Rail to Digital automated up to autonomous train operation

D26.1 – High-level consolidation of prior work and agreement on the Modular Platform specification to happen in FP2-R2DATO

Due date of deliverable: 31/03/2023

Actual submission date: 05/05/2023

Leader/Responsible of this Deliverable: Maik Fox / DB Netz AG

Reviewed: Y

Document status		
Revision	Date	Description
01	28/02/2023	First issue for internal review
02	07/03/2023	Second issue for TMT Review
03	05/05/2023	Incorporation of TMT and Steering Committee Feedback
04	23/02/2024	Incorporation of External Reviewer Feedback

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
Maik Fox	DB Netz AG	Executive Summary, Introduction, Conclusion, Chapter 3: EULYNX, Chapter 4 & 5, Document Editor, Review
Patrick Marsch	DB Netz AG	Chapter 4 & 5, Review
Oliver Mayer-Buschmann	DB Netz AG	Abbreviations and Acronyms, Chapter 4 & 5, Review
Kai Schories	DB Netz AG	Chapter 4
Julian Wissmann	DB Netz AG	Chapter 3: SIL4 Cloud
Lucas Heinke	DB Systemtechnik GmbH	Chapter 3: Shift2Rail
Martin Kochinke	DB Systemtechnik GmbH	Abbreviations and Acronyms, Review
Viviana-Carolina Rivas-Mesa	Deutsche Bahn AG	Chapter 3: Shift2Rail
Piero Petruccioli	Trenitalia	Review, Prior Work Research
Francesco Inzirillo	MER MEC	Chapter 3: Shift2Rail - X2Rail 1-2-3-4-5
Thomas Martin	SBB	Chapter 3: OCORA, Chapter 4; Review
Valentin Inocencio Cuesta	SBB	Chapter 3: OCORA
Axel Träger	Siemens Mobility	Chapter 3: UNISIG Subset 150, CONNECTA-2, Safe4Rail
Sonja Steffens	Siemens Mobility	Chapter 3: SIL4 Data Center, RCA History; Review
Wolfgang Wernhart	GTS Austria	Chapter 3: RCA; Review
Stefan Resch	GTS Austria	Chapter 3: RCA; Review
Patrick Rozijn	NS Reizigers	Review

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

Computers are ubiquitous and their number is increasing, and the railway sector is no different. A huge amount of computing platforms is needed today, and even more will be needed in the future. They are needed in on-board systems, trackside elements and in data centres for efficient, reliable, and safe operation. By increasing the automation in the railway infrastructure including fully automated trains, new requirements and expectations towards these platforms will rise and add complexity, while the goal of safety may never be in jeopardy.

Supporting this growth in complexity and sheer number of computing systems and platforms, this work package, supported by railway companies and industry partners, aims to define a specification for modular platforms and deliver these not only to the demonstrator work package 36, building the "On-Board Platform Demonstrator", but also to the ERJU System Pillar and future Innovation Pillar activities. As a first step towards the goal, this first deliverable captures the state-of-the-art and available learnings, while already providing an outlook to elements of the future specification work to happen in a subsequent task.

A wealth of prior relevant work for modular platforms was found and consolidated, coming out of EU funded, self-driven initiatives or public-private partnership projects. Together with their learnings, they are listed in this deliverable. Furthermore, a first draft and outlook towards specification work needed to reach the stated goals in the context of modular platforms is shown. Both form the essence of this document, building a basis and reference guide.

Important key learnings coming out of the prior work consolidation are, for example, the need to establish a common language and terminology, e.g., by aligning on a concise glossary together with the System Pillar, and the challenges of multi-vendor integration in a functional safety context. Based on the prior work collected and discussions within the work package, a first architectural sketch, consisting of a computing platform hosting functional applications and implementing common external interfaces is discussed, also giving a first definition proposal for common terms. Both were made available to the System Pillar Computing Environment domain as an input for their work.

Based on this first step represented by this deliverable, further work is necessary creating the specification suite for modular platforms, followed later in the work package by studying approaches for certification and acceptance of modular platforms for the future railway as envisioned by ERJU.

ABBREVIATIONS AND ACRONYMS

ADAS	Automatic Driver Assistance System
ATO	Automatic Train Operation
BMS	Bogie Monitoring System
CONNECTA	CONtributing to Shift2Rail's NExt generation of high Capable and safe TCMS PhAse 3
COTS	Commercial Off The Shelf
CCS	Command, Control and Signalling
ETCS	European Train Control System
FDF	Functional Distribution Framework
FOC	Functional Open Coupling
FVA	Functional Vehicle Adapter
GoA	Grade of Automation
HVAC	Heating, Ventilation and Air-Conditioning
IM	Infrastructure Manager
MDCM-OB	Monitoring Diagnostics Configuration Maintenance On-Board
NG-TCMS	Next Generation TCMS
OCORA	Open CSS On-Board Reference Architecture
PI API	Platform-Independent Application Programming Interface
POSIX	Portable Operating System Interface
R2DATO	Rail to digital automated up to autonomous train operation
RCA	Reference CCS Architecture
RTE	Run Time Environment
RTOS	Real Time Operating System
RU	Railway Undertaking
SCP	Safe Computing Platform
SRACs	Safety Related Application Conditions
SW	Software
TCMS	Train Control Management System
TD	Technical Demonstrator
TMS	Train Management System

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The present document constitutes the Deliverable D26.1 “High-level consolidation of prior work and agreement on the Modular Platform specification to happen in FP2-R2DATO” in the framework of the WP 26, Task 26.1, of the FP2 R2DATO.

Computing platforms, either on-board a train or stationary along the trackside or in a data centre, are an integral part to the modernization and digitization of railway systems. While these computing platforms come in many variants and are built to vastly differing requirements targeting many applications and use cases, there are several common functionalities, nevertheless. Especially modern and ubiquitous computing platforms implement a wide variety of applications that depend on standardized interfaces, guaranteeing their interoperability. This interoperability is also critical to enabling competition between railway undertakings, allowing vendor-neutral access to the network without limiting future technical innovations.

Modular platforms aim to provide the standardized interfaces needed for safety related railway applications, independent on where they are located. Modularization is intended to help in (re-)deploying applications to computing platforms and potentially allow new approaches for (re-)certification, especially when considering software and hardware updates.

Targeting different levels of automated railway systems and grades of autonomy, new challenges for computing platforms can be found in – but are not limited to – topics such as availability, reliability, and cyber security. Here, the modularization approach intends to balance overall architecture subsystem complexity and development effort by providing a suitable and tailored set of specifications.

To lay a foundation for the following work, the first task in R2DATO's work package dedicated to modular platforms was to collect and consolidate prior work and to reach a first agreement on the specification framework to be detailed.

Chapter 2 provides an overview over prior work in the field of (modular) computing platforms in the railway sector. Here, relevant results from previous initiatives and projects are summarized and, where possible, lessons learned in the context of modular platforms are noted as well. References to respective project resources are provided.

Chapter 3 captures the current state of discussion for modular platforms, considering Computing Platforms and Functional Applications – terms aligned with the System Pillar Computing Environment domain – as building blocks that need to be equipped with appropriate interfaces.

Chapter 4 discusses areas of future work in this work package.

A summary and outlook conclude Deliverable D26.1.

2 HIGH-LEVEL CONSOLIDATION OF PRIOR WORK

This chapter provides an overview over prior work in the field of railway computing related initiatives and projects. The focus is on modular platforms and possible learnings that could be derived from these previous activities.

A summary is provided at the end of the chapter.

2.1 Prior Work Overview

In Table 1 the prior work that has been identified to be relevant for this R2DATO work package is listed. Each of the entries has a reference to a chapter discussing the item in detail and a list of references.

PRIOR WORK	CHAPTER	REFERENCES
Shift2Rail	Chapter 2.2.1	[1]
CONNECTA-2, Safe4Rail 1+2	Chapter 2.2.2	[2], [3], [16], [17]
Shift2Rail IP2 (X2Rail 1-2-3-4-5)	Chapter 2.2.3	[27], [28]
RCA	Chapter 2.2.3	[4], [11], [12], [13], [14], [15], [26]
OCORA	Chapter 2.2.5	[5], [18], [19], [20], [21], [22], [23], [24], [25]
SIL4@Cloud	Chapter 2.2.6	[8],
SIL4 Data Center	Chapter 2.2.7	[9] and [10]
UNISIG Subset 150	Chapter 2.2.8	n/a
EULYNX	Chapter 2.2.9	[6], [7]

Table 1: List of relevant Prior Work

The following subchapter will outline the project's contents and relevancy to modular platforms. Furthermore, learnings form these projects will be discussed where possible.

2.2 Individual Prior Works

The following subchapters contain individual summaries and learnings of the relevant prior work.

2.2.1 Shift2Rail

Shift2Rail is the predecessor of EU-Rail and therefore a European initiative that aims to improve the competitiveness of the European rail industry and meet the changing transportation needs of the EU by focusing on research and innovation. The initiative was dedicated to integrate new and advanced technologies into innovative rail product solutions that are driven by market demands. The objective of Shift2Rail is to double the capacity of the European rail system, increase its reliability and service quality by 50%, and reduce life-cycle costs by half.

Shift2Rail has a robust framework that is organized around five Innovation Programmes, which cover all aspects of the rail system. Most important for WP26 are Innovation Programme 1 and 2, therefore below is a short overview on these to sub-projects:

The Innovation Programme (IP) 1 was created to enhance the design of trains to make them more appealing to passengers, railway operators, and urban operators. The future generation of passenger trains must be more energy- and cost-efficient while offering a safe, comfortable, and affordable travel experience. To achieve this goal, an innovative approach is necessary. This includes new modular solutions for designing attractive and comfortable trains, higher-performance technologies for traction, command-control, and cabin environment applications, and flexible, reliable, and safe design and production solutions. Innovative solutions must also be developed to extend vehicle lifetime, simplify retrofitting, and ensure that networks can support the operation of these vehicles.

The European Rail Traffic Management System (ERTMS) has become a leading global solution for railway signaling and control systems. However, ERTMS has the potential to offer even greater functionalities by integrating new technologies, such as high-speed data and voice communications, automation, and real-time data processing. Innovation Programme 2 was created to enhance the flexibility of control, command, and communication systems to enable intelligent traffic management and decision support. This improved train movement efficiency, reduce energy consumption and carbon emissions, lower operational costs, enhance safety and security, and provide better customer information. The current lack of standardization and compatibility between systems increases costs and hampers interoperability. Shift2Rail seeks to address these issues by improving system functionalities and standardizing interfaces, while also adapting to the needs of different rail segments and a multimodal smart mobility system.

Important research for modular platforms was done in IP 1&2, which can be used as an input to this work package's future tasks. Two of the most important projects in this context were CONNECTA and Safe4Rail, which will be explained in the next chapter. Also, X2Rail will be explained in 2.2.3.

2.2.2 CONNECTA-2 & Safe4Rail 1+2

Project Reference: CONNECTA-2: "CONtributing to Shift2Rail's NExt generation of high Capable and safe TCMS. PhAse 2" (see [2]), with complementary project: SAFE4RAIL-2 (see [3]).

Executive Summary of CONNECTA-2 (Taken from [16])

> Throughout this second phase of CONNECTA projects series –and based on the specifications and architectures defined within CONNECTA-1 project–, CONNECTA-2 has gone deeper into the refinement and improvement of the different specifications, and it has also addressed the design and implementation stages of the set of technologies comprising the Next Generation TCMS (NG-TCMS):
>
> - Drive-by-Data (DbD) concept, including safe train inauguration, safe data transmission and scheduled data traffic.
> - Function Distribution Framework (FDF) concept based on Integrity RTOS and on AUTOSAR Adaptive Platform.
> - Application Profiles (AP) for a set of selected subsystems (HVAC, doors and BMS).
> - Wireless Train Backbone (WLTB), including the implementation of wireless safe train inauguration.

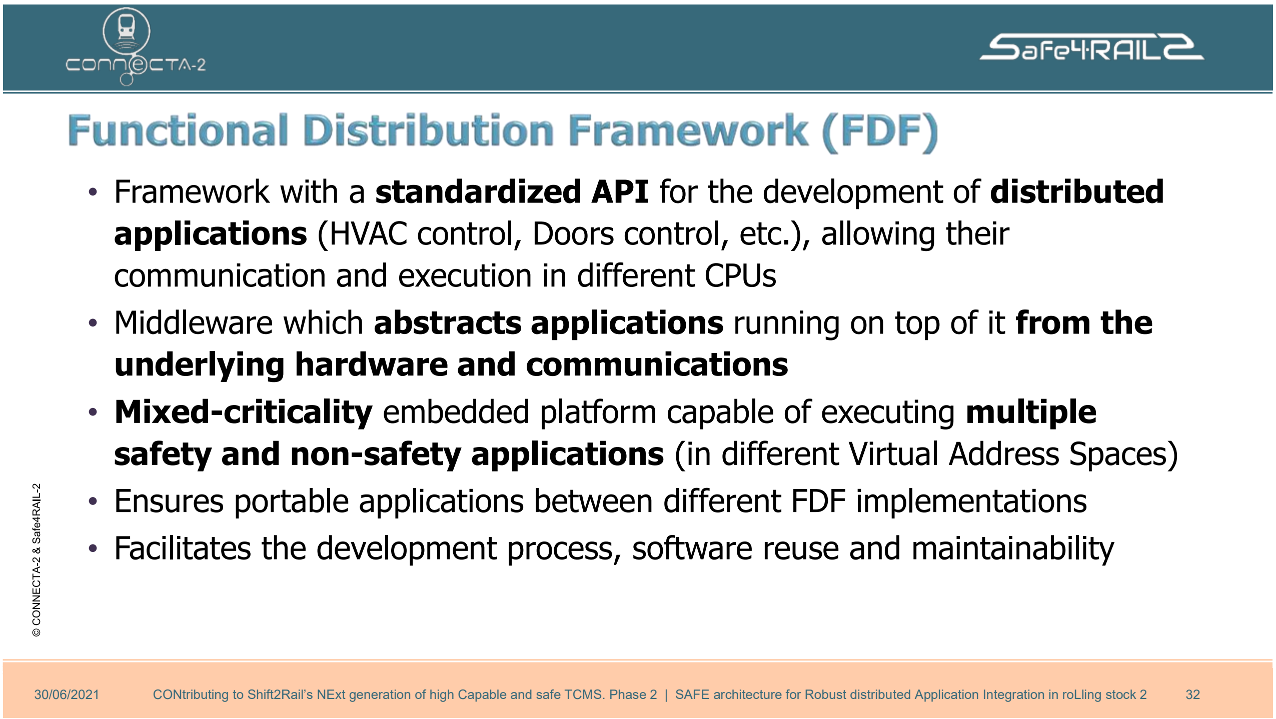
- Wireless Consist Network (WLCN).
- Train-to-Ground (T2G) communication.
- Functional Open Coupling (FOC) for a set of selected functions (HVAC and Doors).
- Simulation and Virtualization Framework (SVF), including the simulation/virtualization of train subsystems, as well as the functionality offered by the SVF to execute automatized tests to validate these NG-TCMS technologies in a virtualised environment.
- Furthermore, CONNECTA-2 has made a great effort in the implementation of components and their integration into two different laboratory demonstrators for urban and regional train applications, respectively.

It is worth noting that the work carried out in CONNECTA-2 during the last 34 months is completely aligned with the activities planned within the Shift2Rail TD1.2 Multi Annual Action Plan (MAAP), in which CONNECTA-2 is not expected to cover the whole life cycle of the Next Generation TCMS, but only some of the first activities in different controlled laboratory environments (TRL 4/5).

In general terms, the project has fulfilled the expectations and has provided the required knowledge and implementations in order to address the system testing phase during the successor project (i.e. CONNECTA-3), not only in a laboratory environment, but also in real train environments (TRL 6/7).

Finally, it must be remarked that the project has put much effort in communicating and disseminating the results, providing valuable inputs to ERA TWG ARCHI, OCORA and IEC WG43.

Project Learnings (taken from [17]):



CONNECTA-2 Safe4RAIL-2

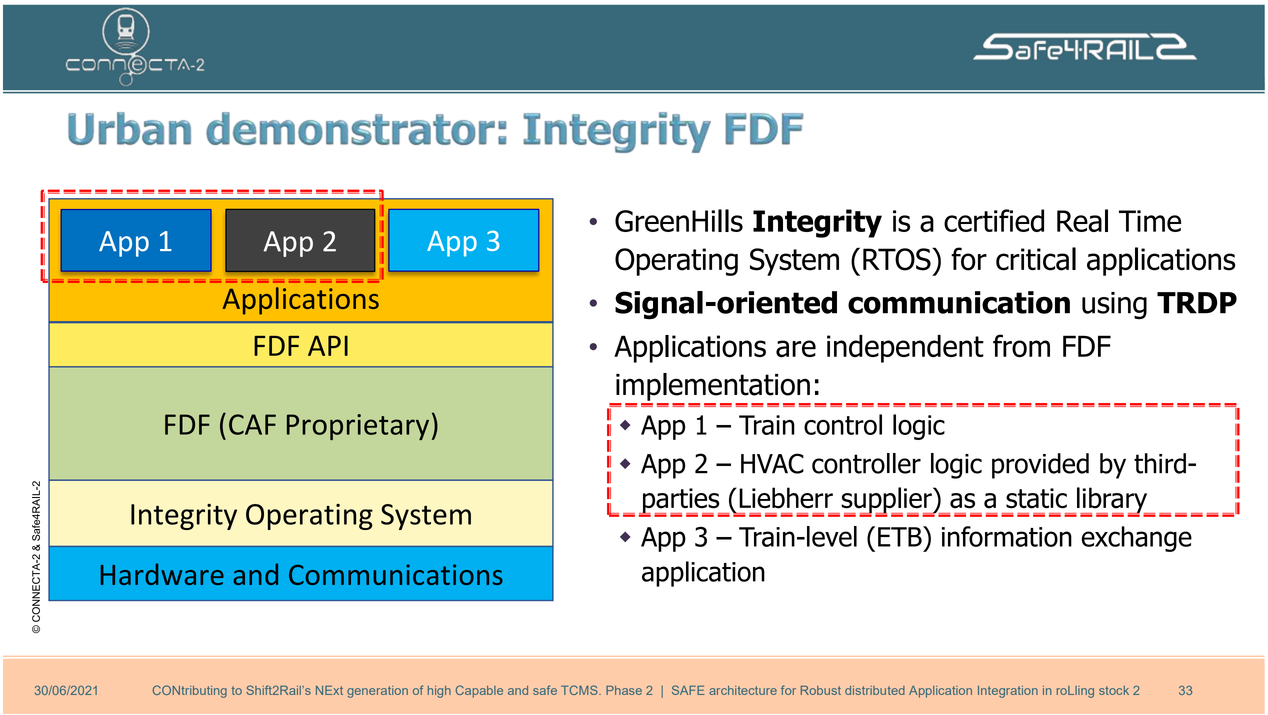
Functional Distribution Framework (FDF)

- Framework with a **standardized API** for the development of **distributed applications** (HVAC control, Doors control, etc.), allowing their communication and execution in different CPUs
- Middleware which **abstracts applications** running on top of it **from the underlying hardware and communications**
- **Mixed-criticality** embedded platform capable of executing **multiple safety and non-safety applications** (in different Virtual Address Spaces)
- Ensures portable applications between different FDF implementations
- Facilitates the development process, software reuse and maintainability

© CONNECTA-2 & Safe4RAIL-2

30/06/2021 CONtributing to Shift2Rail's NExt generation of high Capable and safe TCMS. Phase 2 | SAFE architecture for Robust distributed Application Integration in roLling stock 2 32

Figure 1: CONNECTA-2 and Safe4Rail-2 Learnings w.r.t. FDF [17]



CONNECTA-2 Safe4RAIL2

Urban demonstrator: Integrity FDF

App 1 | App 2 | App 3

Applications

FDF API

FDF (CAF Proprietary)

Integrity Operating System

Hardware and Communications

© CONNECTA-2 & Safe4RAIL-2

- GreenHills **Integrity** is a certified Real Time Operating System (RTOS) for critical applications
- **Signal-oriented communication** using **TRDP**
- Applications are independent from FDF implementation:
  - App 1 – Train control logic
  - App 2 – HVAC controller logic provided by third-parties (Liebherr supplier) as a static library
  - App 3 – Train-level (ETB) information exchange application

30/06/2021 CONtributing to Shift2Rail's NExt generation of high Capable and safe TCMS. Phase 2 | SAFE architecture for Robust distributed Application Integration in roLling stock 2 33

Figure 2: CONNECTA-2 and Safe4Rail-2 Urban Demonstrator [17]

Middleware Learnings (taken from [17]):

- A stable and long-lasting decision on the middleware is not possible at the moment
  - None of the candidates fulfils the major requirements for our domain with SIL4
  - Good evolvability additionally is required to support the rapid changes in our domain for digitalization and decarbonization
  - Even the Automotive for the ADAS domain tried now for more than 5 years without success to define a suitable solution and fast movers use successfully alternative solutions
  - ➢ standardization of the middleware is not feasible at the moment
- Interoperability and evolvability are better achieved by SysML based Application Profiles
  - already represent a considerably engineering cost improvement.
  - Supports direct integration of the standardization in the development tool chain
  - Leaves required degrees of freedom for innovation and standardization
- Binary interoperability is not suitable due to different
  - Operating Systems (OS),
  - Hardware
  - Serialization of Communication protocols
- APIs from application to the FDF is usually implementation dependent and requires code generation to avoid user failures in safety critical systems

2.2.3 Shift2Rail IP2 (X2Rail 1-2-3-4-5)

The Shift2Rail IP Advanced Traffic Management and Control Systems project aims to create demonstrators for the following technical enablers:

- TD 2.1 - Adaptable communications for all railways, see [30]
- TD 2.2 - Automatic Train Operations – ATO (up to GoA4) , see [31]
- TD 2.3 - Moving Blocks, see [32]
- TD 2.4 - Fail-Safe Train Positioning, see [33]
- TD 2.5 - On-board Train Integrity, see [34]
- TD 2.6 - Zero on-site testing, see [35]
- TD 2.7 - Formal methods and standardisation for smart signalling systems, see [36]
- TD 2.8 - Virtually – Coupled Train Sets, see [37]
- TD 2.9 - Traffic management system, see [38]
- TD 2.10 - Smart radio-connected all-in-all wayside objects, see [39]
- TD 2.11 - Cyber Security, see [40]

It follows a short description of every Technical Demonstrator (TD).

TD 2.1 – Adaptable communications for all railways

The main target applications are the existing ETCS (European Train Control System) and CBTC (Communications-Based Train Control), all train-to-ground communications for train control applications including voice applications for Train and Metro. The system shall be based on Packet switching/IP technologies (GPRS, EDGE, LTE, 5G, Satellite, Wi-Fi, etc.), in accordance with the findings of the ongoing NGTC project, other national and European projects and with the requirements of current safety and security standards. The system will enable easy migration from existing systems (e.g.: GSM-R or WiFi), and will provide enhanced throughput, safety and security functionalities to support the current and future needs of signalling systems and voice emergency services, and be resilient to interference and open to radio technology evolution.

TD 2.2 – Automatic Train Operations – ATO (up to GoA4)

Develop an ATO up to GoA4 starting from what is previously done in other projects and in UNISIG.

Automatic Train Operation (ATO) is already proven in urban applications for different Grades of Automation (GoA, up to GoA4). Compared to urban systems, the situation for mains line systems is more complex:

- Larger and interconnected track layout;
- Diverse rolling stock fleet;
- Complex and diverse operating scheme;
- Different responsibilities of IM and RU.

The development carried out in the project takes the points mentioned above and adapts them to the railway needs.

TD 2.3 – Moving Blocks

The objective of the Moving Block is to define a high capacity, low cost, high reliability signalling system, based on Moving Block principles, which is applicable across all railway market segments.

High Capacity is based on the use of Moving Block principles, which permits decoupling of the infrastructure from train performance parameters. Low Cost is achieved by the reduction in the use of trackside train detection and line-side signals. High Reliability is achieved as a consequence of the reduction in trackside equipment associated with trackside train detection and line-side signals.

TD 2.4 – Fail-Safe Train Positioning

The Fail-Safe Train Positioning (including satellite technology) is aimed to become an absolute positioning system, significantly reducing the number of the traditional wayside train detection systems. The solution will be based on a safe on-board multi-sensor positioning concept, where GNSS is the preferred technology. The approach taken to apply the GNSS based localization functionality to ETCS will guarantee the ETCS interoperability concept from one side and will allow the introduction of the state-of-the-art technologies in the use of absolute position technologies (e.g. GNSS and different Augmentation Subsystems) and of kinematic sensor technologies (e.g. inertial sensors, gyroscope sensors, MEMS).

TD 2.5 – On-board Train Integrity

The Train Integrity is an on-board function responsible for verifying the completeness of the train permanently, while the train is in operation. This consists entirely in monitoring the status of the train's integrity by e.g. tail and train length detection: if the last vehicle is regularly advancing in a coherent way in relation to the movement of the remaining train and the train length remains unchanged, then the Train Integrity system can easily deduce that everything is working properly.

TD 2.6 – Zero on-site testing

System- and Integration Test (SIT) is a fundamental method of system verification across many different industrial sectors. Due to the complexity of signalling systems and the differences between sites, a large amount of tests must be carried out on-site. Today's procedures of verification & validation testing differ all around Europe. Overcoming these differences by standardizing the procedures and test scopes will improve the interoperability and reduce the time to market.

TD 2.7 – Formal methods and standardisation for smart signalling systems

This TD addresses the use of formal methods and standard interfaces to meet the following objectives:

- Save cost in signalling system life-cycle (LCC)
- Support independent lifetime of sub-systems in the control command and signalling system
- Increase market competition and standardisation
- Improve interoperability and reliability
- Shorten time-to-market of new products
- Increase know-how of formal methods in railway signalling

Many different types of formal methods exist, with various purpose and benefit, such as increasing confidence in system reliability and correctness, improving specification quality for system development, and reducing the effort for verification and validation. Formal specification refers to mathematically precise techniques to describe a system or its properties, formal development is used to produce software compliant by construction with its formal specification and formal verification is used to verify that a system satisfies given properties (its specification), during development or in safety assessment.

TD 2.8 – Virtually – Coupled Train Sets

This activity addresses the increment of network capacity, beyond the limitation of the current signalling approach for train and units separation. Increased capacity is today needed for many networks in Europe. Building new lines or adding tracks to existing lines is a slow and expensive process. Furthermore, decoupling, shunting and coupling is a key feature of the traditional railway system. Virtual Coupling can help to bolster the competitiveness of rail as regards flexible and timely operation on demand.

TD 2.9 – Traffic management system

The overall objective of the TD is to design a scalable and interoperable Data Layer providing the data exchange between TMS and Asset Management and external services either leveraging on the available Information and/or sourcing required Information e.g. weather forecast to the Integration Layer. The information provided from the different connected applications, clients and services is available on one "Data-Highway" – the Integration Layer. These Information shall have a standardized Data-structure and Interfaces, shall support High-speed Data transmission and shall be able to be extended in the future for Topics (Data-lines) carrying specific Information of different complementing services/clients. The design of the content of the CCS related "Topics" shall be oriented in a first step on the requirements of TMS, Asset Management, conventional train operation, Automatic Train Operation and, Moving Block/ERTMS L3 Operation with focus to reduce complexity of technical solutions needed to be implemented on-board/trackside.

TD 2.10 – Smart radio-connected all-in-all wayside objects

The overall scope of this TD is to contribute to the development of an autonomous, intelligent, maintenance-free smart equipment ("box") able to connect with any signalling wayside object and communicating device in the area (by radio or satellite) in order to foster overall reduction both of installation and maintenance costs. The objective of this task is to arrive at definitions and specification of practical demonstrations of prototype systems. The completion and integration of the Smart Wayside Objects Technical Demonstrators is envisaged to be within the future Shift2Rail IP2 projects.

TD 2.11 – Cyber Security

The main objectives of this TD are:

- The definition of a cyber security system dedicated to railway;
- The definition of a security-by-design standard applicable to railway application;

The definition of a cyber security system consists in the specification of standardised interfaces, monitoring functions, protocol stacks and architectures for secure networks based, among other, on a security assessment of existing railway solutions and of railway networks. Efficiency and robustness of the standardised solution has to be demonstrated through a technical demonstrator. Security assessment, identification of the threat detection, prevention and response processes will be completed.

Demonstrators for cyber security protection/management will be developed.

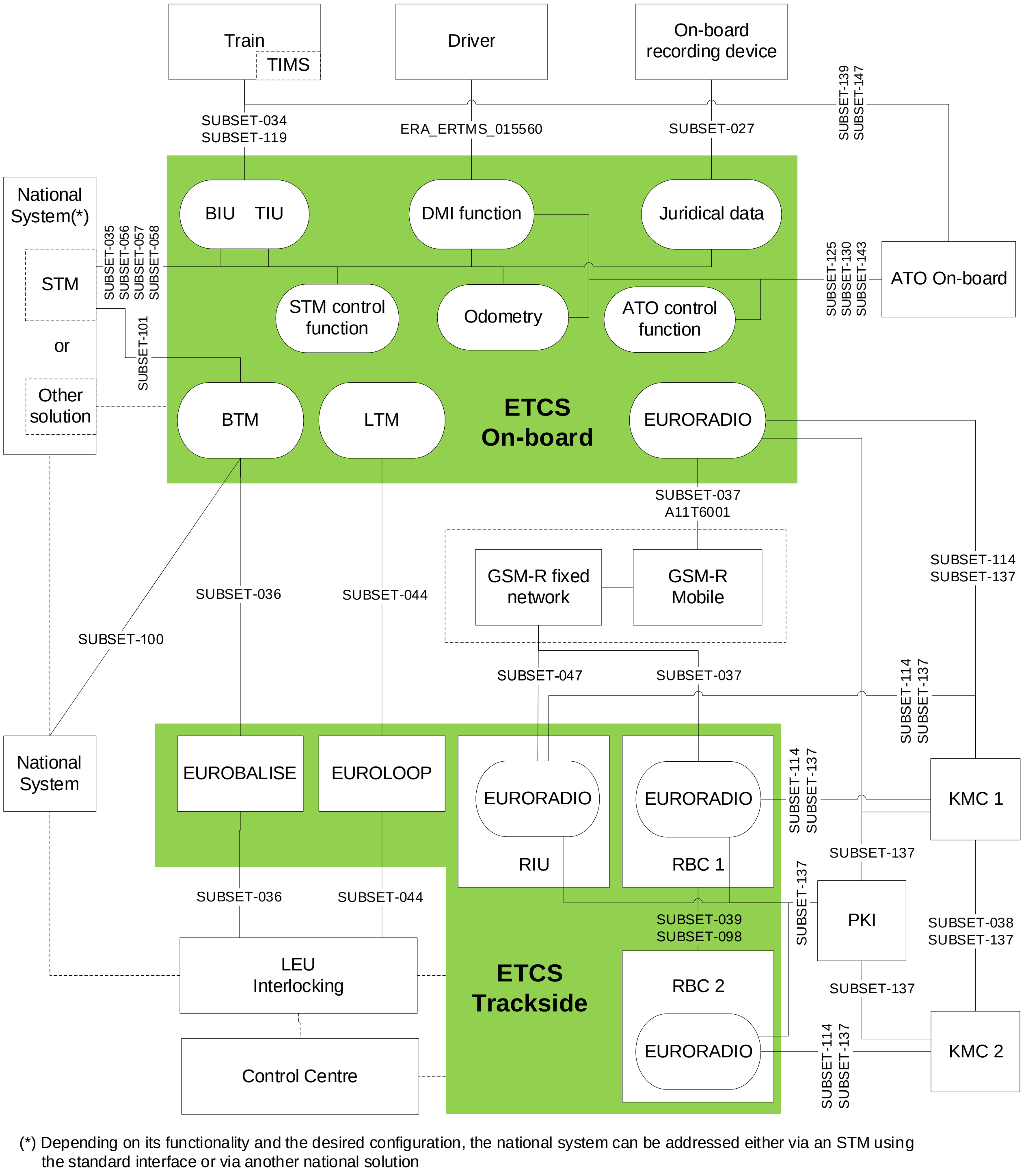
Summary

Each TD, described above, has worked in defining its objectives but the reference architecture has always been ETCS. The result is present in Figure 3.

For each TD (Technical Demonstrator) there is the associated architecture that solves the specific demonstrator. The architecture of the entire system conforms to the one indicated below, and which in this case does not integrate the TMS.

The architecture that has been adopted/developed in Shift2Rail IP2 is from a functional point of view always in line with the TSIs (including those soon to be issued). See [27].

The architecture is also in line with the solutions proposed by UNISIG and ERA.



(*) Depending on its functionality and the desired configuration, the national system can be addressed either via an STM using the standard interface or via another national solution

Figure 3: Proposed Architecture of Subset 026 for next TSI (evolution of [28], going to be published in [27])

This is in line with the Moving Block, ATO functions and ready for the management of the new functions introduced by the TSI.

This architecture lacks the part of Fail-Safe Train Positioning which is being defined in X2Rail_5.

A cyber assessment was also made using this architecture as a reference. Figure 4 (see [28]) shows an example how next generation products and new rail automation systems can incorporate FRMCS, ATO and CONNECTA solutions.

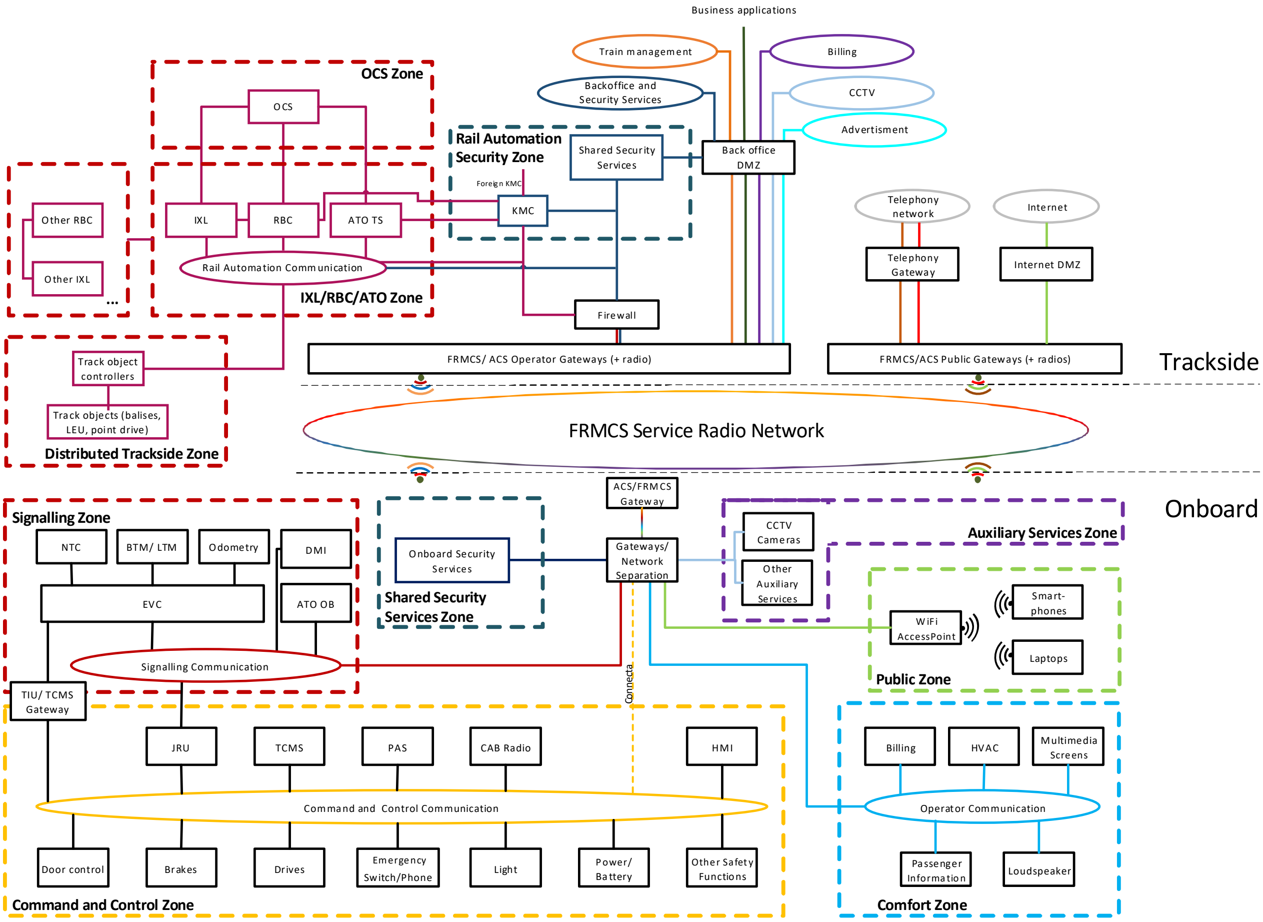


Figure 4: X2Rail Communication Model (from [28])

The pertinent "zones" and the links between them have been identified.

2.2.4 RCA

Overview over RCA

RCA (Reference CCS Architecture), where CCS stands for Control, Command and Signalling, aims to define and standardise future command- and control systems for railway. It is steered by the ERTMS Users Group and the EULYNX Initiative within the RCA Strategy Group.

Currently, a Whitepaper [13] explaining the ideas regarding functions and architecture has been released. Furthermore, details are documented within the RCA baselines [14], where baseline 1 release 0 is the most recent one.

The motivation for RCA is stated in [12] as:

> European infrastructure managers have to optimize cost, reliability, safety, capacity and fast migration. Assets have to be renewed while new technologies are introduced. Incompatibilities create high cost and investment risks. Inflexible old architectures bind the asset owners to technologies and processes which are not making use of the possibilities of today. Expensive migration challenges hinder the infrastructure managers to change their situation fast.
>
> RCA is introducing a modern architecture for the CCS field based on radio based ERTMS and EULYNX.
>
> It creates a flexible and safe investment situation by defining a framework for modular standard products as a set of standard requirements used in CCS procurements.
>
> The digitization of command, control and signalling will only succeed together with a market, that uses standard architectures.

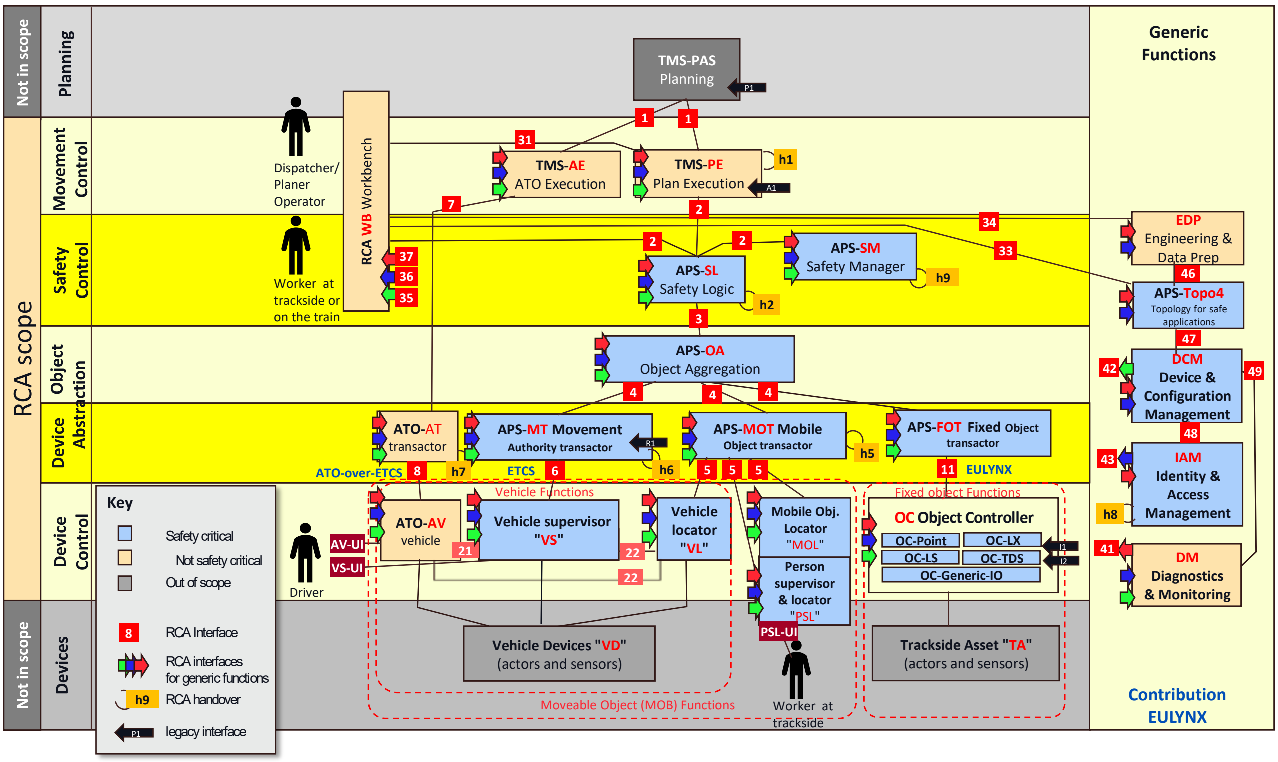


Figure 5: RCA Architecture Overview (from [15])

The overall architecture of RCA is outlined in Figure 5. Its scope is from movement control down to device control with the Advanced Protection System (APS) functions realized on different layers in the centre, building up on the object controller for fixed object functions, as well as the movable object functions such as the vehicle supervisor.

While all of these functions can be implemented independently, the safe functions marked in blue could be perfect candidates to be realized using a common safety-critical platform. Furthermore, such a common platform can be realized not only for data centres or trackside, but also for on-board. This is also reflected by the fact that the latest RCA baseline includes the on-board related OCORA definitions for the Platform Independent API [20], as well as the computing platform [18] and computing platform requirements whitepaper [19] as basis for all these functions also in context of RCA. OCORA is described in more detail in the next chapter.

History and Learnings of RCA

SBB with program SmartRail 4.0 had been a main driver of RCA for several years (2017-2020) until the program was reoriented in 2020 when BAV ("Bundesamt für Verkehr", the Federal Office of Transport of Switzerland) had raised objections (see [26]).

BAV stated out: the program inclusive RCA was too technically oriented, difficult to understand and not enough coordinated with international developments. There was a considerable high risk that the huge complexity cannot be managed.

Because of the risks mentioned, the BAV did not see SmartRail as expedient, for example the development of a new type of interlocking or projects for the development of automated trains that are not based on the existing driver's cab signalling (ETCS Level 2). Meanwhile the SBB SmartRail 4.0 program is no longer pursued.

Additionally, BAV stated, that a holistic coordination with EU is needed to avoid island-like solutions within Switzerland. In fact, this was the reason why RCA is now seen as de-facto continuation of SmartRail on EU level, on a less technical but more common level.

2.2.5 OCORA

OCORA, the "Open Control, Command and Signalling (CCS) On-board Reference Architecture" initiative, aims to reduce life-cycle costs and facilitate the introduction of innovation and digital technologies with standardized interfaces, by establishing a modular, upgradeable, reliable and secure CCS on-board architecture. OCORA deliverables are published under the European Union Public License (EUPL) and are consequently available for all stakeholders.

Today's deployed CCS On-Board systems are proprietary and vendor specific solutions, creating a dependency to the specific vendors.

The OCORA architecture introduces a generic platform concept where core CCS functions can be realised on a generic platform that enables adding, removing or changing functional applications without affecting the computing platform or runtime environment on which they are installed or the state of approval of non-affected parts of the system, e.g. unchanged functions [24]. This will facilitate fast and easy software updates and upgrades of only those applications for which that is necessary, e.g. when requirements demand frequent updates of security software. Safety and non-safety critical applications shall be able to coexist on the same platform and may even run on the same computing hardware. To do everything above, the platform design will apply the following paradigms:

- Satisfy European railway norms in their application up to SIL4,
- Clean separation of concern between the functional applications for business logic only, and the platform for all other required functions related to application execution and control,
- Enable safe and secure communication with external entities like trackside object controllers or other separate on-board systems,
- Follow a modular safety concept,
- Maximize usage of COTS components, tools and Open-Source software to minimize vendor lock-in and leverage advances in other sectors,

- Provide mechanisms to sufficiently isolate applications, especially when the SIL levels are different when running on the same physical platform,
- Implement a harmonized Platform-Independent (PI) API.

The use of an open, unified PI API shall open the door to platform agnostic application development while entrusting the computing platform internals to experienced vendors. A platform agnostic functional application in the context of OCORA is a software that interacts with the platform (devices, operating system, hardware) exclusively via the standardized PI API, allowing the application to be run on all platforms that provide the standardized abstraction (API). It shall follow the guiding principles that are:

- Functional Actors shall not be aware that they are being run by the Platform in Replica,
- The PI API shall maximally leverage already available API specifications to facilitate portability of existing Applications to the new API, and ensure that platform realizations can maximally leverage existing implementations,
- The number of API functions will be restricted to facilitate the portability of applications among realizations and the ease of certification and acceptance,
- Common functions for on-board and trackside shall be maximized (in the work that has led to the first version of the PI API specification, no capabilities or functions of the API have been identified that would be needed only for on-board or only for trackside deployments),
- The evolvability of PI API specs with backward compatibility,
- PI API definition is programming-language agnostic so the PI API may be implemented in any programming language supporting application development up to SIL 4. The selected language must support the CCS application requirements (e.g., real-time demands).

It will take several computing platform iterations to achieve all OCORA goals (e.g., full hardware independency may not be achieved in the first computing platform generation). Along the journey, applications’ portability will be indispensable to leverage implementation and certification effort across the evolving computing platform generations.

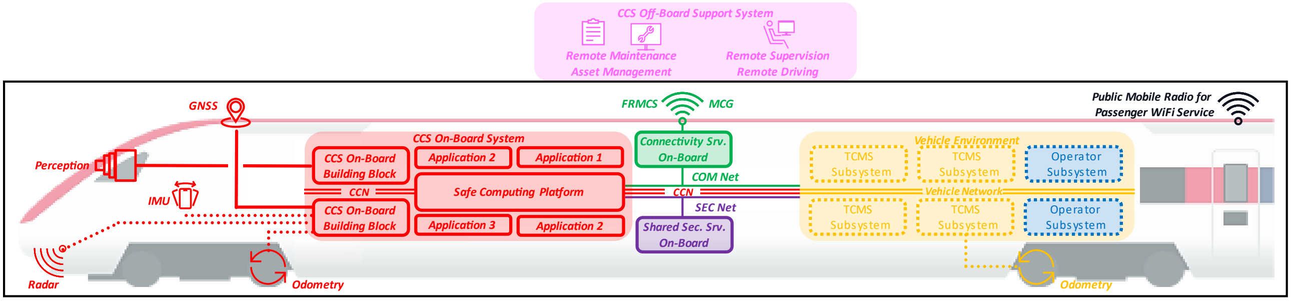


Figure 6: Future view: CCS building block integration supported by vehicle standardisation (from [25])

Already under scrutiny of OCORA and Shift2Rail, CONNECTA is the convergence of vehicle networks, consisting of one or multiple bus systems that integrate the CCS and vehicle bus systems [2]. Train interfaces allowing to connect to legacy bus systems may disappear but standardised secured communication interfaces, to either physical or virtual building blocks, must be anticipated in order to facilitate decoupling. This would ease safety approval, non-regression, cyber-security and maintenance management, while allowing for innovation and fair competition.

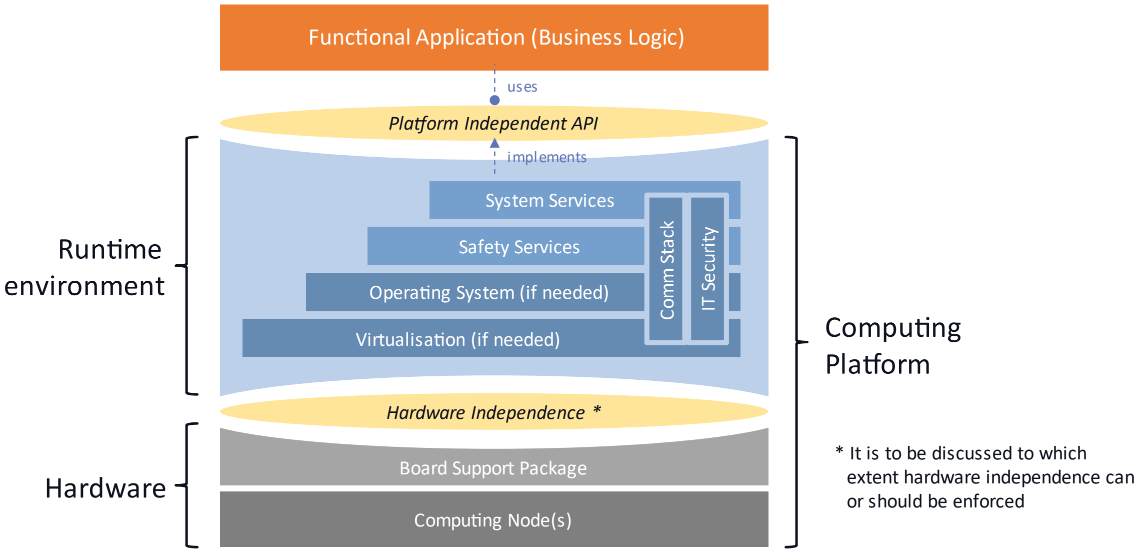


Figure 7: General computing platform principle and terminology (from [25])

Figure 7 is the proposed architecture for the Safe Computing Platform as described in the Computing Platform whitepaper from OCORA.

While the joint specification work among railways and industry suppliers has helped to move the vision of a standardized separation of (safety-related and non-safety-related) railway applications from the underlying IT platforms forward, substantial further specification work and of course prototyping is required.

Additional specifications should be developed, or further considerations are required, for instance related to:

- logging and diagnostics;
- remote updates;
- orchestration;
- standardized tooling and testing;
- modular certification approaches;
- IT Security;
- persistence of application data;
- interfaces "below" to the virtualization (and HW layer);
- juridical recording;
- scalability.

All the previous information has been extracted from three documents where more detailed information on the Computing Platform is available:

- Computing Platform – Whitepaper [18],
- Computing Platform – Requirements [19],
- Initial Specification of the PI API between Application and Platform [20],

- CCS On-Board Architecture [24].

Moreover, it should be highlighted that the PI API specifications have been jointly developed in Workshops held with industry partners DB Netz AG, duagon AG, Nederlandse Spoorwegen, Real-Time Innovations (RTI), SBB, Siemens Mobility GmbH, SNCF Voyageurs, SNCF Réseau, SYSGO GmbH, Thales and Wind River in the time frame between December 2021 and June 2022.

2.2.6 SIL4@Cloud

The research report "SIL4 Cloud" (see [8]) created by Thales, Sysgo, Fraunhofer IESE, University of Rostock and ESE describes a modular, on-premises, private cloud environment for hosting safety-critical rail applications.

Objectives of the report:

- Meeting safety and real-time requirements of CCS (and similar) railway applications
- Usage of COTS hardware based on an open interface
- Flexible mapping of applications to compute resources
- Support of the coexistence of SIL4 and non-SIL applications
- Support of 3rd party applications through an open interface
- The platform shall allow for a modular safety certification process, using pre-certified components

The following platform options were considered in the report:

- TAS Platform from Thales
- PikeOS from SYSGO

Additionally, unsolved challenges were highlighted in the report and need future work:

- Cross-vendor integration processes, roles, and responsibilities
- Standardization of PI API
- Standardized communication between different RTE solution

The conclusion drawn with respect to RCA and OCORA in the report [8] in chapter 13 is:

> Overall, the idea and the architectural concepts were well received by industry partners, and no major showstoppers were identified. SIL4 Cloud approaches based on two platforms (TAS Platform and PikeOS), also fulfil high-level objectives and key design principles of RCA/OCORA Safe Computing Platform. With an intent to achieve full potential of SIL4 Cloud in the long run, the partners have also identified a need for finetuning the extent of technical expectations and identified open points such as limits of dynamic behaviour and integration aspects.

2.2.7 SIL4 Data Center

The research report „SIL4 Data Center"(created by Siemens Mobility and Deutsche Bahn, see [9] and [10]) describes different aspects of a COTS based data centre for safety-critical rail applications with modular architecture and common platforms.

Content of the report is:

- evaluation of the high-level objectives of RCA/OCORA
- basic architecture definition for aggregated running of legacy applications and new applications on common platforms on COTS hardware
- complexity in integration and testing of the modular architecture
- solutions-specific aspects of an API of the runtime environment
- migration concept for legacy applications
- system interfaces to standardize
- maintenance aspects
- geographical redundancy aspects
- security architecture
- dependencies and complexity regarding vendor multiplicity for the modular parts
- certification and homologation

Additionally **unsolved challenges** from process view and technical view are pointed out:

- cross-vendor integration processes, roles, responsibilities
- juridical recording of cross-vendor influences
- solutions specific API aspects
- standardization of test and engineering environments
- bundling of applications
- common Virtualization
- dynamic CPU resource management
- non-safe RTE for non-safety-critical applications

The conclusion sums up the findings at the end of the research report.

2.2.8 UNISIG Subset 150

This chapter introduced the “Concept for the evolution of the on-board CCS architecture” (which will become Subset-150), by Author Hartwig Schuster (Siemens Mobility) [42].

Overview and learnings

> This document describes the UNISIG considerations on the evolution of the on-board CCS subsystem as defined by the CCS TSI in close collaboration with UNIFE members from vehicle manufacturers, UNITEL and Shift2Rail CONNECTA.
>
> The evolution of the on-board CCS subsystem focusses on a system level, which is the level of functional building blocks, but not below (e.g. the component level (safe computing platforms, middleware, antennas etc.)). Therefore, the functional building blocks will be regarded as black boxes.
>
> Nevertheless, this concept will allow an efficient enhancement of the on-board CCS subsystem by creating a transparent platform for the communication (one common on-board CCS bus), which allows an open interface to the on-board CCS subsystem’s core, the ETCS on-board. Easier manageable integration will be an additional positive side-effect of this approach.
>
> Manageable integration comes along with modularity, i.e. the separation of elements of the on-board architecture, their independent specification and complete definition of the interfaces to other elements. Methods shall be put in place to make sure that the new element underwent the same quality assurance as the existing one, even if it is provided by different vendors.
>
> The individual elements can be more easily manageable as their functionality is limited and more specifically defined. Furthermore, the separation of elements according to their higher-order characteristics such as safety-related or non-safety related would make it possible to differentiate the methods on design, verification, validation, certification and authorization in a more efficient manner.
>
> The system design for the future on-board CCS architecture shall be based on the existing functional modules. Break-down of the system into individual elements shall be performed with care to avoid design trade-offs between safety, reliability, and responsibility at the specific application level.
>
> The definition of the standard network of the future system architecture shall also consider requirements on scalability to be future proof for functional enhancements and a better separation of safety-related from non-safety related elements.
>
> Scalability shall in the future be addressed by defining an ETCS on-board API, which means that future elements connected to the ETCS on-board can communicate based on the same bus technology with the application SW of the ETCS on-board. As soon as the SW of ETCS on-board provides the API for a specific enhancement, the related element providing the enhanced functionality can be integrated to the on-board CCS subsystem.

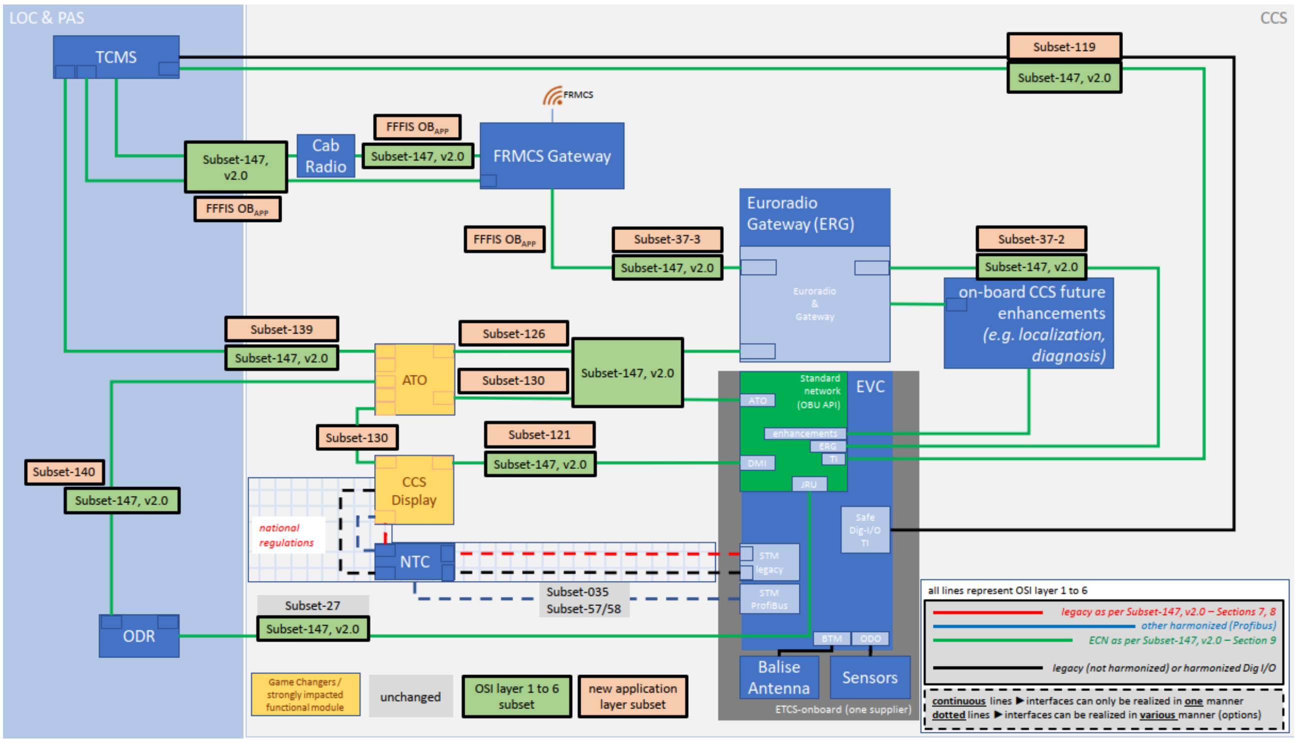


Figure 8: UNISIG Subset 150 Proposed long-term system architecture [42]

2.2.9 EULYNX

The following summary of EULYNX is extracted from an EULYNX general presentation (see [6], EULYNX documents are provided to interested parties under the "EU Public Licence" after registration).

EULYNX is an initiative of European railway infrastructure managers, defining an internationally standardised signalling system with a focus on common standardized interfaces, e.g. for signalling data, monitoring, diagnostics and maintenance. The defined modular signalling architecture aims to separate the lifecycles of the interlocking core and the field elements.

EULYNX is (together with EUG) one of the governing bodies of RCA, while in the scope of architecture, EULYNX is a building block of RCA. EULYNX covers today's architecture and has a goal to serve as preparation step towards RCA. Specifications developed withing RCA will reuse the model-based engineering process defined by EULYNX.

For the modular platform specifications in this work package, relevant previous work items can be found in the EULYNX "SCI Standard Interface", the "SDI Monitoring/Diagnosis" interface and the "SMI Maintenance" interface, and also in the field of IT security.

Key learnings from the partners in this work package are coming from hands on experience and learnings that will be worthwhile in the further progress.

2.3 Summary of Prior Works

The long list of prior work entries summarized in the previous chapter reflects the importance of modular compute platforms and the corresponding interface standardization. Many different initiatives driven by various stakeholders delivered potential solutions and learnings; comparable in some aspects, but different enough to include them all.

The analysis has shown that prior work usually had a focus on either trackside (e.g., SIL4 Data Center, SIL4@Cloud), onboard (e.g., OCORA) or even a concrete functionality (e.g., EULYNX). The stated goal of this work package is to treat onboard and trackside environments ideally within a single modular platform concept.

Also, with a certain focus on applications or field elements (e.g., in EULYNX), for example generic specifications for external interfaces needed for orchestration of multi-application modular computing platforms are not found in the prior work in a way that they could be immediately adopted in the work package. Similarly, not all prior works cover use cases or requirements towards potentially unique features of modular platforms, e.g., adding of functions to safe systems after deployment.

A common learning is the challenge of a multi-vendor project including certification and integration processes. These topics will be addressed in Task 26.3 of this work package, where a study on certification and acceptance methods for modular platforms will be performed.

Another crucial learning is the need to establish a common language for the definition of computing platforms and modular platforms in particular. All relevant terms for the description of these platforms need to be collected in a glossary and aligned with the System Pilar Computing Environment domain.

For the specification of modular platforms, as planned by this work package, the prior work has not shown a clear candidate that would be suitable as a basis for the future work. This does not mean that prior work will be ignored, on the contrary. In the work package’s next task, aligned with System Pillar Computing Environment domain work, the identified learnings for trackside and onboard will be incorporated.

3 POTENTIAL SPECIFICATION CANDIDATES

The work package's subsequent task will be to work on the actual specifications on modular platform. The following high-level interface model and potential specification candidates shall trigger discussions that eventually lead to detailed specification activities.

3.1 Assumed Structure of Platform Related Interfaces

For the following detailed analysis of prior work and derivation of the specific modular platform related specification work to be carried out in R2DATO, a particular structure of platform-related interfaces is assumed that is shown in Figure 9. This figure has been discussed with the System Pillar Computing Environment domain and was considered helpful in providing structure to the topic. As this is still work in progress, please note that the interface structure is still subject to change, as are the names and the scope of standardization.

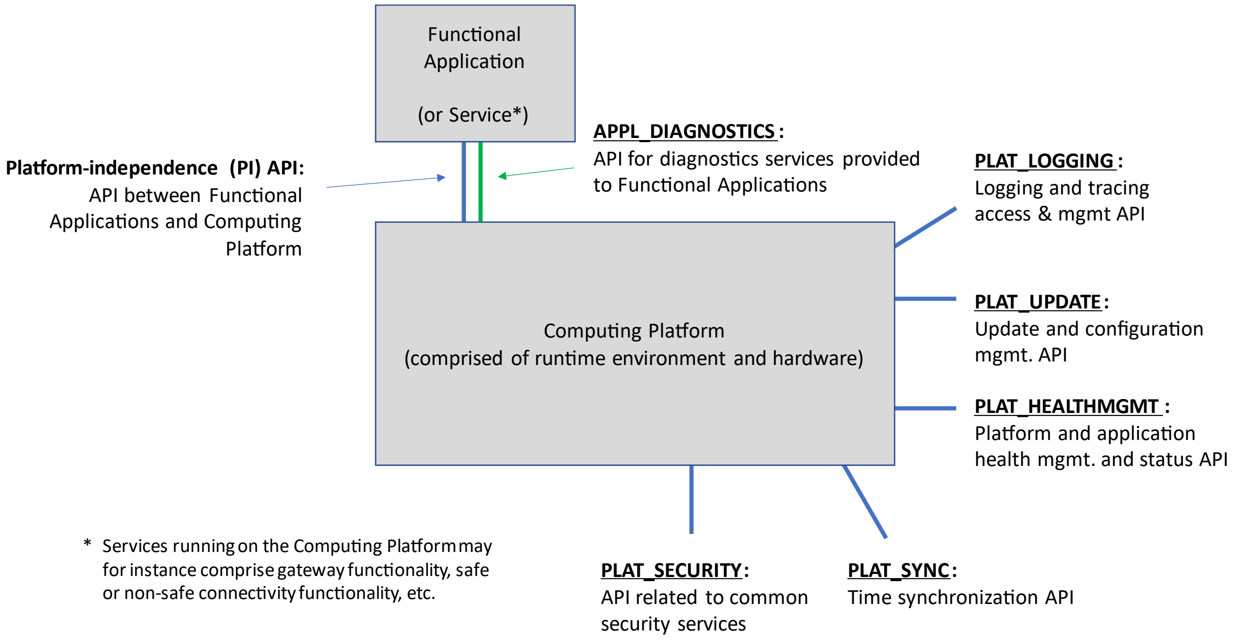

Functional Application
(or Service*)
APPL_DIAGNOSTICS:
API for diagnostics services provided to Functional Applications
Platform-independence (PI) API:
API between Functional Applications and Computing Platform
PLAT_LOGGING:
Logging and tracing access & mgmt API
Computing Platform
(comprised of runtime environment and hardware)
PLAT_UPDATE:
Update and configuration mgmt. API
PLAT_HEALTHMGMT:
Platform and application health mgmt. and status API
* Services running on the Computing Platform may for instance comprise gateway functionality, safe or non-safe connectivity functionality, etc.
PLAT_SECURITY:
API related to common security services
PLAT_SYNC:
Time synchronization API

Figure 9: Assumed structure of Computing Platform interfaces

The terms Functional Application and Computing Platform – as shown in Figure 9 – were agreed with the System Pillar Computing Environments domain [41] and are also provided in Table 2.

Term	Definition
Functional Application	A comprehensive set of self-contained software functions, assumed to be provided as one product by a single vendor. Functions within one application may have different functional safety requirements.
Computing Platform	Refers to an environment on which functional applications are run, comprised of hardware and software (i.e., the runtime environment).
Runtime Environment	A software that acts as intermediary by providing a generalized abstraction of the underlying hardware and software and enabling communication and data management for distributed applications.

Term	**Definition**
	Definition derived from RCA/OCORA (see [18]): An (instance of a) runtime environment, which comprises safety services (e.g., integrity checking, fault tolerance, synchronisation and communication services related to safety, hardware and software monitoring as needed in safety context) and system services (e.g., application lifecycle management, platform and software monitoring, tracing and logging, communication services that are not related to safety, security means incl. authentication, encryption, key storage, etc., provisioning and management of persistent storage) and the communication stack for information exchange between Functional Applications running on the same Platform and with external entities.
Hardware	The physical and electronic parts of a computer or other piece of equipment.

Table 2: Terms used in the context of specification work in work package 26

In the following, the platform related interfaces shown in Figure 9 are described in detail, being grouped into:

- Interfaces between Functional Applications and Computing Platform, see Chapter 3.2
- Interfaces for Interaction among External Entities and the Computing Platform, see Chapter 3.3

3.2 INTERFACES BETWEEN FUNCTIONAL APPLICATIONS AND COMPUTING PLATFORM

The interfaces listed in the coming subchapters are located between the Functional Applications and the Computing Platform.

3.2.1 Platform-independent API (PI API)

Functionality

The PI API allows a Functional Application from one vendor to run on a Computing Platform of another vendor, for instance offering messaging services for inter-Functional-Application communication. It also involves means through which the Functional Application can state whether it needs guaranteed resources provided by the Computing Platform or is only "best effort".

Previous Work

An initial specification of a possible platform-independent API between Functional Applications and Computing Platform was developed in RCA and OCORA in collaboration with 6 suppliers [20]. Key aspects covered in this specification are listed in Table 3:

Aspect	**Description**	**Evaluation from perspective of R2DATO**
Terminology	A detailed terminology related to elements of Functional Applications and Computing	It may be helpful for R2DATO to reuse this terminology.

Aspect	**Description**	**Evaluation from perspective of R2DATO**
	Platforms is provided and complimented with an entity relationship diagram and illustrations of possible deployment options.	
Key Design Paradigms and Guiding Principles	Key design paradigms are listed that have a fundamental impact on the interface design, for instance that Functional Actors (subsets of Functional Applications) should not be aware that there are run in redundant replicas by the Computing Platform, as fault tolerance mechanisms like composite fail safety are transparent to the Applications.	It may be helpful for R2DATO to build upon these key design paradigms and guiding principles.
Messaging	The initial specification elaborates in detail on a potential messaging framework through which Functional Applications could send and receive messages from and to other Functional Applications, including notions of quality of services.	It is suggested that at least the R2DATO on-board platform demonstrator implements and tests the proposed messaging framework.
Execution behaviour	It is elaborated on how Functional Applications can get deterministic CPU resources, e.g. in regular intervals, event based or best-effort.	It is suggested that at least the R2DATO on-board platform demonstrator makes use of the considerations regarding execution behaviour.
Gateway concepts	A specific proposal is developed on how service functions related to safe or non-safe communication protocols could also be hosted on a Safe Computing Platform over a standardized API, and how the end-to-end communication flow could look like.	It is suggested that at least the R2DATO on-board platform demonstrator makes use of the considerations regarding gateway concepts.
Fault, error and failure handling	The initial specification defines and exemplifies the stated terms in the context of modular platforms and takes position which behaviour should possibly be standardized.	It is suggested that R2DATO further expands on this work.
API considerations	The initial specification elaborates on how POSIX could be used as a basis for a standardized separation of application and platform (with a distinction between functions suitable for up to SIL4 or for non-SIL levels), and proposes specific extensions.	It is suggested that at least the R2DATO on-board platform demonstrator makes use of and expands on the API considerations.

Table 3: Previous work on Platform Independent API

Gaps

- Fault, error and failure handling should be further detailed
- Juridical recording of interactions among Functional Application and Computing Platform should be specified
- Further interface for the PLAT_ interfaces to external entities that are not yet part of the prior work
- generic SRACs (as they are currently depending on the RTE solution)
- Bundling of applications for time critical communication (e.g. for request/response mechanism)
- Communication between different RTE solutions

Topics to be addressed in Task 26.2

- Fault, error and failure handling should be further detailed
- Juridical recording of interactions among Functional Application and Computing Platform should be specified
- Existing PI API specification elements should be subselected for evaluation in the On-Board Platform Demonstrator
- Investigate potential SRACs harmonization (also relevant for Task 26.3)
- Further investigation of bundling of applications for time critical communication
- Further investigation of future safe communication protocols and their relation to the computing platform

3.2.2 Application Diagnostics API (APPL_DIAGNOSTICS)

Allows Functional Applications to store diagnostics information or obtain diagnostics information from other Functional Applications of Platform Components. Furthermore, the provision of dedicated interfaces and corresponding services to provide information on the availability and performance of hosted applications, mainly for the purpose of health assessment and remote diagnostics seems possible in this context.

Functionality

The main purpose and functionality of APPL_DIAGNOSTICS is the provision of services for monitoring, diagnostics (and optionally also configuration) of hosted system functions. The envisioned stakeholders and users of APPL_DIAGNOSTICS are operators, integrators and vendors.

Previous Work

- OCORA MDCM-OB, building block of CCS-OB reference architecture
    - See MDCM Introduction [22] and MDCM SRS [23]
    - See MDCM Introduction, ch. 3.1, for particular needs regarding CCS-OB diagnostics
    - See MDCM Introduction, ch. 3.2, for a brief overview of SUBSET-149 (OMS)
    - See MDCM Introduction, ch. 3.4.1.1, for an overview of current recommendations and best practices for on-board diagnostics

- EULYNX MDM, Maintenance and Data Management subsystem
  - See [7] (Maintenance and data management specification Eu.Doc.18)
  - The MDM provides diagnostic information for trackside assets in scope of digital interlockings and compliant to to Eulynx specifications
  - Trackside assests in scope of the MDM are, amongst others: Points, Level Crossings (LC), Train Detection System (TDS), Electronic Interlocking systems, I/O Controllers (Generic I/O)
  - Furthermore, the MDM provides means for configuration and SW updates

Gaps

Today, there are no normative standards for diagnostics interfaces, services and datasets.

3.3 Interfaces for Interaction among External Entities and a Computing Platform

In addition to the interface between Functional Applications and the Computing Platform described in the previous chapter, interfaces for outside access to the Computing Platform are needed for a complete system. The scope of outside access to the platform is ranging from other devices or computing systems in the same network, train or data centre and includes up to over-the-air access to a remote, on-board system.

The following sub chapters give a brief overview over potential candidates for further study in Task 26.2, but of course it is not meant to be limiting Task 26.2 activities. Where known, previous work and potential gaps to be addressed in this work package are listed.

Regarding the suitability of previous work, the detailed analysis is not in the scope of this deliverable and task and will be discussed in future work. However, some general criteria are collected in the following, non-exhaustive, list.

- Flexibility to be integrated into a modular platform concept.
  This criterion would exclude narrow interface specifications that can, for example, only handle a single type of specific configuration or device.
- Open and non-discriminatory access to the specifiation details.
- Not limited to a special kind of platform technology or architecture.
- Ideally scalable over the full range of targets the “Modular Platform Specifications” is concerned with, both onboard and trackside.
- Ideally proven in use, potentially also within another sector.
- Suitable in the IT/OT security context of critial infrastructure and/or onboard.
- Does not prevent future development and ongoing maintenance.
- Where needed, the appropriate server and client side specifications and reference implementations should be available.
- Ideally, test definitions and compliance test suites exist.

3.3.1 Logging and Tracing Access & Management API (PLAT_LOGGING)

The Logging and Tracing Access & Management API allows an external entity to a Computing Platform to access log and trace data (from platform and/or applications) that is stored in the platform, to access juridically hardened log and trace data (from platform and/or applications) stored in the platform, and set log and trace levels.

Previous Work

- Juridical logging has its own subset (Subset-27)

Topics/Gaps to be addressed in WP26

- General purpose logging
- Suitability of Subset 27 in this context

3.3.2 Update & Configuration Management API (PLAT_UPDATE)

The Update & Configuration Management API allows an external entity (possible over-the-air) to install new and upgrade existing software components (related to application and platform), as well as to activate or deactivate software components (both application and platform) and to activate or deactivate hardware components.

Previous Work

- Discussion paper from OCORA was published (see [21])
- EULYNX [6] with a focus on field elements, such as object controllers, that need configuration and software updates

Topics/Gaps to be addressed in WP26

Currently part of ERJU “Transversal CCS components”, to be discussed during Task 26.2 how we can get relevant input from the transversal domain into the computing environment domain, to serve as an input for the discussion in this work package.

3.3.3 Platform & Application Health Management and Status API (PLAT_HEALTHMGMT)

The Platform & Application Health Management API allows an external entity to obtain health information about Functional Applications and Platform components and to define actions to be taken in case of faults, errors, failures (e.g. automated restart of a Functional Application or Platform component).

Additional functionality (e.g. statistics) and if this interface might get merged with PLAT_UPDATE (see Chapter 3.3.2) is going to be discussed in the future.

Previous Work

None known for railway use cases, to be researched in Task 26.2.

Topics/Gaps to be addressed in WP26

General research for suitable prior work in other relevant industries with adequate safety and certification requirements.

3.3.4 Time Synchronization API (PLAT_SYNC)

The Time Synchronization API allows a Computing Platform to synchronize to an external clock (or any other synchronization means that may be needed).

Previous Work

- IEEE-1588 for general purpose computing [29]

Topics/Gaps to be addressed in WP26

- Safety context suitability of previous work.
- Additional prior work research for suitable railway grade solutions.

3.3.5 IT Security API (PLAT_SECURITY)

The IT Security API provides an interface to common security services such as key management, certificate management, etc.

Previous Work

None known for railway use cases, to be researched in Task 26.2.

Topics/Gaps to be addressed in WP26

Security activities to be synchronized with PRAMSS activities in ERJU System Pillar.

4 AREAS FOR POSSIBLE SPECIFICATION WORK AND NEXT STEPS IN R2DATO WP26

In this chapter, a list of open topics is collected based on the previous chapters is show. Next, the alignment and delineation between System Pillar and R2DATO with respect to modular platform topics is discussed and an outlook to the next task is given. However, this is not to be seen as limiting for subsequent activities and can be expanded based on ongoing alignment activities.

4.1 List of Opens and potential Areas of Activity

The following table captures the current state of ongoing alignment discussions. Therefore, it is not final and not limiting the breadth of activities in this work package.

Area	**Potential Activity**
Platform Independent API (PI API)	Address gaps listed in Chapter 3.2.1.
Application Diagnostics API (APPL_DIAGNOSTICS)	Address gaps listed in Chapter 3.2.2.
Logging and Tracing Access & Management API (PLAT_LOGGING)	Address gaps listed in Chapter 3.3.1.
Update & Configuration Management API (PLAT_UPDATE)	Address gaps listed in Chapter 3.3.2.
Platform & Application Health Management and Status API (PLAT_HEALTHMGMT)	Address gaps listed in Chapter 3.3.3.
Time Synchronization API (PLAT_SYNC)	Address gaps listed in Chapter 3.3.4.
IT Security API (PLAT_SECURITY)	Address gaps listed in Chapter 3.3.5.

Table 4: List of potential activities for Task 26.2

4.2 Alignment and Delineation between System Pillar and R2DATO

For the specification work expected to take place in Task 26.2, the following delineation and alignment between the related System Pillar domains and R2DATO is assumed.

Computing Environment Domain

The System Pillar "**Computing Environment**" domain, which is closest related to R2DATO WP 26, aims, according to its remit, to determine and analyse key use cases and operational scenarios related to computing environments, assess these w.r.t. feasability, derive operational and system capabilities, and provide guidance on which interfaces in the context of computing environments would be most relevant to be standardized. As the domain does not aim to produce specifications itself, the foreseen activities in R2DATO Task 26.2 well complement the activities in the System Pillar. Obviously, it is important that the use cases, operational and system capabilities identified in the System Pillar domain, same as the guidance on interfaces to be standardized, are taken well into account in Task 26.2. This should be possible if the input from the System Pillar domain is provided according to the timeline in Figure 10.

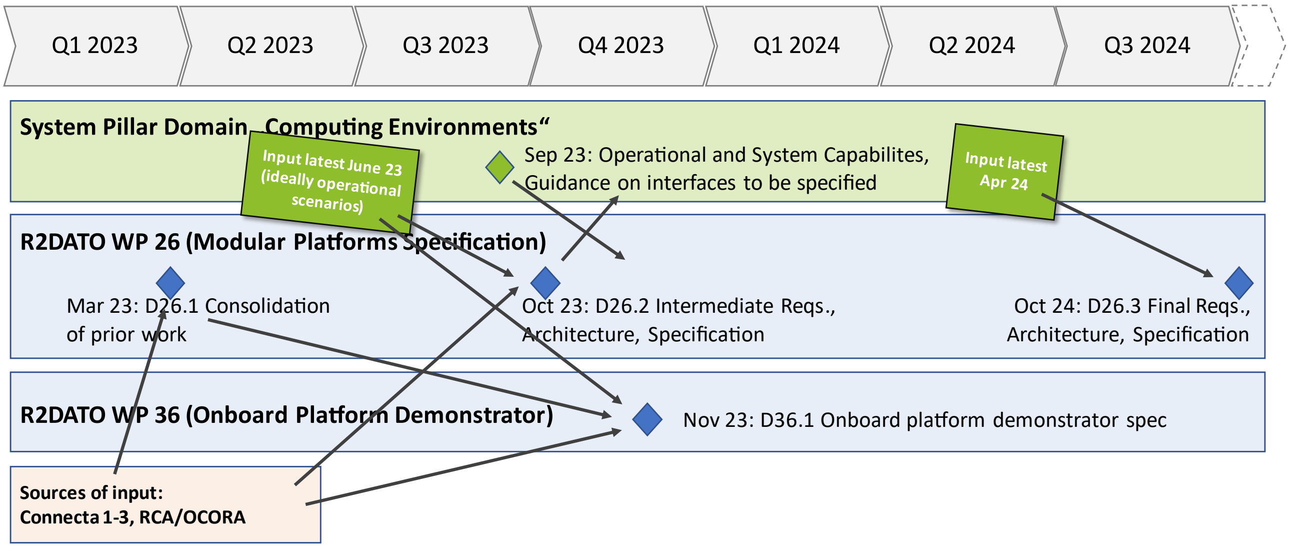

Q1 2023
Q2 2023
Q3 2023
Q4 2023
Q1 2024
Q2 2024
Q3 2024
System Pillar Domain „Computing Environments"
Input latest June 23 (ideally operational scenarios)
Sep 23: Operational and System Capabilites, Guidance on interfaces to be specified
Input latest Apr 24
R2DATO WP 26 (Modular Platforms Specification)
Mar 23: D26.1 Consolidation of prior work
Oct 23: D26.2 Intermediate Reqs., Architecture, Specification
Oct 24: D26.3 Final Reqs., Architecture, Specification
R2DATO WP 36 (Onboard Platform Demonstrator)
Nov 23: D36.1 Onboard platform demonstrator spec
Sources of input:
Connecta 1-3, RCA/OCORA

Figure 10: Proposed alignment of System Pillar Computing Environments Domain, R2DATO WP26, and R2DATO WP36

Transversal CCS Components

The System Pillar domain “Transversal CCS Components” also has some relation to Task 26.2 in the way that it investigates diagnostics and configuration related interfaces. However, according to its remit, the domain mainly focuses on interfaces to collect asset condition data from CCS-external asset management systems and on configuration of application components. The planned specification work in R2DATO Task 26.2 on update and configuration management (related to both platform and application, see Chapter 3.3.2) and that on platform and application health management (see Chapter 3.3.3) hence goes well beyond the envisioned activities in the System Pillar domain. Obviously, a tight alignment of the activities is important, which is expected to be achieved through the same individuals being active on both System and Innovation Pillar side, and through the notion of dedicated System Engineers in R2DATO.

4.3 Outlook to Task 26.2

In particular to achieve the aforementioned close alignment with the System Pillar, the following steps are envisioned for work package 26.

In the beginning of this work package’s subsequent Task 26.2, a gap analysis will be created, taking the prior work collected in this document and their respective learnings, as well as the System Pillar’s viewpoint, as an input. Based on this and the ongoing alignment with other ERJU bodies, the actual structure, naming, requirements and architecture of the interfaces and specifications for modular platforms will be discussed and defined.

5 CONCLUSIONS

In this first and final deliverable of the first task in this work package, a consolidation of prior work, their relevant learnings, and an outlook on the future specification work for modular platforms have been presented.

Especially the ERJU's predecessor project, Shift2Rail, delivered a lot of content relevant to modular platforms, with a long list of subprojects in different areas. Additionally, the open access initiatives RCA and OCORA provide a lot of valuable insights and concepts. Further operator and industry partner collaborations which are noteworthy are the SIL4 Cloud and SIL4 Data Center reports. Current activities in UNISIG have also been evaluated.

A key learning with respect to this work package's goal, creating a unified specification for modular computing platforms for onboard and trackside, is that previous work has not yet fully attempted this unification. There are some notable exemptions, however, such as the OCORA PI-API work in an onboard context that has also been made available in the RCA trackside context [20]. Nevertheless, besides these specialized interfaces, a generalized and common architecture and approach for these unified railway-centric modular platforms is not yet known.

Furthermore, the analysis of the previous work has highlighted a common issue: The development and consequent usage of an aligned vocabulary and glossary is essential for comparison, discussion, and development of technologies among the many stakeholders encountered in the railway sector. As such, aligning this glossary will become essential for Task 2.

While there is some discussion available in previous work around the challenges around certification of modular platforms, the level of detail and, again, a unified approach for trackside and onboard systems will need further research, as planned in Task 3.

Potential specification areas and candidates have been identified in a first, coarse approach using a simplified architecture model. The structure assumed here gave some initial needs for interfaces and specifications, already integrating concepts from previous work, e.g., the aforementioned PI-API. For the operation and maintenance of an actual modular platform instantiation, e.g., the physical device with the appropriate software executing relevant functions, external interfaces are necessary. These external interfaces are explicitly only in the scope of the platform and not to be used by the functions implementing business logic themselves. Some general criteria have been identified, as well as previous work and its gaps, to guide the selection of interface specifications in Task 2.

The next task (Task 26.2) will focus on the actual specifications needed for modular platforms and produce two deliverables, one intermediate and one final. Most notably, this output will be consumed by work package 36 (On-Board Platform Demonstrator), by the ERJU System Pillar Computing Environment Domain and subsequent ERJU Innovation Pillar activities. Furthermore, the work package partners' experience with regards to actual EULYNX project implementation, integration and certification will be a strong basis for future specification work and allows prioritization of certain areas. As such, the work package's final task, Task 26.3, will focus on a study on modular certification and acceptance approaches for modular platforms.

Work package 26 will continue to closely align with ERJU's System Pillar Computing Environment Domain and the Transversal CSS Components Domain to deliver modular platform specifications that are maximally in-line with System Pillar requirements and priorities.

REFERENCES

[1] Shift2Rail website
https://rail-research.europa.eu/about-shift2rail/

[2] CONNECTA-2
https://projects.shift2rail.org/s2r_ip1_n.aspx?p=CONNECTA-2
https://cordis.europa.eu/project/id/826098

[3] Safe4Rail 1+2
https://projects.shift2rail.org/s2r_ip1_n.aspx?p=SAFE4RAIL
https://projects.shift2rail.org/s2r_ip1_n.aspx?p=SAFE4RAIL-2

[4] RCA List of published releases
https://public.3.basecamp.com/p/KeehzqFmXv5R2N7tGDjaEokq

[5] OCORA website
https://github.com/OCORA-Public/Publications

[6] EULYNX website
https://eulynx.eu/

[7] EULYNX List of published documents
https://eulynx.eu/index.php/documents/published-documents

[8] SIL4@Cloud Report
https://digitale-schiene-deutschland.de/Downloads/Report%20-%20SIL4%20Cloud.pdf

[9] SIL4 Data Center Report
https://digitale-schiene-deutschland.de/Downloads/Research%20Report%20-%20SIL4%20Data%20Center.pdf

[10] SIL4 Data Center “Signal+Draht” article
https://digitale-schiene-deutschland.de/Downloads/Signal%26Draht_113_10_21_SIL4DataCenter.pdf

[11] RCA Architecture
https://ertms.be/workgroups/ccs_architecture

[12] RCA Overview
https://public.3.basecamp.com/p/xZcPn1eH54AFTUWf76HNndmN

[13] RCA Whitepaper
https://eulynx.eu/index.php/news/37-reference-ccs-architecture-white-paper

[14] RCA Baselines
https://public.3.basecamp.com/p/jGh4E3ZdE8T1RtoxvbWLCYss

[15] RCA Architecture Poster
https://public.3.basecamp.com/p/xZcPn1eH54AFTUWf76HNndmN

[16] D8.4 – Final report on the contribution of CONNECTA-2 to Shift2Rail
https://projects.shift2rail.org/download.aspx?id=1adf1159-1c27-41d1-9401-0c4174dc3e34

[17] CONNECTA-2, Safe4Rail-2 Learnings
https://projects.shift2rail.org/download.aspx?id=a1ea7b2f-5a06-44dc-b06e-354f9d15d532

[18] Computing Platform – Whitepaper:
https://github.com/OCORA-Public/Publications/blob/master/00_OCORA%20Latest%20Publications/Latest%20Release/OCORA-TWS03-010_Computing-Platform-Whitepaper.pdf

[19] Computing Platform – Requirements:
https://github.com/OCORA-Public/Publications/blob/master/00_OCORA%20Latest%20Publications/Latest%20Release/OCORA-TWS03-020_Computing-Platform-Requirements.pdf

[20] Computing Platform – Specification of the PI API between Application and Platform:
https://github.com/OCORA-Public/Publications/blob/master/00_OCORA%20Latest%20Publications/Latest%20Release/OCORA-TWS03-030_SCP_Specification_of_the_PI_API_between_Application_and_Platform.pdf

[21] OCORA Discussion paper about Configuration and Updates
https://github.com/OCORA-Public/Publications/blob/master/08_OCORA%20Release%20R3/OCORA-TWS07-060_Configuration%20Management-Concept.pdf

[22] OCORA-TWS08-010 MDCM Introduction
https://github.com/OCORA-Public/Publications/blob/master/08_OCORA Release R3/OCORA-TWS08-010_MDCM-Introduction.pdf

[23] OCORA-TWS08-030 MDCM SRS
https://github.com/OCORA-Public/Publications/blob/master/08_OCORA Release R3/OCORA-TWS08-030_MDCM-SRS.pdf

[24] OCORA-TWS01-035 CCS On-Board Architecture
https://github.com/OCORA-Public/Publications/blob/master/00_OCORA%20Latest%20Publications/Latest%20Release/OCORA-TWS01-035_CCS-On-Board-(CCS-OB)-Architecture.pdf

[25] OCORA-BWS02-030 Technical Slide Deck
https://github.com/OCORA-Public/Publications/blob/master/08_OCORA%20Release%20R3/OCORA-BWS02-030_Technical-Slide-Deck.pdf

[26] BAV Mitteilung Juli 2020:
https://www.bav.admin.ch/bav/de/home/publikationen/bav-news/ausgaben-2020/bav-news-juli-2020/artikel-3.html

[27] Proposed architecture for Subset 026 of new TSI (potentially 2023)

[28] EU Agency for Railways (ERA) Subset 026 (ETCS B3 R2 GSM-R B1)
https://www.era.europa.eu/era-folder/set-specifications-3-etcs-b3-r2-gsm-r-b1
https://www.era.europa.eu/system/files/2023-01/sos3_index004_-_subset-026_v360.zip

[29] IEEE-1588
https://standards.ieee.org/ieee/1588/4355/

[30] X2Rail TD 2.1 - Adaptable communications for all railways https://projects.shift2rail.org/s2r_ip_TD_r.aspx?ip=2&td=0ac388f4-f808-484c-9f5e-432447324e36

[31] X2Rail TD 2.2 - Automatic Train Operations – ATO (up to GoA4) https://projects.shift2rail.org/s2r_ip_TD_r.aspx?ip=2&td=b47388a9-b1f8-4ed8-9872-bb7708f7c08d

[32] X2Rail TD 2.3 - Moving Blocks https://projects.shift2rail.org/s2r_ip_TD_r.aspx?ip=2&td=990df592-cb2c-4498-99db-74ddc5aeb147

[33] X2Rail TD 2.4 - Fail-Safe Train Positioning https://projects.shift2rail.org/s2r_ip_TD_r.aspx?ip=2&td=21651fb7-7047-4093-932a-79f4ca9c9652

[34] X2Rail TD 2.5 - On-board Train Integrity https://projects.shift2rail.org/s2r_ip_TD_r.aspx?ip=2&td=061d0fcf-51a6-4358-a74f-a4d34e8dac01

[35] X2Rail TD 2.6 - Zero on-site testing https://projects.shift2rail.org/s2r_ip_TD_r.aspx?ip=2&td=983bce09-d662-47ac-904c-a543f5b73fcc

[36] X2Rail TD 2.7 - Formal methods and standardisation for smart signalling systems https://projects.shift2rail.org/s2r_ip_TD_r.aspx?ip=2&td=0f5a0853-1523-414d-bcb5-5aa91557a3b6

[37] X2Rail TD 2.8 - Virtually – Coupled Train Sets https://projects.shift2rail.org/s2r_ip_TD_r.aspx?ip=2&td=aa1f0995-1402-4e7b-a905-49fba1bc6f71

[38] X2Rail TD 2.9 - Traffic management system https://projects.shift2rail.org/s2r_ip_TD_r.aspx?ip=2&td=a86b26b5-2680-4765-9bc6-c935804d6aa6

[39] X2Rail TD 2.10 - Smart radio-connected all-in-all wayside objects https://projects.shift2rail.org/s2r_ip_TD_r.aspx?ip=2&td=7399937f-ffaa-4d65-b8ee-95007156a154

[40] X2Rail TD 2.11 - Cyber Security https://projects.shift2rail.org/s2r_ip_TD_r.aspx?ip=2&td=1bccaf76-7915-4283-8b52-36afdb1bdf42

[41] ERJU System Pillar – Computation Environment Domain https://rail-research.europa.eu/system_pillar/

[42] UNISIG Subset-150 proposal, not yet published. Courtesy of Hartwig Schuster, Siemens Mobility GmbH. https://www.ertms.net/about-ertms/about-unsig/