Rail to Digital automated up to autonomous train operation

D39.5 – Integration Laboratories Architecture

Due date of deliverable: 31/05/2026

Actual submission date: 15/09/2026

Leader/Responsible of this Deliverable: Michele Bruzzo/Loredana Freda Albanese (HITACHI RAIL STS)

Reviewed: **Y**/N

Document status		
Revision	Date	Description
01	17/04/2026	First issue.
02	06/05/2026	Second issue, release for internal review.
03	13/05/2026	Document release after WP internal peer review.
04	15/09/2026	Figures resolution improved, added reference to D39.7.

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

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

ACKNOWLEDGEMENTS

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

REPORT CONTRIBUTORS

Name	Company	Details of Contribution
Michele Bruzzo	HITACHI	Main Contributor/reviewer all chapters
Loredana Freda Albanese	HITACHI	Main Contributor/reviewer all chapters
Maria Luisa Aiachini	HITACHI	Main Contributor/reviewer all chapters
Nazzareno Filippini	RFI	Contributor/reviewer all chapters
Diana Serra	RFI	Contributor/reviewer all chapters
Lorenzo Abbati	TRENITALIA	Contributor /reviewer all chapters
Leonardo Malesci	TRENITALIA	Contributor /reviewer all chapters
Joelle Aoun	PRORAIL	Contributor /reviewer all chapters
Jonny Gustafsson	TRAFIKVERKET	Contributor /reviewer all chapters

Disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. The project FP2-R2DATO is supported by the Europe's Rail Joint Undertaking and its members.

1 EXECUTIVE SUMMARY

Deliverable D39.5 describes the integration laboratories architecture developed within the R2DATO project to support the validation of automated train operation subsystems. The work addresses the need for a controlled and repeatable environment to verify the interaction of complex onboard systems prior to field deployment.

The integration laboratory combines real target components with simulated external subsystems, enabling the joint testing of the Automatic Driving Module, Repository, and Automatic Processing Module.

Multiple integration environments are defined to address progressively more realistic configurations, including standalone onboard integration, interaction with real ATP, with Remote Driving functions, and with real TCMS.

A simulation-based approach is adopted through the SimATO platform, which emulates train behaviour and external interfaces, allowing fully automated execution of predefined test scenarios.

The use of automated simulation, standardised test formats, and log-based analysis ensures reproducibility, traceability, and consistency of test results.

Integration testing follows a structured methodology based on test scenarios derived from project use cases, covering both nominal and degraded operating conditions. System behaviour is assessed against expected outcomes, while detailed log data are generated to support post-execution analysis using the LogExplorer tool.

It is important to highlight that the purpose of the integration laboratory testing activities is not to perform a full validation of system performance (e.g. timing, capacity, or operational limits).

Instead, the focus is on verifying the correct functional behaviour of the subsystems and the consistency of their interactions through defined interfaces, ensuring proper data exchange and system-level integration.

The results demonstrate that the proposed architectures effectively support subsystem integration and improve test repeatability and traceability.

Interne

ABBREVIATIONS AND ACRONYMS

ATO	Automatic Train Operation
ADM	Automatic Driving Module
APM	Automatic Processing Module
ATP	Automatic Train Protection
CSV	Comma-separated Values
DMI	Driver Machine Interface
ELO	ERTMS Logic OnBoard
ENV	Environment
GoA	Grade of Automation
IM	Infrastructure Manager
ISM	Incident and Safety Manager
LOC	Localization
PER	Perception
RC-OB	Remote Control - On Board
REP	Repository
RDS	Remote Driving System
RU	Railway Undertaking
R2DATO	Rail to Digital automated up to autonomous train operation
SS	Subset
TCMS	Train Control and Management System
TP	Train Protection
WI	Work Item
XML	Extensible Markup Language

Interne

TABLE OF CONTENTS

Acknowledgements 2
Report Contributors 2
1 Executive Summary 4
Abbreviations and Acronyms 5
Table of Contents 6
List of Figures 7
2 Introduction 8
3 Integration Laboratories Architectures 9
3.1 Interfaces Description 10
3.2 Signalling Integration Laboratory 12
3.2.1 ENV1 12
3.2.2 ENV2 14
3.2.3 ENV3 15
3.3 TCMS Integration Laboratory (Alpha Train) 16
4 Tools Descriptions 17
4.1 SimATO 17
4.2 LogExplorer 18
5 Test Script Format 18
5.1 Overview of the Testing Methodology 18
5.2 Structure of the Test Suite 18
5.3 Creation of Test Scenarios 19
5.4 Test Execution Procedure 19
5.4.1 ENV1 19
5.4.2 ENV2 19
5.4.3 ENV3 20
5.4.4 AlphaTrain 20
5.5 Result Analysis and Reporting 20
6 Conclusions 21
References 22

Interne

LIST OF FIGURES

Figure 1: Interfaces 10
Figure 2: ENV1 Architecture 12
Figure 3: ENV2 Architecture 14
Figure 4: ENV3 Architecture 15
Figure 5: Alpha Train Architecture 16

2 INTRODUCTION

This deliverable, D39.5 – Integration Laboratories Architecture, is produced within the framework of the FP2 R2DATO project, funded under the Europe's Rail Joint Undertaking and the Horizon Europe research and innovation programme. The document contributes to the project activities related to the design, implementation, and validation of the integration laboratory environments supporting the development and testing of automated and autonomous train operation functions at higher Grades of Automation.

The primary objective of this deliverable is to provide a comprehensive and structured description of the Integration Laboratories established for Task T39.2. The document details the laboratory architectures adopted within the R2DATO project, illustrating how real target components and simulated subsystems are combined to support robust and repeatable integration testing activities.

The document first introduces the Integration Laboratories Architecture, describing the role and responsibilities of the onboard subsystems, and the interfaces (as defined in [1]) connecting them both to each other and to external train and trackside entities. The architectural description includes a detailed overview of the different integration environments defined within the project, each of them tailored to validate specific integration aspects.

These environments cover the integration of the onboard components in isolation, the interaction with real Automatic Train Protection (ATP) system, with the Remote Driving functions, and with real Train Control and Monitoring Systems (TCMS). This modular approach allows the verification of system behaviour across multiple configurations, ensuring flexibility, and completeness of the validation activities.

Furthermore, the deliverable provides an in-depth description of the testing framework adopted within the Integration Laboratories. This includes the presentation of the simulation and analysis tools used to execute the tests, as well as the structure and format of the test cases and scenarios. Emphasis is placed on key testing principles such as reproducibility, traceability, and automation, which are essential to ensure the reliability and consistency of integration test results.

3 INTEGRATION LABORATORIES ARCHITECTURES

This section outlines the architecture and interfaces of the Integration Laboratories, describing the organisation of the environments dedicated to TCMS and signalling integration, the emulated components used in place of real subsystems, and the communication framework that enables interaction between prototypes and tools.

For the project, three subsystems have been developed and validated during the activities concerning WP9 and WP10:

- ADM (Automatic Driving Module) is the subsystem responsible for the management of all the on-board automatic operation functions.
- REP (Repository) is the subsystem responsible for communicating with trackside to acquire the Journey Profile and infrastructure data (i.e. Digital Map data, e.g. SP-related micromapped nodes) from Infrastructure Manager (IM) or Mobility Provider (MP) and train data from Railway Undertaking (RU). In addition, it collects and reports incidents to trackside by exchanges with Incident and Safety Manager (ISM).
- APM (Automatic Processing Module) is the subsystem responsible for managing mission execution. It emulates driver actions in GoA3/4, reacts to incidents and provides reporting information offering a comprehensive overview of the entire mission.

The Signalling Integration Laboratory is designed and implemented in three different environments addressing three different integration aspects:

- ENV1: Integration between the three developed subsystems
- ENV2: Test environment considering the integration with a real ATP
- ENV3: Test environment considering the integration with a real ATP and Remote Driving

The TCMS Integration Laboratory is based on “Alpha Train”: the HITACHI test environment used to test real TCMS.

The next sections are dedicated to the description of the interfaces and the architecture for the four different integration test environments.

3.1 INTERFACES DESCRIPTION

Figure 1 highlights the interfaces between the three developed subsystems and between them and the external subsystems.

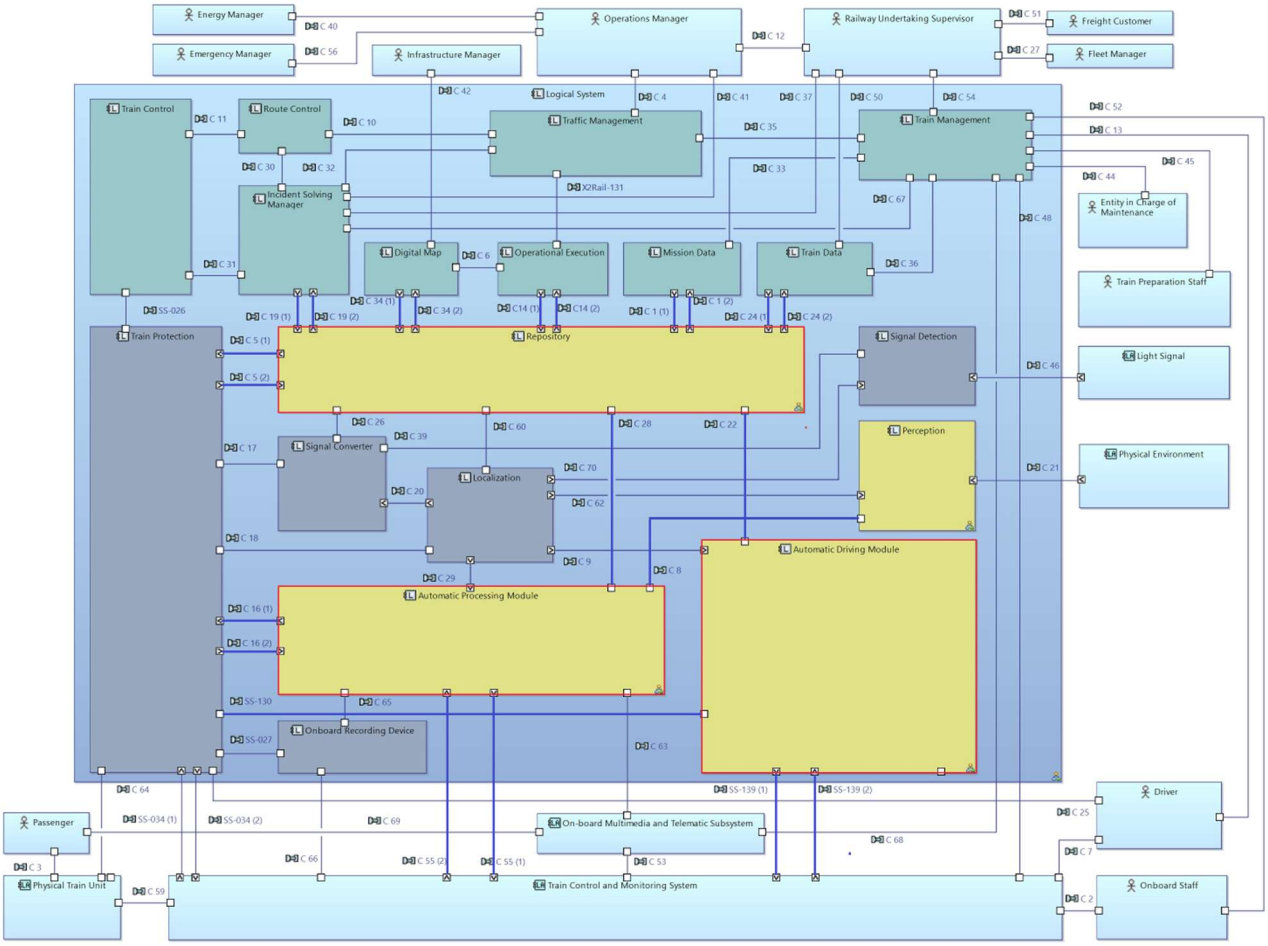


Figure 1: Interfaces

The communication interfaces between the subsystems under test, as well as between them and the external components, are composed of the following elements:

- ADM interfaces
  - C22 ADM-REP: data exchange between ADM and REP.
  - SS-130 ADM-TP: data exchange between ADM and TP.
  - SS-139 ADM-TCMS: data exchange between ADM and TCMS.
- REP interfaces:
  - C22 REP-ADM: data exchange between REP and ADM.
  - C28 REP-APM: data exchange between REP and APM.
  - C05 REP-TP: data exchange between REP and TP.
  - TRACK TRAIN DATA EXCHANGE (C01, C14, C19, C24, C34) data exchange between REP and Train&Traffic transactor.

- APM interfaces:
  - C16 APM-TP: data exchange between APM and TP.
  - C8 APM-PER: data exchange between APM and PER.
  - C28 APM-REP: data exchange between APM and REP.
  - C29 APM-LOC: data exchange generated by LOC.
  - C55 APM-TCMS: data exchange between APM and TCMS.

3.2 SIGNALLING INTEGRATION LABORATORY

The following integration environments can be configured to match the characteristics of the specific railway lines and the train subject to integration. In particular, it is possible to define a track layout including signalling information (e.g. balises, signals), topological features (e.g. gradients, curves), traction system characteristics and a three-dimensional representation of the track. These environments can also be tailored to represent infrastructure and operational conditions withacross different railway domains, including passenger, freight, and mixed-traffic operations, such as complex junctions, constrained nodes. Similarly, the train model can be defined on the basis of detailed train specifications, including traction and braking systems, train length, antenna positions, and aerodynamic characteristics. This level of configurability allows the environment to accurately represent different types of trains with various consist configurations, braking performance, and operational constraints, enabling the testing of system behaviour in a wide range of representative railway scenarios.

3.2.1 ENV1

This section describes the scope and architecture of the ENV1 integration test environment.

The ENV1 architecture, illustrated in Figure 2, has been specifically defined to validate the correct integration and interaction of the three main onboard components: ADM, APM and REP. The primary objective of ENV1 is to verify that these components, when deployed together on the real target platform, correctly exchange data, manage operational responsibilities, and respond consistently to external stimuli. To enable comprehensive integration testing without reliance on real external subsystems, ENV1 makes use of a dedicated simulator. The simulator, implemented through the SimATO test tool, emulates all external subsystems and train-level functions that would normally interact with ADM, APM and REP in an operational environment.

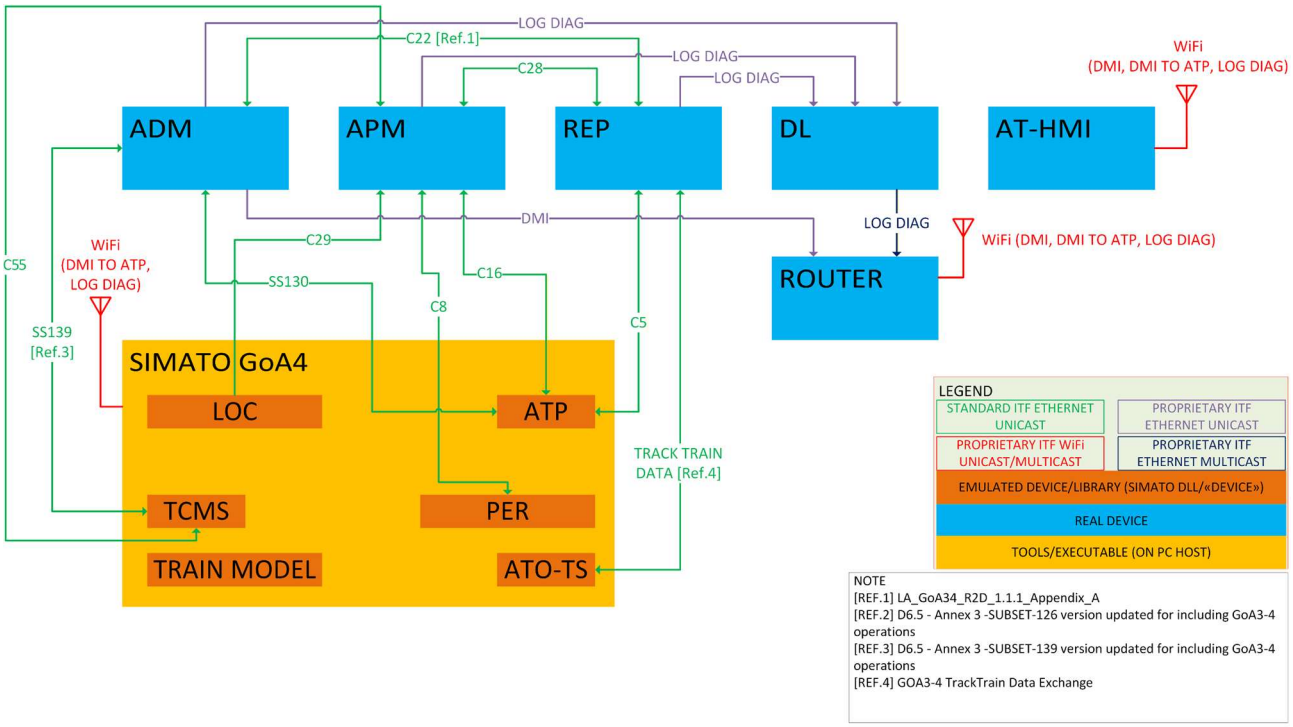


Figure 2: ENV1 Architecture

Interne

Communication between the components and the simulator is performed through the interfaces and protocols described in section 3.1. In this way, ENV1 enables end-to-end verification of interfaces, data consistency and functional integration across ADM, APM and REP, while also validating their interaction with simulated external entities.

3.2.2 ENV2

This section is dedicated to describing the scope and architecture of the ENV2 integration test environment.

ENV2 is specifically designed to validate the correct interaction between the onboard software components and the ATP subsystem under realistic operational conditions. The overall ENV2 architecture is illustrated in Figure 3 and represents a mixed configuration where real target components are integrated with dedicated simulation tools.

The primary objective of ENV2 is to verify the correct integration and data exchange between the three onboard components - ADM, APM and REP - and a real ATP subsystem. In this environment, ADM, APM and REP are executed on the actual target hardware.

Like ENV1, also in ENV2 SimATO has the objective to emulate all external subsystems (except from ATP) that would interact with ADM, APM and REP.

In addition, SimATO is interfaced with Thanos, which constitutes a test environment enabling the correct operation of the Hitachi real ATP system (ELO platform). Thanos manages the train interfaces and provides the air-gap data expected by the ATP, including balise position information, radio messages, odometric data, cab status, and related parameters.

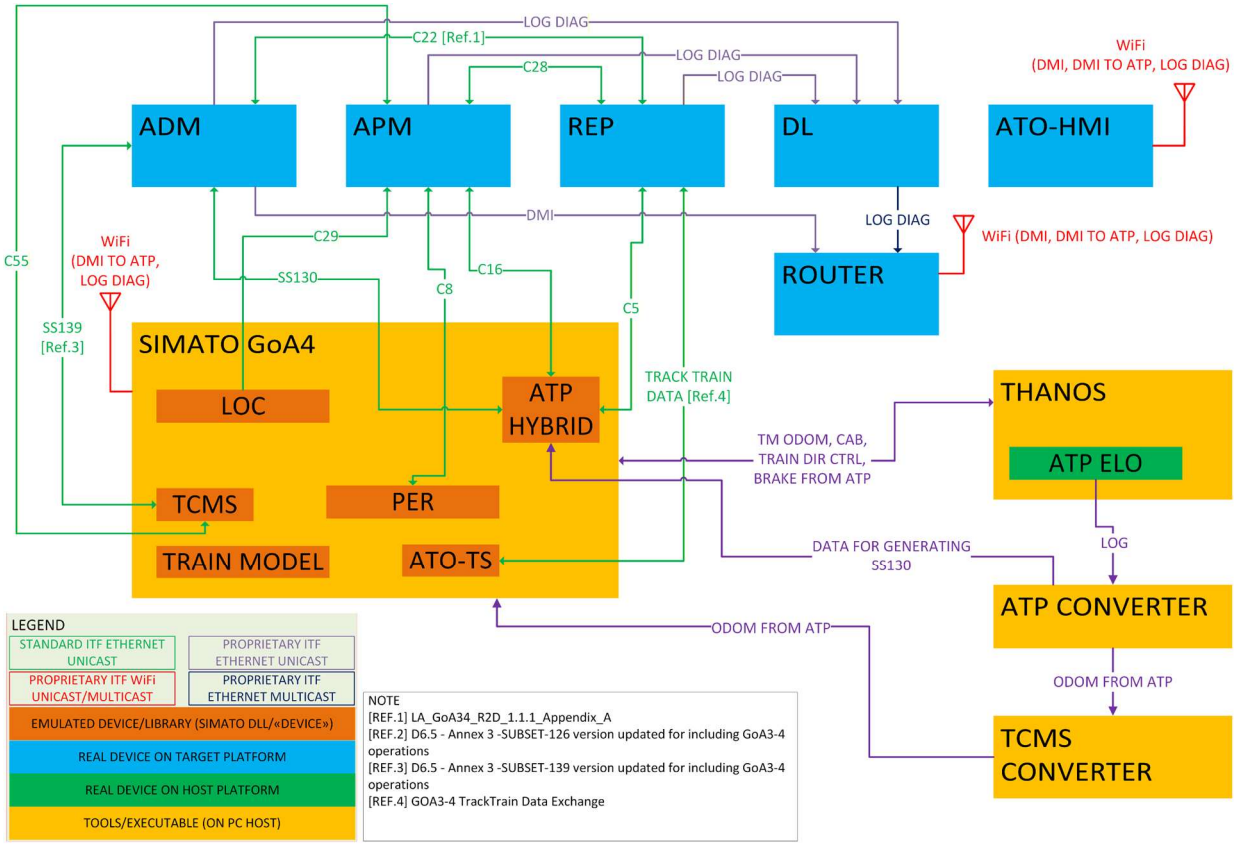


Figure 3: ENV2 Architecture

3.2.3 ENV3

This section describes the scope and architecture of the ENV3 integration test environment.

ENV3 integration environment is specifically designed to validate the correct integration between the Remote Driving System and the real ATP.

The overall ENV3 architecture is illustrated in Figure 4.

Within this configuration, the Remote Driving System is connected to the ATP domain through the Thanos platform, that is able to simulate ATP and the associated DMI. Communication between the Remote Driving System and Thanos is achieved by means of two dedicated modems.

The integration tests executed in ENV3 are designed to verify the correctness, consistency, and integrity of the information transmitted from the real ATP system and received by the Remote Driving System. Particular emphasis is placed on ensuring that all ATP data processed by the Remote Driving Desk and visualised on the Remote DMI accurately reflect the information generated by the real ATP. This includes validation of both the communication chain and the functional behaviour of the interfaces, thereby confirming that the Remote Driving System correctly interprets, displays, and handles ATP-related information under representative operational conditions.

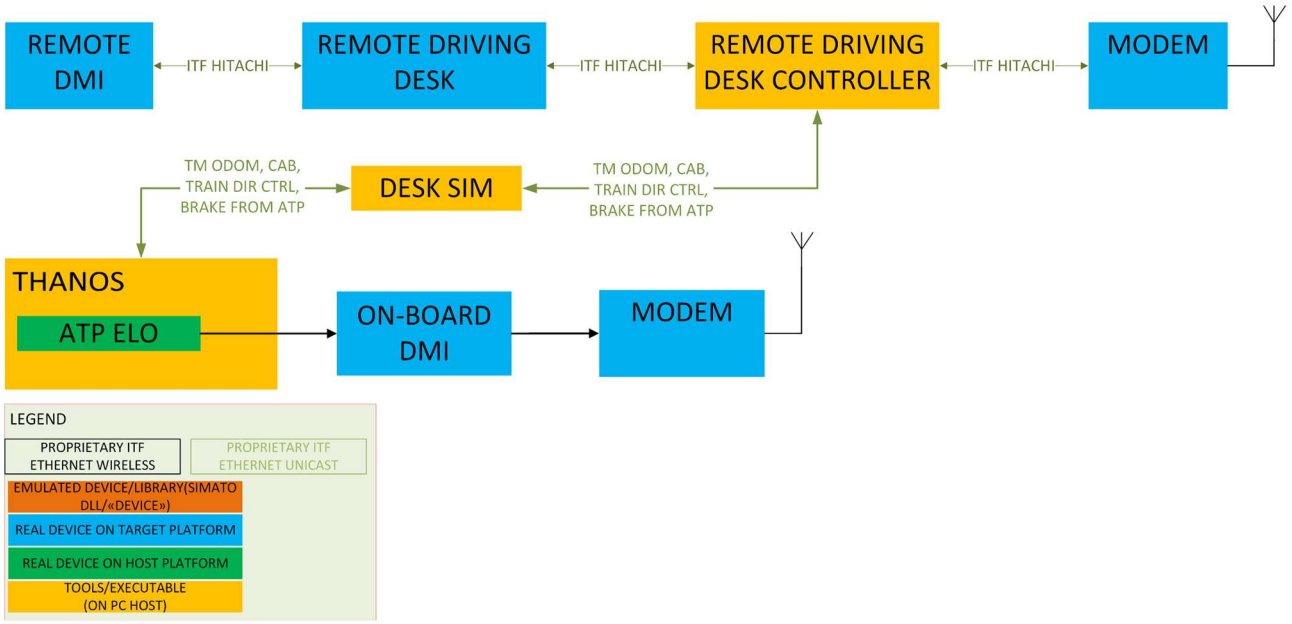


Figure 4: ENV3 Architecture

Interne

3.3 TCMS INTEGRATION LABORATORY (ALPHA TRAIN)

This section describes the scope and architecture of the AlphaTrain integration test environment.

The AlphaTrain environment has been specifically defined to support integration testing activities focused on validating the correct interaction between the Remote Driving System and the TCMS, together with the three developed functional components: ADM, APM, and REP. Although REP does not have a direct interface with the TCMS, it is nevertheless considered in the integration environment.

The overall AlphaTrain architecture is illustrated in Figure 5. Within this environment, the integration tests concentrate on verifying the correct functional and interface-level integration between the Remote Driving System, the simulated ATO components, and the real TCMS. The components ADM, APM, and REP are not deployed as real subsystems; instead, they are simulated using the SimATO tool.

SimATO receives real-time data from the real TCMS via a standard Ethernet connection, allowing the simulated ADM, APM, and REP to process realistic vehicle and operational information. Communication between the Remote Driving System and the real TCMS is established through a modem installed on the Remote Driving System Trackside side. This modem communicates with the Remote Control – OnBoard (RC-OB) subsystem via a wireless communication link. As already stated in Section 3.2, the test environment may likewise be configured to be compliant with the characteristics of the specific train subject to integration.

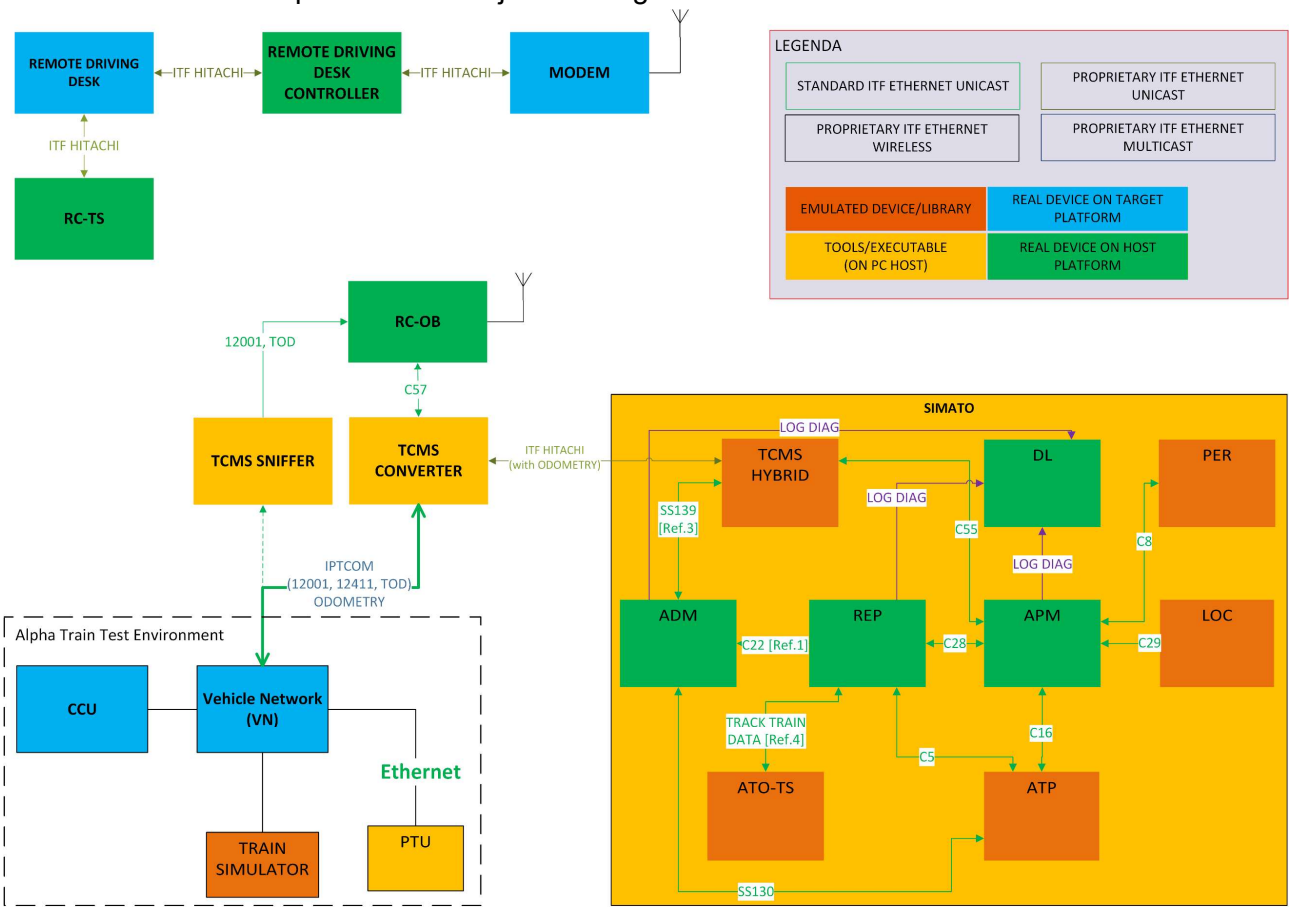


Figure 5: Alpha Train Architecture

Interne

4 TOOLS DESCRIPTIONS

This section provides a detailed description of the capabilities of the tools employed in the laboratory setup, which are used to support the integration tests performed across the different environments described in section 3.

The tools used for the integration test executions are the following:

- SimATO: the simulation software
- LogExplorer: the log analysis software

Both tools are described in the following sections.

4.1 SIMATO

The test environment consists of a real Target system connected to a simulator. The simulator is an application developed by STS, named SimATO, which is used to simulate all the connected subsystems as well as the train behaviour.

SimATO is a software application running on Windows used in target configuration. The configuration

is defined through dedicated configuration files. The simulator, installed on an industrial PC, communicates with the system under test via appropriate physical interfaces, such as serial links and Ethernet links and simulates the external components interacting with the subsystems under test.

In particular, SimATO is capable of simulating Trackside behaviour by generating a SS126 interface, which allows Journey Profile and Segment Profile information to be transmitted to the ATO components at the beginning of the Journey. In addition, SimATO is able to implement a more dynamic behaviour, this is achieved through the transmission of dedicated stimulus packets that introduce controlled perturbations during the execution of a test scenario. These inputs are particularly important to stimulate all the architectural components and verify the related behaviour.

Furthermore, SimATO supports the execution of so-called “change Journey” procedure. This functionality allows the same train to be associated with a sequence of Journey Profiles rather than with a single Journey Profile for the entire session. Multiple Journey Profiles can therefore be applied within the same test execution.

These functionalities are enabled through the following features:

- Automatic inputs (events) defined in the test scenarios. For repeatability, these inputs are time-based or linked to specific events occurring during test execution.
- Automatic outputs (goals) defined in the test scenarios. These are defined as the achievement of a certain objective or the fulfilment of a certain condition.

By means of automatic inputs and outputs, the simulator allows the behaviour of each subsystem to be directly checked during test execution. Additionally, detailed log files are generated at the end of the execution to support an in-depth analysis of the subsystem responses to the applied stimuli.

4.2 LogExplorer

After the execution of a test scenario, the corresponding output log files are extracted and analysed. This analysis is performed using LogExplorer, an internal Hitachi tool designed for the management and evaluation of logged data.

LogExplorer provides a graphical user interface that allows users to analyse logged data through queries and data visualisations as plots. The tool implements the interface protocols of all architectural components, together with the related message definitions. This enables testers to navigate across different interfaces and to analyse the content of each packet exchanged between the system components.

5 TEST SCRIPT FORMAT

This section describes the format and structure of the test scripts, defining the conventions, templates, and execution logic adopted for the integration tests. The focus of the section is to highlight consistency with system requirements, reproducibility of test scenarios, and traceability of test results.

5.1 Overview of the Testing Methodology

The testing approach is based on a structured process in which predefined actions are executed, and the system's responses are observed. Each test is designed to introduce specific stimuli into the system, record the resulting behaviour, and compare the actual outcomes with the expected ones. This method ensures that any deviations or unexpected behaviour are identified and documented.

Each test aims to:

- Provide the system with input stimuli
- Observe and record the software's reactions
- Verify the results by comparing them with predefined expected outputs

5.2 Structure of the Test Suite

All integration tests are organised within a dedicated Test Suite, which contains the full set of test cases. Each case is described in a structured and repeatable way, ensuring completeness and traceability.

Each test specification includes:

- Test Scope: describing the functional area under examination and the purpose of the test
- Test Description: Provides a concise explanation of the purpose and logic of the test
- Preconditions: outlining the initial conditions required before execution, such as system configuration or system status
- Inputs: detailing the actions to be performed during the test
- Outputs: defining the expected results and the criteria for determining whether the test has passed
- Remarks/Justifications: providing assumptions, clarifications, or additional explanatory notes

5.3 CREATION OF TEST SCENARIOS

The tests defined in the Test Suite are translated into XML scenario files tailored for the simulation environment. Each scenario contains the configuration of the track, the georeferenced track elements, the input actions to be triggered, and the goals used to assess the results.

Segments and track elements are referenced via CSV files that are linked to the XML scenario definitions.

The test scenarios are developed in accordance with the use cases defined in “D39.3 – Demo Specifications”.

The complete set of scenarios includes:

- Five nominal scenarios, representing standard operating conditions
- Four degraded scenarios, simulating abnormal or failure conditions to assess system robustness

All scenarios are designed to run fully automatically. Once configured and started, no further intervention by the tester is required; this is to ensure the reproducibility of the tests.

5.4 TEST EXECUTION PROCEDURE

The following sections report the test execution procedure for each integration test environment described in sections 3.2 and 3.3.

5.4.1 ENV1

To execute each test in the target environment using SimATO, the steps below are followed:

1. SimATO is powered on
2. The XML scenario file is loaded into the SimATO test environment
3. The simulation is started by the tester and is carried out automatically
4. Automatic shutdown and log generation

5.4.2 ENV2

To execute each test in the target environment using SimATO and Thanos, the steps below are followed:

1. ATP Converter and TCMS Converter are powered on
2. SimATO is powered on
3. The XML scenario file is loaded into the SimATO test environment
4. Thanos is powered on
5. The MDB scenario file is loaded into the Thanos test environment
6. The simulation is started both on SimATO and Thanos by the tester and is carried out automatically
7. Automatic shutdown and log generation

Interne

5.4.3 ENV3

To execute each test in the target environment using Thanos, the steps below are followed:

1. The Remote Driving Desk is opened
2. Thanos is powered on
3. The MDB scenario file is loaded into the Thanos test environment
4. The simulation is started on Thanos and carried out by the tester
5. Automatic shutdown and log generation

5.4.4 AlphaTrain

To execute each test in the target environment using SimATO, the steps below are followed:

1. TCMS Converter is powered on
2. SimATO is powered on
3. The XML scenario file is loaded into the SimATO test environment
4. AlphaTrain is powered on
5. The simulation is started on SimATO and carried out by the tester
6. Automatic shutdown and log generation

5.5 Result Analysis and Reporting

After the simulation has finished, the generated log files and output data are analysed by means of LogExplorer. All findings are reported into a dedicated Test Report, which documents the outcome of each executed test.

Any discrepancies, unexpected results, or issues encountered during execution is recorded as Work Items (WIs). These are referenced within the Test Report to ensure full traceability and to support subsequent corrective actions.

It should be noted that the purpose of this deliverable is to describe the architecture of the integration laboratories, the supporting tools, and the testing methodology. Representative validation results, execution outcomes and integration evidence generated during the laboratory campaigns are reported in Deliverable D39.7 "Integration Activities Report" [3], which documents the test plans, executed scenarios, obtained results and related findings.

Interne

6 CONCLUSIONS

This deliverable has presented the architecture and operational framework of the integration laboratories developed within the R2DATO project to support the integration and validation of automated train operation subsystems.

The document has described how the laboratories are structured to enable systematic and repeatable integration testing of the main onboard subsystems, namely the Automatic Driving Module (ADM), the Repository (REP), and the Automatic Processing Module (APM), in combination with both simulated and real external systems.

The definition of multiple integration environments has allowed different aspects of system interaction to be addressed in a controlled manner.

Starting from the integration of the onboard components in isolation, the laboratories extend to configurations including real ATP, Remote Driving systems, and real TCMS. This approach supports early fault detection, risk reduction, and confidence building prior to further system deployment phases.

The document has also detailed the role of the supporting simulation and analysis tools, with particular emphasis on SimATO for scenario execution and subsystem emulation, and LogExplorer for post-processing and analysis of test results. Together with a structured test format, an automated execution strategy and the organisation of test cases within a dedicated test suite, these tools enable high levels of reproducibility, traceability, and consistency across the integration test campaign.

REFERENCES

[1] R2DATO APC Baseline 1 – Release Note FP2 T4_1 I ATB 047 01.

[2] FP2-SCRev-R-SNCF-089-01– D39.3 – Demo specifications

[3] FP2-WP39-D-STS-014-04 - D39.7 – Integration Activities Report