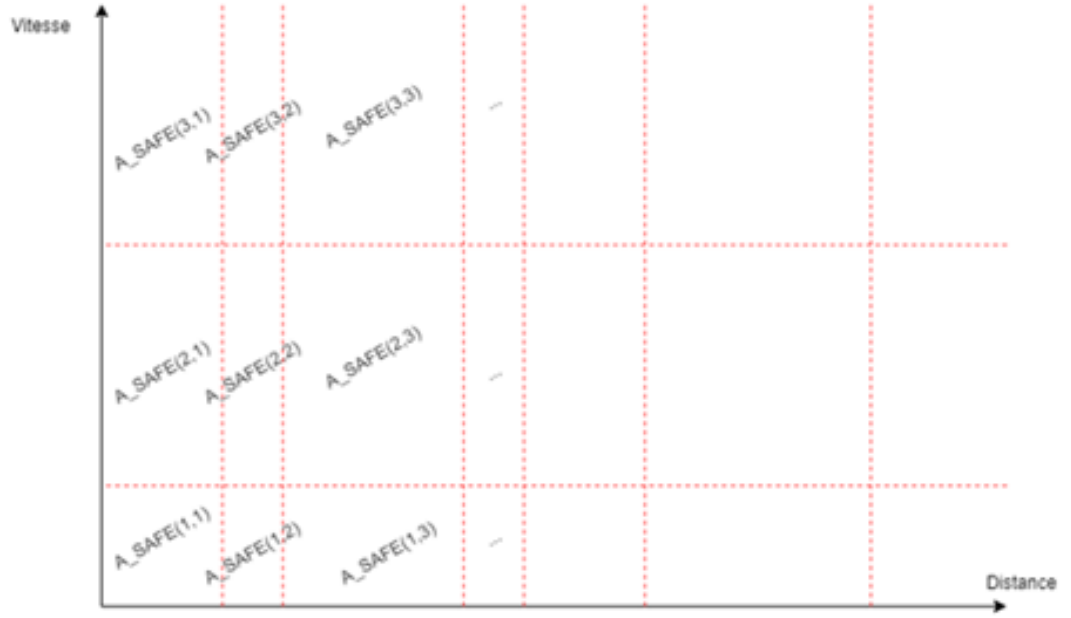


Figure 87: EBD plotting - A_safe

From the A_SAFE grid, which must at least cover the distance from the Max Safe Front End (MSFE) to the Supervised Location (SvL), the EBD curves are plotted for each target point,

following the model described in Subset 026.
Target points can be of three types (Subset 026, section 3.13.8.3):

- Reduction of the Most Restrictive Speed Profile (MRSP)
- Limit of Authority (LOA)
- Supervised Location (SvL)

It should be noted that no EBD curve is plotted when an End of Authority (EOA) replaces the LOA: for this type of point, the ETCS plots a Service Brake Deceleration (SBD) curve instead of an EBD. However, the ETCS does not transmit to the ATO-OB the information required to plot the SBD curve (see Subset 130).

EBI plotting

The plotting of the EBI is carried out in accordance with Subset 026, section 3.13.9.3.2. In the context of the Subset, the EBI is defined as a point on the distance/speed plane, where the speed corresponds to the train's estimated speed at the time of calculation.

The particularity of our approach lies in plotting EBI curves instead of individual points. The method involves calculating EBI points over a sampled range of speeds, as illustrated in the following figure.

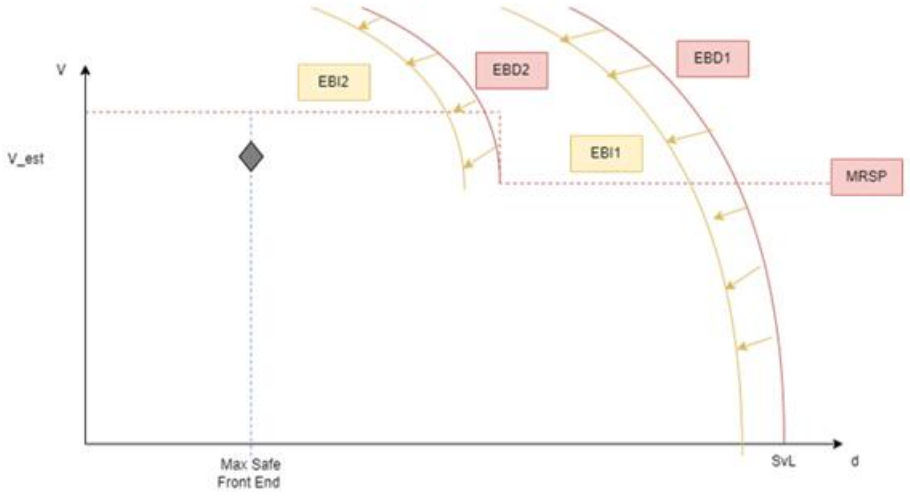


Figure 88: EBI deduction form EBD

SSEM plotting

The SSEM value at a given position x corresponds to the most restrictive speed among the following:

- The MRSP speed at position x, increased by dV_ebi (see Subset 026, section 3.13.9.2).
- The speeds of the EBI curves defined at position x.
- The permitted speed beyond the LOA, if a LOA exists and its position x_LOA is less than x (i.e., downstream of the LOA).

The SSEM profile always terminates at the Supervised Location (SvL).

4.7.3.2.3 Profile fusion

The TTSM, ATSM, and SSEM profile merging block consists of three main steps:

- Aligning the profiles to a common sampling: This step prepares the profile merging algorithm. The profiles are resampled so that they share the same set of x values over their respective definition intervals.
- Merging the profiles: For each position x, the minimum speed among the profiles defined at x is selected.
- Adding a temporal dimension to the merged profile: The integration of the inverse of the speed is applied to introduce a time dimension to the profile, which is essential for the Model Predictive Control (MPC).

4.7.3.2.4 Profile update

The following table lists the conditions under which a recalculation is triggered for each of the following modules:

- Calculation of the TTSM profile
- Calculation of the ATSM profile
- Calculation of the SSEM profile
- Merging of the profiles

Module	Trigger condition for the module	Rationale
TTSM calculation	Departure from a stopping point	The TTSM profile is calculated from one stopping point to the next.
TTSM calculation	Modification of the Journey Profile during the mission	Adjustment of timetable constraints for traffic regulation.
TTSM calculation	Modification of train characteristics (e.g., isolated bogie)	Change in train performance parameters.

Module	Trigger condition for the module	Rationale
TTSM calculation	Significant delay compared to the TTSM reference x(t) (condition to be specified) **and** the current profile being followed is TTSM	If the train is significantly delayed compared to the TTSM forecast, MPC tracking alone is not sufficient. A new profile must be calculated to achieve a more energy-efficient trajectory. The second condition prevents multiple unnecessary recalculations during ATSM and especially SSEM braking phases.
TTSM calculation	Switching from ATSM or SSEM tracking back to TTSM tracking	Transitioning back to TTSM "resets" any accumulated delay (see second condition under the merging module). Thus, MPC no longer considers previous delays.
ATSM calculation	Departure from a stopping point	The ATSM profile is calculated for the approach to the next stopping point.
ATSM calculation	Modification of the Journey Profile during the mission	Adjustment of timetable constraints for traffic regulation.
ATSM calculation	Modification of train characteristics (e.g., isolated bogie)	Change in train performance parameters.
SSEM calculation	Significant change in the data sent by ETCS to ATO-OB (e.g., extension of the MA)	New information impacts the supervised speed profile.
Profile merging	Recalculation of any of the TTSM, ATSM, or SSEM profiles	Modifying one of the profiles may result in changes to the Operational Speed Profile (OSP).
Profile merging	Switching into ATSM or SSEM mode	Allows a "re-temporalization" of the profile based on the current train status, effectively cancelling any previous delays to focus solely on precise stopping.

Table 18: Speed envelopes recalculation triggering conditions

4.7.3.3 Speed profile tracking

The Optimum Speed Profile (OSP) calculation module was then integrated into a simulator developed in Simulink, with the architecture shown in Figure 2.

The purpose of the controller is to track the OSP. A Model Predictive Controller (MPC) was chosen to make the best use of the available reference information we have until the next stop. This type of controller generates a command that anticipates future variations in the reference trajectory. At each timestep, an optimization problem is solved to find the series of commands over the time horizon that allows to minimize a cost function under several constraints.

This cost penalizes:

- poor tracking of the reference trajectory in both position and velocity (as determined in the process described in the previous section).
- large variations in the commands, in order to ensure passenger comfort.

This optimization is subject to the following constraints:

- the linearized dynamics of the train, which allow for predicting the future states of the train based on the applied commands.
- bounds on the command, due to maximum traction and braking efforts that can be applied.
- speed limitations, which depend on the train's position.

The solution of the optimization problem is a series of commands over the time horizon. Only the first command is applied during the current timestep, and then the optimization problem is solved again, with an updated time horizon.

The dynamic and energetic model of the train is validated by comparing the energy consumption in our simulation with the energy meter data for the same real speed profile. Any discrepancy between the two energy values may be due to an inaccurate estimation of the train's mass or the omission of certain disturbances in the model (e.g., wind).
The modelling tool can be used to evaluate the impact of those inaccuracies on energy consumption. It also facilitates the analysis of the performance requirements for localization, which allow to have a good driving behaviour.

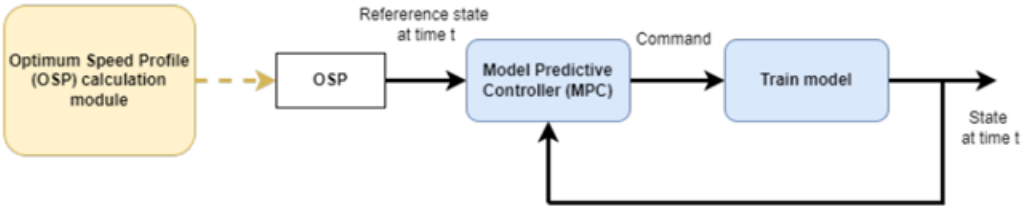


Figure 89: Speed profile tracking

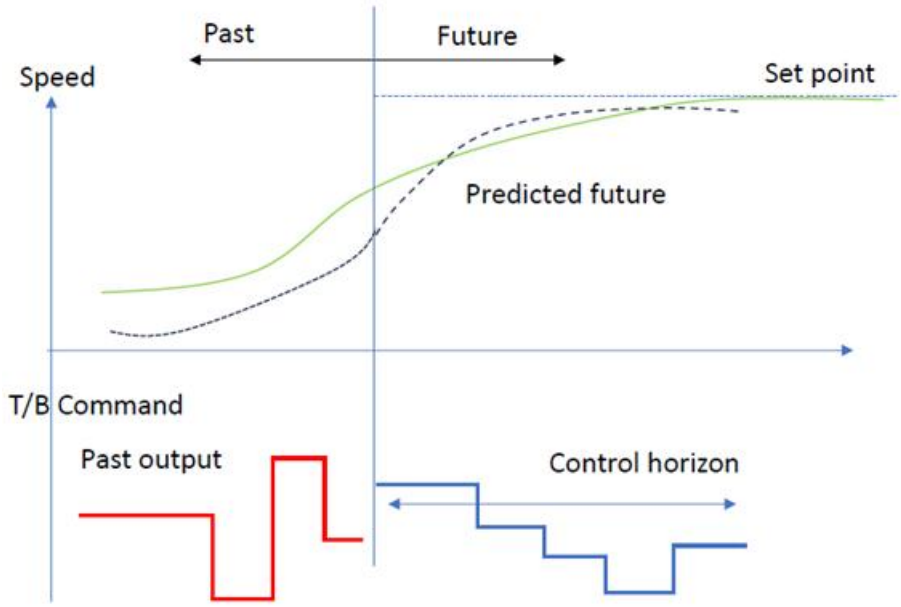


Figure 90: MPC principles

4.8 LINK WITH OTHER ERJU DELIVERABLE

An analysis based on published documents has been carried out to identify commonalities and differences in approach with other working groups.

WP	Deliverable	Similarity / Difference
R2DATO - WP5	D5.2 – Documentation of use cases for Perception system	Applicable use case: -UC5.2-0036: Long range detection and recognition of signals applicable to train route Other UC identified in WP5 are related to obstacles detection. So there are not applicable in our UC.

WP	Deliverable	Similarity / Difference
R2DATO - WP5	D5.4 – Documentation of use cases for Remote Driving	On-Boarding Box allows the driver to get on train safely by providing a light indication In the WP31 concept, the onboarding process relies on direct communication between the driver and the traffic manager. No specific onboard system is required. This communication can be carried out through a formal voice exchange or via a digital system. A dedicated ground-based system can support the offboarding process by verifying the physical absence of the driver from the train.

Figure 91: Link with R2- DATO delivrables

5 MODEL

This chapter describes the model that will represent the ATO system. It presents the objectives of the model and its consistent implementation. It also introduces the development environments and defines the simulation plan.

5.1 OBJECTIVES

The primary objective of the model is to enable the simulation of the GoA4 use case for technical movement between the depot and the station, under various signalling conditions and external system conditions (train status, weather conditions).

The model's capabilities to achieve this objective are:

- Ensuring the transition from GoA1 to GoA4
- Controlling traction and braking in accordance with the operational speed envelope
- Acquiring signalling information and adapting speed regulation accordingly
- Ensuring precise station stopping and completing the end-of-mission sequence
- Managing operational degraded mode:
    - Signalling fault
    - Train fault
- Managing system degraded mode: lack of signal detection

The secondary objective of the model is to simulate a GoA2 mission between two stations, while respecting scheduled timing and stop locations, and minimizing energy consumption.

The model's capabilities to achieve this objective are:

- Ensuring the transition from GoA1 to GoA2
- Controlling traction and braking in accordance with the operational speed envelope
- Ensuring precise station stopping

5.2 DEVELOPMENT ENVIRONMENT

Two modelling environments are used:

- Simulink toolbox attached to scientific software Matlab form MATHWORKS is used to model the ATO functions, the systems interacting with the ATO, as well as the environment supporting the simulation execution.
- Unreal Engine from EPIC GAMES is a graphics engine used to model a 3D environment.

5.2.1 Simulink

Simulink allows to model dynamic behaviour of the system. It's a graphical programming language based on a bloc library which enable generic function. Specific function can be coded in a script way. It enables simulation but also code generation which can be implemented in a hardware target.

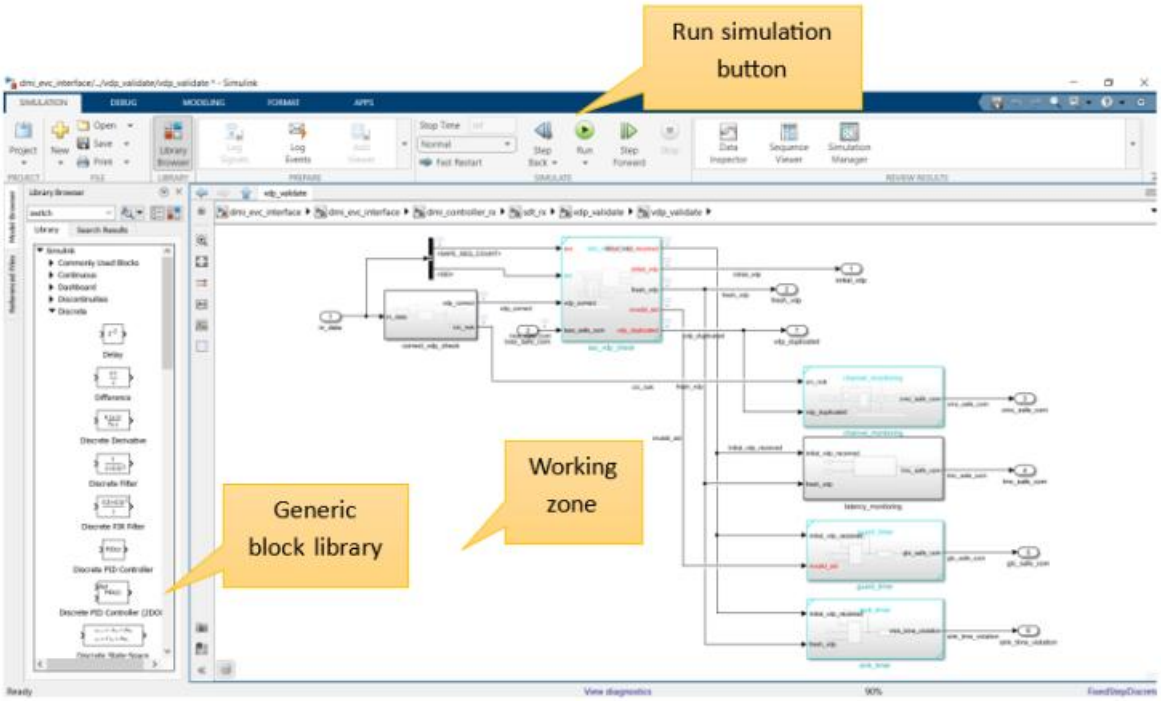


Figure 92: Simulink interface

The base software used is Matlab/Simulink. Matlab is a mathematical computing software, and Simulink is a simulation environment.

The main functions used are:

- Matlab:
  - Built-in mathematical functions
  - Scripts
  - Model management functions: configuration, change management (linked with Git)
- Simulink:
  - Data Dictionary: a file used to define data structures, associated with a Simulink model.
  - Bus: a data structure used to exchange information between two Simulink subsystems

- Blocks: predefined functions (e.g., signal generators, integrators, built-in Matlab functions, gains, signal summation, etc.)
- Referenced model

The base software can be extended through toolboxes. The toolboxes used in this project are:

- Simulink 3D Animation: enables data exchange between Simulink and Unreal Engine
- Deep Learning Toolbox: allows integration of AI models into Simulink

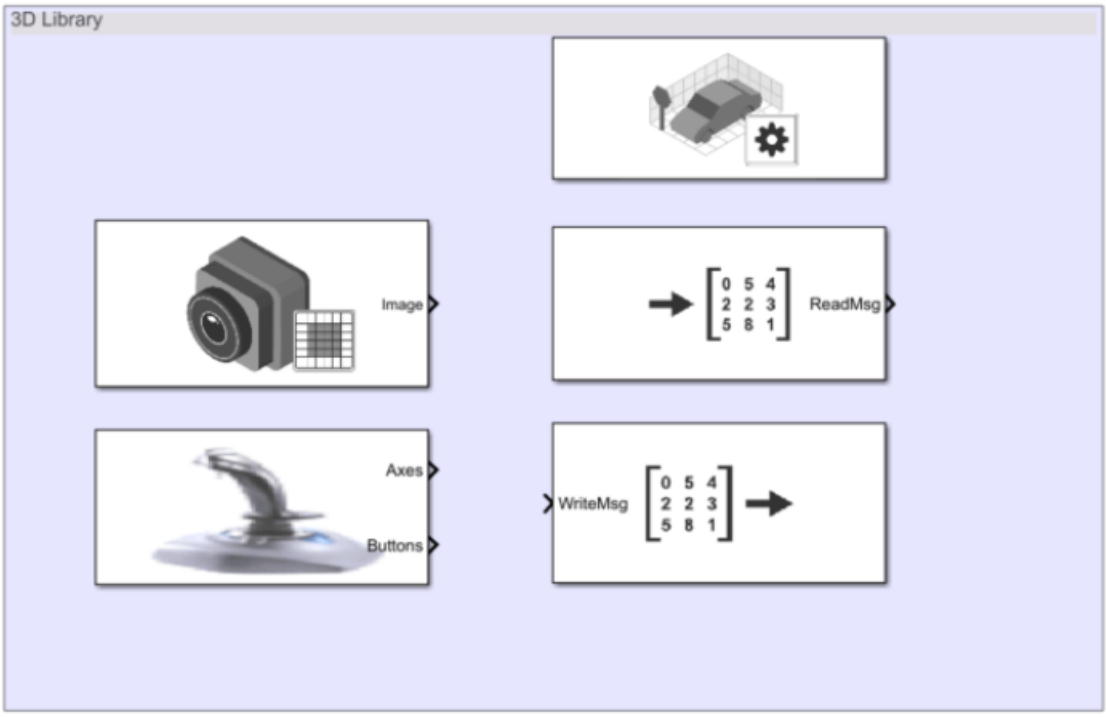


Figure 93: Simulink 3D animation interface block with unreal

5.2.2 Unreal Engine

Unreal Engine is a real-time 3D graphics engine developed by EPIC GAMES used for the modelling and simulation of virtual environments. It enables the creation of realistic and immersive 3D worlds by allowing users to either model directly within the platform or import 3D elements and assets from external sources. These 3D elements can be enhanced with materials and textures, which are applied to define their visual appearance.

The engine integrates animation capabilities, allowing objects within the environment to be animated and interact with their surroundings.

Unreal Engine provides a development environment, including a visual editor that allows developers to explore the modelled level, navigate through integrated components, and manage the scene layout. It also supports the creation and management of an object database, where elements such

as actors, materials, blueprints, and other assets can be stored, organized, and reused across different scenes or simulation scenarios.

The software can be extended by enabling plug-ins. The MathWorks plug-in is used to enable data exchange between Unreal Engine and Simulink.

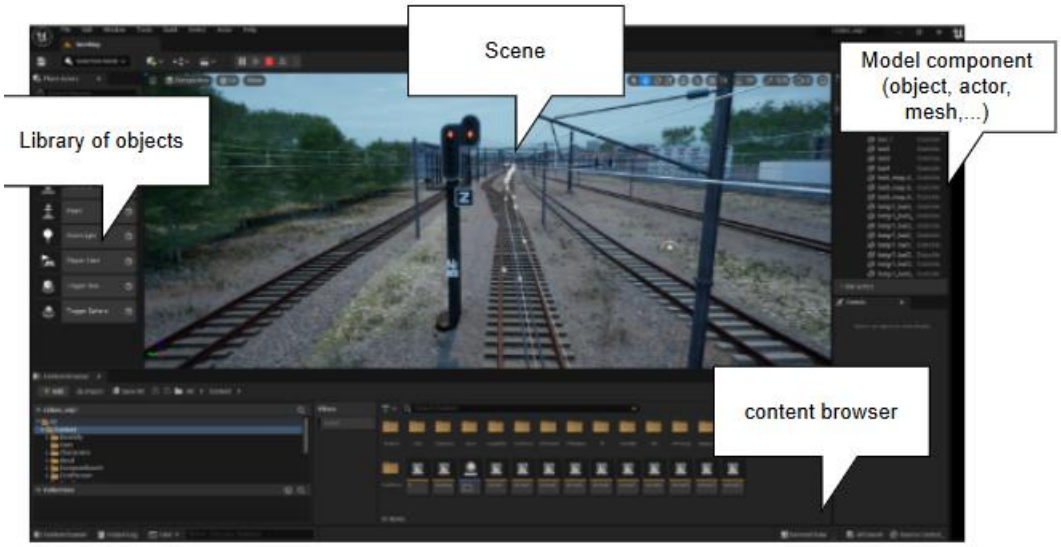


Figure 94: Unreal Engine interface

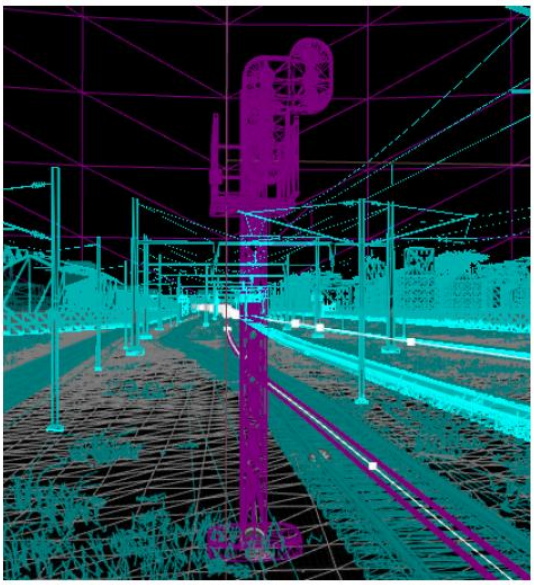


Figure 95: Signal 3D model

5.3 MODELLING PROCESS

The structure of the model follows the structure of the ATO system as defined in the logical architecture.

The interfaces between logical components are implemented using structured data within buses. Each subsystem (e.g., ADM, PER, SCV) is developed independently. The prior definition of interfaces ensures proper integration at a later stage.

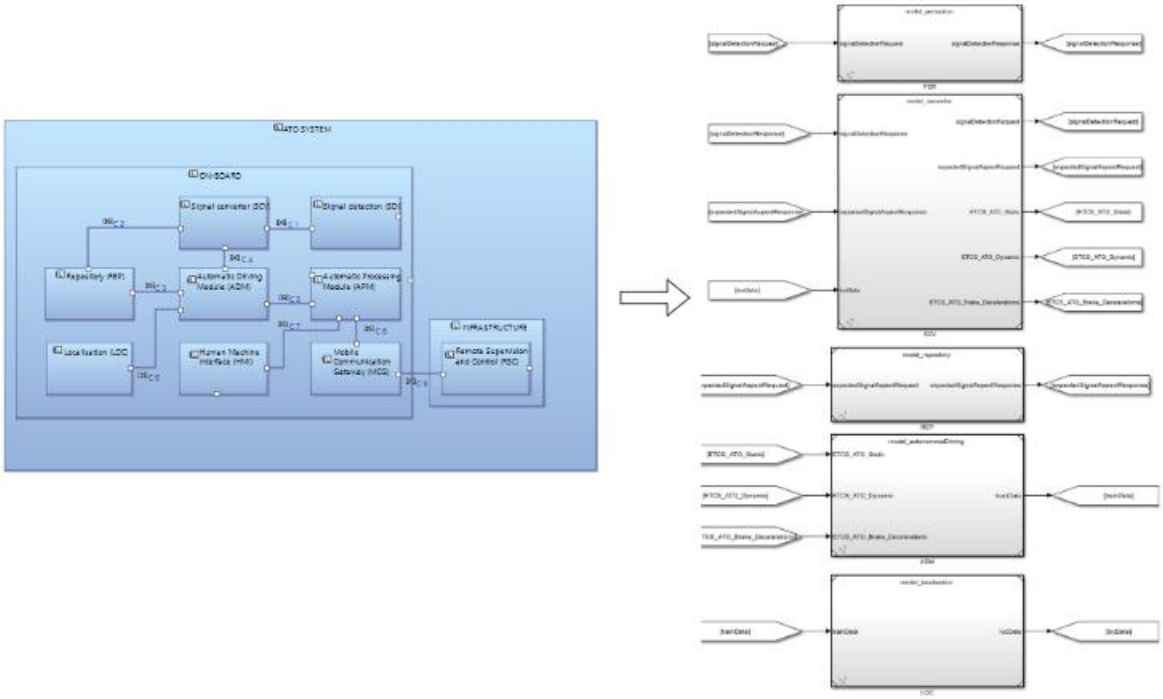


Figure 96: Transposition from architecture model to simulink model

Once the subsystems are developed and tested individually, they are integrated into the overall model, where system-level tests are conducted.

Regarding the graphical environment, a portion of the generic 3D elements (such as buildings, vehicles, etc.) is sourced from existing libraries. Railway-specific components (signals, tracks, overhead lines, etc.) are modelled in a dedicated development environment. These 3D elements are then imported into the graphics engine and assembled within a level, which represents the 3D scene.

Some 3D elements are built from basic components. For instance, Track is based on a 3D rail section.

The animation of certain components is managed using Blueprints, Unreal Engine's visual scripting system. This includes:

- The movement of the camera along the track, simulating the driver’s point of view
- The state changes of railway signals

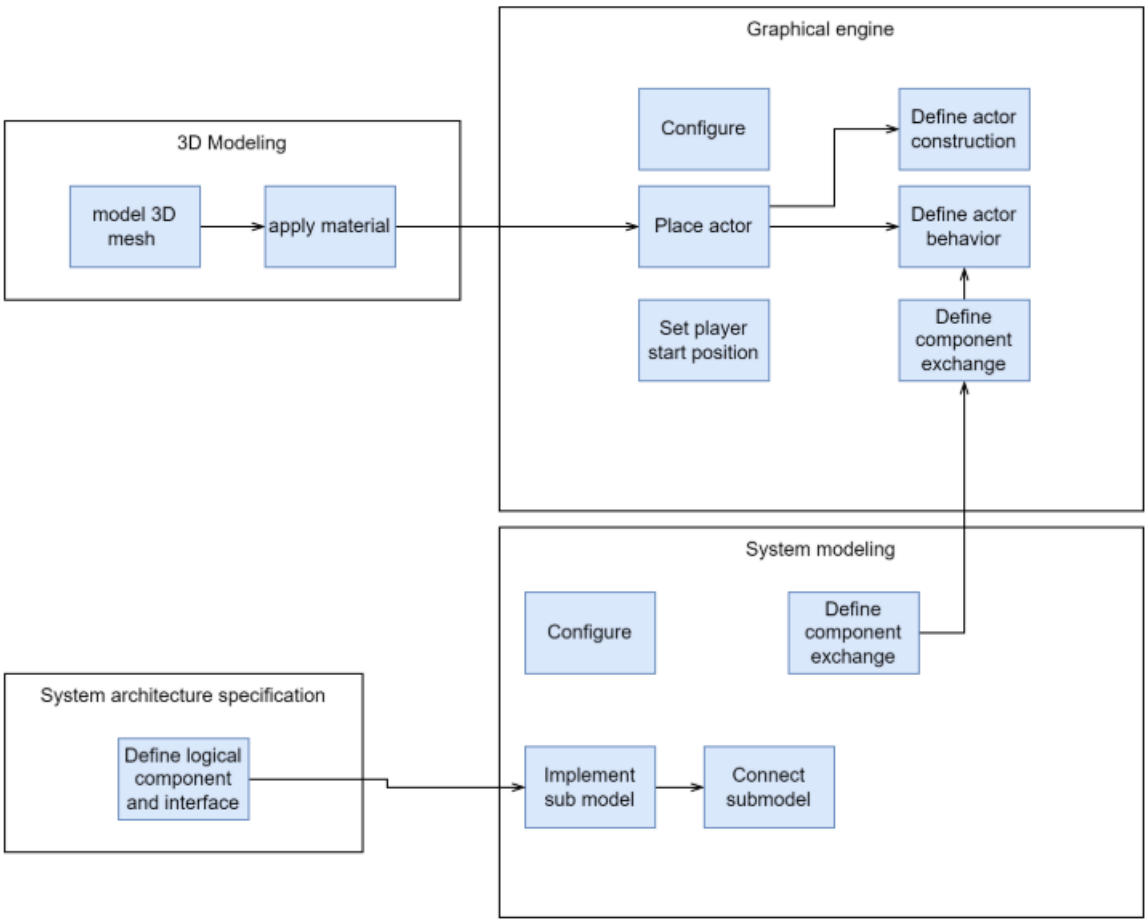


Figure 97: Modelling process

5.3.1 Define actor construction

Using Blueprints, it is possible to "construct" an actor based on basic graphical elements. For instance, a 3D rail section can be duplicated and positioned along a predefined curve to form the complete railway track on which the train operates.

5.3.2 Landscape and foliage

The software includes built-in features that make it easy to create landscapes and integrate vegetation into scenes.
Landscapes can be generated either from Geographic Information System (GIS) files, allowing for realistic terrain based on real-world data, or they can be manually sculpted within the editor to create custom environments.

Similarly, vegetation can be easily added using a plant asset library, which provides a variety of trees, bushes, and grasses. These elements can be "painted" directly onto the terrain using intuitive

tools, with control over parameters such as density, distribution, and variety, allowing for both natural-looking and performance-optimized environments.

5.3.3 Define actor behaviour

Blueprints are a visual scripting system in Unreal Engine, based on interconnected blocks that represent various programming elements.

They provide a user-friendly interface for implementing logic without writing traditional code, making them accessible to both programmers and non-programmers.

Blueprints support a wide range of functionalities, including:

- Event-driven activation: Triggering actions based on in-game events (e.g., collision, user input, time-based triggers)
- Function definition and execution: Creating reusable logic blocks that can be called from other parts of the system.
- Actor instantiation and manipulation: Spawning, modifying, or destroying objects (actors) within the 3D environment.
- Variable handling and state control: Managing data and conditions across the simulation.
- Debugging tools: Visual debugging options to trace execution flow and monitor variable states during runtime.

This system allows for rapid prototyping and development of complex interactions, animations, and behaviours within the 3D scene, all within an intuitive and highly visual development environment.

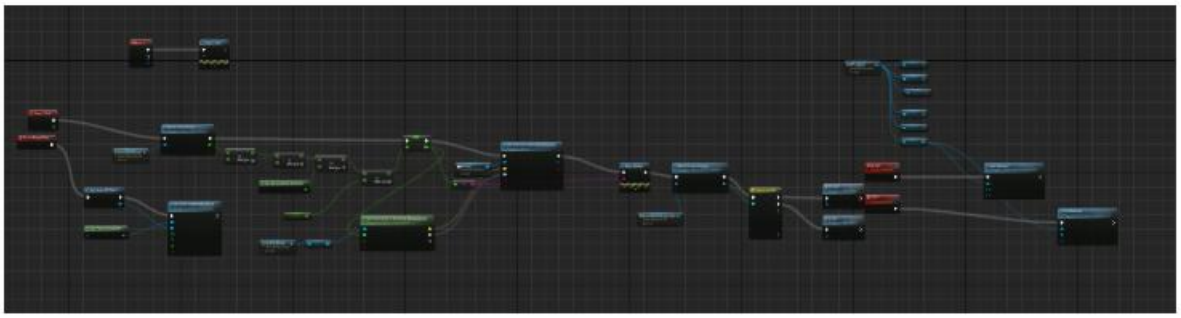


Figure 98: Camera movement along the line

The driving view is generated from the video feed of a camera placed on the track and moving along a predefined curve.

The position information provided by the ATO Simulink model is used to move the camera along the curve in a way that reflects the train’s dynamics and motion accurately.

Figure 99: Camera movement along the spline

5.3.4 Weather conditions

Weather conditions are simulated by applying a post-processing filter to the generated video image. This approach allows for the realistic rendering of various atmospheric effects, such as rain or snow without modifying the underlying 3D models or scene geometry.

Figure 100: Rain generation

5.3.5 Light conditions

The lighting is controlled by the position of the artificial sun, which is determined based on the time of day and the season.

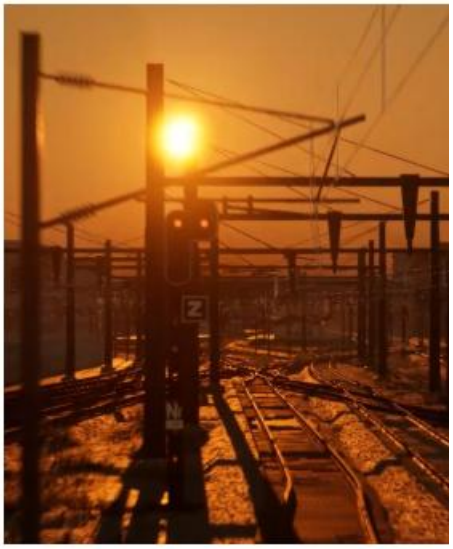


Figure 101: Light management

5.4 HIGH LEVEL MODEL ARCHITECTURE

The model is composed of three main sub-models:

- Simulation orchestration sub-model: generates non-modelled triggers, launches configuration and data logging scripts.
- Environment sub-model: models systems external to the system of interest (ATO) that are necessary for the simulation, such as the TCMS and train dynamics.
- ATO sub-models

The selection of a specific scenario by the operator enables the model to load the correct test configuration files. At the end of the simulation, the simulation environment computes graphical visualizations and writes log files that include variations of selected variables over time.

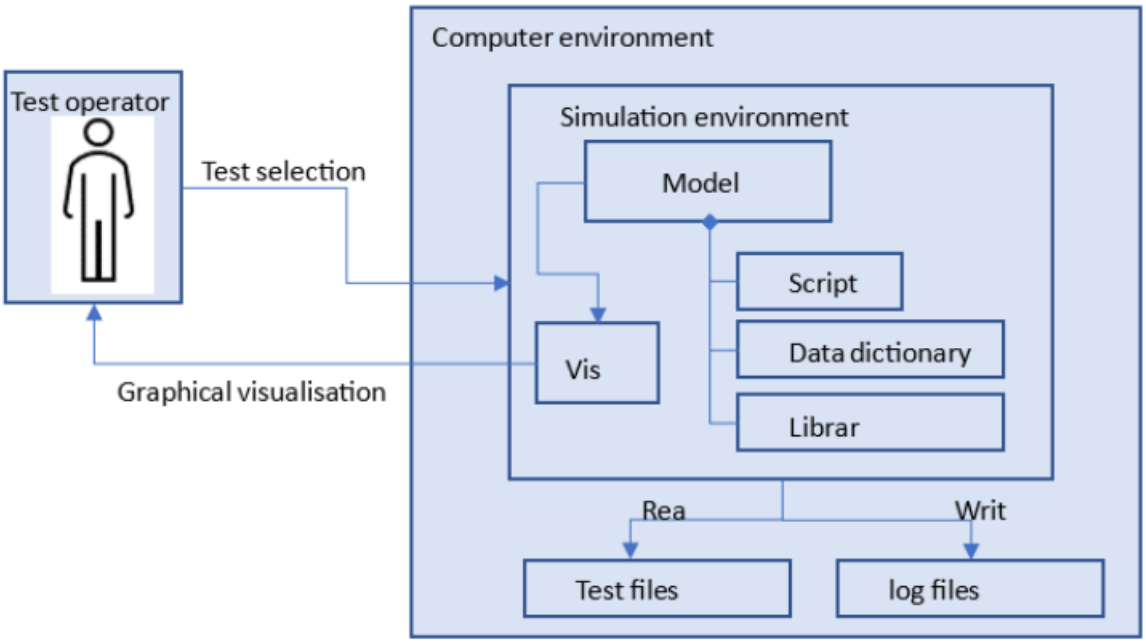


Figure 102: Model high level architecture

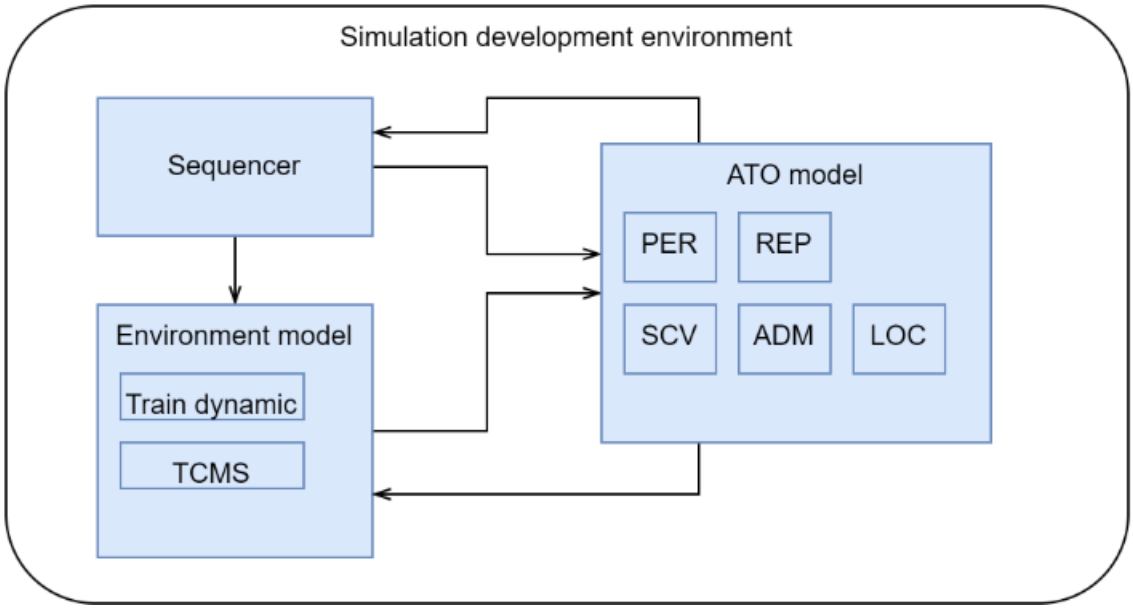


Figure 103: Model development environment

5.4.1 ATO model structure

The model follows the ATO logical architecture:

PER/SDC	Detect and acquire the signal
SCV	Convert signal to movement authority
REP	Save and share data
ADM	Elaborate speed envelope and traction/brake command
LOC	Elaborate train localization

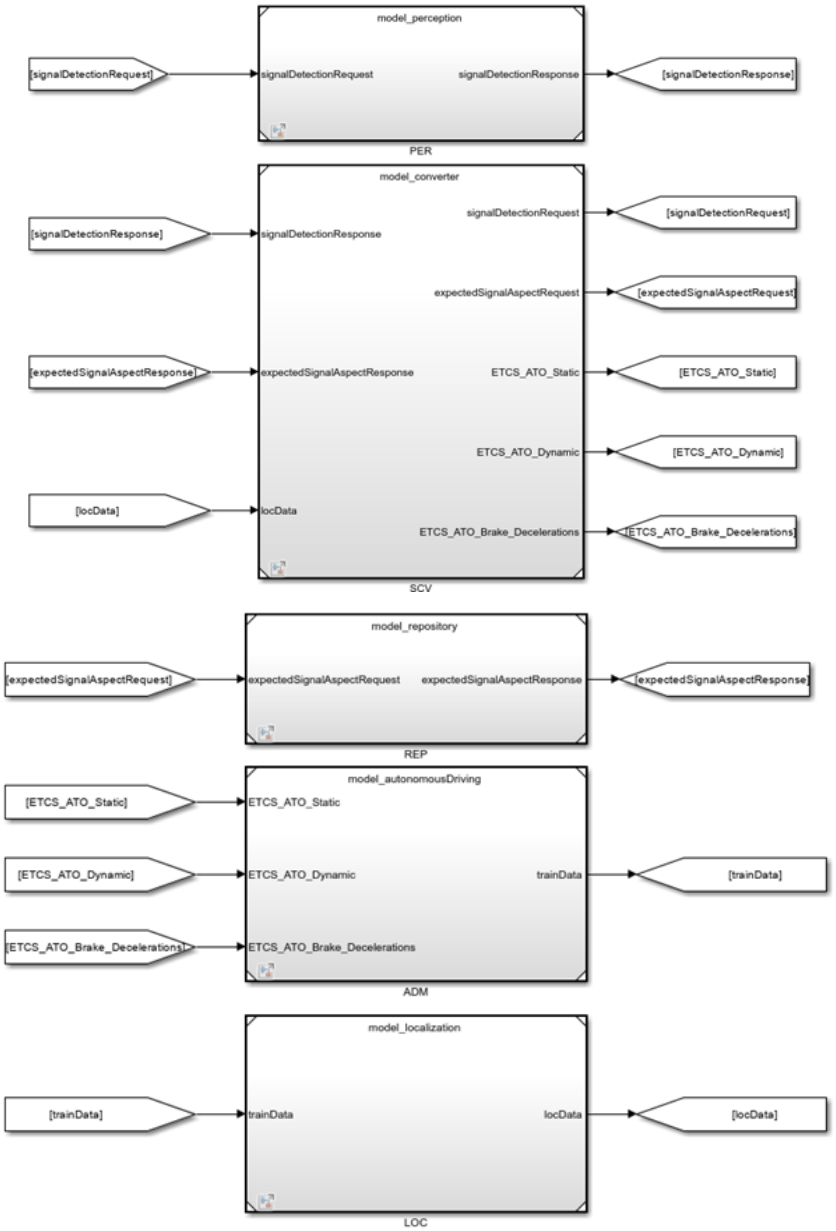


Figure 104: ATO model architecture

5.4.1.1 Subsystem interfaces

	PER	SCV	REP	ADM	LOC	
signalDetectionRequest	in	out				Request to detect a signal in the area
signalDetectionResponse	out	in				Detected signal
expectedSignalAspectResponse		in	out			
locData		in			out	Train localization data
expectedSignalAspectRequest		out	in			Request signal aspect expected
ETCS_ATO_Static		out		in		Subset 130
ETCS_ATO_Dynamic		out		in		Subset 130
ETCS_ATO_Brake_Decelerations		out		in		Subset 130
trainData				out	in	

Table 19: model subsystem interfaces

Interface is defined through structured format in Bus component in Simulink. For e.g.:

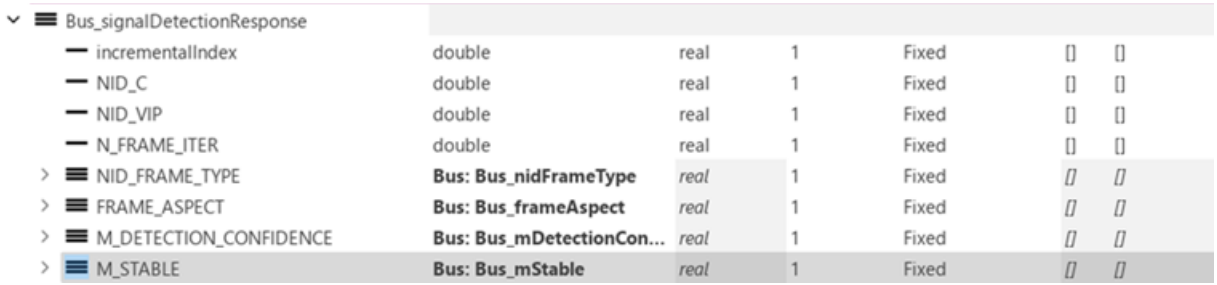


Figure 105: interface between SDC and SCV

Bus_EtcsAtoDynamic						
Q_RC	double	real	1	Fixed	[]	[]
Q_DIRSOLR	double	real	1	Fixed	[]	[]
Q_DSOLR	double	real	1	Fixed	[]	[]
N_LOC_REF	double	real	1	Fixed	[]	[]
T_LOC_REF	double	real	1	Fixed	[]	[]
N_LOC_REFBALISE	double	real	1	Fixed	[]	[]
NID_ACTIVE_ANTENNA_SOLR	double	real	1	Fixed	[]	[]
NID_REFBALISE	**Bus: Bus_NidRefbalise**	*real*	1	Fixed	[]	[]
T_LOC_REFBALISE	double	real	1	Fixed	[]	[]
V_EST	double	real	1	Fixed	[]	[]
V_DELTA0	double	real	1	Fixed	[]	[]
A_EST	double	real	1	Fixed	[]	[]
L_UNCERTAINTY_OVERREADING	double	real	1	Fixed	[]	[]
L_UNCERTAINTY_UNDERREADING	double	real	1	Fixed	[]	[]
V_MRSP	double	real	[1000 2]	Fixed	[]	[]
V_MRSP_size	double	real	1	Fixed	[]	[]
A_GRADIENT	double	real	[51 2]	Fixed	[]	[]
A_GRADIENT_size	double	real	1	Fixed	[]	[]
A_MAXREDADH	double	real	[21 2]	Fixed	[]	[]
A_MAXREDADH_size	double	real	1	Fixed	[]	[]
N_LOC_EBI	double	real	1	Fixed	[]	[]
N_LOC_EOALOA	double	real	1	Fixed	[]	[]
V_EOALOA	double	real	1	Fixed	[]	[]
N_LOC_SVL	double	real	1	Fixed	[]	[]
T_TRACTION	double	real	1	Fixed	[]	[]
T_TRACTION_SPEED	double	real	1	Fixed	[]	[]
T_BE_REACT	double	real	1	Fixed	[]	[]
T_BEREM	double	real	1	Fixed	[]	[]
T_BEREM_SPEED	double	real	1	Fixed	[]	[]
V_PERMITTED	double	real	1	Fixed	[]	[]
V_RELEASE_ATO	double	real	1	Fixed	[]	[]
N_LOC_RSM	double	real	1	Fixed	[]	[]

Figure 106: interface between SCV and ADM

5.4.2 Model data

Model data are stored in a structured format within a data dictionary. The WP6ository mainly includes:

- A signal database, detailing signal types, their possible states, and associated movement authority characteristics (e.g., MA, gradient profiles, speed profiles)
- A mission profile database, containing predefined mission scenarios and parameters

Bus_JourneyProfile						
NID_PACKET_ATO	double	real	[1 1]	Fixed	[]	[]
NID_SP	double	real	[1 5]	Fixed	[]	[]
NID_TP	double	real	[1 11]	Fixed	[]	[]
T_Latest_Arrival_Date	double	real	[1 11]	Fixed	[]	[]
T_Latest_Arrival_Seconds	double	real	[1 11]	Fixed	[]	[]
T_Arrival_Window	double	real	[1 11]	Fixed	[]	[]
Q_Stop_Skip_Pass	double	real	[1 11]	Fixed	[]	[]
T_Departure_Date	double	real	[1 11]	Fixed	[]	[]
T_Departure_Seconds	double	real	[1 11]	Fixed	[]	[]
N_TP	double	real	[1 5]	Fixed	[]	[]

Figure 107: Journey profile model data stored in REP

5.4.3 Signal Detection subsystem

The Signal detection (SD) (also called “perception”) subsystem is responsible for detecting railway signalling. It takes as input a frame from the 3D environment, processed frame by frame, as well as the detection request, 'signalDetectionRequest'. The Signal Detection block receives the detection request, performs detection over multiple frames to create a sliding window, and obtains a stable and reliable result. The detection process consists of four main steps: rail detection, chassis detection of the signal light, detection of active light spots, and processing these detections to generate a stable and formatted detection response.

5.4.3.1 Global SD subsystem

The SD subsystem is divided into 3 subsystems, detailed below:

1. **Pre_processing_SD Block**: Extracts relevant information from the detection request and stores key data for use throughout the detection process.
2. **SD Block**: Composed of subsystems that perform the detection based on the request data and the 3D environment.
3. **Converter Block**: Adapts the detection response to the required format.
4. **Unit delay block**: Introduce a one-step simulation delay, which is essential in our system modelling to allow the SCV block to send a request to the SD system.

The SD system has the following interfaces:

- **Input**:
  • *signalDetectionRequest*: Detection request sent by the SCV to the SD system.
- **Output**:
  • *signalDetectionResponse*: Detection response sent by the SD to the SCV system.

Figure 108: Signal Detection subsystem architecture

5.4.3.2 Detailed SD subsystem

For detailed information about the input and output signal components, please refer to Section 11.4.6.1.1 "Acquire Signal State.". The same applies to the Pre_processing_SD and Converter subsystems, which are also described in that section.

- The Pre_processing_SD block consists exclusively of MATLAB Function Blocks and is fully described in Section 11.4.6.11.
- The Converter block is also entirely presented in that same section.

5.4.3.2.1 SD block

Composed of subsystems that perform the detection based on the request data and the 3D environment.

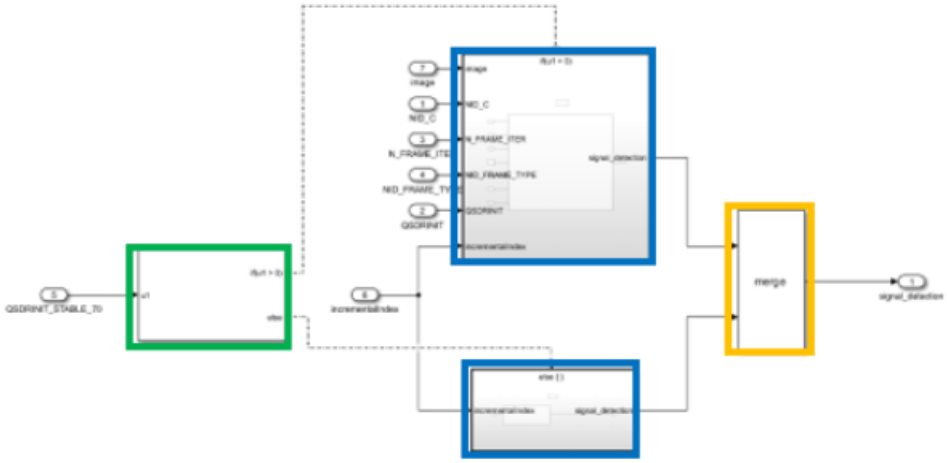


Figure 109: Signal detection subsystem

This subsystem contains an *If* block, *If Action Subsystem block*, and a *Merge* block.

- The ***If block*** is responsible for checking a predefined variable.
- The ***If Action Subsystems*** are controlled by this variable to determine whether detection is required. If detection is needed, *Subsystem 1* is activated to perform the detection. If detection is not required, *Subsystem 2* is activated, which sends an empty response.

- The ***Merge block*** combines the outputs of both *If Action Subsystems* to ensure that only one output is generated, as either *Subsystem 1* or *Subsystem 2* is active at any given time, sending the response signal.

5.4.3.2.2 If Action Subsystem 1

Only the ***If Action Subsystem 1*** will be detailed, as If Action *Subsystem 2* consists solely of a MATLAB Function Block that generates an empty response.

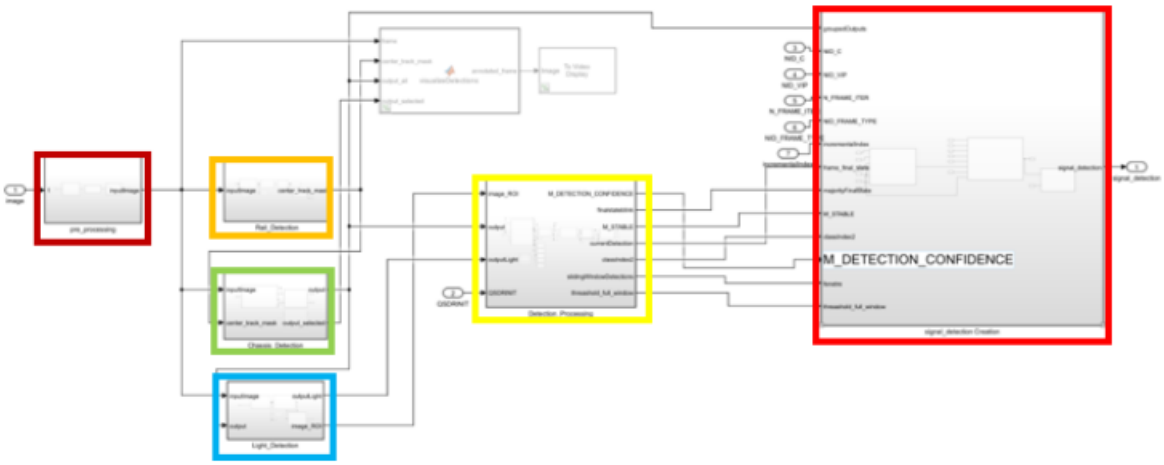


Figure 110: if action subsystem

If Action Subsystem 1, responsible for performing detection, is further divided into 6 subsystems:

1. **A *global image preprocessing subsystem*** to prepare the image for use by deep learning models.
2. **A *rail detection subsystem*** that identifies the rail being travelled on to determine its position.
3. **A *chassis detection subsystem*** that focuses on detecting the chassis related to our track, obtaining its type and position.
4. **A *lamp detection subsystem*** that identifies active lamps on the chassis and determines their positions.
5. **A coordination block** that consolidates all detections, establishes the signal state for a given frame, and incorporates it into a sliding window to ensure a stable and accurate signal state over a sequence of images.
6. **A *subsystem for assembling the gathered information*** and creating the final response signal in the format expected by the SCV block.

All of these subsystems are composed of a series of MATLAB Function Blocks, whose logic is explained in Section 11.4.6.1.1 "Acquire Signal State," so you can refer to that section for further details.

The only blocks with a slightly different structure are those containing deep learning models, as they

use a *Predict Block* to perform inference with pre-trained models. To illustrate, we will detail one of these subsystems as an example.

Example Detail: *"Rail_Detection Subsystem"*

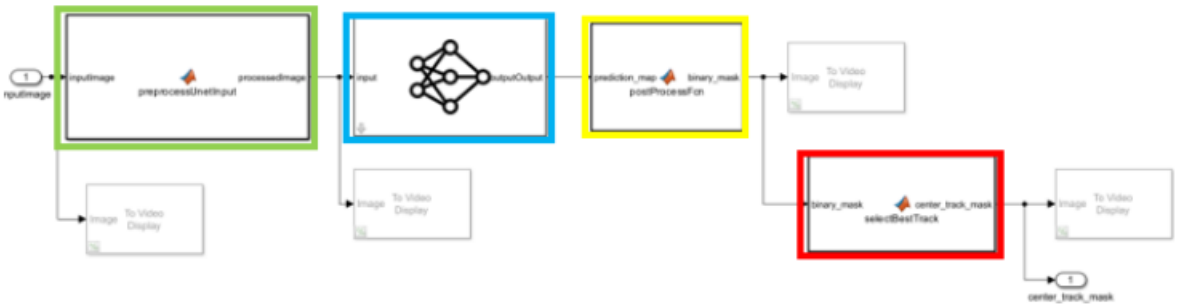


Figure 111: Rail detection subsystem

The "Rail_Detection" subsystem takes the pre-processed image input and outputs a binary mask highlighting the track under the train. Internally, it uses three MATLAB Function Blocks and one Predict Block:

1. **Image Preprocessing Function Block**: Applies model-specific steps, such as grayscale conversion and normalization to adapt the image for the segmentation network.
2. **Predict Block**: Runs inference using a pre-trained deep-learning model to generate a raw segmentation map.
3. **Post-Processing Function Block**: Cleans the raw map by applying geometric and morphological filtering to remove noise and small artifacts.
4. **ROI Selection Function Block**: Selects the primary rail region via simple region-of-interest heuristics to ensure a single, stable output.

MATLAB Function Blocks

- **Purpose:** Embed custom MATLAB code for tasks not covered by standard Simulink blocks.
- **Why use them:** They let you prototype algorithms quickly using MATLAB's built-in functions and toolboxes.
- **How they work:** You write a function y = fcn(u) script inside the block, Simulink then calls it at each time step.
- **Benefits:** Full access to MATLAB language and libraries, seamless integration with Simulink logging and debugging.
- **Limitations:** Can slow simulation if the code is complex, less efficient than hand-optimized S-functions or C code for high-performance needs.

Predict Blocks

- **Purpose:** Perform out-of-the-box deep-learning inference on inputs within a Simulink model.
- **Why use them:** They wrap pretrained networks (imported via MATLAB's Deep Learning Toolbox) into a reusable Simulink block.
- **How they work:** During simulation or code generation, the block loads network weights and runs a forward pass, outputting class scores or regression values.
- **Benefits:** Simplifies deployment of neural networks in real-time systems, supports GPU acceleration if available, integrates with Simulink Coder for embedded targets.
- **Limitations:** Inference speed depends on model size and hardware, updating network architectures requires regenerating the block, adds dependency on the Deep Learning Toolbox.

5.4.4 Train dynamic

The train dynamics are computed by applying the fundamental principle of dynamics, taking into account the components of gravity (track gradient), resistance to motion, and traction/braking effort along the longitudinal axis of the track.

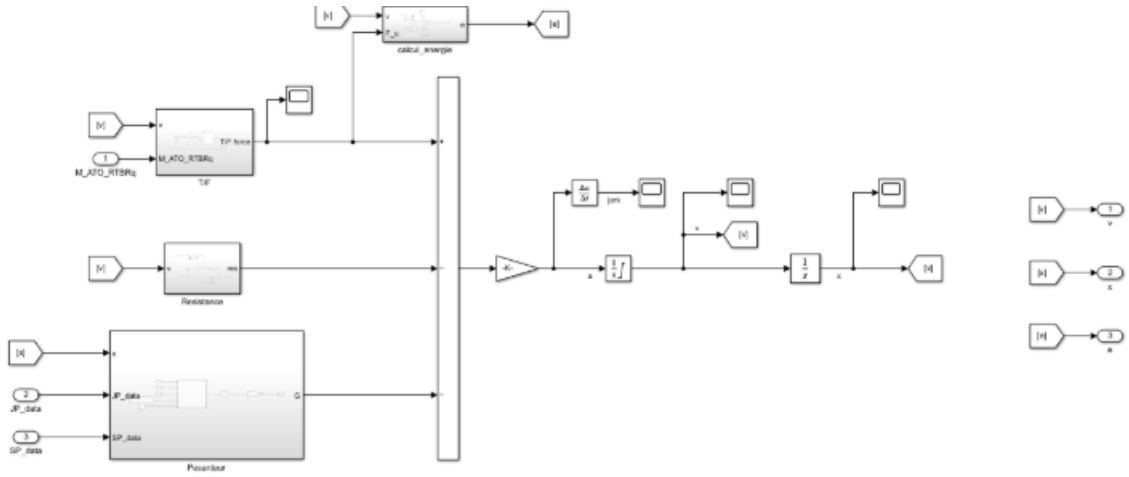


Figure 112: Train dynamics subsystem

5.4.5 ADM

A state machine manages the transitions of the ADM between different modes, each defining the applicable speed envelope (e.g., TTSM, ATSM, or stop-hold mode).
This state machine activates the appropriate envelope determination blocks, enabling the traction/braking controller to consider the correct speed envelope based on the current driving phase—whether the train is in line running or station stopping.

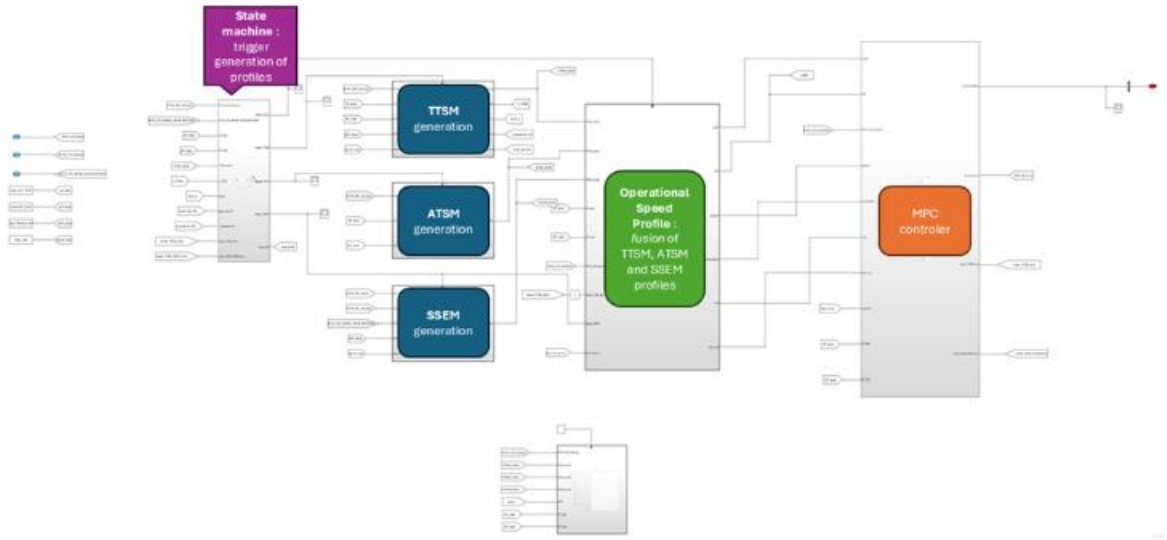


Figure 113: ADM subsystem

The Automatic Driving Module generates traction/braking (T/B) commands to ensure adherence to a reference speed/distance profile. This reference profile, known as the Operational Speed Profile (OSP), is computed based on three distinct speed profiles:

- TimeTable Speed Management (TTSM) profile: this profile is the result of an optimization for which the goal is to minimize energy consumption while respecting timetable constraints and speed limits.
- Supervised Speed Envelope Management (SSEM) profile: this profile constraints the speed of the train to avoid the intervention of the ETCS due to the crossing of an intervention supervision limit.
- Automatic Train Stopping Management (ATSM) profile: profile that allows precise stopping at train station.

Under specific conditions, a state machine triggers the recomputing of one or several of these profiles. For instance, if the movement authority has been extended, the SSEM shall be updated.
Based on those 3 profiles, a single profile is built by “merging” information contained in TTSM, ATSM and SSEM profiles. The resulting profile is called Operational Speed Profile (OSP).
Finally, a Model Predictive Controller (MPC) is designed to track the OSP despite external disturbances and sensors and actuators noises. This type of controller takes into account the reference profile on a finite time horizon in the future to build the current command, hence anticipating future variations of the reference profile.

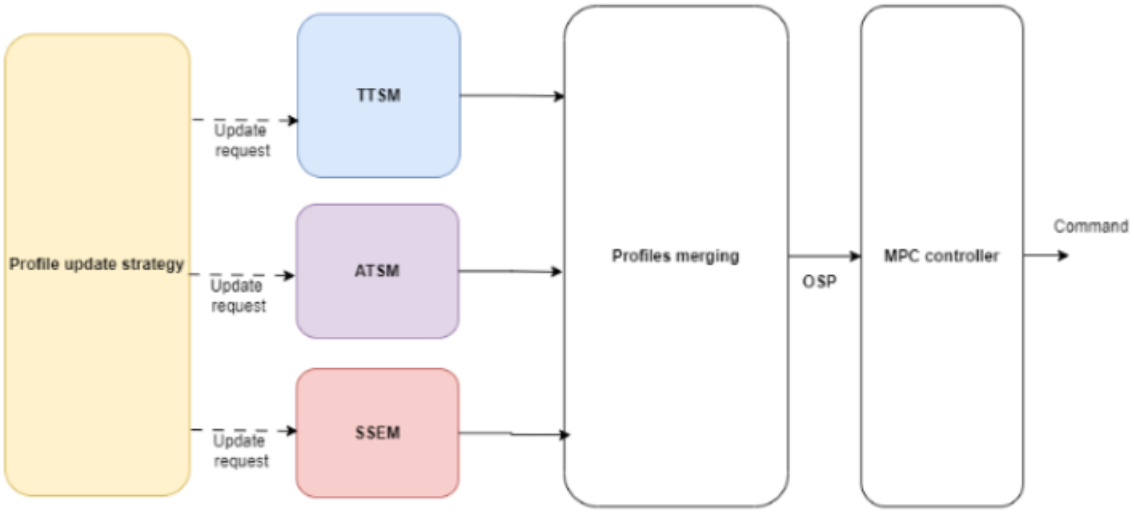


Figure 114: ADM architecture

Data Sources:

- The Journey Profile (JP) and Schedule Profile (SP) for the mission are used for TTSM and ATSM profiles. They remain valid throughout the entire mission. The format of these two packets is specified in the subset 139 on ATO-OB / ATO-TS interface.
- Data for train modelling: mass of the train, resistance to motion parameters, ...
- ETCS data (transmitted by SCV) allows to take into account the Movement Authority (MA) and the emergency brake curves considered by the ETCS. It includes:
- SvL (Supervised Location) and EoA (End of Authority) location
- MRSP (Most Restrictive Speed Profile), which combines the SSP (Static Speed Profile) and TSR (Temporary Speed Restrictions, per §11.2.2)
- Reachable deceleration values along the MA, for EBD calculation

5.4.6 Signal converter

Signal-converter subsystem is composed of several function blocks.

Two Simulink blocks are dedicated to unit and format conversion, in order to ensure flexible data transmission and exchange between subsystems. Signals exchanged with other subsystems are embedded into bus format. The use of this standard, recommended by TAURO specification, allows to easily keep track of signals, as they are referenced by their name. No matter their position in the genuine bus creator, their name ensures a clear identification through the various subsystems.

Check_signal_coherence function block runs several comparisons between Perception data and Repository data. For a given signal, it ensures signalling state reported on trackside is listed among the many theoretical states declared for this very signal. To accomplish this task, data about the incoming signal to be found on the trackside is pulled from Repository database. Various comparisons between expected frames, light signals displayed and signs and the current video acquisition are run, in order to confirm acquisition validity. Once validity is confirmed, a Movement Authorisation is built. This authorisation allows the train to cover up to two blocks ahead, before encountering a stop instruction.

Safe_Deceleration provides acceleration due to gradient, and the various control curves SBD and EBD associated respectively to EOA and SvL/LOA targets. For acceleration resulting from slope profile on track, train length is taken into consideration. Over the entire length of the train, depending on its location, the strongest gradient is considered, even though only a small fraction of length is involved. This calculation is very conservative and prevents underestimating gradient effect on the train physics.

Supervision_Limit_Monitoring block consumes SBD and EBD curves data and converts it to deceleration grid and various speed and distance data, necessary to generate speed envelopes in the next subsystem, a.k.a. Autonomous Driving Module (ADM). Based on the nature of each speed curve to produce, specific function blocks are used to apply various formulas. These formulas are all derived from technical subset documents and specifications.

Input and output ports are managed by bus of individual signals. There are three main input ports:

- locData gathers data from localisation subsystem. Information about current Virtual Information Point, antennas position and estimation of speed and train location are provided at a regular rate.
- signalDetectionResponse is provided by Perception subsystem. It contains signalling attributes acquired by camera and various sensors for trackside detection and acquisition. This bus is updated only once in a while, when the train enters an area preceding the location of a signal and runs the acquisition of a video stream.
- expectedSignalAspectResponse is received from Repository subsystem. It contains a set of signalling attributes, corresponding to all possible states that can be observed on trackside for the next signal. These data are only delivered on request, which means that it is only updated once in a while, on demand.

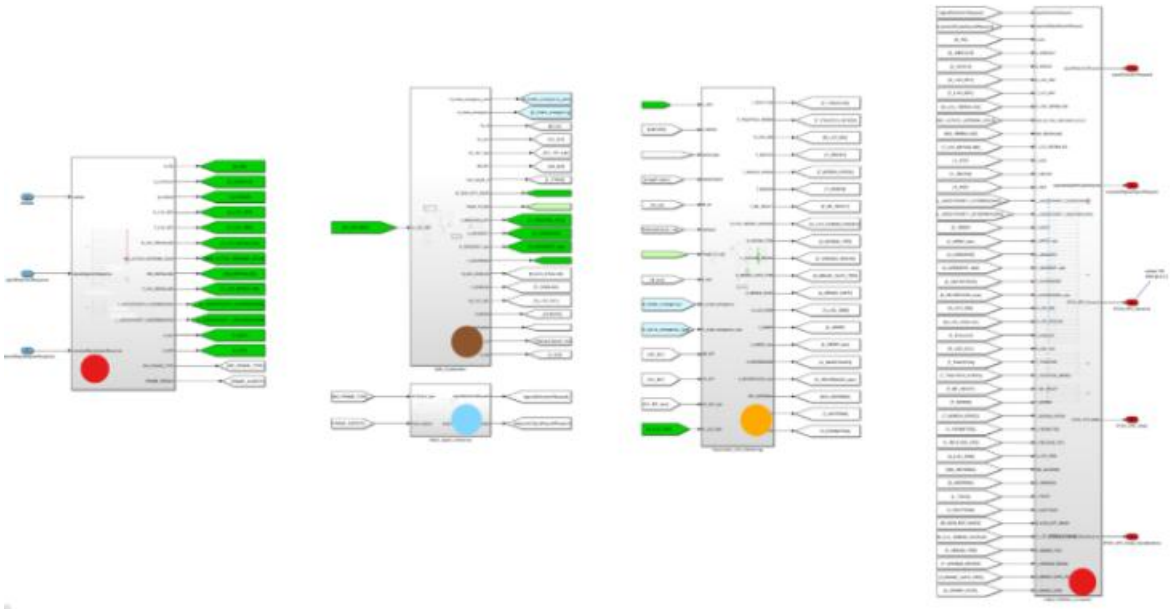


Figure 115: Signal converter subsystem

Legend:
Interface function blocks for input and output conversion and datatyping
Check_signal_coherence function block
Safe_Deceleration function block
Supervision_limit_monitoring function block

Output ports are divided into five buses:

- signalDetectionRequest: a set of information is sent to Perception subsystem, in order to request a video acquisition and analysis, as the next signal is approaching. Some additional information is sent too, specifying in which area of the image frame to look, what type of frame to look for, etc...
- expectedSignalAspectRequest: a set of information is sent to Repository subsystem, in order to request data regarding the next signal. The aim is to get a comprehensive scope of all the states to expect from the next signal, from a theoretical perspective.
- ETCS_ATO_Dynamic: it contains all the dynamic data regarding braking points, acceleration slopes and EBD curves construction. These data will be used by Autonomous Driving Module to generate speed envelopes.
- ETCS_ATO_Static: it contains all the data related to static train specifications, such as antennas position, train max speed, train length.
- ETCS_ATO_Brake_Decelerations: this bus contains all data arrays related to deceleration grid. This table takes into account gradient change locations and speed transitions.

5.5 DRIVING SIMULATOR

The implementation of an ATO model and a 3D environment forms the foundational building blocks of a driving simulator. This type of simulator, intended as an engineering tool, serves as a design aid. It allows real-time evaluation of the system's behaviour and comparison with manual driving scenarios. The modelling approach follows a white-box methodology, granting access to all internal model variables—both in real-time and during post-processing.

5.5.1 Objectives

A driving simulator can serve multiple purposes.

The first and most established use is driver training, where the focus lies on line familiarization and mastering operational procedures. Industrial suppliers offer a wide range of simulators, with varying levels of realism—from fully equipped mock-up cabins to simplified desktop environments. For operators responsible for driver training, the key challenges include:

- ensuring an appropriate level of realism relative to the training objectives,
- maintaining scalability and adaptability as rolling stock and systems evolve,
- and guaranteeing traceability and transparency in the simulator's processes and assessment logic.

In this context, the Model-Based Design (MBD) approach proves particularly relevant. Its modular structure, full access to internal model variables, and support for rapid prototyping make it well suited to developing flexible and transparent training tools.

A second, complementary use case for the simulator is as a system design and validation platform for systems interfacing with the driver.

Such a simulator offers several advantages across the development lifecycle:

- It enables early validation of functional requirements through the simulation of complex operational scenarios, prior to hardware deployment.
- It supports rapid prototyping and iterative design, allowing engineers to adjust algorithms or control strategies and immediately evaluate their effects in a realistic environment.
- It facilitates testing of failure scenarios, performance optimization, and interface verification between onboard subsystems (e.g., ATO, TCMS, signalling).

A driving simulator is particularly valuable for assessing human factors in GoA2 ATO systems, where certain interactions between the train and the driver are retained. These include, for example:

- engagement/disengagement procedures,
- the display and interpretation of speed limit gauges,

Human factors evaluation requires immersing the driver in a realistic operational context to observe,

characterize, and assess behavioural responses in both nominal and degraded scenarios [ref. ERJU-763].

5.5.2 Architecture

The simulator consists of the following components:

- Modelling PC: A high-performance workstation equipped with powerful graphics capabilities. It hosts the Simulink and Unreal Engine development environments. This PC also handles visual outputs displayed across multiple screens and interfaces with a joystick.
- Joystick: Emulates the driver's traction/braking controller.
- Screen 1 – Driver Interface: Displays a representation of the ETCS screen (based on ERA 15560 specifications), providing the driver with standard train control information.
- Screen 2 – 3D Environment: Shows the immersive 3D simulation of the railway environment, enhancing realism and providing context for system behaviour.
- Screen 3 – Simulation Control Interface: Allows management of the simulation (e.g., start/stop commands, visualization of results).
- Screen 4 – System Monitoring & Analysis: Displays real-time and historical system behaviour through various plots, including:
    - Mission characteristics: gradient, speed profile, and timing points
    - Current and historical speed of the train
    - Current MPC command: traction/braking control and predicted control horizon

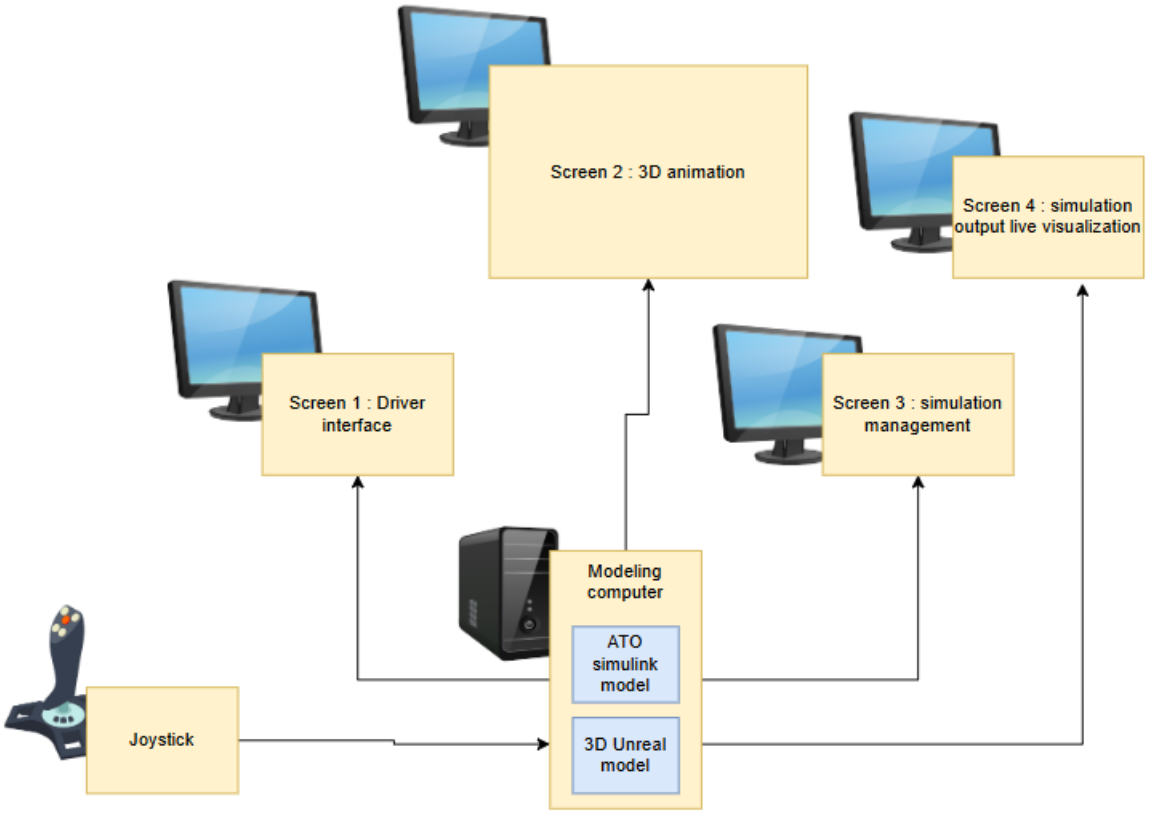


Figure 116: Simulator architecture

From a logical standpoint, the model implemented in Simulink transmits information to the 3D environment in order to generate the video stream consistent with the test scenario. This video stream is then captured by the perception system.

Human-Machine Interfaces (HMIs) are generated using the App Designer tool integrated into MATLAB. Additionally, HTML components are embedded to produce views compliant with the ETCS standard, in accordance with the ERA DMI 15560 specification.

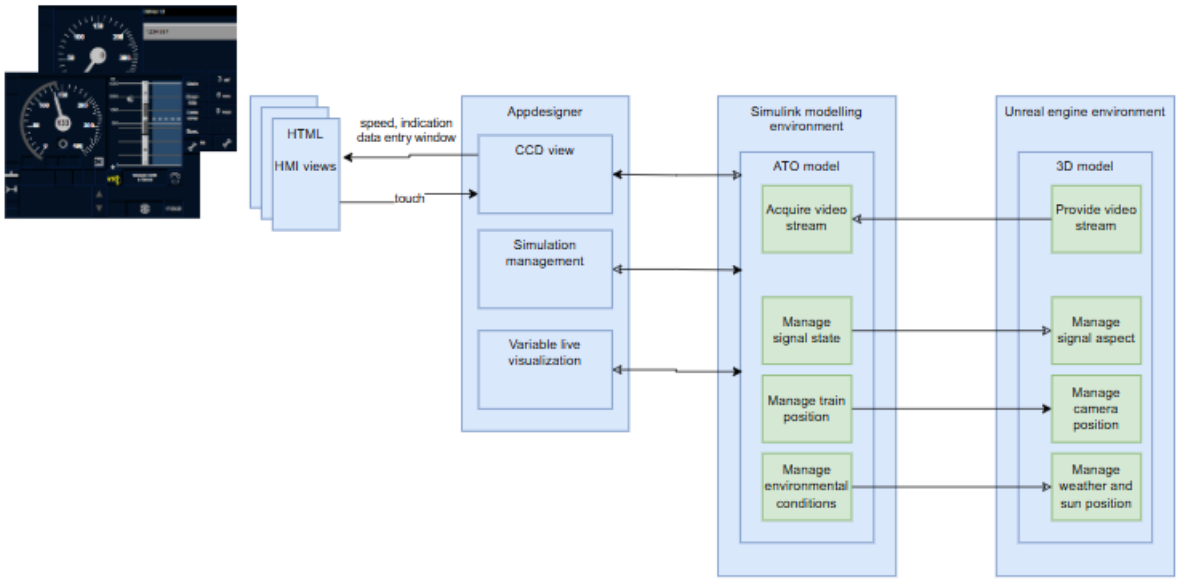


Figure 117: Simulator logicical architecture

5.6 MODELLING FEEDBACK REPORT

The ATO system modelling integrates the reference frameworks defined by the TSI (e.g., Subset 125, etc.).
The implementation in a model, which required clarifying topics not covered by the standard, raised several issues that are addressed in this chapter.

Subject	related subset	Description
Supervised Speed Envelope integration in operational envelope	subset 125 subset 130	The braking curve associated with the EOA was not taken into account, as it is a Service Brake Deceleration (SBD) curve, and the data transmitted from the ETCS to the ATO-OB under Subset 130 is insufficient to plot such a curve. This presents a problem, as the ETCS will intervene if the intervention curves related to the EOA are exceeded. The Release Speed was not taken into account. The Release Speed is a speed limit that replaces the MRSP near the EOA, up to the SvL. It allows the train to move slowly to reach an information point (e.g., a signal or balise). Subset 130 does provide for the communication of the Release Speed value and the position from which it must be considered (start of the RSM). However, this start RSM point is located on an SBI curve, which we are unable to plot (see previous point). Therefore, it is not possible to plot a braking curve leading to the start RSM point. Traction curves should be added when the MRSP speed increases, to avoid "jumps" in the SSEM profile. These jumps are particularly problematic in the case of Temporary Speed Restrictions (TSRs), as TSRs are not taken into account when plotting the TTSM and ATSM profiles.
Stopping point time window	subset 125 subset 126	Although time windows are defined for passing points, they are not specified for stopping points. To enable effective optimization, the algorithm must account for a time interval—however minimal—to ensure convergence.
Presence of stopping point	subset 125 subset 126	Including a stopping point in the Journey Profile (JP) is not mandatory. However, this complicates the implementation of an ATO system, which benefits from having a clear target point (position and speed). Since speed is not constrained at passing points, they cannot serve as reliable targets. One possible workaround is to assign a target speed at the last passing point of the JP (e.g., 80% of the line's nominal speed), but this approach may lead to suboptimal driving behaviour.
Odometer	Subset 125 Subset 130	The primary positioning source is the ETCS, as defined in Subset-130. However, Subset-125 (§7.10.1.8) allows for direct access to onboard sensors. The performance of the ATO is closely tied to the accuracy of the positioning data. When relying on the ETCS's odometry—which is subject to drift between periodic corrections (see Subset-091)—the controller may respond inappropriately to sudden re-localizations, sometimes involving shifts of several or even tens of meters. This raises the question: could a more accurate, though potentially less safe, odometry source be more suitable for ATO optimization? This also implies that the ATO designer must have access to deployment assumptions, such as the distance between balises. Such constraints could be documented in a dedicated specification, for example within Subset-041.
EBI distance	Subset 130	Why does the ETCS transmit the distance to the EBI? Which requirement in Subset-125 is it related to?

Subject	related subset	Description
Clarity	subset	Some points in the subsets could be improved to enhance clarity. This would make them easier to understand and implement. It would also facilitate error detection, helping to reduce the spread of change requests over time. In this regard, the support of an MBSE approach could be applied, making it easier to link related requirements that are spread across different specifications. Some formulations are difficult to grasp, for example §10.2.5.4. *"All the position counter values shall be such that the absolute value of the difference between the current position counter and the position counter at the time when the Location Reference Balise of the ETCS SOLR was detected corresponds to the estimated front end as used by the ETCS-OB, when it is augmented/diminished (depending whether a movement in the direction of the train orientation corresponds to an increase or a decrease of the raw counter respectively) by the distance between the antenna active at the time when the Location Reference Balise of the ETCS SOLR was detected and the front end of the train."*

Table 20: Feedback collected during the modelling activities

5.7 SIMULATION PLAN

The simulations must enable the evaluation of the ATO system's behaviour as specified, and verify that its functional coverage meets the operational requirements.

5.7.1 GoA2 simulations

The GoA2 simulations must address the following objectives:

- Functional validation in accordance with the applicable standards, including Subset-125 and relevant interface subsets (126, 130, 139).
- Energy efficiency improvement, targeting an average energy consumption reduction of 10% compared to GoA1 operation.
- Configuration of simulation parameters representative of a real operational scenario, including:
    - Static speed profile
    - Gradient profile
    - Rolling stock characteristics: acceleration, deceleration, and mass
    - Timing points: location, scheduled passage times, and tolerances
    - Stopping points: stopping position and stopping tolerance

- GoA1 driving energy consumption reference

The simulation observables are defined to verify the achievement of these objectives:

Observable	Expected results
Speed	Compliance with the theoretical static speed profile
Traction/brake profile Traction/brake command history	Profile tracking gap
Energy consumption	Consumption reduction
Distance travelled Timetable	Stopping accuracy Keeping to the Timetable

Table 21: GoA2 expected results

5.7.2 GoA4 simulations

The GoA4 simulations must address the following objectives:

- Functional validation in accordance with the system definition
- Signal detection and evaluation robustness by perception system.
- Configuration of simulation parameters representative of a real operational scenario, including:
  - Static speed profile
  - Gradient profile
  - Rolling stock characteristics: acceleration, deceleration, and mass
  - Timing points: location, scheduled passage times, and tolerances
  - Stopping points: stopping position and stopping tolerance
  - Lineside signalling: signal position, possible states

The simulation observables are defined to verify the achievement of these objectives:

Observable	Expected results
Speed	Compliance with the theoretical static speed profile
Traction/brake profile Traction/brake command history	Profile tracking gap

Energy consumption	Consumption reduction
Distance travelled Timetable	Stopping accuracy Keeping to the Timetable
Detected signal	false/right positive/negative

Table 22: GoA4 expected results

5.7.2.1 Line configuration

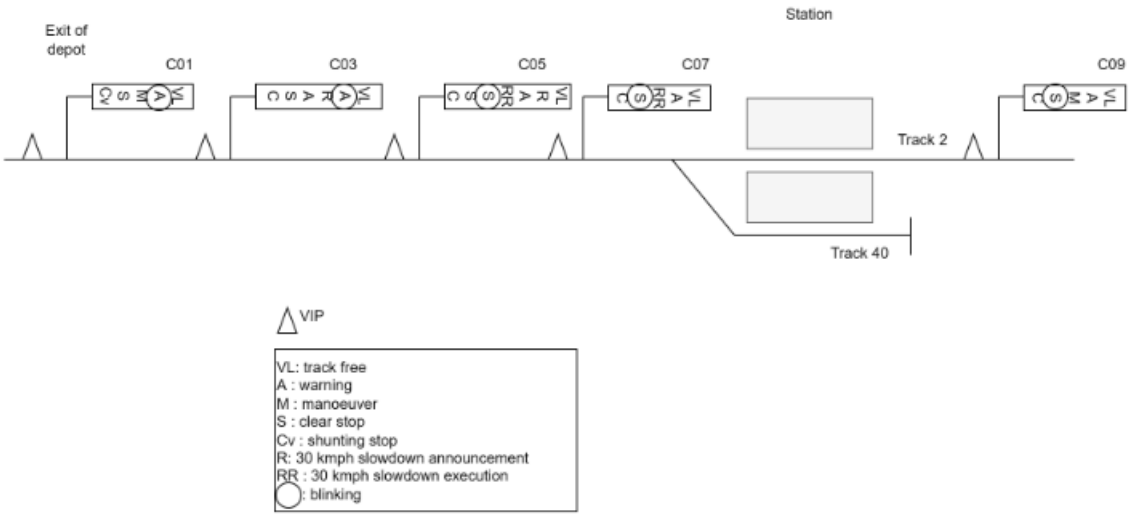


Figure 118: Signalling configuration

5.7.2.2 Scenario

5.7.2.2.1 Nominal operation

ID: 01 Track free
Description: Depot to station track free, direct track reception

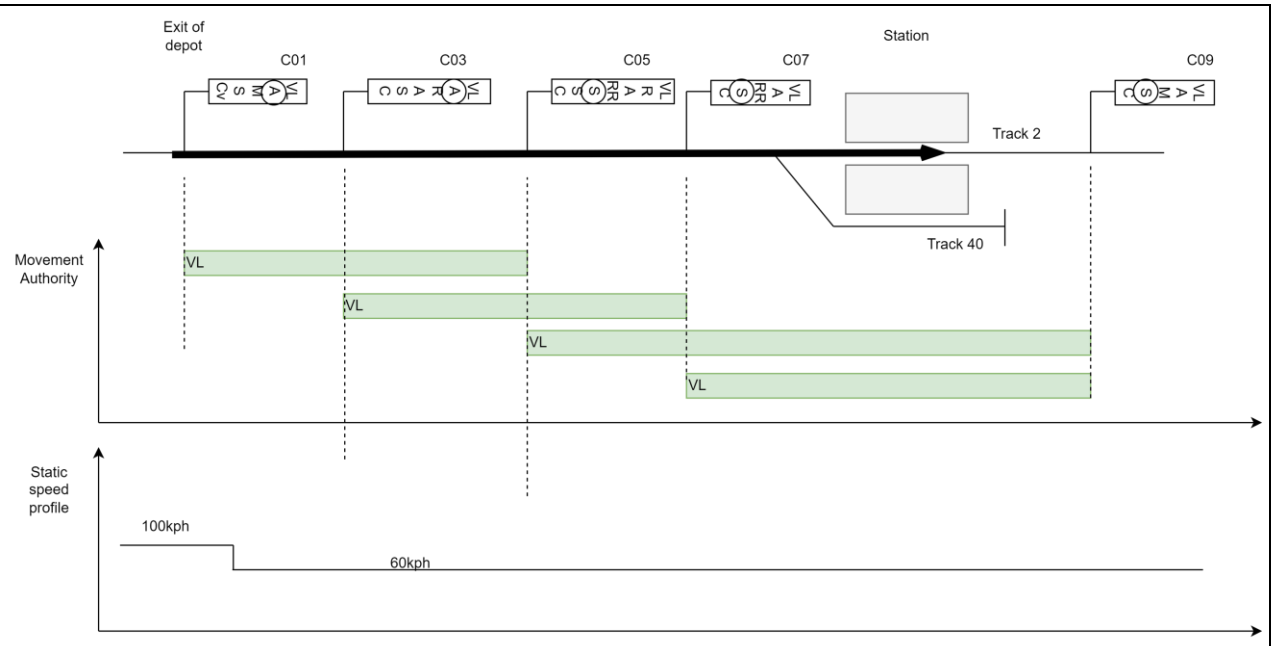


Figure 119: Track free scenario

Expected result:

- Set in motion
- Maximum static speed respect
- Detection of track free and no brake
- Stop at platform mark at station

ID: 02 stop
Description: Depot to station warning and non-crossable stop, signal opening, direct track reception

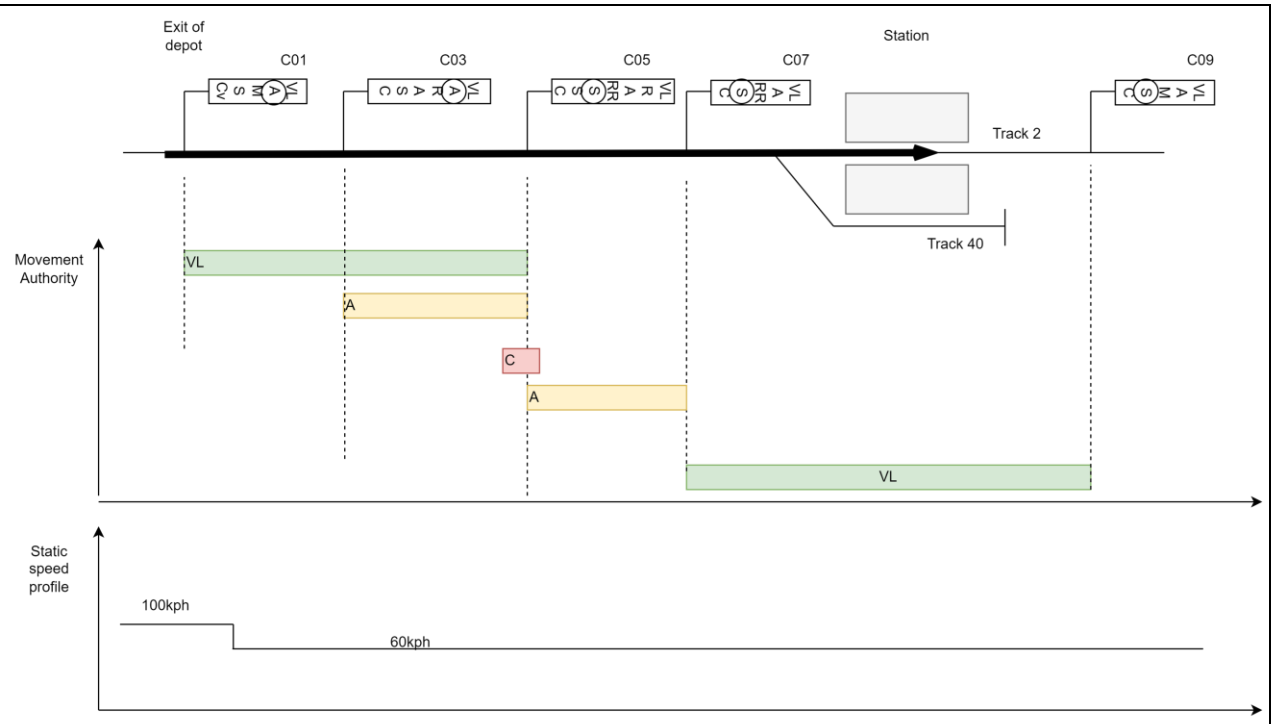


Figure 120: Stop scenario

Expected result:

- Set in motion
- Warning detection (C03) and speed decrease
- Non crossable stop detection and stop before signal
- Detect change: stop to warning
- Detect track free (C07)
- Stop at platform mark at station

ID: 03 switch crossing
Description: Depot to station 30kph announcement, 30kph execution, reception on deviated track

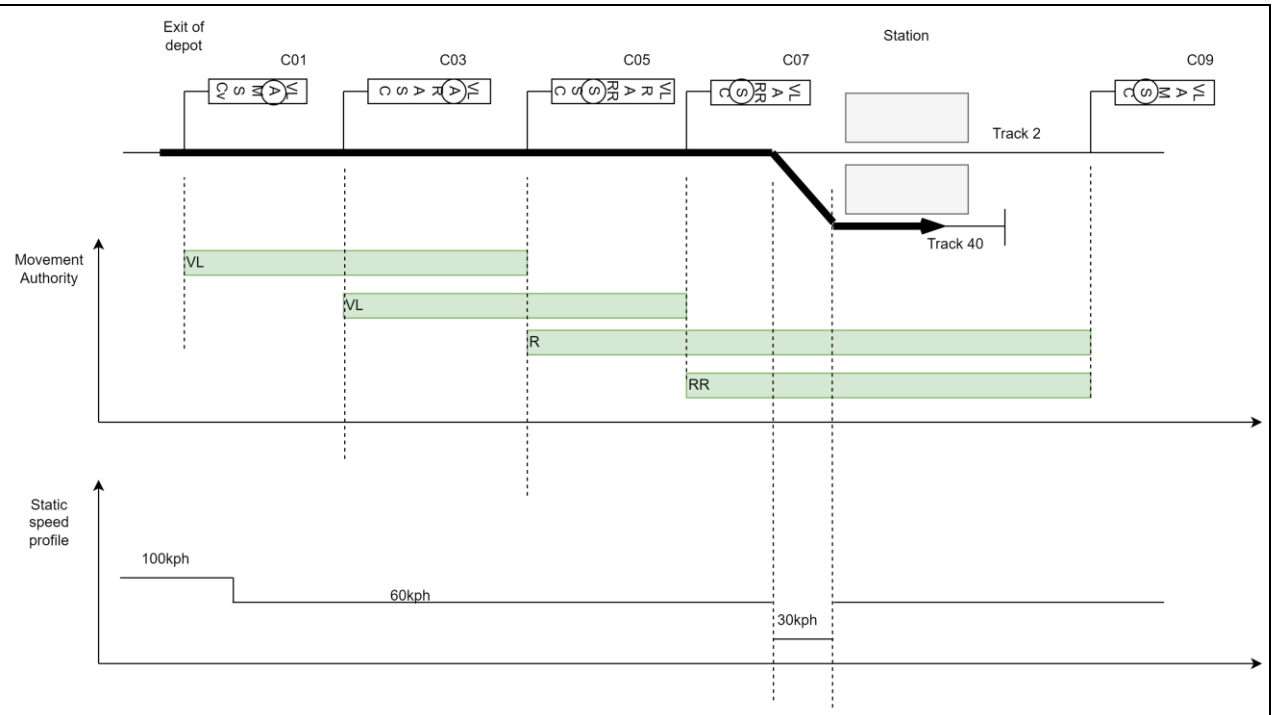


Figure 121: Switch crossing scenario

Expected result:

- Set in motion
- Detect track free
- Detect signal announcement for speed reduction to 30kph: reduce speed
- Detect signal speed execution: maintain speed at 30kph on switch
- Stop at platform mark at station

5.7.2.2.2 Degraded operation

Nominal scenarios are replayed with the implementation of failure modes, including:

- On-board failure (e.g., fire detection)
- Signalling failure (e.g., signal remains in stop aspect)
- ATO system failure (e.g., missed detection of a lineside signal)

5.7.2.3 Combination parameters

These scenarios are repeated with combination of specific parameters:

- **Perturbation**

Perturbation type		**Parameter**
Environmental condition	Sun	Sunrise Noon Dusk
Environmental condition	Luminosity	Cloud presence: light, medium, heavy
Environmental condition	Weather	Fog, rain (light, heavy), snow
Environmental light	Front light of crossing train Rear light of crossing train Platform screen Sun in front	

Table 23: Perturbation parameter

- **Camera position** on the train
- **Camera characteristic:** Focal, optical centre, image size, radial distortion, tangential distortion, axis skew

6 PERSPECTIVES

The objective of the work carried out was to focus on the modelling approach, highlighting the connection between MBSE and MBD, and how MBD can be applied transversally across the development lifecycle. The intention was not to delve into detailed implementations or explore alternative approaches. As such, this work provides a foundation from which other modelling or design domains for ATO systems can be further explored.

The description of the modelling activities and their results will be detailed in Deliverable D31.5.

6.1 MODELLING TECHNIQUES

Part of the model of the use case implemented within the WP31 framework is deployed on a real-time target. While the current real-time target enables rapid deployment of MATLAB Simulink models, the use of an industrial real-time target could be further investigated. This would not only demonstrate the end-to-end digital continuity of the model but also highlight operational challenges (like communication performance on representative hardware).

Design methods leveraging artificial intelligence—for automatic architecture generation, software code generation, or automated verification—could represent relevant research directions to improve development efficiency. Using AI in the design process of the model does not necessarily introduce AI based components in the model.

6.2 USE CASE

The use cases have been limited to specific applications of GoA2 and GoA4. This work could be extended by considering additional use cases, both in terms of the operational scenarios addressed and the ATO architecture considered.

6.2.1 Scope

Based on the current modelling, the scope can be extended to other use cases such as DAS, GoA2, and GoA4.

6.2.1.1 Driver Advisory System

The operational speed envelope planning component is common across all grades of automation and serves as a foundation for further evaluations.
Regarding the DAS: the standardization of the planning component aims to ensure modularity, harmonize user ergonomics, and enable integration into a multi-display system concept. The evaluation of the DAS follows an approach positioning it as a facilitator for ATO deployment, through assessing potential benefits and defining an architecture designed for future integration.

Ergonomic aspects are also a topic of consideration, as each European DAS currently features its own user interface design, which often deviates from the layout proposed by the ERA_DMI 15560 specification.

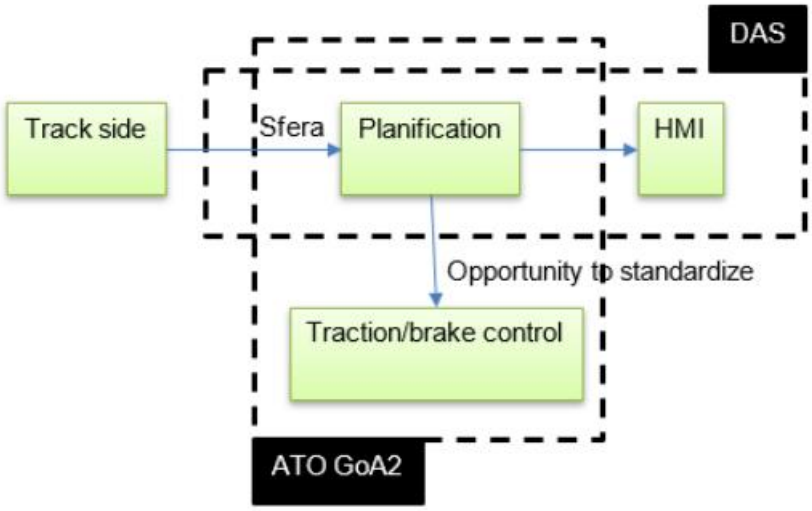


Figure 122: DAS interface standardization opportunity

6.2.1.2 GoA2

The use case addressed in WP31 focused on energy efficiency (see KPI in 704.3.4). However, ATO can offer additional benefits. Headway stability—ensuring a robust time interval between two trains on the same line—could be a relevant use case, demonstrating the ability to maintain robustness at higher traffic throughput levels than those achievable with manual driving.

6.2.1.3 GoA4

The designed model serves as a foundation for extension to other similar technical movement operations, such as automatic movements within depots, coupling at station or turnaround operations at terminal stations. Additional capabilities could be integrated, such as obstacle detection, to complement signal recognition functionalities.

6.2.1.4 Remote control

The use case to be modelled includes a significant number of elements already present in the modelling of the remote-control use case, such as the driving simulator and the ground-to-train control interface.

6.2.2 Architecture

Scientific literature suggests considering a train-level architecture to achieve greater energy efficiency beyond the ATO system itself. In particular, it is proposed to explore the integration of ATO with onboard energy storage systems, such as batteries, to enable energy recovery during braking.

The level of modelling should be tailored to the targeted simulation outcomes. A more detailed modelling approach allows for an expanded scope of simulation objectives.

6.3.1 Train model

Train dynamics have been modelled using the fundamental principle of dynamics equation, which is the standard approach for energy assessment studies. However, a more refined model could provide additional insights by accounting for internal dynamic phenomena within the train (particularly relevant for variable-consist trains), reaction times (e.g., pneumatic braking), and external factors (e.g., wheel slip or slide). Such modelling enhancements could be achieved through Multiphysics modelling and/or co-simulation techniques.

6.3.2 Optimal speed profile

In line with this perspective of improving train modelling, energy recovery could be considered in the definition of the reference speed profile. Additional criteria could also be incorporated into the path cost used in the graph traversal. For example, passenger comfort could be taken into account by penalizing frequent changes in driving modes along the path.

Performance analysis of the TTSM algorithm reveals a significant increase in computation time as the graph size grows. To mitigate this effect—particularly with a view toward on-board implementation—an iterative approach could be explored: the problem is first solved using a coarse-grained graph, and then a refined solution is searched for within a finer-meshed graph constructed around the initial solution. This would help reduce the overall graph size and consequently the computation time.

The state-of-the-art review highlighted a wide variety of methods used to address this type of problem. In particular, recent years have seen significant development of optimization approaches based on metaheuristics (such as Particle Swarm Optimization) and artificial intelligence algorithms (such as genetic algorithms). It could be valuable to compare the performance of these approaches with that of the implemented method.

6.3.3 Computer vision enhancement

A key improvement to consider is enhancing the tracking module responsible for computing M_STABLE (stability measurement). Currently, it relies on a basic probabilistic algorithm, which could be significantly improved for better accuracy and reliability.

In addition, the chassis detection model struggles under rainy conditions and would benefit from further refinement to improve robustness. The model was developed during a previous project using an older version of YOLO (version 5). It would be relevant to retrain the model using the latest version, YOLOv11, to benefit from improved performance and features.

Another important area is optimizing the performance of the signal detection (SD) model to reduce processing delays during simulation.

Currently, the comparison between expected light positions and detected lights is done using a resized version of the image. It may be more effective to adapt the size of the filter's regions of interest based on the actual image dimensions instead.

By enabling rapid implementation through Model-Based Design, even more innovative architectures could be tested, including:

- Exploring recent deep learning advances, such as Transformer-based models.
- Data enrichment and adaptation, leveraging synthetic images (e.g., using GANs) and self-supervised learning techniques to make use of large volumes of unannotated video streams.
- Improving real-time processing pipelines by integrating network pruning techniques or deploying lightweight model variants. Neural network outputs could be combined with low-level image processing algorithms to rapidly verify signal states (e.g., blinking detection).
- Progressive monitoring and continuous learning, by implementing online learning mechanisms that log ambiguous cases, later corrected and used to retrain and refine the model over time.

7 CONCLUSION

The activities carried out as part of Task 31.1 have led to the deployment of an MBSE (Model-Based Systems Engineering) methodology adapted to the design of Control-Command and Signalling (CCS) systems. This work proposes a practical, transposable modelling framework illustrated through specific GoA4 and GoA2 ATO use cases (see section 4), contributing to enhanced design efficiency, reduced time-to-market, and improved system integration quality in the railway sector.

Building on established reference frameworks, this work extends them by proposing an integrated modelling approach that details the activities associated with each modelling phase, as well as the levers to activate the full benefits of the methodology.

For modelling purpose dedicated GoA4 ATO architecture has been proposed to support technical movements between depots and stations (sections 4.5 and 1.1). This architecture synthesizes various reference models and incorporates specific adaptations that leverage simplification opportunities offered by this particular use case—characterized by a constrained operational domain, a reduced functional scope, and limited associated risks.

Since the architecture was defined for illustrative purposes, it is not intended to serve as a reference. In this case, the reference architectures are those published as deliverables under WP6.

In parallel, an algorithmic definition for GoA2 has been developed (presented in the TTSM and ATSM calculation: 4.7.3.2.1), aiming to meet performance objectives such as schedule adherence and precision stopping, while minimizing energy consumption. This work also served as a basis to explore the implications of implementing such a system under the constraints of the Subset 125 functional specification and the Subset 126,130 and 139 interface definitions, highlighting the trade-offs and performance impacts of the design choices left to suppliers in algorithm implementation. This systematic exploration enhances industry understanding of optimization potentials and facilitates more informed design choices.

A general modelling framework is proposed, which can be transposed to other CCS system modelling use cases. The definition of the system model and the associated simulation plan provide a reference baseline for model implementation and the execution of system tests, the results of which will be presented in deliverable D31.5. This transposable framework broadens the applicability of MBSE approaches across the railway sector, potentially driving wider adoption of digital engineering practices.

A driving simulator architecture has also been proposed, based on the developed models, to support system design activities (section 5.5). This simulator serves as a versatile tool for validation, rapid prototyping, and human factors assessment, offering a real-time, immersive environment to evaluate ATO behaviour under realistic conditions.

Several perspectives (section 6) for future enhancements are identified to further strengthen the modelling approach not only for the ATO system design, but also more widely for the CCS scope (cf. §6 - Perspectives):

- Investigate the integration of Artificial Intelligence techniques in model development and decision-making processes.

- Enrich the 3D modelling layer with railway-specific components and tools for automated scenario generation.
- Human factors assessment for GoA2, based on a driving simulator, to evaluate the transition phase during GoA2 disengagement, limited supervision mode, or operation over lineside signalling.
  The driving simulator can also be assessed as a tool for driver training, supporting the evolution of ATO driving styles to improve energy consumption efficiency.
- Explore how modelling activities can be integrated into the safety approval and homologation process, with the goal of reducing on-track testing requirements.

8 REFERENCES

8.1 MODELLING TECHNIQUES

ERJU-722 - CCS TSI mandatory specifications
https://www.era.europa.eu/era-folder/1-ccs-tsi-appendix-mandatory-specifications-etcs-b4-r1-rmr-gsm-r-b1-mr1-frmcs-b0-ato-b1

8.1.1 System engineering

ERJU-687 - INCOSE, system engineering handbook SEBOK
https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)

ERJU-688 - NASA, 2007, SYSTEM ENGINEERING HANDBOOK
https://www.nasa.gov/wp-content/uploads/2018/09/nasa_systems_engineering_handbook_0.pdf

8.1.2 MBSE

ERJU-705 - Donovan Roodt, Malaeka Nadeem, Lam-Thien Vu MANAGING COMPLEXITY ON DIGITAL SYSTEMS; A MODEL-BASED SYSTEMS ENGINEERING APPROACH

ERJU-706 - M. Blumenfeld, E.J.C. Stewart, E. Larios, 2016, Using model-based systems engineering to increase the robustness of monitoring systems in railways

ERJU-724 - Jean Luc VOIRIN, Conception architecturale des systèmes basée sur les modèles avec la méthode Arcadia, 2018, ISTE Editions Limited

ERJU-725 - Pascal ROQUES, Modélisation architecturale des systèmes avec la méthode Arcadia, 2018, ISTE Editions Limited

8.1.3 MBD

ERJU-697 - A Amendola, A Becchi, R Cavada, A Cimatti, A Griggio, A model-based approach to the design, verification and deployment of railway interlocking system, Applications of Formal Methods, Verification and deployment of railway interlocking system, 2020
https://es-static.fbk.eu/people/griggio/papers/isola20.pdf#:~:text=Italian%20Railway%20Signalling%20network,is%20intended%20to%20support%20a

ERJU-698 - Baumgart Andreas, Reinkemeier Philipp,Rettberg Achim, Stierand Ingo, Thaden Eike, Weber Raphael,2010/10/13, A Model-Based Design Methodology with Contracts to Enhance the Development Process of Safety-Critical Systems, Software Technologies for Embedded and Ubiquitous Systems https://www.researchgate.net/publication/215863007_A_Model-Based_Design_Methodology_with_Contracts_to_Enhance_the_Development_Process_of_Safety-Critical_Systems#:~:text=In%20this%20paper%20a%20new,example%20we%20conclude%20that%20our

ERJU-700 - Mathworks customer feedback, Bombardier Transportation Implements Model-Based Design to Accelerate Rail Propulsion System Development https://fr.mathworks.com/company/user_stories/bombardier-transportation-implements-model-based-design-to-accelerate-rail-propulsion-system-development.html

ERJU-701 - Thomas Bochot, Pierre Virelizier, Hélène Waeselynck and Virginie Wiels, 2009, Model Checking Flight Control Systems: the Airbus Experience https://www.researchgate.net/publication/224503769_Model_Checking_Flight_Control_Systems_the_Airbus_Experience#:~:text=This%20paper%20presents%20experiments%20realized,formal%20verification%20at%20design%20level

ERJU-702 - Emeka Eyisi,Zhenkai Zhang,Xenofon Koutsoukos, Janos Sztipanovits, 2013, Model-Based Control Design and Integration of Cyberphysical Systems: An Adaptive Cruise Control Case Study

ERJU-703 - Sukru Yaren Gelbal, Bilin Aksun-Guvenc, Levent Guvenc, SmartShuttle: Model Based Design and Evaluation of Automated On-Demand Shuttles for Solving the First-Mile and Last-Mile Problem in a Smart City https://arxiv.org/pdf/2012.12431

ERJU-704 - Mathworks customer feedback, Airbus Develops Fuel Management System for the A380 Using Model-Based Design https://fr.mathworks.com/company/user_stories/airbus-develops-fuel-management-system-for-the-a380-using-model-based-design.html

ERJU-723 - Stephane CALLET, Thierry NAVARRO, Damien LEDOUX, SaïdEL FASSI, Hervé FEDELER, Utilisation d'une approche Model Based Design sur un système sol ERTMS/ETCS Niveau 2, Revue Generale des Chemins de Fer N°250 Juin 2015

ERJU-726 – FMI standard https://fmi-standard.org/docs/3.0.2/

8.2 ARCHITECTURE

8.2.1 Concept

ERJU-679 - JR East, 10/09/2024, JR East's plans for driverless Shinkansen operation, Press release
https://www.jreast.co.jp/e/press/2024/pdf/20240910.pdf

ERJU-681 - Metro report international, 11/10/2021, Driverless train demonstrated on the Hamburg S-Bahn
https://www.railwaygazette.com/technology-data-and-business/driverless-train-demonstrated-on-the-hamburg-s-bahn/60077.article

ERJU-686 - DGITM, note de synthèse sur le plan de deploiement du système européen de gestion du trafic
https://www.ecologie.gouv.fr/sites/default/files/documents/202411%20Consultation%20DGITM%20-%20ERTMS/ETCS.pdf

ERJU-684 - German national ETCS implementation plan
https://transport.ec.europa.eu/document/download/3e233929-da50-4ce2-9276-aff640e939d4_en?filename=NIP_ERTMS/ETCS_2024_DE.pdf

ERJU-720 - French national ETCS implementation plan
https://transport.ec.europa.eu/document/download/0494f7a3-30a1-446b-aec4-ccc1d67d4714_en?filename=NIP_ERTMS/ETCS_2024_FR.pdf

ERJU-719 - Timetable reliability for french regional train
https://ressources.data.sncf.com/explore/dataset/regularite-mensuelle-ter/table/?sort=date

ERJU-727 - Arrêté du 4 janvier 2016 - Arrêté relatif à la nomenclature de classification des événements de sécurité ferroviaire
https://www.securite-ferroviaire.fr/reglementation/arrete-relatif-la-nomenclature-de-classification-des-evenements-de-securite-ferroviaire

ERJU-728 - ENISA Threat landscape
https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024

8.2.2 GoA2 algorithm

ERJU-693 - J. T. Haahr, D. Pisinger et M. Sabbaghian. “A dynamic programming approach for optimizing
train speed profiles with speed restrictions and passage points”. In : Transportation Research Part B :
Methodological 99 (2017), p. 167-182. doi : https://doi.org/10.1016/j.trb.2016.12.016.

ERJU-692 - S. Irnich et G. Desaulniers. “Shortest Path Problems with Resource Constraints”. In : Boston, MA :
Springer US, 2006, p. 33-65. doi : http://dx.doi.org/10.1007/0-387-25486-2_2.

ERJU-691 - L.S. Pontryagin et al. The Mathematical Theory of Optimal Processes. Wiley Interscience, 1962.

ERJU-690 - J. Richalet et al. “Model predictive heuristic control : Applications to industrial processes”. In :
Automatica 14.5 (1978), p. 413-428. doi : https://doi.org/10.1016/0005-1098(78)90001-8.

ERJU-689 - S. Su, Z. Tian et R.M.P. Goverde. Energy-Efficient Train Operation. Springer Cham, 2023.

ERJU-774 - Braulio Aguiar, Denis Berdjag, Bernard Demaya, Thierry-Marie Guerra, A robust and fault tolerant approach for automatic train stop control system design, IFAC-PapersOnLine, Volume 50, Issue 1, 2017

ERJU-775 - Valerio De Martinis, Mariano Gallo, Models and Methods to Optimise Train Speed Profiles with and without Energy Recovery Systems: A Suburban Test Case, Procedia - Social and Behavioural Sciences, Volume 87, 2013

ERJU-776 - Miyatake, Masafumi & Member, Hideyoshi. (2010). Optimization of Train Speed Profile for Minimum Energy Consumption. IEEJ Transactions on Electrical and Electronic Engineering. 5. 263 - 269. 10.1002/tee.20528.

ERJU-777 - Domínguez, María & Cucala, Asunción & Fernández, Antonio & Pecharromán, Ramón & Blanquer, J. (2011). Energy efficiency on train control: Design of metro ATO driving and impact of energy accumulation devices

ERJU-778 - Zhu, Xiaomin, Qian Pu, Qian Zhang and Runtong Zhang. “Automatic Train Operation Speed Profile Optimization and Tracking with Multi-Objective in Urban Railway.” Periodica Polytechnica Transportation Engineering (2019): n. pag.

ERJU-779 - Achilleos, Achilleas & Anastasopoulos, Markos & Tzanakaki, Anna & Iordache, Marius & Langlois, Olivier & Pheulpin, Jean-Francois & Simeonidou, Dimitra. (2019). Optimal Driving Profiles in Railway Systems based on Data Envelopment Analysis. 254-259. 10.5220/0007878000002179.

8.2.3 GoA4 architecture

ERJU-694 - ERJU WP5, D5.1 - WP5 GoA3/4 Specification v1.0.0 https://projects.shift2rail.org/download.aspx?id=351d8551-cf6a-49ca-b213-8ce4e37892f8

ERJU-699 - TAURO Projet, D4.2 - Updated GoA3/4 Specification https://projects.shift2rail.org/download.aspx?id=6ad5ec60-e828-4e68-9cc7-beb362acde09

8.3 MODEL

ERJU-763 - Yang Sun, Guy André Boy, Marc Sango, Anne Barros, Function analysis for human-machine teaming for semi-automated trains, Human Systems Integration conference, Jeju, Korea 2024/08/27-29

9 APPENDIX

9.1 COMPUTER VISION PATTERN

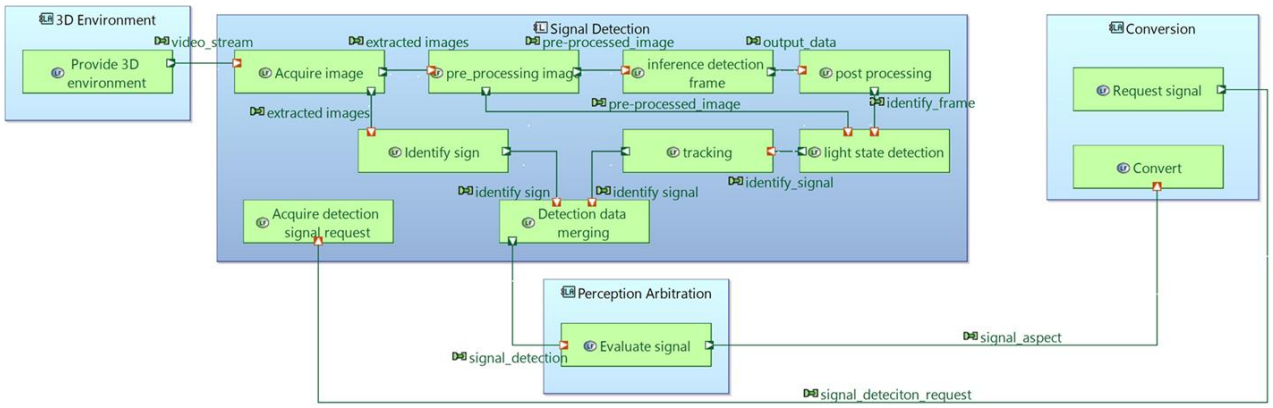


Figure 123: Computer vision architectural pattern

9.2 AUTOMATION SYSTEM PATTERN

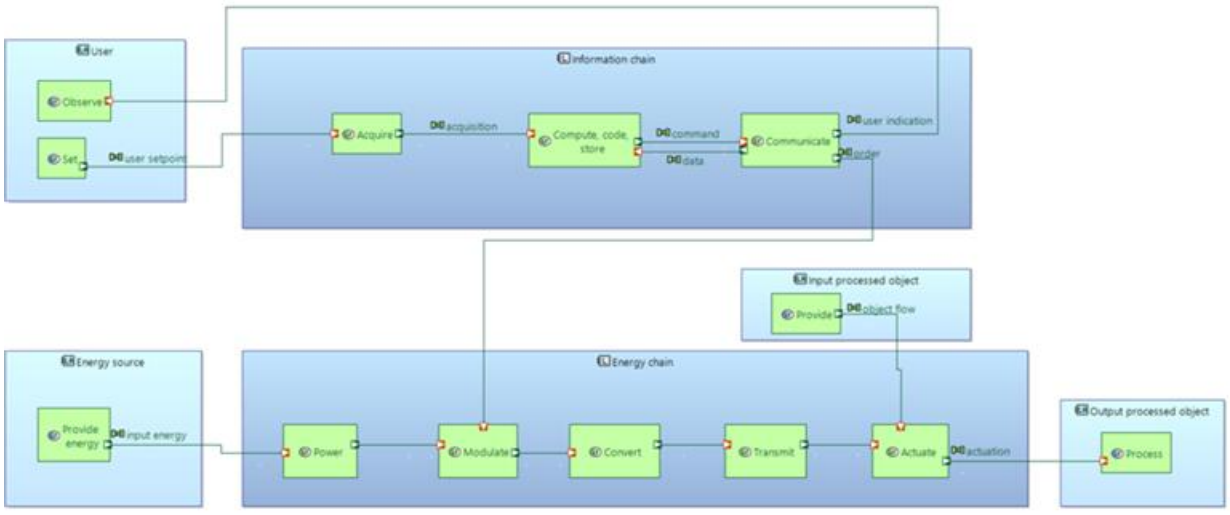


Figure 124: Automation system architectural pattern

9.3 GoA2 Optimization algorithm scientific literature

Category	Method	Publication
Data based	Data Envelopment Analysis (DEA)	ERJU-779
Deterministic	unknown input observers (UIO): to compensate the “train system” with the estimation of dynamic friction. Fault Detection, isolation and estimation (FDI): detection of jamming and brake disable. Proportional integral (PI): comparison of train system and reference	ERJU-774
Deterministic	starting and ending point of the coasting phase control of travel time and distance to be covered “as soon as possible” coasting strategy (on the contrary to driver usual coasting localisation based on line knowledge)	ERJU-775
Deterministic	dynamic programming (DP) and bilinear interpolation: discretization of space domain [distance, speed], find the optimal control input at each lattice point. gradient method: iteratively adjusts model parameters in the direction of the steepest increase in the objective function to find the minimum or maximum of the function. sequential quadratic programming (SQP): solve general non-linear problem; iteratively solves quadratic subproblems, approximating the constrained optimization problem, to converge towards a local minimum or maximum	ERJU-776
Deterministic	parameter: deceleration rate, speed holding, coasting speed, target speed decision theory: domination, sensitivity, uniform distribution bellman’s dynamic programming	ERJU-777
Genetic	genetic algorithm to find the reference velocity, fuzzy logic, predictive control	ERJU-778

Table 24: Type of algorithm in publications

Article	Objectives Energy consumption	Punctuality	Stopping accuracy	Passenger comfort
ERJU-774			X	
ERJU-775	X			
ERJU-776	X			
ERJU-777	X			
ERJU-778	X	X	X	X
ERJU-779	X			

Table 25: Objectives identified in publications