RESEARCH REPORT

SIL4 CLOUD

Digitale Schiene Deutschland

A REPORT BY THALES, SYSGO, FRAUNHOFER IESE,
UNIVERSITY of ROSTOCK, ESE AND DB Netz

THALES SYSGO EMBEDDING INNOVATIONS Fraunhofer IESE Universität Rostock

ESE Engineering und Software-Entwicklung Digitale Schiene Deutschland DB NETZE

List of Authors and Contributors

Authors	Rasmus Adler	Fraunhofer Institut für Experimentelles Software Engineering IESE
	Denis Uecker	Fraunhofer Institut für Experimentelles Software Engineering IESE
	Reinhard Hametner	THALES
	Stefan Resch	THALES
	Wolfgang Wernhart	THALES
	Zeeshan Ansar	SYSGO GmbH
	Holger Blasum	SYSGO GmbH
	Don Kuzhiyelil	SYSGO GmbH
	Frank Golatowski	University of Rostock, Germany
	Benjamin Rother	University of Rostock, Germany
	Frank Breitschaft	DB Systel GmbH
	Arthur Martens	ESE GmbH
	Ulrich Velte	ESE GmbH
	Rouven Scholz	ESE GmbH
	Oliver Guenther	ESE GmbH
	Prashant Pathak	DB Netz AG
	Julian Wissmann	DB Netz AG
	Alexander Heine	DB Netz AG
	Patrick Marsch	DB Netz AG
Contributors	Oliver Mayer-Buschmann	DB Netz AG

Jens Franke	DB Netz AG
Andreas Jelen	DB Netz AG
Reza Anbarestani	DB Netz AG
Frank Eschmann	DB Systel GmbH
Mario Brotz	SYSGO GmbH

Version History

Version	Date	Changes
1.0	14-Sep-2022	First published version

Table of Content

1 Introduction

Less traffic, less congestion, less particulate matter – and more people and more goods on the rails: The rail sector in Europe is on the verge of a technological leap into the digital future. The sector initiative "Digitale Schiene Deutschland" [1] is taking advantage of this opportunity and bringing future technologies into the rail system. This benefits not only passengers, but also the climate and Germany as a business location. And all this without having to construct a single new track.

The foundation for this is being laid with the fundamental modernisation and digitalisation of the infrastructure through the consistent introduction of digital control and safety technology. To achieve this, DB is working with other European Railways and industry partners on a far-reaching digitalisation of the railway system. This includes a system architecture that details the tasks of individual components of the railway system and how they should work together.

On this basis, numerous digital technologies will then be tested and further developed for use in the system: for example, an AI-based traffic and incident management system will provide intelligent and automated control of trains in the future. These will then run fully automatically and at an optimal distance from each other. The latest sensor technology for environment perception, coupled with high-precision train location and an automated interruption detection, are further technologies that will play an important role in the digitalisation of the railway system. Overall, a significant improvement in capacity, reliability and efficiency of the railway system will be achieved, all of which are requirements for more traffic on the railway and a strengthening of the railway as the climate friendly mode of transport of the future. On this foundation, numerous digital technologies will then be tested and further developed for use in the system e.g.

- an AI-based traffic management system to schedule timetables
- a fully automated rail operation up to GoA4
- a novel architecture for Command and Control Systems (CCS) that enables train operation with minimal distance through an ETCS Level 3 moving block approach
- intelligent environment perception to identify obstacles
- fully automated incidence prevention, mitigation, and resolution

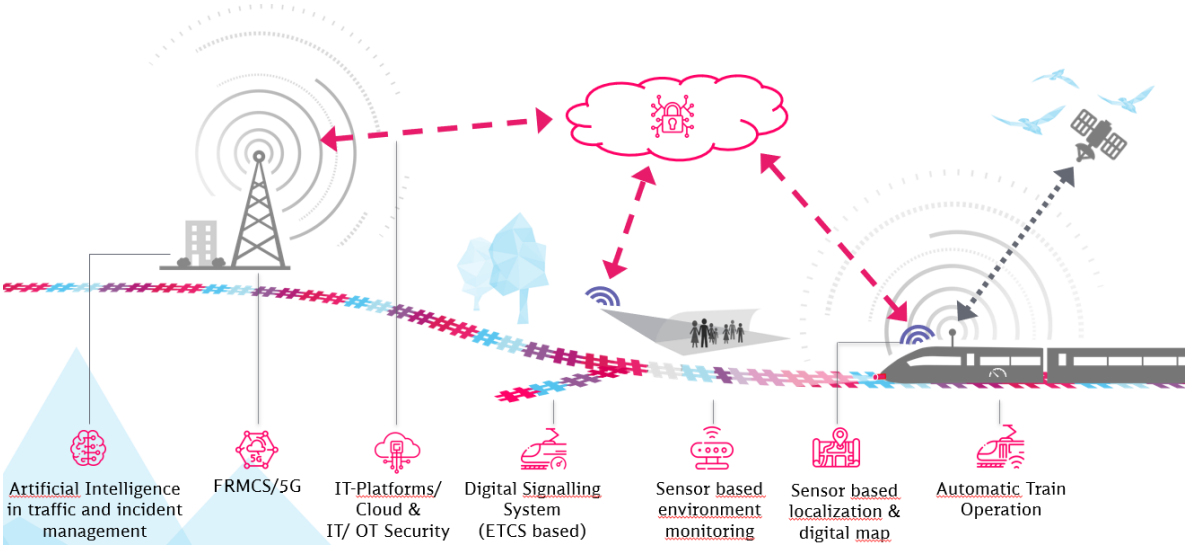

Figure 1: Essential technologies needed for future rail operation

A key prerequisite for the envisioned digitalisation of rail operations is a highly performant IT infrastructure that allows to decouple the different life cycles of railway applications, middleware and hardware, and explicitly leverages the latest developments in the IT sector regarding, e.g., virtualisation and orchestration for safety-critical applications, while still complying with railway norms such as the CENELEC EN 50126 and following, or IEC 62443.

1.1 Motivation of DB

For trackside subsystems, DB plans to design and implement a on-premise private Cloud[1] infrastructure for different types of digital subsystems required for future rail operations, particularly safety-relevant applications such as a further evolved ETCS Level 3 moving block approach. A key aim of DB is to base future rail operations on Commercial off-the-shelf (COTS) Cloud technologies, to the extent that this is possible, which are tailored well to the specific needs of safety-relevant railway applications (up to SIL4) and ultimately compliant with related railway norms. Further, railway applications may have to be designed differently or adapted to run in COTS Cloud environments, and there may be some impact on distinct life cycles, integration, portability and homologation processes related to the more complex vendor setup expected. DB plans to have future-proof computing platforms, along with support for legacy systems and lowered total cost of ownership (TCO) over longer lifetime (e.g., 20+ years). All the points stated above are considered necessary and need to be discussed in detail with industry, academia, and research partners in the current early stage of development.

1.2 Motivation of partners

Keeping the importance of railway vision in mind, Thales, SYSGO, Fraunhofer IESE, University of Rostock, ESE and DB have collaborated on this research project to identify potential challenges and solutions at an early stage. While for DB the necessity and pursuit of a Safe Computing Platform (SCP) [2] approach with a modular separation of functional application, middleware and hardware is unquestionable, it is clear that this comes with significant challenges. It is hence essential for railways, suppliers and industry experts to collaborate at an early conceptual stage in order to:

- determine if there are any potential showstoppers regarding the vision of the railways
- obtain an early understanding of the requirements along with the associated effort
- analyse how the legacy platform solutions of suppliers can possibly be adjusted so as to maximally leverage standardized developments of suppliers, for instance Platform Independent Application Programming Interface (PI API), Safety Related Application Conditions (SRACs)
- conceptualise migration strategies from today's platform approaches to the target architecture

As an existing CCS vendor, domain expert and SIL4 platform provider, it is important for **Thales** to collaborate with railways to analyse new needs and share experiences on potential showstoppers such as gaps in technical expectations, migration, and so on. **SYSGO**, as a vendor of certified embedded system and virtualization platforms for the avionics and automotive sectors, brings safety-critical system expertise to develop a SCP that runs up to SIL4 railway applications in data centres.

Furthermore, the architecture of an SCP realization based on a virtualisation and separation kernel for developing safe railway applications in the data centre environment will be presented. Thales and SYSGO also evaluates the requirements of the SIL4 Cloud and defines the design, development and verification processes followed in the SIL4 software construction. **Fraunhofer IESE** brings experience from different application domains as well as engineering disciplines. Activities such as research to find the gaps, develop solutions and transfer them into practice require close cooperation with industry partners, who have the knowledge of the application domain and the targeted use cases. Fraunhofer's special focus in this cooperation was on the introduction and research of modular safety concepts for SIL4 Cloud. **University of Rostock** shares its computing experiences on communication protocols and analyses modern protocols applicable for SIL4 Cloud. For **ESE,** as an independent safety assessor for safety-critical systems in railway, it is important to identify major changes and challenges posed by new and innovative platform designs, in particular on homologation and IT/OT Security aspects. Through this project, DB has collaborated with vendors and domain experts from different

[1] The term cloud loosely means the platform of servers used to run software infrastructure which is in full control of DB, and that can take many forms such as Infrastructure as a Service, Platform as a Service.

backgrounds to analyse trackside Safe Computing Platform needs and potential challenges the future may pose on overall architecture. Though it is envisioned to have a common platform design paradigm, this particular project was focused on infrastructure-relevant subsystem hosting in private cloud environments.

1.3 Other Initiatives

Like DB, other railways in Europe are also aiming to introduce a larger degree of digitalisation in rail operation. In this context, the railway initiatives Reference Control Command and Signalling Architecture (RCA) and Open Control Command and Signalling Onboard Reference Architecture (OCORA) are driving a functional architecture for, respectively, the trackside and onboard functions for future rail operation from scratch. More precisely,

- RCA [3] is an initiative by the members of EUG [4] and EULYNX [5] to define a harmonised architecture for the future railway control command and signalling (CCS), including a definition of components and interfaces among these, with the main goal of substantially increasing the ratio between performance and total cost of ownership (TCO) compared to today's implementations.
- OCORA [6] is first and foremost a platform for cooperation to the benefit of the European railway sector. Recognising that a coherent, modular, upgradeable, interchangeable, reliable, and secure onboard architecture is paramount to overcoming the challenges of the overall CCS system, the intent is to establish the OCORA onboard architecture in coherence with and complementarily to the trackside control command and signalling.

As a first step, the railway initiatives RCA and OCORA have consolidated their key objectives and requirements and developed a first high-level concept for a so-called SCP approach and requirements list. This is envisioned to provide the conceptual basis for both trackside and onboard computing platform deployments for safety-critical railway applications.

2 Requirements

A SIL4 Cloud will have to fulfil requirements and constraints from many different stakeholders and sources. These include legislative norms on safety, standards from European Railway standardization groups for interoperability, data security, and organizational objectives. Figure 2 summarizes the flow of information in the form of requirements.

The image below represents a non-exhaustive list of standards and additional standards may apply e.g., EMC standards, Sektorleitlinie (Annex 5, Chapter 1.3, the code of practices (a.R.d.T), technical regulations need to be complied with).

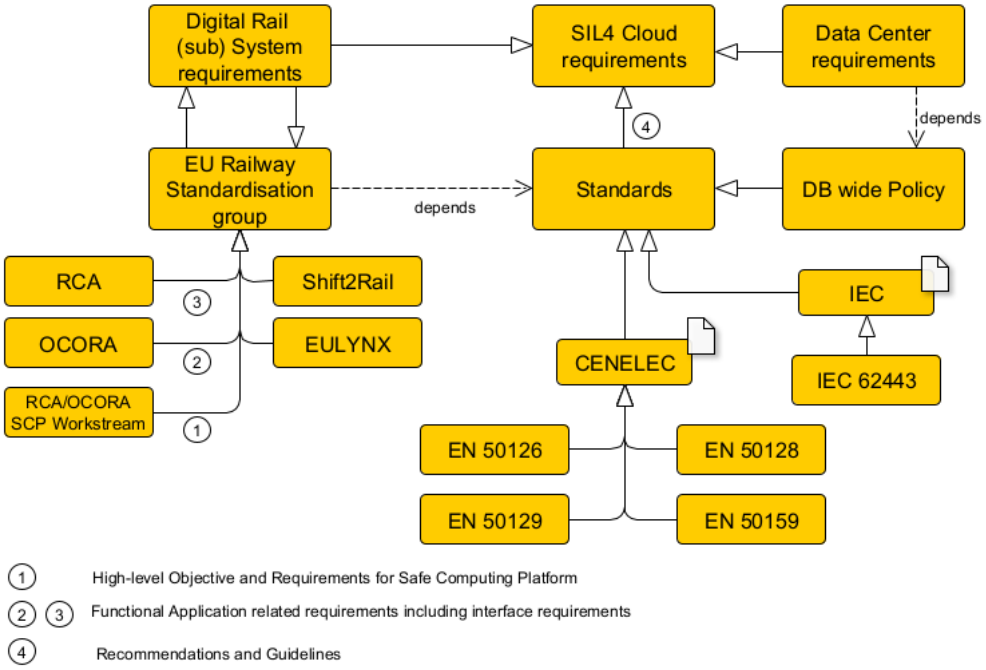

Figure 2: Overview of SIL4 Cloud requirements' sources

The following section summarizes the key requirements that need to be considered for the SIL4 Cloud. In Section 2.6 a usage scenario for the SIL4 Cloud is described, which will be taken as guideline in the design process and for architectural decisions.

The following table presents DB organizational goals and their relevance to computing infrastructure and associated requirements.

Table 1: High level goals of DB for computing platforms

DB Goals [1]	Ways to achieve goals
Reduce carbon emissions [7] by shifting traffic to rail • By avoiding congestion and braking/acceleration • Shifting passengers to rail means more traffic demands	Minimise quantity of computing infrastructure while in operation to reduce power consumption, heat generated, etc. Also, less infrastructure would lead to less future e-waste. *Possible approach could be to host higher number of Functional Applications on lesser number of hardware components (support for mixed criticality, usage of virtualization, hardware independence)*
Improve train punctuality	High-performing computing infrastructure for applications (Platform evolvability)

Increase passenger capacity by increasing number of trains (per distance, per unit time)	Improved performance while keeping the safety level, higher dependability of computing infrastructure to ensure safe travel (safety)
Improve availability of railway operation services	Higher guarantees of reliability and availability factors of underlying computing and connectivity infrastructures. (Reliability, availability, scalability)
Reduce overall cost for economical sustainability	Economical compute platform designs sustainable for multiple decades (20-40 years), affordable service model for railways over the life cycles. (Sustainable design) *Vendor shall support the Platform design and further evolution of platform elements (e.g., upgrades) for 40 years.*

2.1 Requirements inherited from RCA/OCORA SCP Workstream

The SIL4 Cloud builds on the identified requirements and prior work done in the RCA and OCORA railway initiatives, which are driving a functional architecture for trackside and onboard functions for future railway operations. In the RCA/OCORA Safe Computing Platform technical workstream (TWS - 03), various European railways[2] have consolidated their high-level objectives in 16 points, which are recreated in Annex 0. Furthermore, the following figure shows all high-level objectives and their key motivation which would further help to realize the vision of digital rail.

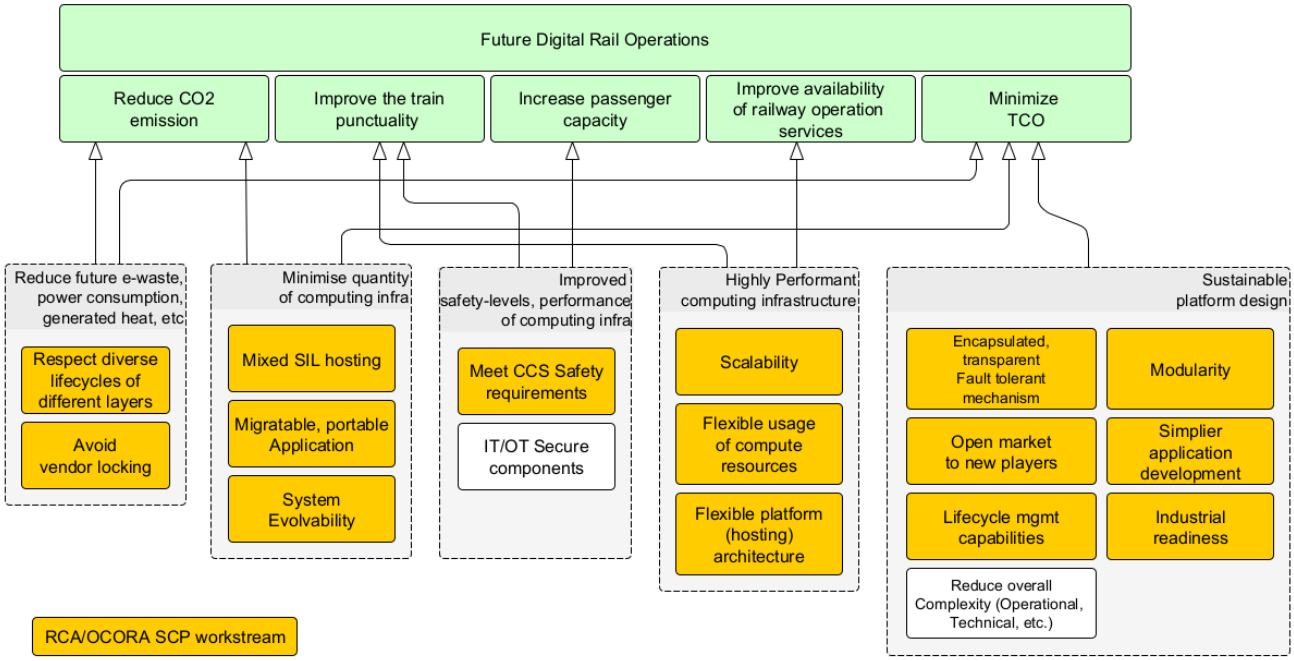

Figure 3: Overview of the high-level objectives of SIL4 Cloud

In addition to high-level objectives, RCA/OCORA also proposes key **design paradigms** to be followed for novel platform design**.** Following is a list of overall design paradigms identified for a SIL4 Cloud.

[2] SBB, SNCF, and NS

- Clear separation of concerns between the Functional Application and the platform: Notably, Functional Applications shall handle only business logic, while all other required functions related to application execution and control (incl. mechanisms for safety, fault tolerance, persistence, communication, and application management) shall be handled by the SCP in a way that is transparent to the business logic. This increases reusability of applications running on railway platforms, by providing a base platform ecosystem API that safety-critical applications can use, similar to approaches in automotive (AUTOSAR) and avionics (ARINC-653);
- Provide a harmonised Platform Independent API (hereafter PI API), possibly common for onboard and trackside systems;
- Follow a modular safety concept: to minimise homologation efforts including minimisation of numbers of safety-certifiable components;
- Maximise usage of COTS components (CPU, I/O, SW, etc.), tools and open-source software (where applicable): to minimise vendor lock-in and leverage advances in other sectors;
- Consider utilising virtualisation techniques or similar means of abstraction of computing resources: for better evolvability, scalability, the support of mixed SIL constellations and a more flexible mapping of applications to compute resources;
- Use a composite fail-safety approach to achieve safety level (up to SIL4) over other mechanisms;
- Make SIL4 Cloud capable of hosting mixed criticality applications, in parallel;
- Deploy and distribute Functional Applications as much as possible, mostly utilising Geographically redundant SCP;
- Flexible and dynamic application deployment and management in the context of functional safety.

2.2 Key Actors interacting with SIL4 Cloud

As a system, SIL4 Cloud will be interfaced and interacted by many different actors, which also may vary according to the different life cycle stage of the system. A high-level picture of important actors is shown in Figure 4.

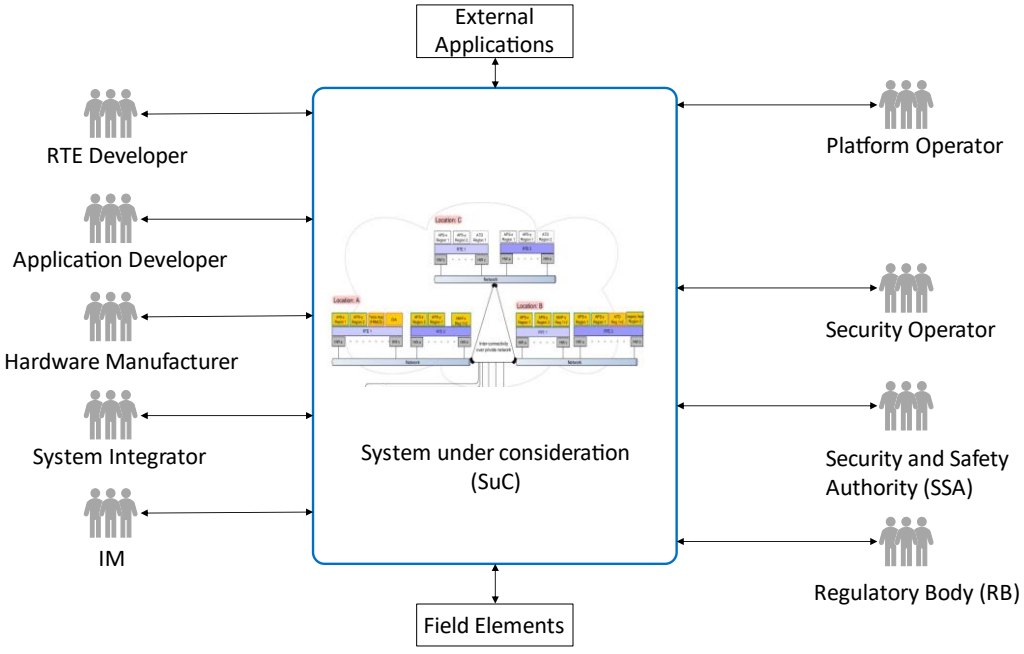

Figure 4: System Actors

The description of each actor interacting with the SIL4 Cloud is given below:

Actor	RTE Developer
Actions	• Develops the runtime environment product as per customer (i.e., IM) needs (functional, non-functional, and technical specifications) • Provides periodical changes, updates to the platform, depending on the needs of business and technical demands • Upgrades platform to improve the lifecycle of the overall system (e.g., in case of hardware (HW) end-of-life) • Informs customer (i.e., IM, Security Operator) about any security flaws, vulnerabilities in the RTE product in timely manner • Delivers training to Functional Application developers and exchange specific guidance (e.g., which COTS hypervisor to choose, how to configure and qualify the setup) • Evaluates and provides updated artefacts such as updated RTE version, guidance documentation, if changes are requested by customer (e.g., if a specific COTS hypervisor is requested by customer, RTE developer evaluates how to integrate it and provides the applicable guidance documentation) • Supports on the SRACs definition for the selection of hardware and aligns with the railway operator • Provides documentation, tests, generic product certification etc. to support homologation process
Demands	SIL4 Cloud requirements
Example	Thales, SYSGO

Actor	Application Developer (or Engineering Team)
Actions	• Develops the Functional Application as per DB functional specification • Develops the Functional Application using Software Development Kit (SDK), toolchain provided by RTE Developer • Provides documentation, test results, etc. to support homologation process
Demands	• Documentation and constraints (e.g., SRACs to fulfil, development/API guides) • Proper training • Necessary tools / description, which tools to use etc.
Example	A software development vendor, DB's development team

Actor	Hardware Manufacturer
Actions	Manufactures the computing hardware (motherboards, servers, storage, etc.) or hardware components (chipsets, etc.)
Demands	Available MTBF values
Example	EMC, DELL, Hewlett-Packard (HP), duagon

Actor	System Integrator

Actions	• Integrates Functional Application and computing platform provided by different vendors • Ensures Functional Application is seamlessly integrated into the SIL4 Cloud as a whole • Provides overall E2E homologation certification(s) • Obtains overall authorisation for both Functional Application and SIL4 Cloud based on the fulfilment of SRACs • As per current assumption and understanding, the railway (i.e., IM) is responsible for the overall integration but the responsibility might be delegated to another vendor.
Demands	System Integrator needs to get all relevant information and documentation from the subsystem vendors.
Example	DB or 3rd party company

Actor	Railway Infrastructure Manager Organization (IM)
Actions	• IM is considered an entity operating the railways which hosts the computing platform • May provide specifications for the functional software(s) and computing platform(s) • Provides suitable facility and physical environment for the computing platform as specified by the platform/RTE developer • May do E2E system integration (see actor System Integrator)
Demands	• Fulfilment of high-level objectives defined and requirements specification
Example	DB

Actor	Platform Operator

Actions	• May work on platform on daily basis and monitor the SIL4 Cloud to ensure it is functioning in accordance with defined KPIs
	• May perform maintenance work of the overall SIL4 Cloud
	• Platform Operator's responsibilities can further be differentiated in:
	  ○ **Facility Maintenance**: maintains an operational environment for computing systems like racks, servers, etc. (This could create substantial considerations on Security)
	  ○ **Infrastructure Maintenance**: Responsible for replacing hardware in the data centre
	  ○ **SW Maintenance**: a team that maintains SW services, platform software, etc.

Demands	• Get a real-time picture of current situation of HW and virtual servers • Access server infrastructure from remote in a secure way • React on problems by e.g., restarting or manually re-balancing.

Example	DB (or internal divisions of DB)

Actor	Security Operator
Actions	• Performs regular and event-triggered security operations in SIL4 Cloud (e.g., across multiple data centres) • Operates shared security services (e.g., IAM, PKI, BKP, LOG/IDS/SIEM) • Operates security infrastructure components (e.g., firewalls, IDS sensors) • Monitors and reacts to incidents caused due to cyber risks
Demands	Detailed information about assets and integrated software components etc.
Example	Security team of DB (or internal divisions of DB)

Actor	External Applications
Actions	• Connects to SIL4 Cloud/data centre from outside of the boundary and vice-versa • Communicates using SCI with respective central computing platforms • Could be categorized in legacy and new Functional Applications
Demands	Interface definitions (e.g., SCI)
Example	• Adjacent RBC, IXL systems (Legacy) • CTMS, Scheduling system • External Alert System represents a system providing alert messages from different alarm systems: fire warnings (tunnel, station, platform), landslide warnings)

Actor	Field Elements

Actions	• Field Elements are railway fixed equipment on/or adjacent to track
	• The Field Elements may be of two types, based on communication protocols used
	  ○ EULYNX-Controllable Track Elements (main focus)
	  ○ Non-EULYNX-Controllable Track Elements

Demands	Compatible standardised communication protocol must be supported by functional applications and RTE
Example	Light signal, Point machine, Level Crossing, Train Detection System (TDS), Balises

Actor	Security and Safety Authority (SSA)
Actions	• Embraces different public entities responsible for the security and safety within its area of authority: • Railway safety authority • Railway investigation authority

	• SSA represents different entities with the following responsibilities:
	• Railway safety and investigation authorities:
	  ○ Investigation of dangerous events and hazardous events pursuant to Article 20 (1) and (2) of Directive 2016/789
	  ○ Enforcing railway safety standards
	  ○ Supervising the correct application of the principles of railway operations
	  ○ Analyses of log data and safety records in case of accidents and incidents
	  ○ Homologation of vehicles according to national and international rules
	  ○ Homologation of infrastructure assets according to national and international rules
	• Police and security entities:
	  ○ Customs: Ensuring the correct application of international border laws
	• SSA may be relevant in the following two situations:
	  ○ When railway (e.g., DB) needs preventive assessment
	  ○ After an incident

Demands	• Proper description/safety case according to respective standards • Experts to explain if needed
Example	• EBA = Eisenbahnbundesamt • EUB = Eisenbahnunfalluntersuchungsbehörde. The new name for EUB is BEU - Bundesstelle für Eisenbahnunfalluntersuchung • BSI = Bundesamt für Sicherheit in der Informationstechnik (German Federal Office for Information Security

Actor	Regulatory Body (RB)
Actions	The role of RB is also highlighted by CENELEC standards.
Demands	As defined within the standards.
Example	https://www.bundesnetzagentur.de/

2.3 Target Functional Applications

One of the key design principles of SIL4 Cloud is to run SIL applications along with basic integrity functional applications side by side. This helps to consolidate different computing nodes in a data centre. In the following section, some example applications are described, which could run on the SIL4 Cloud.

2.3.1 Advanced Protection System (APS)

The Advanced Protection System (APS) is part of a novel architecture for the future control, command and signaling (CCS) system developed by the RCA initiative. More precisely, APS is a group of subsystems responsible for safe track usage and for control and supervision of the railway production. It includes the subsystems Safety Logic (SL), Safety Manager (SM), Object Aggregation (OA), Movement Authority Transactor (MT), Mobile Object Transactor (MOT) and Fixed Object Transactor (FOT). As

per RCA architecture, each APS subsystem (realised in the form of safety-critical Functional Applications) has defined functionalities and interfaces. The following figure shows the current status of a high-level view of subsystems and names of logical interfaces in RCA-defined architecture.

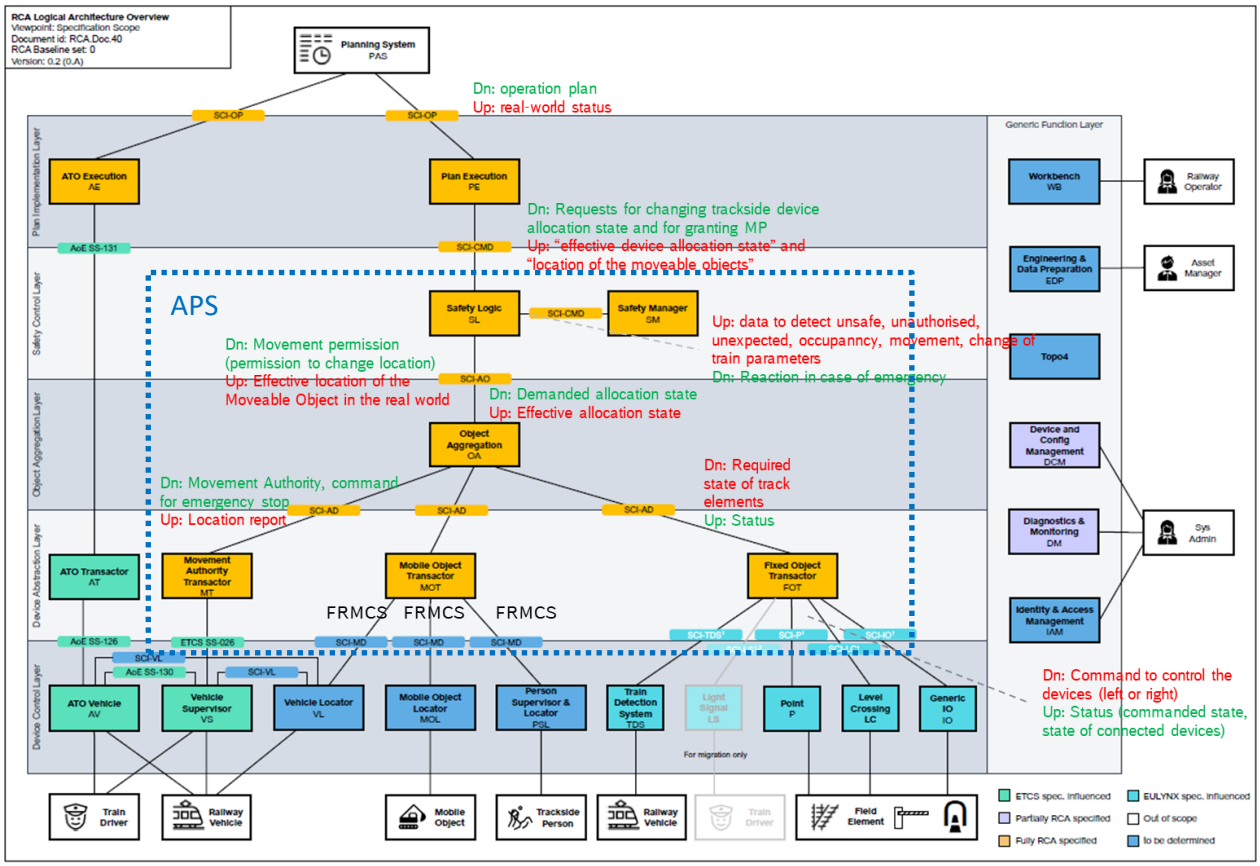

Figure 5: APS logical architecture [*blue dashed line]* (Jan 2021)

In a nutshell, it is understood that those subsystems in RCA defined architecture:

- contain pure functionality/business logic(s)
- have proprietary internal architecture and implementation
- have RCA-defined logical interfaces to (elements of) railway applications, which talk to each other by exchanging defined sets of inputs and outputs. RCA-defined interfaces (SCI-OP, SCI-CMD, SCI-AO, SCI-AD) are seen as quite important for runtime environment specifications, since they are newly defined, yet unfinished
- EULYNX defined interfaces, which are already standardized for following field elements:
  - SCI-TDS (Train Detection System)
  - SCI-TSS (Trackworker Safety System)
  - SCI-LS (Light Signal)
  - SCI-LEU (Lineside Electronic Unit)
  - SCI-PM (Point Machine)
  - SCI-LX (Level Crossing)
  - SCI-IO (I/O Controller)
- have interfaces to the following generic functions:

- Identity and access management (IAM)
- Topo4 (contains topology data)
- Diagnostics and monitoring, updates

2.3.2 Automatic Train Operations (ATO) applications

In Grade-of-Automation 2 (GoA2) rail operation, ATO is responsible to provide trains with so-called segment and journey profiles that allow an automated acceleration and deceleration of the train. In GoA2, ATO functionalities on the trackside and on the train are usually rated (at most) as basic integrity. In the context of fully-automated rail operation (GoA4), ATO will likely be complemented by functions with a SIL rating. Irrespective of whether ATO is rated as basic integrity or (perspectively) contains components with a SIL rating, it should be possible to host ATO functions on the SIL4 Cloud, for instance to consolidate trackside computing infrastructure.

2.4 DB Cloud Infrastructure

At its core, the envisioned infrastructure required to host the SIL4 Cloud consists of multiple data centres, their network interconnection as well as network connections between the data centres and the field elements that need to be controlled. For the data centres as well as the networks, a very high degree of availability and stability is required in order to satisfy the requirements of a SIL4 Cloud. In addition, low latencies and scalability must be guaranteed for a large number of connected field elements.

In order to satisfy these needs, DB Netz has started the construction of a series of regional datacentres, the TSO (Technischer Standort), that are located in comparatively close proximity to the tracks (around 50-100 Km away from a track). These are designed to host redundant data centre compartments for fire safety reasons, with feature redundant and disjoint electricity supply, as well as redundant connectivity to field elements and the core network, alike.

The TSOs are built to EN 50600 and TSI.STANDARD V4.2 standards and satisfy availability level 3 as defined therein.

DB Netz, as the infrastructure provider, also operates a fibre optic network on the rail network for the connection of field elements to control command systems. The network providing WAN access between the TSOs and the GFKs (Gleisfeldkonzentrator), which act as gateway locations to concentrate individual field element lines is provided by the bbIP Network (bahnbetriebliches IP Netzwerk). The construction of this network is also under way. It is structured into several hierarchical layers consisting of:

- a redundant core network consisting of two rings connecting the operations centres DB Netz uses to operate the national rail network
- redundant single ring regional networks connecting individual TSOs to the core network via their corresponding operations centre
- redundant access networks connecting GFKs to TSOs

This hierarchy is depicted in Figure 6.

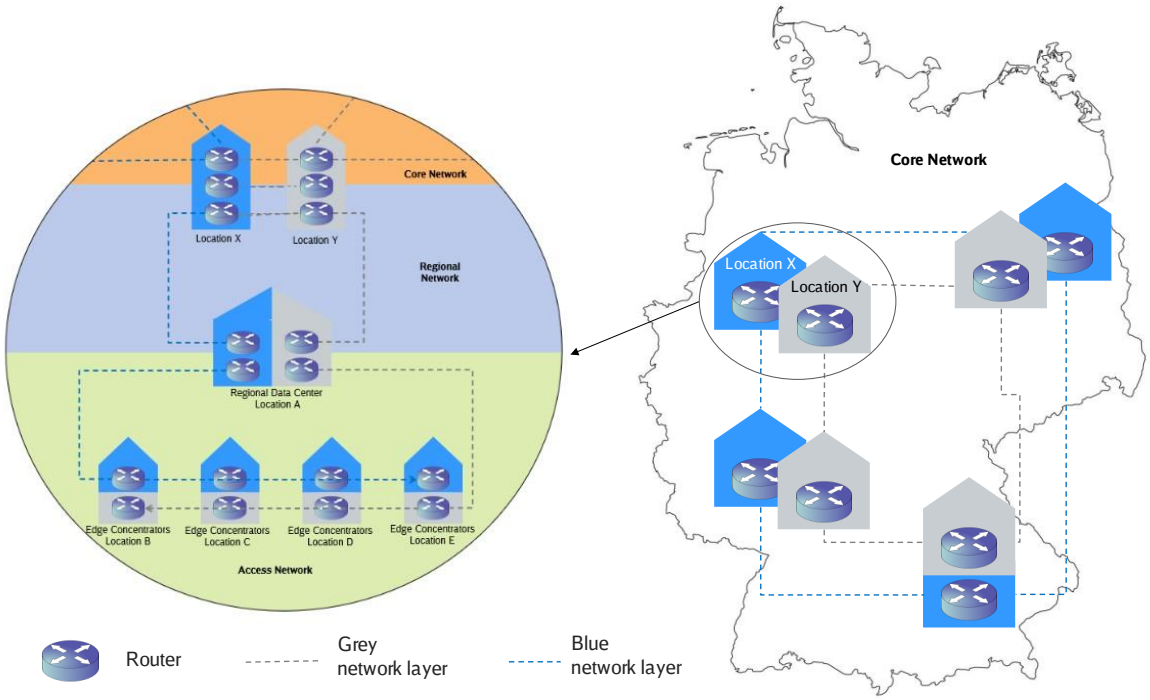

Figure 6: Example of the bbIP hierarchy

An additional network layer is needed to connect the field elements themselves to their corresponding control systems located in the TSO, e.g., digital interlocking. For this purpose, an additional infrastructure component is provided on the trackside, the FeAk (Feldelemente-Anschlusskasten), where the object controller and additional cryptographic components are situated. This way, all communication between the object controller and the central control logic, e.g., a digital interlocking system is encrypted at the source and the underlying network treated as an untrusted WAN.

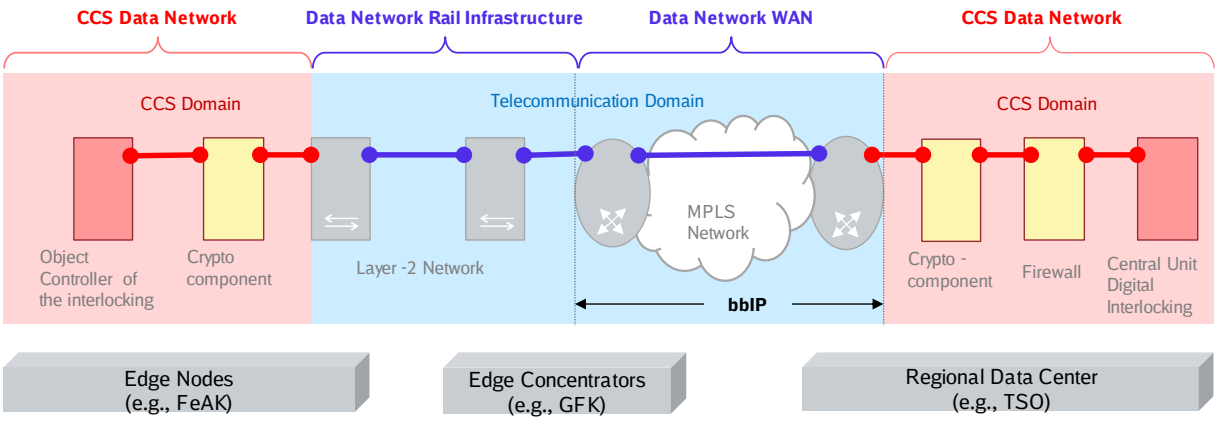

Figure 7: IT Security in bbIP for CCS systems

2.5 Key requirements for SIL4 Cloud

Coming from the target applications, a SIL4 Cloud should fulfil the following capabilities and requirements.

SIL4 Cloud shall:

- host safety-critical Functional Applications using a standardised API between Functional Applications and the underlying Platform.
- host migrated legacy safety-critical Functional Applications.

- host non-safety-critical Functional Applications (Basic Integrity).
- host mixed-critical Functional Applications.
- provide Publish-Subscribe oriented communication mechanisms for managing communication within the SIL4 Cloud.
- provide Communication Gateway mechanisms to manage communication with 3rd party external systems or Field Elements.
- provide Standardized API between Functional Applications and underlying SCP.
- provide guaranteed deterministic behaviour for Functional Application in terms of 1) timing, 2) computing resources.
- provide guaranteed freedom from interference for Functional Applications from external events (such as reconfiguration of other Functional Applications) to avoid interruptions in safe and secure execution.
- host safety-critical Functional Applications (up to SIL 4) in a distributed manner across multiple data centres so that a blackout at one location would not interrupt the safety functionality.

[Dynamism demands]

- SIL4 Cloud shall automatically restore Functional Applications to healthy state when there is an event of failure in Functional Applications or associated computing services.
- SIL4 Cloud shall automatically restores the communication between a sender and a receiver when there is an event of network changes (such as network fails and restores or networking changes) or communication failure.
- While in runtime/operation, safety-critical user functionality[3] should not be interrupted if a hardware is added or removed.
- While in runtime/operation, safety-critical functionality shall be independent from any interference caused by any updates or reconfiguration of SIL4 Cloud or/and Functional Application (s).
- Non-safety critical Functional Applications can be updated without a re-qualification (safety and security) of Safe Computing Platform (s) or application.

[(Controlled) Scalability]

- The SIL4 Cloud can be expanded while in operation by adding additional number of hardware elements.

Constraints as Requirements:

- The Platform shall comply with following railway norms/standards
  - EN 50128 (Software)
  - EN 50129 (Hardware and equipment)
  - EN 50126-1/-2 (RAMS)
  - EN 50159 (Safety related Communications)
  - IEC 62443, especially IEC 62443-3-3 and 62443-4-2 (cybersecurity for operational technology)

[3] Related to business logic such as APS Safety Logic (APS SL) functionality which is realized by Functional Application (s)

- EULYNX and NeuPro (for interoperability)

- The SIL4 Cloud shall use non-vital COTS computing hardware
- The SIL4 Cloud shall use wide range of computing hardware boards (such as differentiated by different processor architecture of computing hardware) to avoid dependency on a particular hardware manufacturer/make.
- It shall be possible to replace the SIL4 Cloud hardware independently of the RTE Developer, such as by the System Integrator.

2.6 System under consideration (SuC)

To keep the focus on concepts relevant to SIL4 Cloud within this research project, a representative example – a so called System under consideration (SuC) – has been defined. As the concrete setup is yet to be developed and implemented, some relevant assumptions were made while defining the SuC:

- The SuC consists of several data centres, which host SIL4 functional applications and are geographically distributed to avoid any impact of natural events. Pictorially it is shown as three geo data centres in Figure 8.
- The SuC furthermore contains interfaces to field elements, which are distributed over the whole railway grid. RaSTA is considered as main safe communication protocol to communicate with these field elements.
- The data centres in SIL4 Cloud and the field elements are connected via a private, encrypted and redundant network.
- The network within the data centre might be a so-called software defined network (SDN), providing virtual communication channels to the systems and/or the RTEs and Functional Applications.
- A virtualization layer (not shown in the Figure 8) might be present between hardware (HW) units and RTEs, allowing for shared usage and flexible allocation of hardware. A constellation with virtualization layer is shown in Figure 11.
- An additional consensus layer (not shown in the Figure 8) to enable replication and voting support might also be present – especially, when these functions are not provided by the RTE.
- In the SuC, Runtime Environments (RTEs) and Functional Applications are from different vendors (multi-vendor constellation).
- The Functional Applications include with different safety requirements (SIL) and those which are not safety-relevant at all (mixed-criticality).[4]
- The RTE might also host legacy applications, which were not developed for the SIL4 Cloud and cannot be changed.
- Each data centre in SIL4 Cloud also provides applications for the administration of all resources within the SCP (not shown in the Figure 8).
- Each data centre in SIL4 Cloud also provides the common security services defined by X2RAIL (not shown in the Figure 8).
- Each data centre in SIL4 Cloud also provides an availability of at least 99,982 % (EN 50600 level 3).

[4] Applications might be grouped by safety and/or security requirements (SIL/SL level) and segregated from each other. The need for this and options for implementation were investigated in the project.

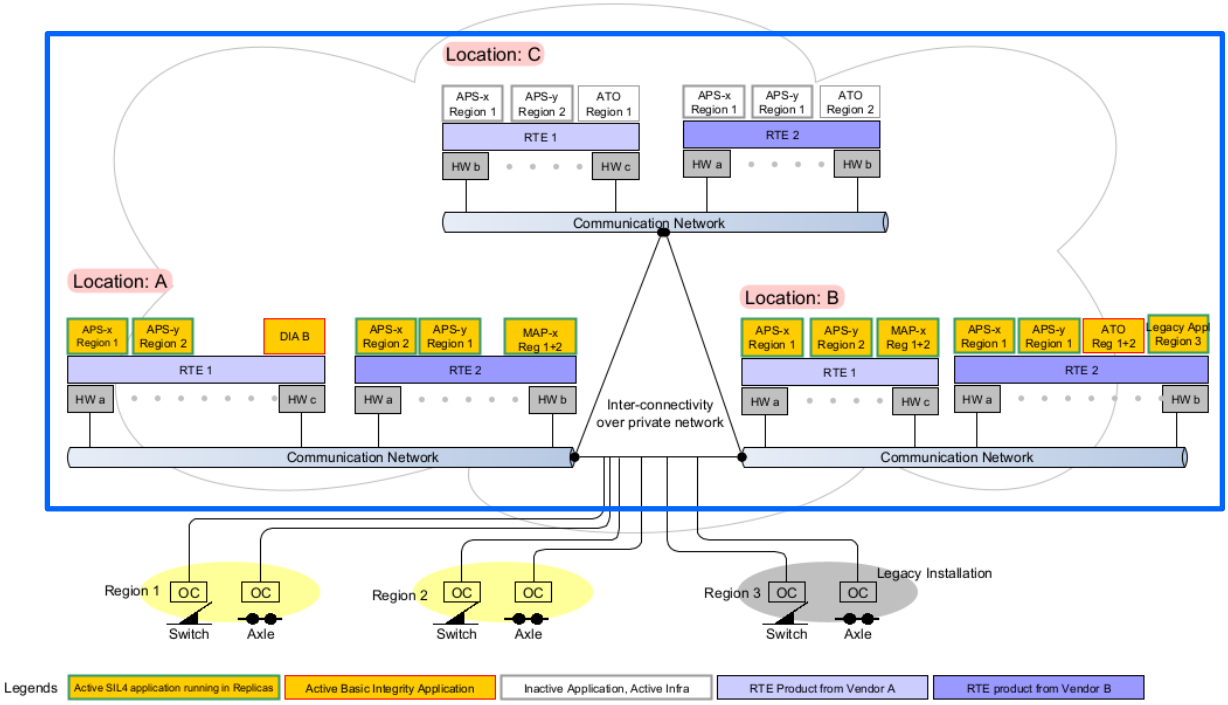

Figure 8: System under consideration (SuC) (inside blue line boundary)

3 Generic architecture of SIL4 Cloud

3.1 Basic concepts

Within this report, mostly the nomenclature from RCA/OCORA Safe Computing Platform is used. However, some additional terms are used in technical discussions and architecture description. A layered approach has been used to set up a modular and flexible architecture (see Figure 9 and Figure 10).

- The lowest layer of the SCP is **Hardware** which provides basic computing functionalities such as processing, memory and data storage. In SIL4 Cloud, standard/COTS server(s) are considered, which do not provide special safety mechanisms and are hence non-vital from the point of view of safety. Two computing hardware may have a different architecture and physical resources (e.g., number of processors, memory capacity, etc.).
- (Optionally) A **Virtualization layer (Hypervisor)** can be used, which is responsible for abstracting the underlying hardware, and provides an execution environment (e.g., partitions) as well as virtual machines (VM) to host the further layers above. For the top layers, it will be transparent whether they are executed on physical hardware or on a virtual machine.
- On top of the hardware or virtual machine is the **Runtime Environment (RTE)**. The RTE layer is composed of Safety Services, System Services, the communication stack for information exchange among applications on the same platform and with external entities, and, depending upon the actual platform implementation, an operating system (Guest OS). The RTE is responsible for providing a safe execution environment for the application(s), i.e., to ensure the guarantees demanded by functional applications such as CPU time, freedom from interference etc.
- The RTE provides a standardized Platform Independent Application Programming Interface, the **PI API**, to support interoperability and portability of applications.
- On top of the RTE, the **Functional Application** or **Application** is executed. The Functional Application refers to a software implementing the business logic of a railway function (e.g., that of a so-called Advance Protection System – APS, as an example of CCS functions according to the RCA architecture).

The following terms, used in the report to describe SIL4 Cloud architecture, are also depicted in Figure 9 and Figure 10.

- A **Computing Element (CE)** is one instance of hardware (virtual machine and RTE), which executes one replica of the Functional Application. The replica of the Functional Application is included in the Computing Element.
- A **Computing Node (CN)** includes several Computing Elements (CE), which together ensure the required safety level MooN for exactly one Functional Application. From the Functional Application perspective, a Computing Node includes all relevant replicas to run the application.
- A **Computing Group (CG)** is a grouping of Computing Nodes (CN) to achieve high availability of exactly one Functional Application. This can be achieved by running the CNs redundantly, as hot standby or similar. If one CE breaks down, the other CN of the CG continues operation. So, the CG is formed in configurations such as 2x2oo2, 3x2oo2, etc. In case no additional redundancy is necessary, the CG is identical to the CN.
- A **Safe Computing Platform (SCP)** is the platform on which Functional Applications can be hosted. It consists of the hardware, hypervisor, virtual machines, and RTE, which can be set up into a different configuration to run Functional Applications in a safe way. In contrast to the other definitions, it doesn't include the Functional Applications.

- A **SIL4 Cloud** refers to a group of SCPs which are deployed across multiple, geo-located data centres. A SIL4 Cloud will contain different SCP realizations from different vendors, including some basic integrity (non-SIL) components as well. The orchestration will be provided through a common orchestrator. It must be noted that the degree to which an orchestration will be standardised is up for future investigation.

Figure 9 illustrates an example of SCP in a 2x2oo2 redundant architecture which runs “bare metal”, i.e., without hypervisor. In this case, there is only one application shown running on the Computing Group. Each replica has its dedicated hardware to run on. The RTE is responsible for safe execution and orchestration.

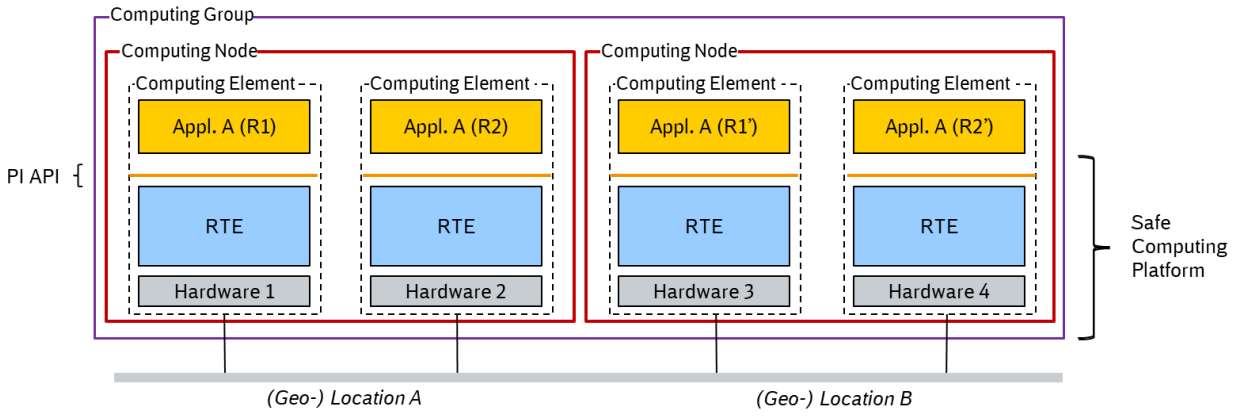

Figure 9: Layers without Virtualization (in 2x2oo2 scenario)

Figure 10 illustrates a setup of SCP with a hypervisor. In this case each hardware runs multiple applications with the help of virtual machines provided by the underlying hypervisor. For safety reasons, at most one replica of each application must be executed on one physical hardware. Nevertheless, different CEs can be run on the same hardware, as long as they host different applications.

For complexity reasons, it is assumed that the RTEs within one computing group will be from a single vendor. Otherwise, many more and very complex interfaces for inter-RTE communication need to be standardized in depth.

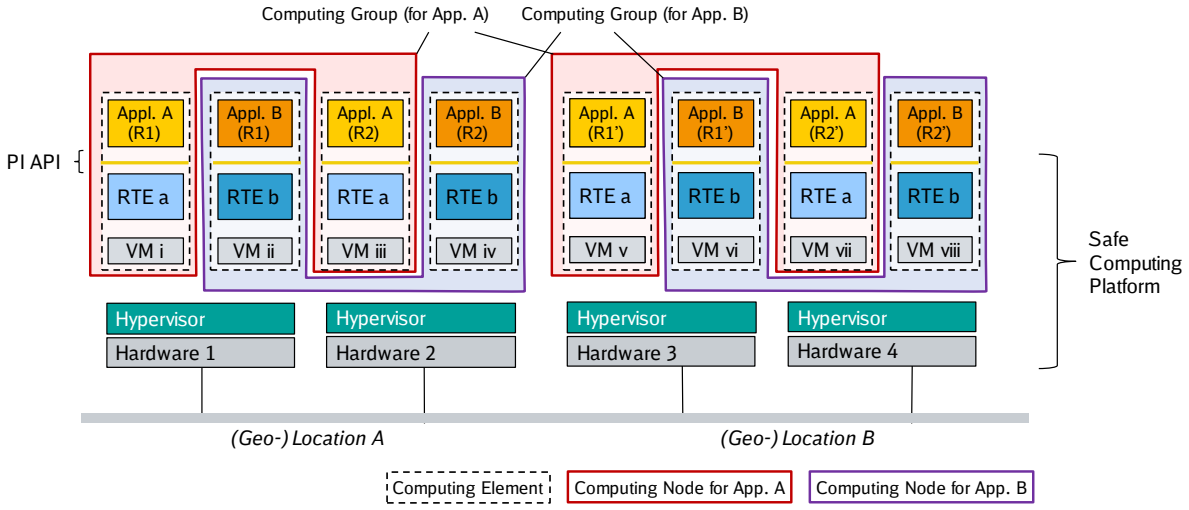

Figure 10: Layers with Virtualisation/Hypervisor for 2 applications each in a 2x2oo2 configuration

Additionally, the following terms are often used in the context of the SCP:

- **Communication Gateway (or Gateway)**, which interfaces with safety-relevant, external entities like Field Elements (OT devices such as Point Machines, Light Signals, etc.). With the help of a Gateway, one can implement communication protocols used for communication with external entities (safety-relevant and/or non-safety-relevant), e.g., the RaSTA protocol for controlling field elements and OPC UA for performing diagnostics. The Communication Gateway allows reusability of application keeps the complex protocol implementation outside of the application itself, hence making it simpler.
- **Voter** is a part of the RTE layer and is responsible for comparing the computed results of the replica. If the results are inconsistent, the voter has to decide whether a result is considered to be invalid and consequently the Functional Application shall transition into the safe state. Also, the voter has to ensure the integrity of the voting itself, i.e., safe voting. For instance, the voter needs to be run replicated on multiple RTEs to ensure freedom from single point faults.
- **Replica** to enable the safe execution of an application, the application is cloned and run in parallel. The voter checks the consistency of the execution. These clones are called replicas.

3.2 Functional Decomposition

At a high level of abstraction, the relationship between Functional Applications and the underlying SIL4 Cloud can be described as a deployment, as shown in Figure 11. The Functional Application is deployed to the SIL4 Cloud and executed there. In addition to definitions mentioned before, the SIL4 Cloud is an abstract construct for combining the distributed functionalities for managing and operating Functional Applications and the IT infrastructure required for it. The SIL4 Cloud is realized here by the totality of all the SCPs involved.

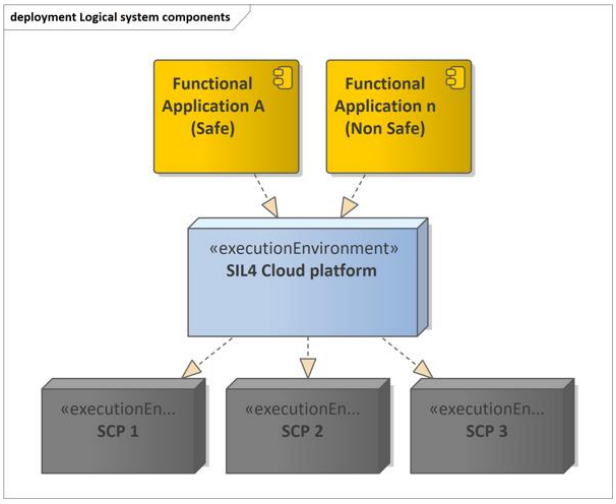

Figure 11: Deployment of Functional Application to the SIL4 Cloud

The key functionalities mentioned for managing and operating applications are shown in Figure 12. The following sub-sections describe the individual logical components, their functions, and relationships to other components.

3.2.1 Package Management

Functional Applications are rarely delivered as standalone executables. There are generally dependencies on other basic applications, which are necessary for the operation of the Functional Application, e.g., databases, frameworks, etc. These dependencies, including release versions and execution conditions (such as replication requirements), can be specified as a package. To avoid redundant storage, the package could be seen as a list of applications and the reference to the location where the images of the application are stored (i.e., image repository). A Package Manager uses these packages to create the deployment configuration for the *Orchestrator*.

3.2.2 Package Repository

The Package Repository is a central directory of packages. Packages contain a list of related applications that must be run together. In addition, the versions of the applications and other execution conditions can be specified in a package. The repository can support other functions, such as version management, signatures and encryption, depending upon the requirements.

3.2.3 Orchestration

To deploy a Functional Application, an Orchestrator allocates a free resource to the Functional Application in its managed computing group (set of computing nodes). It then launches the image as a *virtual unit* (container or virtual machine) and continuously monitors its alive status. When a *virtual unit* becomes unavailable, the Orchestrator automatically starts a new instance of the *virtual unit* on another free resource in the computing group. V*irtual units* can be easily replicated either to implement different voting mechanisms or to scale up the capacity of the maximum number of requests. In general, Orchestrator should not get any functional safety related responsibilities.

Orchestration capability is one of the open research questions in the context of the SIL4 Cloud and shall be investigated further beyond this project e.g.:

- Safety relevance and responsibility of the orchestrator
- The best way to manage and control SCP realisations developed by different vendors (multi-vendor SCP constellation in data centres)
- Need and feasibility for Standardised API specification for orchestration
- Applicability of utilisation of COTS Orchestrator(s) for Safety-relevant systems (Functional Applications and underlying SCP)

3.2.4 Image Builder

An image builder takes the compiled application executables and the libraries that are necessary for its execution and packages everything into an image. These images could be deployed on a computing node. It is possible to sign the image to assure its authenticity and integrity but comes along with certificate management.

3.2.5 Image Repository

Images are stored in a central location in the cloud, the so-called registry or image repository, so that they can be obtained from there.

3.2.6 Cyber Security

Some form of cyber security functionality is required to ensure security against cyber threats e.g., a firewall checks incoming and outgoing traffic to the internal network from outside. Furthermore, SIL4 Cloud will require Identity and Access Management (IAM), patch and vulnerability management, means for system hardening, network segregation and encryption of traffic, incident handling, and others. Please refer to Chapter 5 for more details on Security.

3.2.7 Load Balancer

In the case where multiple replicas are operated to meet availability or performance requirements, a load balancer is needed to distribute incoming requests among the available replicas. If a load balancer is used, there are also safety requirements that the load balancer must fulfil, as the forwarding of requests must be deterministically ensured.

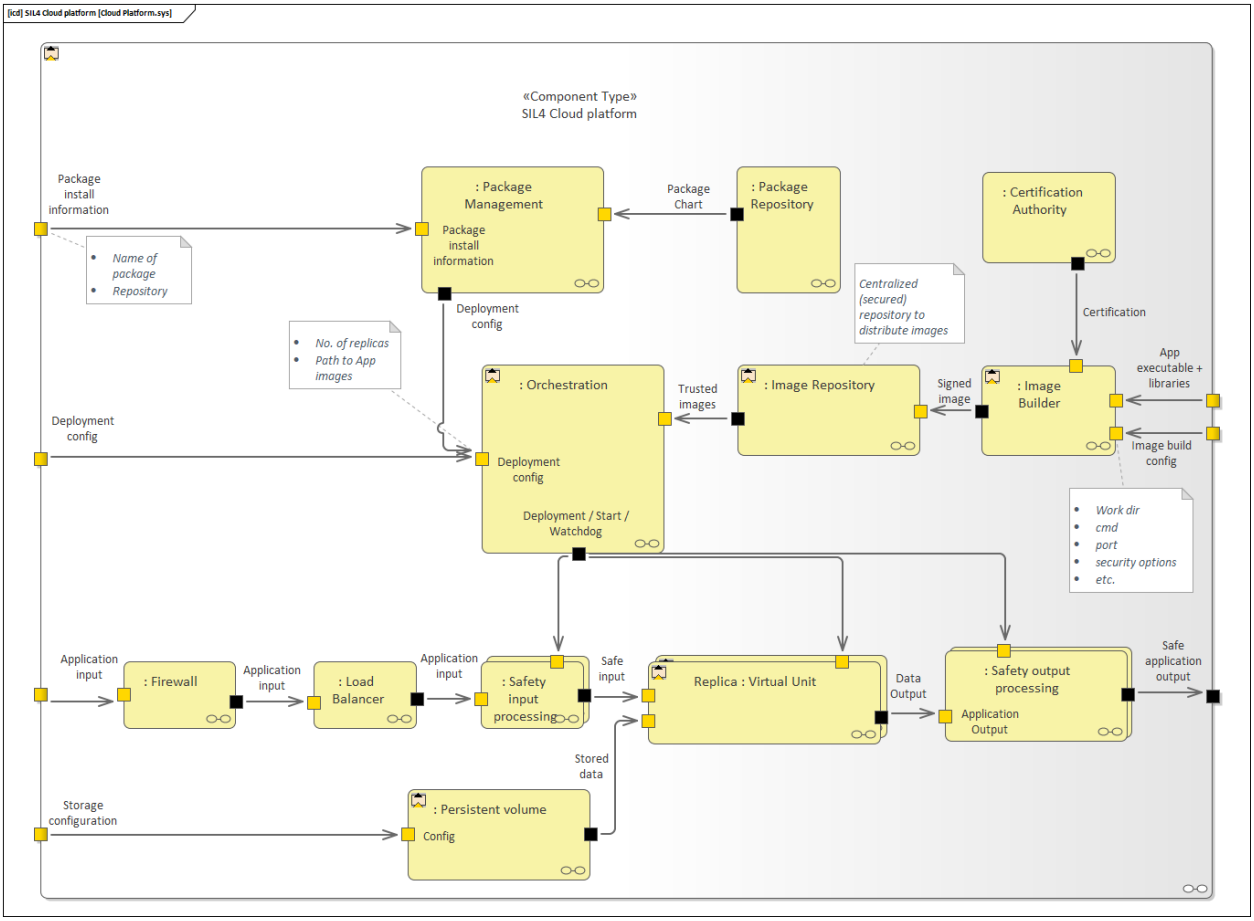

Figure 12: SIL4 Cloud functionalities

3.2.8 Safety input processing

Incoming data is checked according to applicable safety procedures before processing it in the safety-critical applications. These checks can be performed on the protocol data according to the measures defined in the protocol that has to comply with EN 50159. Additionally, suitable checks on application level are performed to ensure that the received data is valid.

The forwarding of data to all replicas participating in the voting process must be ensured to guarantee that the checks are indeed being performed by all replicas.

3.2.9 Safety output processing

Outgoing data from replicas are compared, checked, and supervised in the voting to detect failures in the underlying SW and HW.

3.2.10 Virtual Unit

When it comes to providing virtual units of computing resources to an application, there are fundamentally three approaches:

- Virtual Machines (VM)
- Containers
- Emulators

Generally, VMs and Containers can be considered viable approaches to application hosting.

VMs provide very strict isolation guarantees, enforced by a hypervisor and support functions provided by the CPU. Here a complete, virtual representation of a computer hardware is provided, on which an operating system – and hence Functional Applications on the operating system – can be

run. This way, one computer's hardware can be subdivided into multiple, virtual computers, each with their own operating system and virtual resources like disks or network interfaces.

Containers, on the other hand, make use of the existing operating system for the purpose of isolation. This means it is the operating system's responsibility to provide resource constraints as well as process, user and network namespaces to the Container in order to isolate it. Thus, multiple Containers running on the same computer will make use of the same underlying operating system, which allows for more dynamic resource restriction than virtual machines, however not providing the same quality of isolation constraints.

Due to these vastly different approaches to how isolation is realized, VMs and Containers have different strengths and weaknesses which make them suitable for different use cases. See Table 2 for a comparison of VMs and Containers with an emphasis on the safety context.

Table 2: Comparison of VM and Container on different properties

Property	**Virtual Machine**	**Container**	**Relevance to Safety**
Isolation (required for independence of underlying system)	Very strong, as the VM is isolated from the underlying system through the hypervisor.	Very weak, as it is only isolated through abstraction mechanisms provided by the operating system.	✓
Dynamic Resource Allocation	Typically, not supported	Typically, on the fly	-
Resource and Timing Guarantees	Resource Guarantees can be provided through the hypervisor. Special support for Timing Guarantees is not strictly needed in a hypervisor but can be supported through systems providing partitioning.	If the host operating system supports these properties, it can enforce them	✓
Independence	State is fully encapsulated in VM, therefore they are independent from the underlying system.	State is held in the Kernel, therefore Containers are tightly coupled to the host system.	✓
Monitoring of faults	Fault monitoring is partly handled on the hypervisor level and partly inside the VM, as visibility into a VM by the host system is limited to basic resource allocation.	Fault monitoring, with the exception of application specific fault monitoring, is wholly in the domain of the host system as Containers are merely processes.	✓
Security	As isolation is hardware supported, isolation from the security perspective is theoretically very strong, however a number of high-profile vulnerabilities such as Meltdown and Spectre have in the past shown that it is not without issues. Attack surface however is limited to	As isolation is provided through the host operating system only, any operating system API exposed to a container must be considered as potential attack surface.	✓

Property	Virtual Machine	Container	Relevance to Safety
	the hypervisor and hardware.		

It must further be noted that approaches exist, where VMs and Containers are mixed. Typically, Container filesystems are then mounted into VMs that provide the operating system kernel themselves. This alleviates some of the disadvantages that Containers have in the context of isolation, however it limits the ability to dynamically allocate resources. It is to be noted that these approaches typically use a highly optimized operating system inside of the VM to lessen the resource overhead and boot time overhead typically incurred when comparing these two solutions.

3.2.11 Persistent volume

In principle, modern Cloud environments, whether based on VMs or Containers, do not offer persistent storage for instances directly. Rather, instance storage is ephemeral and in order to use persistent storage, a separate service is used for this purpose. This means, instance storage itself is only suitable for stateless applications which do not store any information about their state after termination. This means that the application is always in the same state when it is restarted. However, some applications are stateful and their state information must be stored persistently. For this purpose, a persistent storage service may offer persistent storage via an interface to a virtual unit. In order to satisfy different requirements of applications, the persistent storage service may offer different categories of storage, ranging from very slow to very fast storage or even storages giving different data safety guarantees. In order to do this, it needs to provide storage to a virtual unit via a standardized interface that hides the actual implementation details of the storage solution used. Implementations can range from hard disks of the server or a network storage to a data base.

3.2.12 Redundancy

Redundancy architectures are used in cloud systems to achieve different goals, such as increasing availability, detecting and masking faults, and to scale up. In redundancy architectures, critical components or functions of a system are replicated or duplicated on different Computing Elements (CEs). There are broadly three kinds of redundancy possible. Redundancy design also helps to avoid Single Point of Failures in the system.

- Redundancy for High Availability systems: In this configuration, all redundant nodes are active at the same time and execute the critical functions. Load is distributed between the active Computing Nodes (CNs) and if one CN fails, the load from the failed CN is transferred to the remaining healthy nodes. Such configuration provides High Availability systems.
- Redundancy for Fault Detection: Redundancy system in MooN or MooM configuration provides fault detection. For example, in a 1oo2 configuration, the same input is processed by the function executed on two separate CEs and the result produced by the function is compared by a voter. When the voter detects a result mismatch, the result is tagged as a faulty one and corresponding fault handling actions will be taken by the platform.
- Redundancy for Fault Tolerances: Hardware Fault Tolerance (HFT) is the ability of a component or subsystem to continue to be able to undertake the required safety function in the presence of one or more dangerous faults in hardware. A HFT of one means that there are two independent devices, and the architecture is such that the dangerous failure of one of the two components or subsystems does not prevent the safety action from occurring. A three-channel system, where a single channel can continue to perform the safety function in the case of a fault in each of the other two channels is considered to have a HFT of two. The HFT can easily be calculated if the architecture is expressed as M out of N (MooN). In this case

the HFT is calculated as HFT = N – M. In other words, a 1oo3 architecture has a HFT of two. This means it can tolerate two failures and still continue to work. There are also 1oo2 architectures in practice that have different safety measures e.g., coded mono processing, diversity for detecting certain classes of faults which can be utilized for lower MooN configurations such as 1oo2.

Table 3 provides an overview of redundancy architectures and their HFT properties.

Table 3: Hardware Fault Tolerance (HFT) of different redundant Architecture

Architecture	HFT
1oo1	0
1oo2	1
2oo2	0
2oo3	1
1oo3	2

3.3 Modular Safety

In a highly flexible and modular system, safety requirements and capabilities must be described in a modular way as well. The **Con**ditional **S**afety C**er**tificate**s** (ConSerts) introduced by Fraunhofer IESE [8] could be one way to achieve this. Here, the capabilities of a component are described as *Guarantees*, whereas safety requirements of a component are described as *Demands*. Using the system definition with its components and sub-components described in Figure 11 and Figure 12, the following figures describe the modular safety model of the SCP.

Every ConSert-model could be described for a single component. Following the paradigm of a Service Oriented Architecture (SOA), the service of the component is defined and augmented with an integrity level as well as additional safety properties that can be guaranteed by the component. In Figure 13, a Functional Application is under consideration. Here, in the illustrating example, the service of this Functional Application is called “App Service”, which can be guaranteed with an integrity of SIL4. Since it is important for the safety-critical use of a service to know the addressed failure modes, these are specified in the form of safety properties, i.e., “*Provision.Commission*” with a certain failure rate. This means, the Functional Application assures that the service will not be commissioned unintentionally, respecting the specified failure rate. Which safety property (or covered failure mode) is relevant, must be derived from the safety concept e.g., *Provision.Commission* is a safety-critical failure mode in a movement authority service, whereas *Value.Deviation* is safety-critical for a train location service. The definition failure modes to be considered must be specified in a domain specific failure type model.

The Functional Application cannot achieve this guarantee by its own, furthermore it has some safety demands towards other system components. i.e., it requires an execution service, a memory service, a communication service, and a storage service. These required services can be specified as *Demands,* similar to the *Guarantees,* with a required integrity level and safety properties.

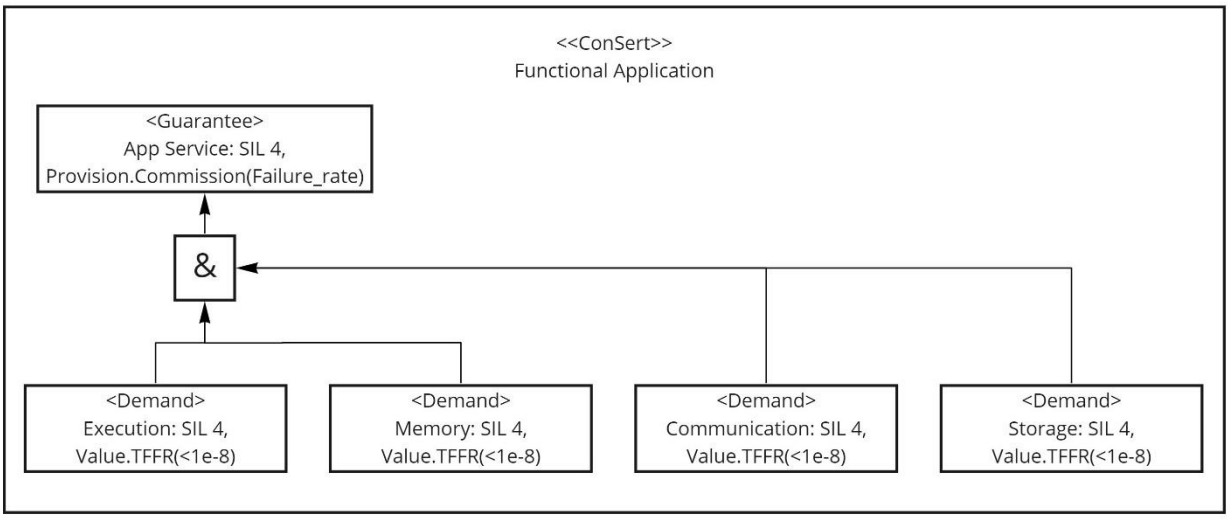

Figure 13: ConSert of the Functional Application

The next component under consideration is the Cloud Platform, for which *Guarantees* and derived *Demands* are specified as well. During run-time, a match making could evaluate whether *Demands* are fulfilled by *Guarantees* of other components in the system. As ConSert-models allow modelling the relationship between *Guarantees* and *Demands*, different configurations are possible. Figure 14 shows the ConSert-model of the Cloud Platform in a 2oo3 configuration. A second model could specify the *Guarantees* and *Demands* in a 2x2oo2 configuration. Based on the fulfilment of the *Guarantees* at run-time, the most appropriate one could be selected, and configurations switched.

In this 2oo3 configuration, the Cloud Platform can guarantee an execution service with SIL4 and a Tolerable Functional Failure Rate (TFFR) of $< 10^{-8}\ h^{-1}$ if the following conditions are fulfilled:

- There are three different voters and their independence could be proven,
- There are three different replicas to execute the application code and their independence could be proven,
- The demands towards the voting services are fulfilled (SIL 4 and TFFR $< 10^{-8}\ h^{-1}$),
- The demands towards the execution service of the replicas are fulfilled (Basic Integrity/SIL0 and TFFR $< 10 * 10^{-4}\ h^{-1}$).

A similar modelling of the relation between *Guarantees*, *Demands*, and necessary conditions at run-time (*Runtime Evidences*) has to be done for the other services of the platform: Memory, Communication, Storage.

The demand of the Cloud platform of a voting service can be provided by the component "Safety output processing". As shown in Figure 15, the voting service can be guaranteed with SIL4 integrity and a TFFR $< 10^{-8}\ h^{-1}$ if this voter instance gets data from three replica (Data R1, Data R2, Data R3) and the consensus of the data is approved by two other voter instances (Voting 2, Voting 3). Note that the independence of the voters and data sources, the integrity of the execution environment, as well as protection against subsequent faults in the value and time domain have to also be ensured.

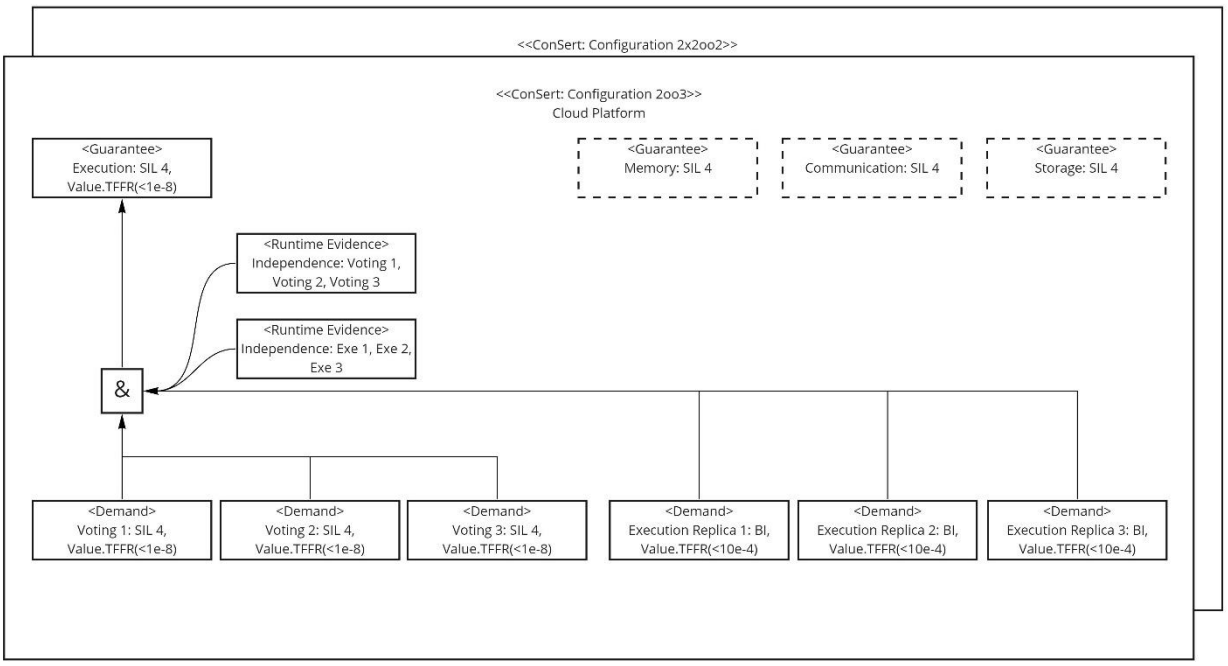

Figure 14: ConSert of the SIL4 Cloud Platform

The data sources, e.g., Data R1, are allowed to deliver their service with Basic Integrity/SIL0 and the voter is able to raise it to SIL4 by comparison of the data and the other voter results.

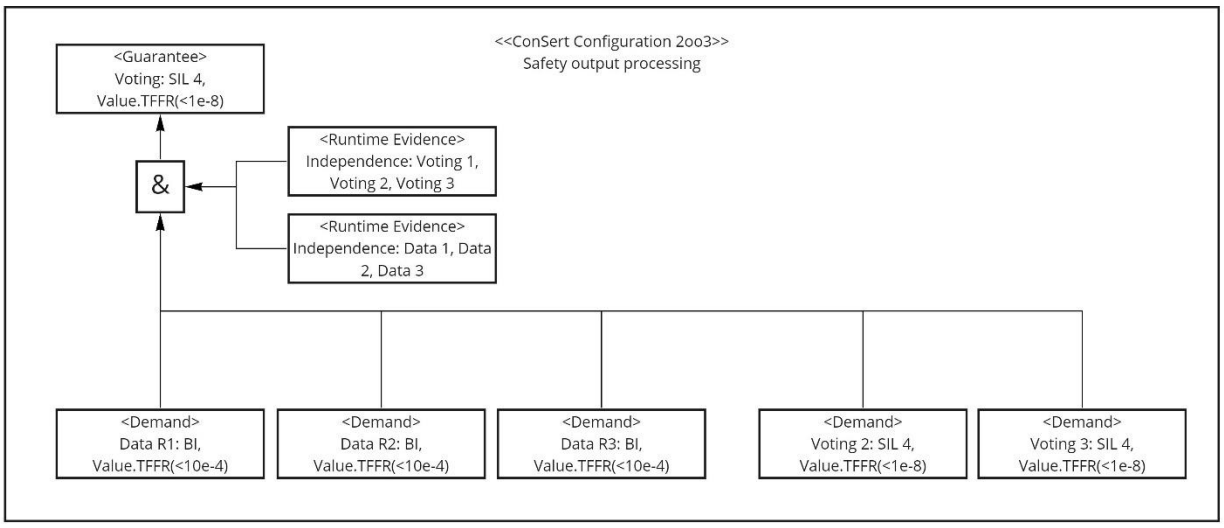

Figure 15: ConSert of Safety output processing

For completeness also the safety requirements for the memory, communication and storage services have to be fulfilled. We will not demonstrate here the argument, how this could be achieved, but rather assume that it will be achieved. Note that safety mechanisms may be used to cover several aspects of the services.

Figure 17 shows the ConSert-model for one instance of a replica. Each replica will have its own model, which will be evaluated at run-time. As the data output service of this replica is only guaranteed with Basic Integrity/SIL0, no further safety measures are specified in the scope of the replica. However, the replica itself requires execution, memory and a communication service for normal operation.

These services are provided by a Computing Element (CE). As this CE is expected to be commercial-off-the-shelf, only Basic Integrity/SIL0 can be expected. Figure 18 illustrates the respective ConSert-model of a CE.

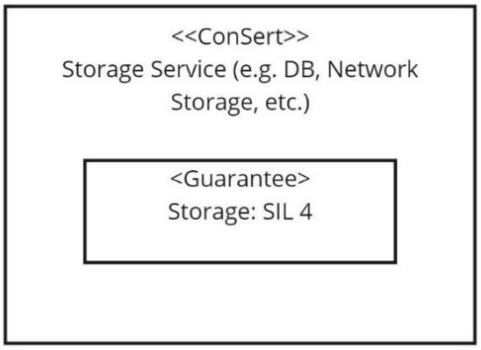

Figure 16: ConSert of Storage

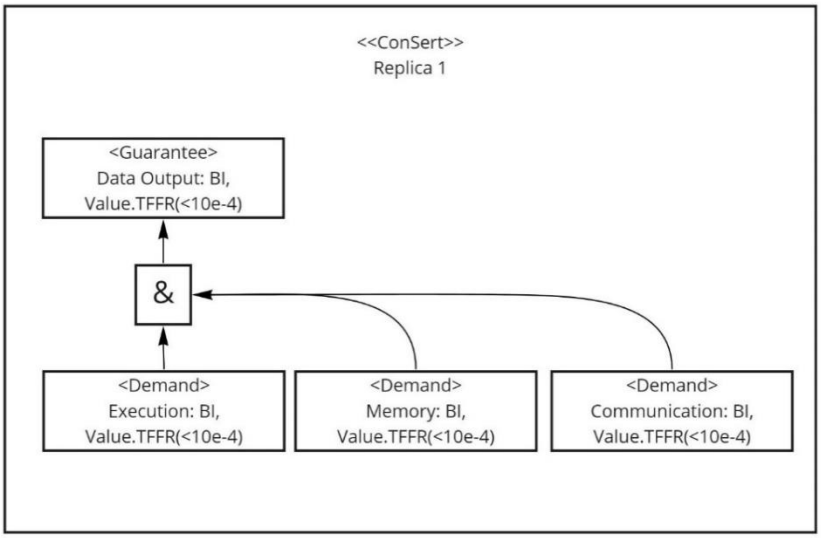

Figure 17: ConSert of a Replica

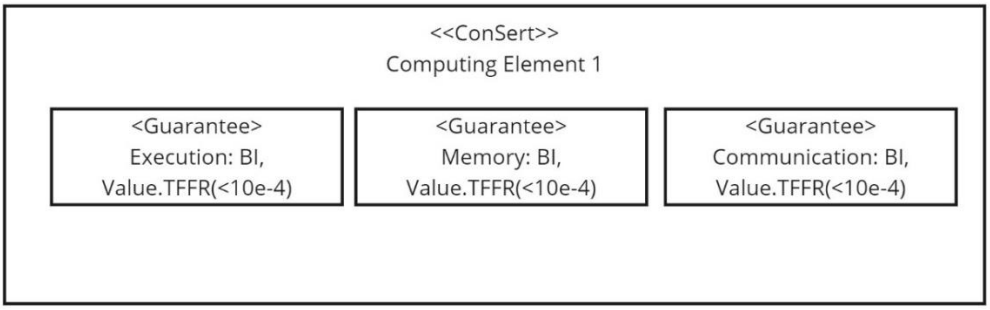

Figure 18: ConSert of a Computing Element

ConSerts can be used to communicate modularised safety requirements across different components of different vendors. Therefore, for each component, a ConSert-model has to be specified, that certifies this component to achieve the requirements under specified conditions.

The fulfilment of the conditions and requirements can be shifted to run-time, where the system can take over the evaluation. This enables an arbitrary number of different constellations and changes during run-time.

4 Communication[5]

As with computation, communication in railway systems is heterogeneous and incorporates safe and unsafe communication. An example for unsafe communication is a command requesting the setting of a route from the automatic train routing system, e.g., the CTMS to the IXL system. This is not safety-critical since the interlocking system guarantees the safe setting of routes.

However, a command from the operator to the IXL system changing the state of a switch marked as occupied is safety-critical. This command overrules the route protection of the IXL system; thus, its safe execution must be guaranteed [9].

For the communication data that is safety-relevant, the communication protocol used on top by the respective safety-relevant applications must be compliant to EN 50159. Errors such as transmission errors, repetitions, deletions, insertions, re-sequencing, corruption, and delays of messages must be considered. So far, most of the standardized protocols in use, developed according to EN 50159, are peer-to-peer protocols involving two communication partners. This is well fitted in the traditional railway architectures that are hierarchically organized and distributed along the railway network. However, the question arises as to which communication protocols will be suitable for the on-premises SIL4 Cloud in the future. To answer this question, the first step is to determine the requirements for the communication infrastructure and to find protocols based on these requirements. Furthermore, we analyse potential application-layer communication protocols used in other domains such as industrial or automotive and find their suitability for railway-specific safety-critical use cases.

4.1 Key Challenges

With regards to communication, following key demands and challenges have been identified in SIL4 Cloud:

Demands from SIL4 Cloud and/or Functional applications	*Challenges for Communication technologies*
Communication over widespread network (between data centres across Germany and EU)	Mix of network (Type, topology, security zones) Realtime constraints (Time-out issues etc.)
Large volume of message exchange among functional application (software), platforms, and other services	Guaranteed high throughput Message delivery ensuring order and content
Zoo of different communication protocols (e.g., OPC UA[6], RaSTA[7], new protocol for Publish-Subscribe support)	Support of protocols by other systems/components (protocol conversion) Interoperability among protocols
Need for dynamism and auto-heal communication features	Safety argumentation
Need for flexible communication patterns (P2P, P2MP, Publish-Subscribe, Broadcast)	Support by procured products, e.g., RTE
Compliance to CCS standard compliance (e.g., EN 50159)	CENELEC certification

[5] Parts of this chapter were first published in [Volume 13294, page number 211, year 2022] by Springer Nature

[6] OPC UA is used in today's CCS systems for collecting diagnostics data (non-safe communication)

[7] RaSTA as standardised protocol is used in today's CCS systems to communicate to external field elements

4.2 Black Channel Communication Concept

In black channel communication, only the endpoints are considered safety-relevant, and the transmission is protected via a special safety protocol. This means that only the safety protocol has to be developed according to safety standard (e.g., IEC 61508-3, EN 50159) and executed in a safe context.

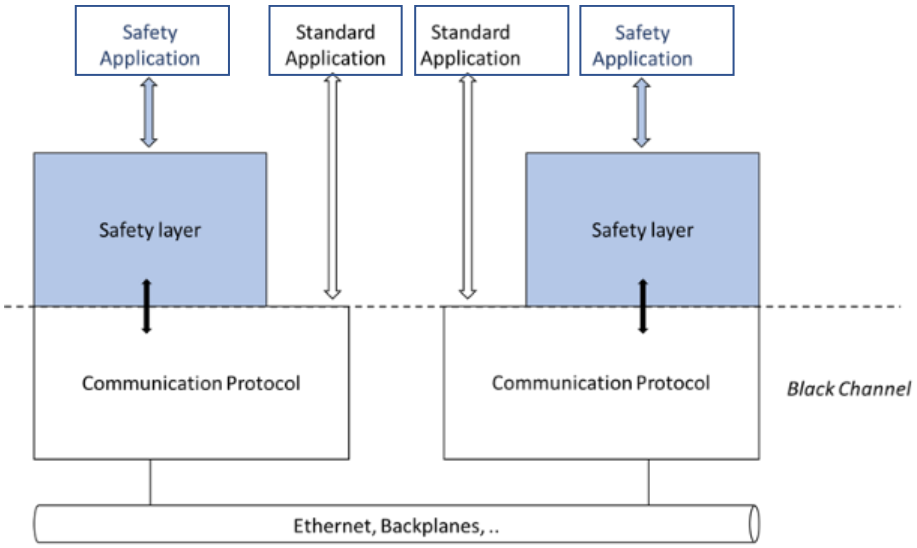

Figure 19: Black channel approach

The name black channel is derived from the concept of a black box. The advantage of a black channel is that the network itself appears as "not being there." The bus system is a non-trusted transmission system, so as a result, the regular network hardware can be reused for safety networks. The safety measures are added as a safety layer on top of the OSI protocol Layer 7. This new layer is responsible for the transport of safety-relevant data. The remainder of the application layer is responsible for the acquisition and processing of user or process data. The safety layer performs safety-related transmission functions and checks the communication to ensure that the integrity of the link meets the assigned SIL requirement.

4.3 Communication Categories

In the railway domain, communication can be grouped into the following five categories, as depicted in Figure 20:

1. Communication within one SCP implementation
2. Communication between two different SCP implementations in the SIL4 Cloud
3. Communication between an SCP and other IT systems within an SIL4 Cloud boundary
4. Communication between a SIL4 Cloud and other IT systems outside of the SIL4 Cloud boundary
5. Communication between an SCP and external systems (e.g., Point machines in the field)

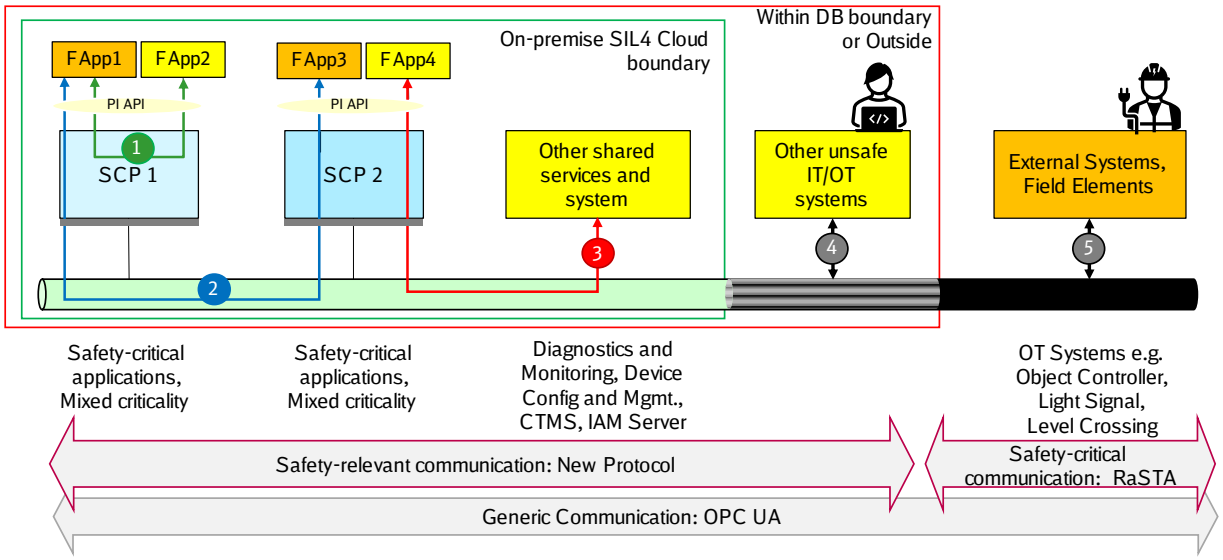

Figure 20: Communication categories

The categorisation is based on the boundaries of communication and there might be different ways of categorizing them. According to the architecture of the SCP, Functional Applications are decoupled from the underlying platform and isolated from each other. The communication between two Functional Applications on the same platform is considered in the domain of the platform vendor, meaning that it can be established via PI API commands with platform as a middleware. The vendor can decide which communication protocol to use, as it is hidden behind the PI API. There is the possibility of distributing the redundant applications, executed on the platform, on different geographical locations. This will be needed in future in SIL4 Cloud (i.e., data centres). Communication protocols are required to be SIL4 capable, if SIL4 applications want to exchange data safely with each other. The same applies to communication with other safety-critical IT systems. Non-safety-critical but nevertheless important communication connections to other systems in the private and external network must be enabled. In today's systems, communication with external systems, such as object controllers or signalling systems, is enabled by the RaSTA communication protocol.

4.4 Requirements for Communication Protocols

Safe communication is subject to EN 50159. This standard defines the requirements for communication between safety-related railway equipment. The key properties for safe communication are:

- Authenticity
- Integrity
- Timeliness
- Sequence

The standard classifies communication network in the following three categories:

Category 1 applies to closed transmission systems with a fixed number of participants, negligible risk of unauthorized access, and static physical characteristics of the transmission system during its life cycle. Categories 2 and 3 deal with open transmission systems, which may have a changing set of participants and possibly unknown participants, that are not part of the railway application, and may generate

- Arbitrary communication loads
- Changing properties of the transmission media
- Changing message routes through the system.

Only in category 3, the open transmission system may also be subject to unauthorized, malicious access. Based on this categorization, the standard identifies possible threats and lists measures and methods that can protect the safety-related communication against these threats. Appropriate measures have to be implemented in an independent layer above the transmission system according to EN 50128 and EN 50129 [10].

According to the OCORA requirements for SCP, the following non-exhaustive list of requirements arise for future communication infrastructure:

COM1: The communication protocol evolves independently from a specific computing platform realization.
COM2: The computing platform shall support point-to-point, point-to-multipoint and publish-subscribe communication model to support different application communication models. The Publish-Subscribe model helps to achieve location transparency for applications running on platform(s).
COM3: Safe communication should be applied end-to-end, so that the whole communication link between remote Functional Applications can be considered safe.
COM4: Safe communication protocols will be transparent to Functional Applications
COM5: The computing platform provides a communication protocol which is based on open and standardized specification to achieve interoperability.

4.5 RaSTA Protocol

Rail Safe Transport Application (RaSTA) is a network protocol that is tailored to the specific needs of railway signalling systems. The special properties of RaSTA include reliable transmission of messages without unnoticed packet loss, monitoring of the channel quality using heartbeat messages, a guaranteed delivery of messages within a time window, and the use of multiple transport channels to increase reliability. Furthermore, it also fulfils the requirements such as message integrity, message authenticity and message sequence, providing protection against unintentional errors. As the MD4 algorithm used by RaSTA is totally outdated, intentional manipulations by an attacker are possible. Protection against hackers can only be reached by using external state-of-the art cryptography (like VPNs).

RaSTA is the transport protocol specified for NEUPRO / EULYNX standardization initiatives. This means, that a broad usage of this protocol can be expected particularly when more and more railway infrastructure managers request fulfilment of this standard. RaSTA is expected to be used heavily to communicate with object-controllers and between several different rail-control systems (e.g., IXL-IXL, IXL-RBC, RBC-RBC).

In SIL4 Cloud, it is required to exchange data among Functional Applications hosted across geo-locations and RaSTA could be one of the straightforward candidates for connecting two geo-located Functional Applications hosted on CN or CG. However, in the preliminary analysis, it was identified that RaSTA could not be the strongest candidate for such communication due to inherent limitations:

P2P: RaSTA is a Point-to-Point protocol with the necessity of static configuration at each endpoint. This means that with RaSTA, it is not possible to add or remove unforeseen connections at runtime.

Cyber security: Cyber security aspects supported by RaSTA are quite weak to defend against modern cyber-attacks. Message integrity and authenticity are ensured only by providing protection against unintentional errors. As the MD4 algorithm used by RaSTA is quite outdated, intentional manipulations by an attacker are possible. This could be overcome by means of additional security devices or protocol-layers or external state-of-the art cryptography (like VPNs), which however would also bring additional overhead to overall system.

Communication overhead: The communication overhead required for heartbeat exchange is relatively high in RaSTA. This means that considerable bandwidth is wasted just to keep the status of the connection up to date.

Network configuration: For the use of networks owned by the operator, the task is to assign unambiguous receiver and sender identifications for the RaSTA instances. This task is very difficult for large networks and poses responsible people with organizational requirements that are hard to fulfil.

The protocol supports safe data transmission in networks classified as category 1 or 2 (according to EN 50159). If transmission over a category 3 network is necessary, additional means of encryption need to be foreseen. This could be within the upper layer (application layer) or the lower layer (e.g., IPsec).

From the experience of railway supplier Thales, it is clear that the use of the protocol in a cloud environment is restricted. In particular, the lack of flexibility of the peer-to-peer protocol and communication restrictions with non-SIL participants are highlighted in this context. Other safe and secure protocols therefore have to be investigated or designed for suitability in the cloud environment, especially for communication between SCPs.

In the following sections, DDS and OPC UA will be presented as potential candidates for such a multipoint communication, since they are widely used in industrial control systems (ICS). Of course, multipoint communication changes some communication paradigms used in current safety relevant system.

4.6 Data Distribution Service (DDS)

Communication middleware such as DDS provide reliable, real-time, interface scale, data scale and new generation industrial IoT architecture for decentralized data communication over the edge-cloud continuum. For using such middleware on an unreliable transport layer, we need to have a safety measure that detects the possible communication errors.

DDS shares and manages data distribution between peers with flexible Quality of Service (QoS) that includes reliability, system health and security. DDS middleware deals with the system and topology change and employs intelligent filters to transmit relevant data to the endpoints. Furthermore, errors such as data corruption, unintended repetition, incorrect sequence number, message loss and delay can be detected and, in some cases, rectified by middleware through the topic/data model design, architecture and configuring QoS parameters such as reliability, liveliness and deadline. For security-critical applications, DDS employs additional features such as access control, data flow path enforcement and data encryption, and can handle integrity issues that may be caused by error types such as message insertion and masquerade.

4.7 OPC UA Safety

Open Platform Communication Unified Architecture (OPC UA) is an upcoming standard in the industrial automation domain provided by the OPC foundation and dedicated to machine-to-machine (M2M) communication. It supports the communication scalability between distributed systems. OPC UA also addresses security issues and provides reliable data transmission between devices. With its flexibility, interoperability, scalability, and other features, it is quickly addressed as the reference standard to meet all the requirements and trends in Industry 4.0. OPC UA provides publish/subscribe over UDP and client/server over TCP communication pattern. OPC UA over TSN provides deterministic communication via Ethernet.

According to the roadmap of OPC UA for 2021 and beyond, the following excerpt of features are under consideration:

- Deterministic communication using 5G
- Additional protocol mappings for deterministic communication, using Wi-Fi 6 and Wi-Fi 7
- Deterministic Layer 3 routing over wired and wireless segments

The specification OPC UA Safety extends OPC UA to fulfil functional safety requirements as defined in IEC 61508 and IEC 61784-3 standards. According to IEC 61508, safety-related applications up to

SIL4 can be built [11]. OPC UA has been developed for M2M communication and is therefore well-suited for communication with field elements and external systems in the context of railway systems. Due to its application-independence, OPC UA Safety does not pose requirements concerning the length or structure of the application data [12].

4.8 One Channel Safe

The fundamental idea of the *One Channel Safe* (OCS) communication concept is to establish a safe communication cannel via one physical link. This channel is established without assumptions on the properties of this link, other than the failure rate and failure behaviour. The application can be certain that data received via an OCS link is correct in the value and time domain.

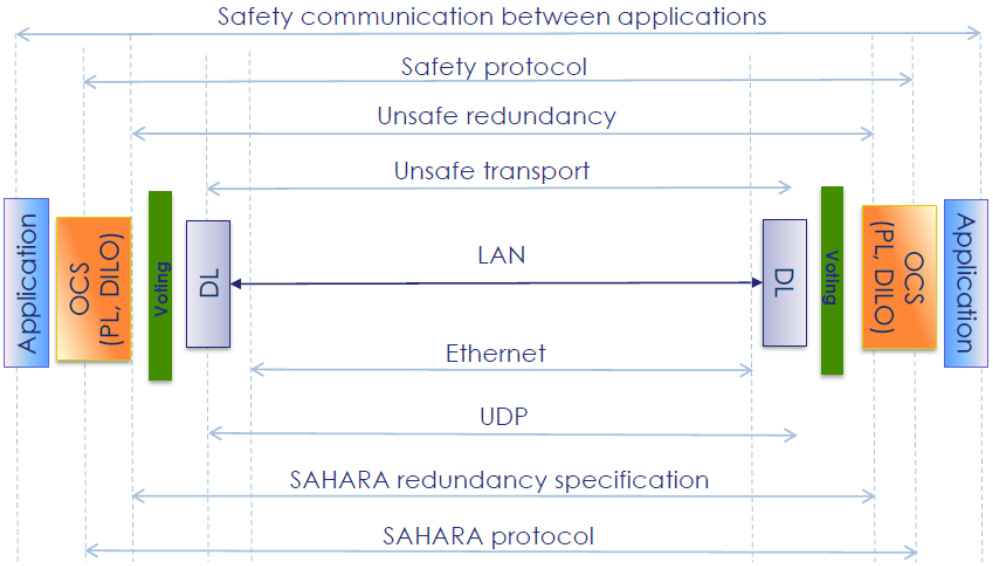

Figure 21: One Channel Safe (OCS) Communication

To reach this level of assurance, the OCS communication must first implement the required measures of EN 50159 in accordance with the nature of the underlying link. Second, the construction of these messages by the sending party has to be consistent with the sending parties' architecture and safety concept. This enables to interconnect applications having different architectures and safety concepts. However, guarantees with respect to timeliness are still achieved by the applications and OCS together and must be treated as such in each application's safety concepts.

The OCS concept can be extended with availability by applying redundancy. Due to the nature of OCS, this is possible either by introducing redundancy transparently on the underlying link or by adding another link, which is handled by OCS itself. In the latter approach, the messages can be sent via one or both links, depending on the replication model. Due to the safety property according to EN 50159 of OCS messages (sequence number, ordering, delay, etc.), duplicate messages received on different links can simply be ignored.

Depending on the architecture used and the safety concept, different approaches may be used for safe message creation. It is important to note that voting and safety code creation before sending the message may not be sufficient, as the voter might fail. One approach for replica-deterministic 2oo2 or 2oo3 configurations is to ensure that the message sent via the network contains at least artifacts of two replicas. That way, no replica on its own can create a valid safety code, and the detection of faults is ensured. [9]

4.9 Safe Communication Architecture using Separation Kernel approach

This section describes the design and architecture for safe communication on a black channel, applicable for safety-critical applications deployed on distributed data centre environment.

Figure 22 shows a system architecture, where communication middleware such as DDS or OPC UA runs in a separate safety-critical partition with POSIX runtime along with the safety application. TCP/IP stack and Ethernet driver run inside a non-safety-critical virtualized Linux partition, which provides the black channel. The black channel stack running inside the non-safety-critical Linux partition implements a software switch that either directly transfers the data by memory copy to the destination port, when it is located on the same CPU, or calling a socket write using an Ethernet driver, when it is on a different CPU.

Separation kernel isolates the safety-critical partition from non-safety-critical partition, and applications running inside these partitions are allowed to communicate via inter-partition queuing/sampling ports provided by the separation kernel. Due to the strong separation provided by qualified separation kernel, the certification efforts can be limited to the components in the safety-critical partition and an unqualified, off-the-shelf network stack can be used inside non-safety-critical partition.

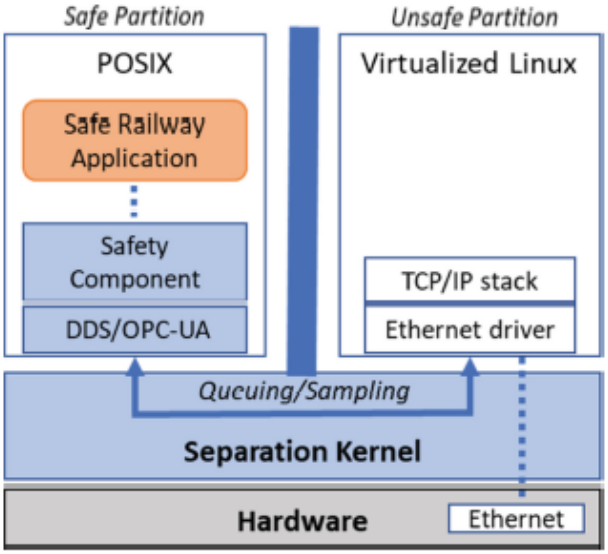

Figure 22: Example of a safe communication system architecture based on DDS/OPC UA and Separation Kernel

4.10 Comparison of Potential Protocols relevant for Safety-Critical Systems

Table 4 presents a comparison of the three protocols RaSTA, DDS and OPC UA.

Table 4: Comparison of safe protocols

	RaSTA	DDS	OPC UA Safety
Basic architecture	RaSTA is a centred around client-server	DDS abstraction is centred around a decentralized data space that decouples applications in time and space.	OPC UA abstraction is centred around client-server
Publish-Subscribe architecture	No, P2P	Yes, P2P and P2M can be implementable	Yes, P2P and P2M can be implementable
App-to-App interaction		DDS applications interact by anonymously and asynchronously reading and writing data in the global data space.	OPC UA applications interact by invoking requests on one or more UA servers
Location Transparency	No location transparency as	DDS applications feature complete location transparency. Data	OPC UA applications have to undergo a two-step resolution process: first, they need

	RaSTA	DDS	OPC UA Safety
	statically configured	goes automatically to where there is interest.	to look up servers and, second, browse data in their address space.
EN 50159 key properties (Authenticity, Integrity, Timeliness, Sequence)	Supported	Supported	Supported
Open Standard with strong international support	No (Used mostly in Railway industry)	Yes	Yes
Usage in existing railway systems/products	Heavily used in IXL and RBC products	No references found to any implementation	OPC UA is used for non-safe communications (e.g., collecting diagnostics data)
Safety Features (excerpt)	• Black channel principle • Detection of communication errors	• Black channel principle • Changing communication partner during runtime • Detection of communication errors	• Black channel principle • Changing communication partner during runtime • Detection of communication errors • Safety Multicast
Security features (excerpt)	Limited (Secure Code)	Extensive (Data encryption, Comprehensive information-centric security framework for data in movement as well as data at rest)	Adequate (Secure Channel with OPC UA Security)
QoS	-	DDS allows information to be annotated with Quality of Service (20+ configurable QoS policies), so as to capture non-functional properties.	OPC UA does not support QoS specification.
Standard compliance	EN 50159, EULYNX support	No info	No info

It would be fair to say that the current, state-of-the-art protocol is RaSTA. However, considering the demands from novel functional applications and DB's needs, it is worth trying, and analysing in detail, the more modern, multi-point supporting communication protocols.

5 Cyber Security of SIL4 Cloud

Railways being a part of critical infrastructure (KRITIS), security is the area of common concern surrounding the protection of systems, networks and assets, whose continuous operation is deemed necessary for ensuring the security of railways, and the health and/or safety of the public. In this chapter, a high-level risk analysis is carried out concerning SIL4 Cloud.

5.1 Security approach for generic systems

The usual approach to develop a security concept for a new system would be following the security lifecycle defined by IEC 62443-3-2. See Table 5 for a mapping of cybersecurity activities on the phases of the lifecycle according to EN 50126.

Table 5: Mapping of cybersecurity activities on the phases of the lifecycle according to EN 50126

Phase according to EN 50126		Phase according to IEC 62443-3-2	Cyber security activities	Synchronization points and deliverables (→ Input Documents, ← Output/Deliverables)
0	Pre-requisites		Manufacturer's and integrator's secure development process is established	→ Railway operator's security program is established → Legal and regulatory framework is identified
1	Concept	IEC 62443-3-2 ZCR-1 Identify the SuC	Review of the level of security achieved up to now Analysis of the project's security implication and context (incl. generic threats) (see 5.4) Alignment with railway operator / asset owner and stakeholder's security goals Consideration of security life cycle aspects (patch management, monitoring etc.) (see Clause 10) Plan cyber security-related activities	→ Operational environment incl. existing security-related controls and High-Level zone model (see Ch. 4) → Applicable security standards → Purpose and scope ← Project cybersecurity management plan (incl. cybersecurity context, goals and lifecycle activities (see Annex G.2)
2	System definition and operational context	IEC 62443-3-2 ZCR-2 Initial cyber security risk assessment	Initial Risk Assessment for the SuC (see 6.3, may also be conducted in phase 3)	System definition: → System boundaries → Initial system architecture, incl. list of functions, interfaces and generic systems → Logical and physical network plans ← Logical and physical network plans review Operational context and criticality:
		IEC 62443-3-2 ZCR-3 Partition the SuC into zones and conduits	Review of the logical and physical network plans Partitioning of the SuC into zones and conduits (see 6.4, may also be conducted in phase 3)	

Phase according to EN 50126		Phase according to IEC 62443-3-2	Cyber security activities	Synchronization points and deliverables (→ Input Documents, ← Output/Deliverables)
			Documentation of components, interfaces and characteristics for each zone and conduit (see 6.5, may also be conducted in phase 3)	→ Essential functions ← Initial risk analysis results ← Zones and conduits
		IEC 62443-3-2 ZCR-4 Risk comparison	It is assumed that for railway applications, a detailed risk assessment is almost always necessary. In the event that the outcome from the initial risk assessment is that all risks are sufficiently mitigated without any additional countermeasures (e.g., because there is very strong physical and organizational protection, see ZCR4 of EN IEC 62443-3-2), the detailed risk assessment could be skipped but for the documentation of the cyber security requirements, see 7.2.5.	
3	Risk analysis and evaluation	IEC 62443-3-2 ZCR-5 Perform a detailed cyber security risk assessment -> Definition of the security level (SL-T)	Detailed Risk Assessment (DRA) (see Clause 7) Derive technical (e.g., SL-T), physical and organizational countermeasures or assumptions for zones and conduits Consider business continuity aspects (incl. incidence response and recovery) for the SuC	DRA: → Functional Requirements (linked to essential functions) ← Initial Threat Log ← Potential Updates (zones and conduits, network plans)
4	Specification of system requirements	IEC 62443-3-2 ZCR-6 Document cyber security requirements, assumptions, and constraints -> Definition of system security requirements according to IEC 62443-3-3 based on SL-T from previous phase	SuC-specific refinement of normative requirements (see Clause 8) Definition of organizational and physical requirements Definition of security-related application conditions (see Clause 7)	CRS release: ← System Cybersecurity Requirements Specification (CRS) incl. security-related application conditions

Phase according to EN 50126		Phase according to IEC 62443-3-2	Cyber security activities	Synchronization points and deliverables (→ Input Documents, ← Output/Deliverables)
5	Architecture and apportionment of system requirements	-> Break down of system requirements to subsystems -> Definition of compensating countermeasures	DRA update, incl. assessment of the SL-C for components and definition of compensating countermeasures including security-related application condition (see 8.3) Assigning the technical security requirements to components (see 8.3) Assigning responsibilities for the organizational and physical requirements for the railway operator/maintainer or asset owner Establish third party management for security aspects, including supplier security capabilities and support contracts	CRS breakdown: → System architecture breakdown to components, incl. SuC inventory ← Subsystem Cybersecurity Requirements Specification incl. security-related application conditions, technical and organizational compensating countermeasures
6	Design and implementation	Development of subsystems and component according to product lifecycle of IEC 62443-4-1 and 4-2	Follow product lifecycle (EN IEC 62443-4-1 and 4-2 or other adequate cybersecurity standards) Consider and prevent potential conflicts between component cyber security functionality and functional architecture	No synchronization points defined
7	Manufacture or procurement	Development of subsystems and component according to product lifecycle of IEC 62443-4-1 and 4-2	Follow product lifecycle (EN IEC 62443-4-1 and 4-2 or other adequate cybersecurity standards) Consider and prevent potential conflicts between component cybersecurity functionality and functional architecture	No synchronization points defined

Unfortunately, precise information about the system's use case and its operational environment (cyber security context defined by the operator) are crucial input for many of the cyber security activities listed, e.g.:

Establishing the project's security goals requires input from the asset owner (usually the operator), e.g., operator's security program, CIA (Confidentiality, Integrity, Availability) Classification, etc.

Performing the cyber security risk assessment requires knowledge about the specific application and its operational environment, which also determines the project's threat context and attack surface.

Evaluating risks requires risk acceptance criteria and a risk matrix, which are usually defined by the asset owner.

As the SuC is currently on a very generic development level, much of the required input information is not yet available. Thus, it is not possible to derive an adequate Target Security Level (SL-T) for a specific application, develop a suitable security architecture, define zones and conduits and to derive the cyber security requirements.

However, for a generic system, the interface between the cyber security risk assessment (phases 1-3) and design activities (phases 4 ff.) may be reduced to the SL-T. That means it is in principle possible to design a generic system only based on a given SL-T by using the system security requirements catalogue provided in IEC 62443-3-3. The same is true for subsystem and component requirements if using the security requirements catalogue provided in IEC 62443-4-2. System design and implementation may then be conducted according to the product lifecycle of IEC 62443-4-1, which also uses a given SL-T as major input.

Thus said, the security concept for the SIL4 Cloud may be derived from a preliminary SL-T, which has been postulated or derived from some example applications. But several aspects must be kept in mind:

For a particular application all relevant assets must be included in the definition of the SuC (System under Consideration). This includes not only the application itself operated in the SIL4 Cloud, but also all interfaces to external systems like trackside components, user workstations and operator systems as well as external networks needed to communicate with them. The risk analysis for this complete SuC might result in a higher SL-T leading to elevated security requirements for the SIL4 Cloud also.

As the SIL4 Cloud should be able to host several functional applications, which might have different criticality, the SL-T of these applications and the resulting security requirements for the SIL4 Cloud might also differ. This could be handled by splitting up the SIL4 Cloud into separated independent instances suitable for different SL-Ts. But considering, that the SIL of the applications might also differ and suggests separate SCP instances, this approach might lead to so many SCP instances, that the hoped-for efficiency effects are totally lost. Therefore, we assume that the SIL4 Cloud should be suitable to fulfil the security requirements of all applications hosted.

Beside the different security requirements / SL-T for the individual applications we also have to consider cumulative effects. If risks relevant for the SIL4 Cloud affect not only a single application, but all or most applications hosted on the SCP the impact might be much higher than the maximum impact from one application. This is especially true for threats against the availability of the SCP and the essential common services. Therefore, the resulting SL-T for the SCP might even be higher than the maximum SL-T of any single application. It is recommended that the SL-T for the SIL4 Cloud must be reassessed when further information is available regarding all applications hosted on the SCP and their security requirements.

5.2 Preliminary SL-T for the SIL4 Cloud

Within the EU research project X2Rail, risk analyses for a generic high-level trackside architecture derived from ETCS and EULYNX architecture models have been conducted. For the trackside zones they resulted consistently in a SL-T of 3, with all foundational requirements (FR) at level 3. These results have also been used to develop a generic Protection Profile for trackside components based on the IEC 62443-4-2 requirements, which is suitable for establishing this security level. Furthermore, it is shown that the common security services are mandatory or highly recommended to fulfil the security requirements for the trackside zones. For more details, see [X2RAIL-1 D8.7], [X2RAIL-3 D8.2-2] and [X2RAIL-3 D8.2-3b].

From this work, it seems plausible to choose a (preliminary) SL-T of 3 for the SIL4 Cloud. These risk analyses are, however, based on generic assumptions, not taking into account the Cyber Security Management System at railway infrastructure manager (e.g., Deutsche Bahn / DB Netz) and their specifications for risk assessments, risk tolerance level and risk reduction guidelines. Furthermore, the hosting of several railway applications on the SIL4 Cloud may lead to cumulation effects and to an even higher SL-T of 4.

For the final risk analysis, the following aspects should be considered:

- In the longer term, the SIL4 Cloud will host applications essential for safe railway operation in Germany, being part of Germany's critical infrastructure. Therefore, potential threat actors might also include attackers with highest resources (skill, money) and motivation, up to state-sponsored hackers.
- In the context of applications providing safety-relevant functionality with SIL requirements up to SIL4, one of the most important objectives for security in SCP must be to provide a secure environment for these applications and to ensure the availability and integrity of the safety functions provided by these applications.

Beside this, there are additional and/or concurring security objectives to be considered:

- Security should also protect the availability and integrity of the business process supported by the applications hosted on the SIL4 Cloud (i.e., in general railway operation). This might lead to a paradox: If an attacker can trigger the safety function adversely (e.g., by denial-of-service attacks or tampering of messages), this might in the end lead to a breakdown of railway operation. Even if the safety is still ensured, the attack is successful from a security point of view.
- Confidentiality and the protection of intellectual property might also be an important objective within SCP, as applications and RTEs from different vendors will be operated on a shared environment. While these applications were operated on separated infrastructure in the past, now there might be the risk that one vendor gets access to classified information of other vendors when providing services.

5.3 Protection needs and SL-T for the SIL4 Cloud

Eventually, the protection needs and the SL-T for a SCP in a productive environment must be determined based on the protection needs and the SL-T of the specific applications hosted by the SCP instance under consideration. Furthermore, it may not be sufficient to just define the highest SL-T of the hosted applications as *the* SL-T for the SCP. Due to cumulative effects, the SL-T of the SCP may in fact be higher than that of any single hosted application.

Take the case of a SCP running the interlocking for five medium-size stations. The loss of service for each of the interlockings has been rated to be below a critical threshold by the operator, because it only has impact on a local scale. However, failure of the SCP disables interlocking on all five stations, resulting in a loss of service on a supra-regional scale. The impact of a failure of the SCP is much more

severe than the failure of only one of the hosted applications. Thus, the event may have consequences lying above the critical threshold, resulting in higher protection needs and potentially a higher SL-T for the SCP.

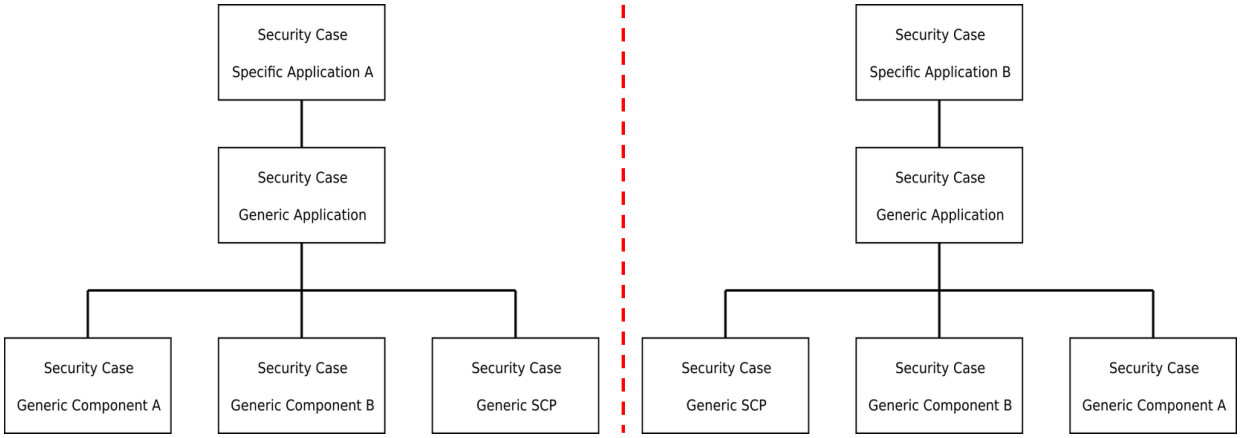

Figure 23: Security cases established for specific applications without interdependencies and not sharing any components.

The aforementioned has consequences for establishing the security case too. In the well-established approach of EN 50126, the safety case is usually established for a specific application. In this approach, the SCP would be treated as a generic component coming with a generic safety case, just as it is usually done for any generic software or hardware component that is part of the application. The safety cases for Specific Application A would be established completely independent of the safety case for the Specific Application B (See Figure 23). Any change to any part of the Specific Application A requires an update of the whole causal chain that safety case A is based upon, but it would not affect the safety case for Specific Application B in any way.

However, this approach only holds as long as application A and application B are indeed independent of each other, i.e., they do not share any component or service. For two specific applications deployed on the very same SCP, it is no longer true (See Figure 24). For the *specific SCP*, the protection needs must be defined based on those of the specific application, as well as by considering the cybersecurity context of the wider railway system. This requires that all risks identified for each of the hosted applications—irrespective of whether they have been rated as critical or not — are collected and re-evaluated with respect to the railway system that the individual applications are part of.

Due to said cumulative effects, the protection needs for the specific SCP may go beyond the protection needs of the individual applications hosted thereon. As a consequence, the overall SL-T for the specific SCP may be higher than the SL-T for any of the individual applications. In essence, it has to be expected that an SL-T of 3 will most likely not be sufficient for a SCP running multiple instances of highly critical applications like interlockings.

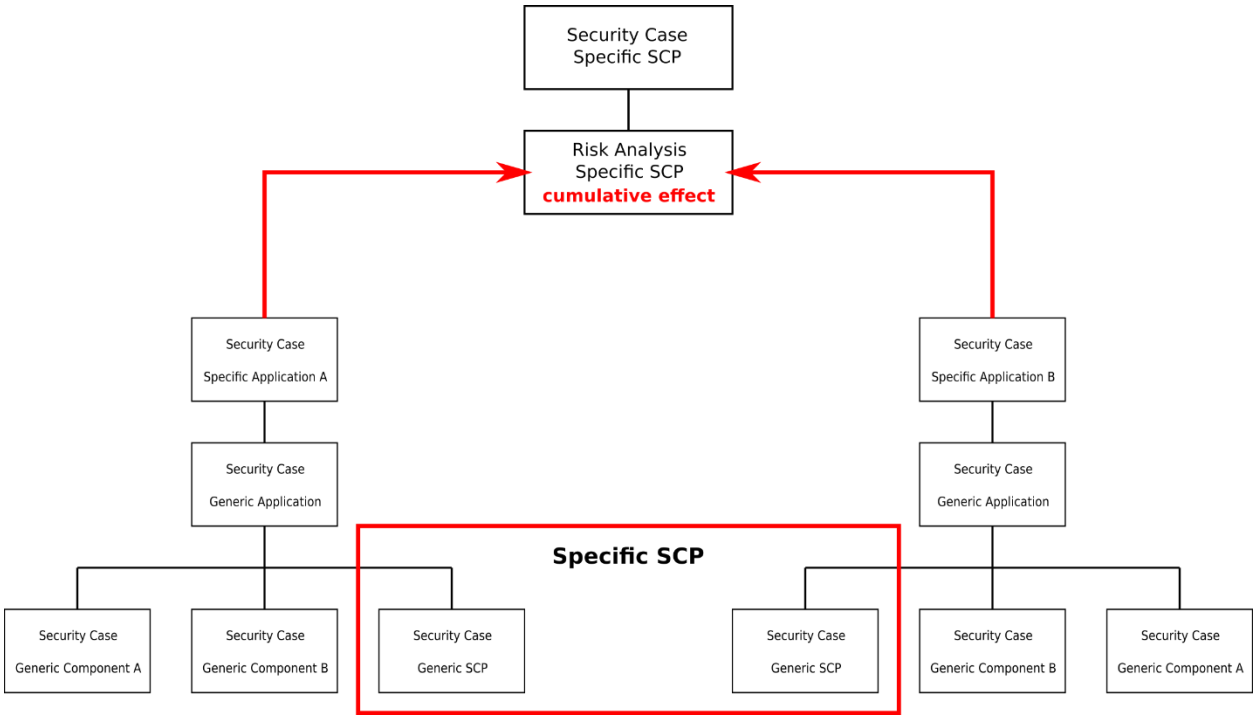

Figure 24: Additional effort required for specific applications being hosted on the same SCP

5.4 Essential security requirements for the SIL4 Cloud

To fulfil the security requirements for the SIL4 Cloud, the following conditions must be met:

Requirement (SEC1)	The SCP data centres must be segregated and private without any non-SCP usage. *(To be complied by IM)*
Explanation	The infrastructure for the SIL4 Cloud must be separated from any non-SCP infrastructure in terms of physical infrastructure, hardware, networks, and software. Therefore, the SCP data centres should be segregated from any other data centre (or be a completely segregated part of a larger data centre) and should not share any common infrastructure with others or even public data centres. Any asset in the SIL4 Cloud and the SCP data centre must fulfil the SCP security requirements.
Rationale	Any component within the SIL4 Cloud not fulfilling the SCP security requirements or having an unknown security status may impose an unpredictable risk for the security and safety of the applications hosted within the SIL4 Cloud. Even when using advanced security controls (e.g., advanced cryptographic methods to protect the confidentiality and integrity of SCP system and data, advanced access controls, and advanced (logical) network segregation and data flow control), availability of the SCP platform and services remains the main issue. As many incidents and vulnerabilities have shown, errors in software or misconfiguration of assets may lead to impacts on the availability of an underlying shared infrastructure.

Requirement (SEC2)	The SCP data centres must be under full control and responsibility of the operator of the railway grid. *(To be complied by IM)*
Explanation	The SCP data centres are under full control and responsibility of one body. This body should be the operator of the railway grid (e.g., DB Netz AG) or a service provider commissioned and supervised by it.
Rationale	As the operator of the railway grid has the ultimate responsibility for the safety and security of the railway grid, and the SIL4 Cloud provides essential functions and services for the safe and secure operation, the operator of the railway grid must have full control over the SCP data centres. This implies that he must set the security controls and enforce and monitor their implementation. This does not limit his right to commission a service provider to operate the SCP data centres, as long as he still carries out this responsibility and supervises the service provider accordingly.

Requirement (SEC3)	The inter-connectivity network should be a private network under full control and responsibility of the operator of the railway grid. *(To be complied by IM)*
Explanation	The network connecting the SCP data centres should also be under full control and responsibility of the operator of the railway grid (e.g., DB Netz AG). Therefore, it should be a non-shared, private network.
Rationale	The security and availability of the inter-connecting network is essential for the security and availability of the applications hosted within the SIL4 Cloud. This is especially true for safety-relevant applications using a geographically distributed configuration providing essential services for railway operation. The unavailability of network communication between the data centres could trigger a safety condition and lead to a breakdown of railway operation. When using shared or public networks, the confidentiality and integrity of the data can be protected sufficiently by using advanced cryptographic methods. Also, the SCP data centres, and internal networks can be protected against illegitimate access from the inter-connectivity network. But, as many incidents have shown, the availability of a shared network can be affected by other users, even when the operator has protecting measures (like bandwidth control) in place. On the other hand, using private networks to inter-connect the SCP data centres is, in our opinion, a simple, feasible and affordable measure to mitigate this risk.
Remark	In our opinion, the connection to a part of the field elements may be via shared/common networks as long as the impact of an unavailability of these networks on the overall train operation is limited. A careful risk analysis should be carried out before doing so.

Requirement (SEC4)	All components within the SuC must have an SL-C (capability security level) of at least 3. Ideally, they fulfil the requirements of the protection profile for trackside components described in [X2RAIL-3 D8.2-3b]. *(To be complied by RTE vendor)*

Explanation	The SL-C defines the security level a component or system can provide when properly configured. If the SL-C does not meet the target security level (SL-T), additional compensating controls are necessary. The EU research project X2Rail has developed a protection profile for trackside components based on the generic trackside architecture and risk analysis. This protection profile ensures an SL-C of 3 and further specifies the implementation of the security requirements, the usage of the shared security services, and measures to ensure inter-operability.
Rationale	To fulfil the security requirements and the SL-T of 3 in all zones of the SuC, all components within the SuC must have a SL-C of 3. Although IEC 62443 allows for compensating measures, when a component does not meet the security requirements of SL3, in the complex, heterogeneous SIL4 Cloud, this can lead to a situation which is difficult to oversee. Therefore, at the current state of planning, this option should not be taken up, unless it is really necessary. Currently we see this as a given only for legacy applications (see below).

Requirement (SEC5)	The RTE might provide compensating measures for applications to ensure that RTE and application have a resulting SL-C of 3. (*To be complied by RTE vendor)*
Explanation	Some applications intended to be run on the SCP might not fulfil all security requirements needed for an SL-C of 3. This will be the case particularly for most of the legacy applications. In such cases, the RTE should provide compensating measures, so that RTE and application together meet the security requirements. Example: SL3 requires unique identification of all users and additional measures to ensure the authentication and the use of strong passwords. While a new developed application will simply integrate with the central IAM service to fulfil this requirement, for existing / legacy applications, the integration of the IAM service or the fulfilment of this requirements may require substantial changes and in consequence the need to undergo the homologation process again. In such a case, the RTE could integrate with the IAM service, provide the individual authentication of the user, and forward the authenticated user to the application itself.
Rationale	To ensure the security of the whole SIL4 Cloud, all applications must meet the defined security requirements. If this is not possible for an application, compensating measures must be taken. These measures may be provided by the RTE, which hosts this application.

Requirement (SEC6)	All components/products used within the SuC must be certified ensuring the required SL-C. *(To be complied by IM)*
Explanation	The proof of evidence that the needed security requirements are met can be provided in two ways: The component/product is certified according to IEC 62443-4-2 ensuring the required SL-C of 3, ideally including the X2Rail protection profile for trackside components.

	The component/product is certified according to common criteria, EAL 4+ and an additional matching table is made available showing that the security requirements of SL3 were part of the specification of the component/product.
Rationale	In the complex and heterogenous SIL4 Cloud, an individual security assessment of each component by the operator will be too elaborate. Therefore, only certified products should be used. While the best way is to provide a certification according to IEC 62443-4-2, this will not be available for many COTS products planned to be used to form the SIL4 Cloud itself. Therefore, we consider a certification according to common criteria with EAL4+ as sufficient, as long as a proof is provided that the security requirements of SL3 were part of the specification of the component/product.

Requirement (SEC7)	The network infrastructure provides secure communication channels, which must be considered as black channels from a safety point. *(To be complied by IM)*
Explanation	The network infrastructure of the SuC (i.e., the inter-connectivity network and the internal networks of the SCP data centres) provides secure communication channels for the: • communication of RTE/application/services within one data centre • communication of RTE/application/services from one location to another Additionally, interfaces are provided to connect to field elements. The external communication is not in the scope of this project.
Rationale	The confidentiality and integrity of the communication channels and appropriate protection of the endpoints can be achieved by security controls available within the SuC, e.g.: • advanced cryptographic methods • advanced access controls • advanced (logical) network segregation • advanced network data flow control Availability and adherence to timing constraints are much more difficult to ensure in the context of the heterogenous SIL4 Cloud, especially when it comes to typical requirements of safety-relevant applications. Therefore, the application should consider the network channels provided as black channels and implement appropriate measures to detect and handle the loss or lateness of data packages to always maintain a safe state. This, however, might conflict with availability objectives of the application. If an application is not capable of using black channels on its own (e.g., legacy applications), the RTE should provide appropriate support.

Requirement (SEC8)	All components (including the applications and RTEs) must be integrated with the common security services of the SIL4 Cloud.

Explanation	The EU research project X2Rail has specified the following common or shared security services (see [X2RAIL-3 D8.2-2]): • System-wide time service (TIME) • Central logging (LOG) • Identity and Access Management (IAM) • Backup (BKP) • Asset inventory (INV) • Intrusion detection/continuous security monitoring (IDS) • Security Incident and Event Management (SIEM) • Public Key Management (PKI) • SW Update (SWU) All components within the SIL4 Cloud must be integrated with these services. The RTE might provide the application with interfaces to these services, when direct integration is not possible, e.g., in the case of legacy applications.
Rationale	The EU research project X2Rail has provided evidence, that the common security services are mandatory or highly recommended for fulfilling the security requirements for trackside zones in the generic railway architecture. For more details see [X2RAIL-1 D8.7], [X2RAIL-3 D8.2-2] and [X2RAIL-3 D8.2-3b].

5.5 Freedom from interference

The three main goals of security are confidentiality, integrity and availability, i.e., CIA. In this section, we will discuss aspects of the availability. Availability is one of the key security goals. When freedom of interference is not provided, security issues in one application could lead to non-availability/latencies in other applications. This is especially true for safety-relevant applications that might lead to a high impact, when the system goes down to the safe state.

Assume we have a simple cruise control application train-side. As long this application receives a signed “safe cruise” (SC) message from trackside any two seconds, it will not initiate an emergency brake. In normal operation mode, the trackside sends a SC message every second to the train.

In the safety world, this example is safe.

- If the trackside decides that it is not safe to go on driving, it can trigger an emergency brake just by stopping the sending of SC messages to the train. In this case, the cruise control will initiate an emergency brake after at most two seconds.
- If for some reasons the messages are getting lost, the train cruise control will initiate an emergency brake after two seconds.

The result of both events is that we reach the safe state. From the safety point of view, this behaviour is good. In contrast, from the security point of view, this behaviour could harm the security goal availability. If an attacker is able to drop / delay the SC messages, s/he could enforce an emergency brake and take down the availability.

Applications use many different resources while computing:

- CPU
- Main memory

- I/O devices
- Background Memory (hard disks etc.)
- Communication (network)

In multithreaded and multicore environments, the applications have to share these resources. If an application wants to use an exhausted resource, it has to wait until the resource is available again. For example, if an application could use 100% of all CPU cycles, this would harm the availability (maybe a security goal for a specific application) of all other applications. Modern Operating Systems like Windows or Linux can restrict the access to these resources. An application's access to the main memory can be restricted by the use of a paging mechanism, which can also protect applications' memory spaces from each other. The use of networking resources can be partitioned with some kind of traffic shaping and other IO workload can be partitioned in a similar way. For network traffic, the operating system can control only outbound traffic.

Another key resource here is the CPU cache. The gap between computer processing speed and main memory access is today more than 1,000x. In Figure 25, we can see a modern CPU processor and cache architecture and its connection to the main memory. The CPU in the figure has four cores and each has two virtual processors (e.g., Intel hyperthreading). The virtual processors inside a core share the L1 and L2 caches. All cores inside a CPU package share the L3 cache. The cache is small in comparison to the main memory. Only a small portion of the main memory can be mapped into the cache. If an application requests data which is not present inside the cache, an old value inside the cache will removed. Afterwards the cache will be filled with the requested data.

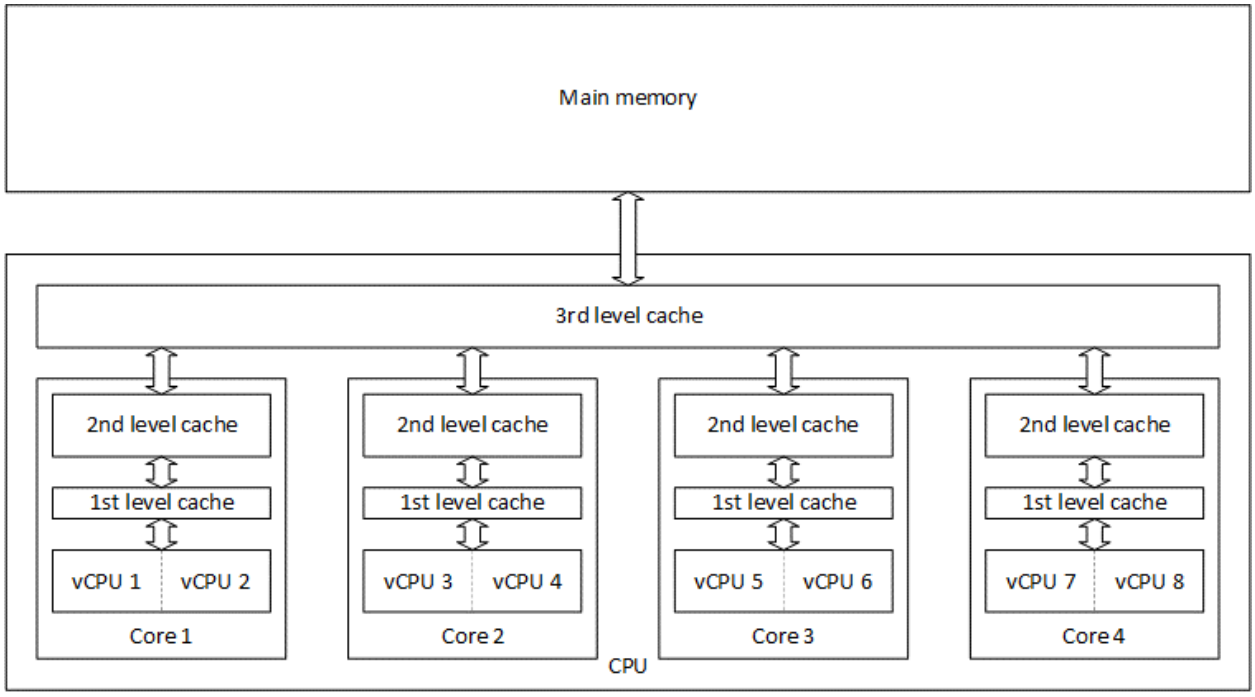

Figure 25: Modern CPU architecture with its caches and the main memory

On multiprocessors, different applications use the cache concurrently. In this environment, an application can influence the calculation speed of another application on the CPU without accessing the other's application data. This opens an attack surface on multicore CPUs. Assume we have the real-time application App 1 and a malicious (or erroneous) application App 2. Both applications are running on the same CPU at the same time. App 2's purpose is a denial-of-service attack on the cache. This could easily be reached by loading many different addresses from the main memory. Any single load operation leads into a substitution of a current value inside the cache system. While the attack is running, many data requests will result into a cache miss. If a cache miss for App 1 occurs, the requested data must be transferred from the main memory to the cache again. While the CPU is

waiting for this data, it cannot proceed with its calculation. This attack could reduce the performance of the victim application by up to 90% [13]. Without any hardware support, a real-time OS can only guarantee a fraction of CPU time to an application. The application can use the CPU for this fraction of time but stalls any couple of instructions.

Some embedded processors can partition the cache to single applications. On some Intel processors with amd64 instruction set (COTS hardware), we can find a technology called Cache Allocation Technology (CAT) [14]. However, CAT only allows controlling the last level cache and it is Intel-specific. Applications on the same core still share the 1st and 2nd level cache.

If technologies like hyperthreading are in use, there are some more shared resources (e.g., calculation units, register file). Some other possible attacks [15] are side channel attacks against the victim's secret data. If technologies like hyperthreading are in use, there are some more shared resources (e.g., calculation units, register file). Some other possible attacks [15] are side channel attacks against the victim's secret data.

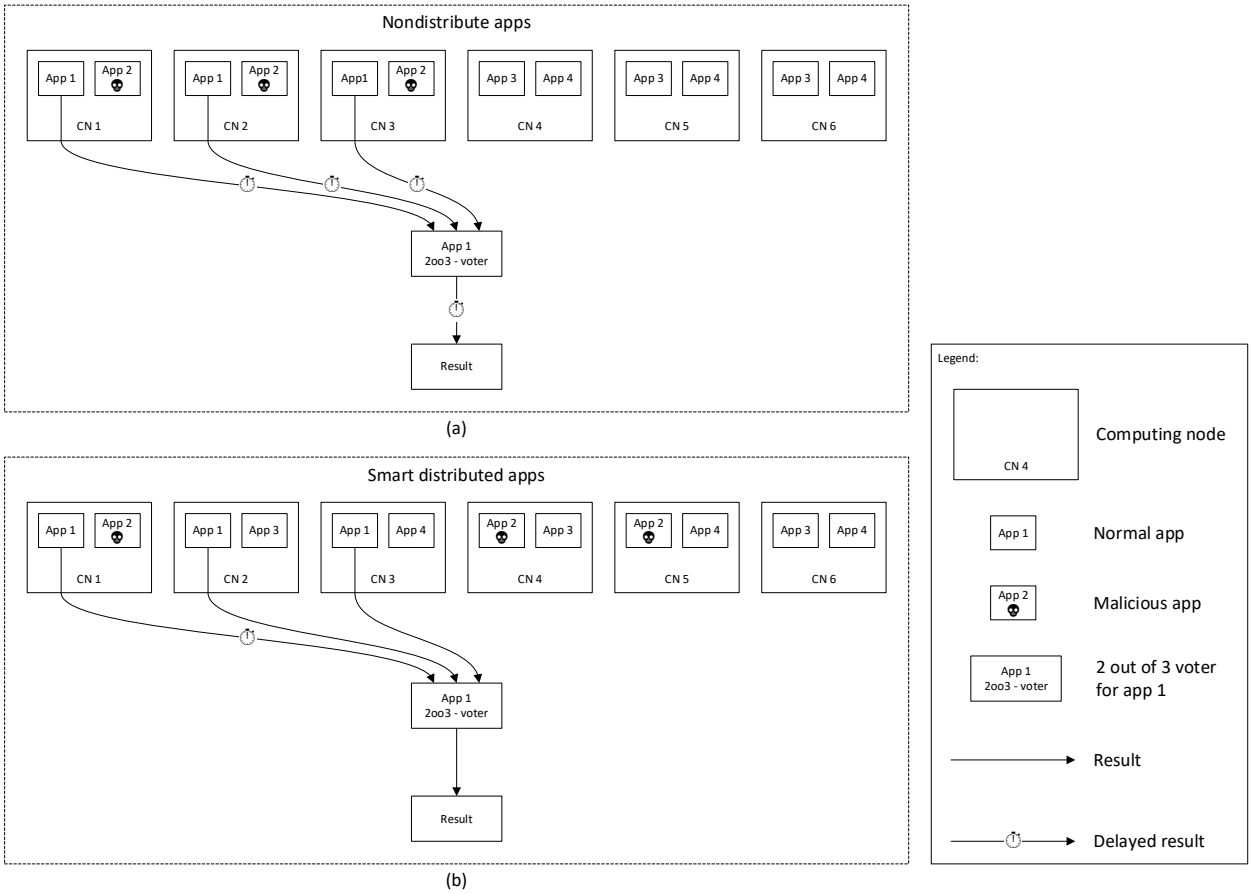

Figure 26: Distribution of applications to mitigate some DoS cache attacks[8]

The denial-of-service (DoS) attacks can be addressed with a special distribution of applications. For example, we have a computing cluster with six computing elements (CE) and each CE can run two applications at the same time. We have the applications 1 to 4 and use a 2oo3 voter for safety reasons. For simplification, in Figure 26 (a), only one voter is shown for App 1 outside the CE. In a real-world scenario, we should replicate this voter inside each CE that is running App 1, possibly with some kind of byzantine fault tolerance. However, the exact implementation or position of this voter will not change the outcome of this example in general. One simple distribution is in Figure 26 (a). The applications App 1 and App 2 are running on the first three CE and the other applications are running on CE 4 to 6. In our example, App 2 is a malicious app, which tries a denial of service against the cache. In this configuration, App 2 slows down the computation speed of CE 1-3 and harms any instance of App 1. The 2oo3 voter receives all results of App 1 delayed. In this case, the result of the voter will be delayed, too.

If we add the requirement, that two applications are only allowed to run on one CE together, we will get a distribution as in Figure 26 (b). If now the App 2 slows down the computation speed, it could only harm the App 1's instance on CE 1. The 2oo3 voter will still receive two results just in time and can promote the result.

Remark on freedom of interference: In the aforementioned approach of establishing the safety/security case, any change to any part of the system requires an update for the whole causal chain the

[8] For simplification in this figure is only one voter is shown. If any CN has a voter, we have exactly the same problem. In (a) three voters scenario, each voter would receive three delayed results. In figure (b) these three voter would only receive one delayed result each.

safety case is based upon and thus an update of the safety/security case itself. If freedom of interference cannot be guaranteed at the generic level, the deployment of an additional application and even changes to existing ones constitute a change to one of the components, requiring an update to the safety cases of all application hosted on the SIL4 Cloud.

5.6 Security homologation

Most of the applications which will be hosted on the SIL4 Cloud need homologation by the federal authority, the Eisenbahnbundesamt (EBA). Homologation regarding safety is quite well-defined through CENELEC standards and a proven and practiced process with the EBA – also including the influence of IT security on safety. But with the introduction of modern digital solutions, it is becoming more and more clear, that IT security itself is also becoming the subject of homologation by the EBA and that the standards and processes are not yet set and proven. Most experts expect that the standard IEC 62443 will be the most relevant or even required standard for homologation of railway systems from the security point of view.

While applying IEC 62443-3-2 for a single application is quite a straight-forward (though elaborate) process as described in Table 5, it increases complexity when many applications are hosted on the SIL4 Cloud as mentioned in previous chapters. Furthermore, the whole environment (SIL4 Cloud and applications) will undergo frequent changes, e.g., by

- adding, modifying or removing an application
- adding, modifying or removing any component of the SIL4 Cloud (hardware, software, RTE, common services, network, etc.)
- functional or security updates.

As each of these changes might influence the security (as well as safety) of any (other) application hosted on the SIL4 Cloud, this could lead to a claim by the authorities to re-do the certification process each time. It is obvious that this would lead to excessive expenses and a total loss of flexibility, thus diminishing all the advantages of SCP.

Therefore, we recommend developing a certification strategy based on the following points:

The SIL4 Cloud (basically the SuC as defined above) should be considered a building block, which can be used by any application, and which has defined security properties (based on the essential requirements discussed above). Then, generic applications, field elements, user workstations and other relevant elements of an application should be added to form an overall, generic SuC. For this SuC, a security concept should be developed following IEC 62443-3-2.

This overall security concept must include mechanisms to separate every application from each other, so that the residual risk resulting from any interference between applications is negligible. Furthermore, the security concept for the SIL4 Cloud should consider all operational processes like integrating new or changed applications, functional and security updates of the SIL4 Cloud, and so on.

For any single application, the SuC consists of the application itself, the relevant external elements (field elements, workstations, etc.) and the SCP as a building block. The security context for this application should now only consider the specific aspects of this application, the correct integration into the SIL4 Cloud (as defined in the overall security concept), but not the specific aspects of the SIL4 Cloud itself.

The basic idea behind this is to separate the homologation of the SCP from the homologation of any application and to allow for an independent homologation of any application. This would reduce the homologation efforts and costs drastically.

This approach should be discussed with proper official stakeholders e.g., Eisenbahnbundesamt, as their acknowledgement on the key concepts would be required for the success of the SCP strategy.

To further reduce homologation efforts it should also be discussed, which category of changes lead to:

- the need to completely re-do the certification process
- the need for just an update of the documents provided for the certification
- no need for any re-certification

Especially for the second case, it might be very helpful, if these documents could be updated in an automatic way – just by providing the information regarding the relevant changes.

Common Criteria (CC) specify a delta analysis process to re-evaluate the product by performing the analysis and evaluation of the impact of updates on the changed modules and its properties [16]. Dupont et al. proposed a semi-automated incremental certification process integrated with DevSecOps for collecting evidence and documenting certification activities [17]. From SYSGO's experience, it is possible to perform delta analysis through automated tools for minor updates of its system software for CC certification.

6 Architecture based on TAS Platform (Thales)

The TAS Platform is based on state-of-the-art software technologies. It uses an open, scalable software architecture based on well-established industrial computing standards and supports real-time multi-tasking applications in a computing environment. Depending on the applications' needs, the platform runs on off-the-shelf components supplemented by elements designed specifically for railway control systems, off-the-shelf hardware boards suitable for use in the railway environment or off-the-shelf server hardware. The core of TAS Platform contains a set of software components including the operating system, the communication system and the fault tolerance system. The TAS Platform's communication system provides various standard services with extended semantics for safety applications, as well as safety relevant protocols consistent with CENELEC standards. The fault tolerance system offers several different redundancy configurations as well as fault management services. The platform meets stringent dependability requirements and provides application transparent redundancy handling and fault management service enablers for safety-critical, real-time applications up to CENELEC SIL4.

The TAS Platform has been assessed according to safety and security standards by independent assessors and federal assessment institutes (EBA, BAV, etc.). Its modular architecture, standard application programming interface and well-defined adaptation layers ensure that the platform will keep pace with technological advances in hardware components and system software components in a controlled manner. It has been used in the field successfully for over 20 years with safety responsibility.

6.1 Details on TAS Platform

Electronic railway control systems share common requirements concerning safety, dependability, and real-time responsiveness. TAS Platform is a generic CENELEC EN 50129 SIL4-certified HW and SW product, providing common fault tolerance architecture, communication interfaces, operating-system services, HW boards, and equipment practice. It serves as a common and generic product kernel for vital railway signalling and on-board applications. This generic kernel can be used for all vital railway applications.

The key benefits are:

- Reduced life cycle costs by using a railway-wide technology Platform covering HW and equipment practice, OS services, communication services and fault tolerance services;
- Ability to use the same customer-specific application when updating to a new HW generation;
- Reduced complexity through the usage of the same type of equipment for all different railway control applications and market segments;
- Usage of COTS building blocks for HW, OS and compilers;
- Railway-wide common safety architecture;
- Usage of the CENELEC-SIL4-certified TAS Platform as a building block to build-up a CENELEC-SIL4-based overall system;
- Usage of the Security Level 3 (SL3)-certified TAS Platform as a building block to build-up an overall secure system;
- State-of-the art architecture for railway architecture for railway applications independent of the specific computer technology; and
- Ready for the SIL4 Cluster / Cloud system

The TAS Platform provides building blocks used by the applications to create CENELEC SIL4 compliant applications. The component overview of TAS Platform is shown in Figure 27 and described in the following.

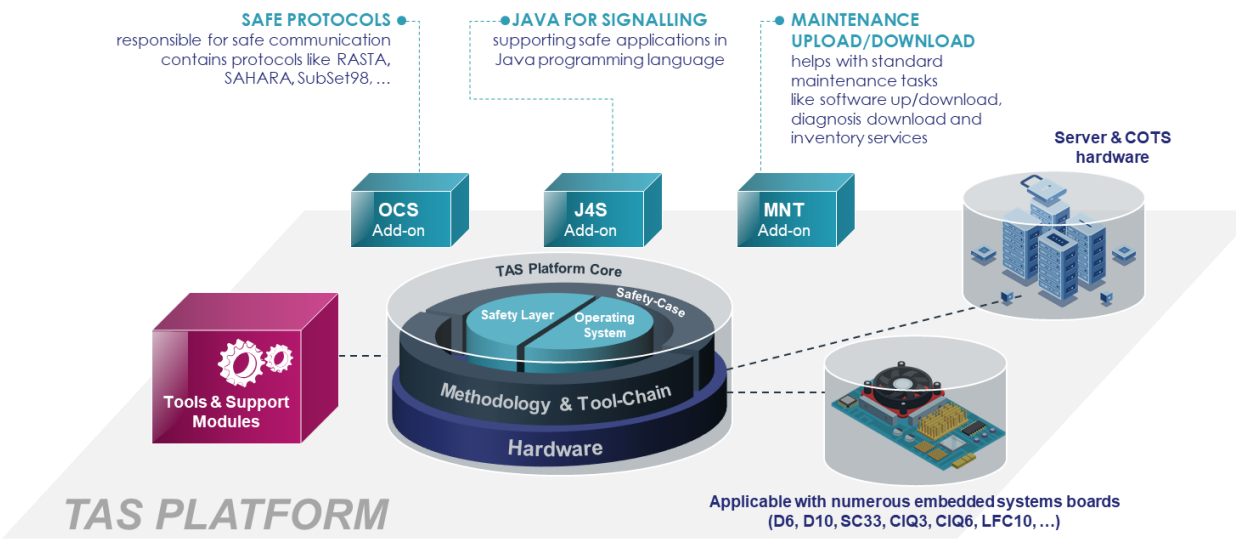

Figure 27: Overview of the TAS Platform architecture

6.1.1 TAS Platform Core Software

The TAS Platform Core Software consists of the following components:

- **OS**: Open-source OS (e.g., Linux). Due to the high stability of Linux and accessible experts, it is easier to change and improve the OS system. The creation of such an OS is necessary to fulfil the needs of the safety critical applications and obtain major benefits such as:
  - introducing mechanisms for safety, reliability, availability requirements,
  - tailoring Linux to the needs for embedded safety-critical applications, one tailoring sufficient for all applications that run on TAS Platform,
  - targeting the real time constraints of the applications,
  - providing drivers for (Thales) developed devices (e.g., CAN driver),
  - providing additional packages used for customers,
  - dealing with the board specific devices for safety reasons,
  - managing operative/maintenance mode handling as well as safety reactions.
- **Safety Layer:** provides specific functionality which is not available in the market. Here (N times M Redundancy layer) will provide a redundancy layer for applications requesting a 2x2oo2, 3x2oo2 or 4x2oo2 architecture in addition to a 2oo2 or 2oo3 architecture. A corresponding safety case is developed for all those redundancy architectures.
- **Compiler Tool Chain**: The widely used gnu-based compiler GCC[9] is used. In case of required maintenance activity for the compiler, issues are resolved via the open source gcc community. The TAS Platform provides validated and assessed compilers (C, C++, Ada). Online and offline debug and tracing support is given.
- **Virtualization**: TAS Platform supports virtualization for several purposes: security, reduced hardware dependency and the extended support of mixed criticality. It is available for embedded systems, as well as for industrial servers. In all contexts the isolation properties are used and a generic safety concept is provided.
- **Safe Protocols**: One channel safe (OCS) provides safe communication between two Platform Computing Nodes and between a Platform Computing Node and external non-Platform based

[9] gcc.gnu.org

components, according to CENELEC EN 50159 (class 1 and class 2, for class 3 additional measures have to be considered). Various communication protocols and physical communication media are supported and in use. Protocols such as SAHARA, RaSTA and many more are supported in this communication framework.

- **MNT**: MNT is the Maintenance and Update framework of TAS Platform. The major focus of MNT is to enable secure up/download to the target system (embedded systems configuration and industrial server solution). Support of SNMP access techniques using special MIB for diagnosis in addition to secure download and upload service are provided. Download/update of the OS is also important and a major step for future maintenance activities, which is solved with MNT. Future applications will urgently need such technical support amongst others due to security updates. MNT can be used locally or in a remote environment.
- **J4S**: The J4S (Java for signalling) framework provides support for Java-based applications (developed in Java language) for SIL4 field usage.
- **Tools & Support Modules**: TAS Platform provides a separate tooling package for application development, which can help analysis during the development of applications in addition to development tooling like compiler. This tool chain is called POST (for **P**latform **O**ffline **S**upport and **T**ools). This provides triple debugger and associated terminal as well as tracing support. The capabilities to provide field tracing data (with limited tooling support) are available. POST is also a mechanism for the TAS Platform support to provide additional elements (like tools) to customers ensuring tracing and optimal support of customers.

6.1.2 Available and Supported Hardware

The TAS Platform supports two main operational areas:

- Embedded COTS solutions like on-board systems, axel counters up to operating centres. For this class of solutions, the TAS Platform provides a set of validated HW components (CPU-board, Power-supplies, rack) conforming to the relevant CENELEC HW standards. The TAS Platform supports ARM32, x86/x86_64, ppc32, and ppc64 CPU architectures.
- COTS Industrial server solutions for services which can be concentrated in managed data centres (i.e., railway applications executed in non-railway environment).
- For both areas of operational execution layers, the TAS Platform deploys SW components and a system safety case to be the generic HW and SW platform to the customers.

6.1.3 TAS Platform Layered Architecture

The TAS Platform has a layered architecture. The top layer presented in Figure 28 represents the functional aspects of the system, the applications. The middleware section provides an implementation of the API and a bridge between the programming models included in the safety layer, and the operating system on top of the hardware layer.

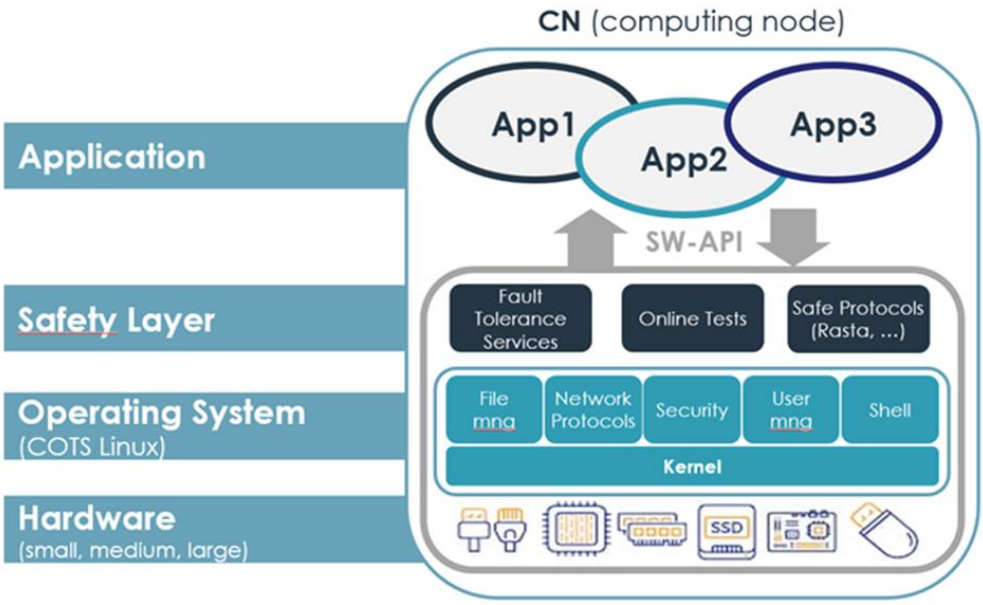

Figure 28: TAS Platform Layered Architecture – current state-of-the-art

Application Layer: The application layer is split into two, very distinguishable parts: the safe applications and the non-safe applications part. This strict separation between non-SIL and safety related applications allows executing applications with different SIL levels on the same HW (mixed criticality).

Safe applications: The most important added value of the TAS Platform is the safe application RTE. This safe environment allows executing applications up to SIL4 according to EN 50129. The major property is that safe applications have to reside on the Safe Platform API. All safety-relevant interfaces to lower-level SW components are developed as SIL4 Platform services. The TAS Platform OS provides SIL4 APIs to several library functions like configuration database (CDB) or diagnostic services (SNMP). For execution, the safe application uses the API and communication provided and supervised by the Safety Layer.

Non-safe applications: The non-safe part of the TAS Platform system applications is executed as ordinary Linux programs. Here the way is opened to run Linux services like diagnosis, non-safe protocols or security applications as provided by the Linux community. Additionally, services like MNT (maintenance/download services) are implemented according to their needs for safety integrity. Also, non-safe applications are executed in this way. Applications running in this partition execute on one CE (e.g., HW board, Server) without support of the safe voting and redundancy mechanisms of the **Safety Layer** of the TAS Platform. They interface directly to the POSIX/Linux API and can utilize all available services of the underlying OS.

Safety Layer: The Safety Layer consists of services such as deterministic scheduling, voting and fault management, redundancy management, and hardware supervision. To fulfil the safety requirements requested by EN 50129 concerning HW faults in CPU or memory, the Safety Layer provides integrity checks for data between the single replicas at start-up and operational phase. The Safety Layer implements the functionality of the POSIX/Linux API in a SIL4 developed and supervised way. Due to the synchronization and voting mechanism, the behaviour of the Safety Layer itself is checked frequently to detect failures of the underlying SW and HW components. Hence, each potential failure mode of e.g., pre-existing Linux functionality, is detected, and the safe state can be guaranteed. For availability reasons, the safe applications are pushed to high real-time priorities of the Linux OS. This enforces decoupling of safe and non-safe applications which are executing on lower priority levels. Therefore, the safe applications claim the near real-time behaviour whereas non-safe applications utilize the unused free computation power with a best effort approach.

Operating System Layer: The TAS Platform delivers an own OS distribution based on open-source Linux. TAS Platform tailors the Linux kernel and services for the releases of TAS Platform individually, regarding functionality, size, CPU family, board specifics, etc. The key focus in the OS is security.

Hardware Layer: The TAS Platform Core provides a broad variety of HW boards fulfilling the environmental requirements of different systems: starting with the most stringent requirements of trackside environments (i.e., field elements), going to on-board systems leading equipment used for CENELEC defined indoor systems (i.e., interlocking systems, Radio Block Centre, etc.). As a next step, the safe application computation can execute in environmental conditions that do not need to follow the stringent conditions of a CENELEC defined environment: safe computation on COTS industrial server boards, which are running in data centres without the physical requirements requested by CENELEC, assuming conditions of typical railway environments like relay rooms.

6.2 Safety and Security Approach for Communication

The TAS Platform supports applications up to SIL4 according to CENELEC by applying the composite fail-safe principle. Safety is ensured by end-to-end consideration while security can be separated to use state-of-the-art solutions within TAS Platform OS layer (see Figure 29). Safe data transmission bases on the norm EN 50159 supporting class 1 and class 2 networks. To access class 3 networks (open networks), additional measures like firewalls and cryptography (IPSec) have to be applied by the application. The required capabilities are provided by the TAS Platform. Examples are provided to the users of TAS Platform.

The TAS Platform Core considers additionally the security requirements according to the IEC 62443 standard.

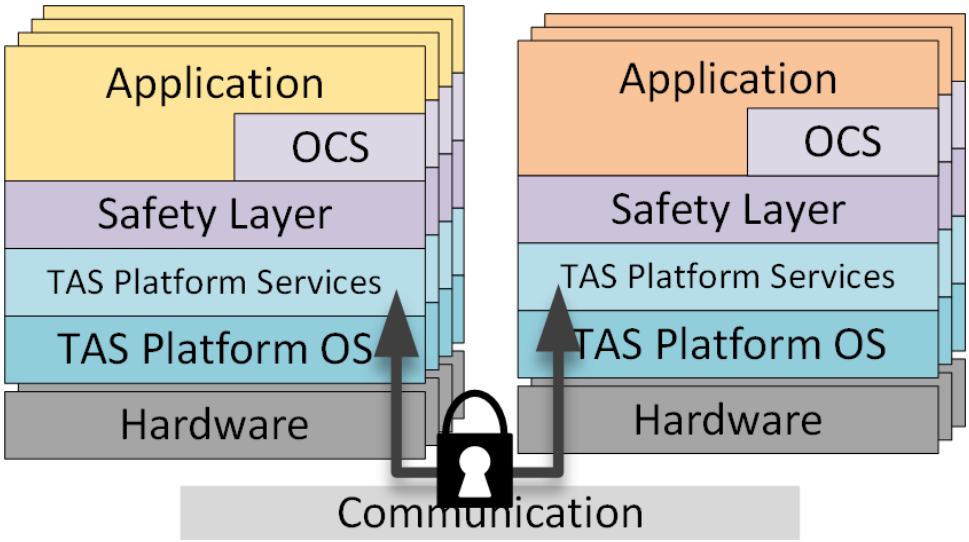

Figure 29: Safety and Security concept for communication.

6.3 Safe Redundant Architectures

With TAS Platform, several redundancy architectures are possible. Figure 30 gives a graphical overview of the used and supported redundancy configurations of TAS Platform: a set of Computing Elements (CEs) forming the safe total system.

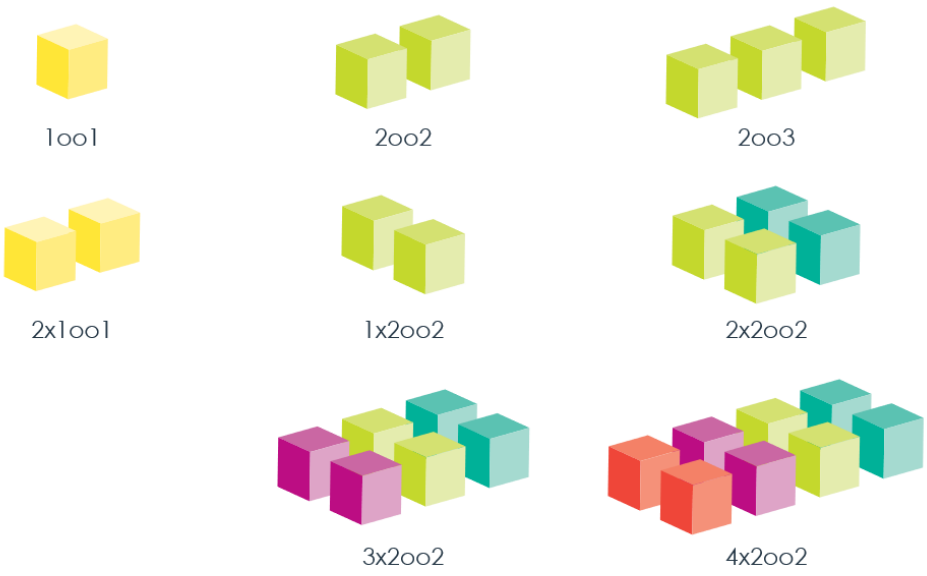

Figure 30: TAS Platform supported redundancy architecture

This configuration forms a Computing Node (CN).

- 1oo1: one CE, fault detection via diverse software channels

  Use case example: field element using diverse software channels for safety.

- 2oo2: two CEs with fault detection via hardware channels

  Use case example: field element using redundancy for safety.

- 2oo3: three CEs with fault tolerance via hardware channels

  Use case example: radio block centre.

In case of high availability requirements, a 2oo3 or an NxMooM (NMR[10]) architecture can be used, with a set of Computing Groups (CG) forming the safe system. NMR based architectures define a collection of 2oo2 systems as a CG.

An NMR system can be specified as:

- 2x1oo1: two redundant single HW architectures (safe with diverse SW channels). Such systems provide high availability compared to single HW channel applications – either for test simulation purpose, non-SIL applications or in combination with 2 diverse SW channel even as high available and safe system.

  Use case example: a non-SIL logger replicated for availability.

- 1x2oo2: two CEs with fault detection via HW channels based on NMR technology. Such systems are dedicated to applications needing synchronization and voting. The NMR configuration allows a lower latency for data exchange with safety guarantees.

  Use case example: a safety-critical field element replicated for safety, no further redundancy for availability.

- 2x2oo2: two redundant safe 2oo2 systems. This kind of redundancy provides two safe CNs with one redundancy.

  Use case example: interlocking system replicated for safety and availability.

[10] N-times M-out-of-M Redundancy

- 3x2oo2: three redundant safe 2oo2 systems. The third CN of this class enhances the availability further. Such a system also permits to take one CN offline and upgrade it concerning hardware and software changes while providing the availability of a 2x2oo2 system.

  Use case example: geo-redundant interlocking system replicated for safety and high availability. Such a system also permits to take one CN offline and upgrade it concerning hardware and software changes while providing the availability of a 2x2oo2 system.

- 4x2oo2: four redundant safe 2oo2 systems. This architecture is provided for applications with a very high degree of availability. This redundancy degree will allow realizing geo-redundancy between long distances.

  Use case example: geo-redundant interlocking system replicated for safety and very high availability.

6.4 SIL4 Cloud architecture based on TAS Platform architecture

Applying the TAS Platform approach to the Cloud architecture enables further approaches for availability and managing obsolescence by fostering the flexibility such an architecture brings concerning hardware management. [18]

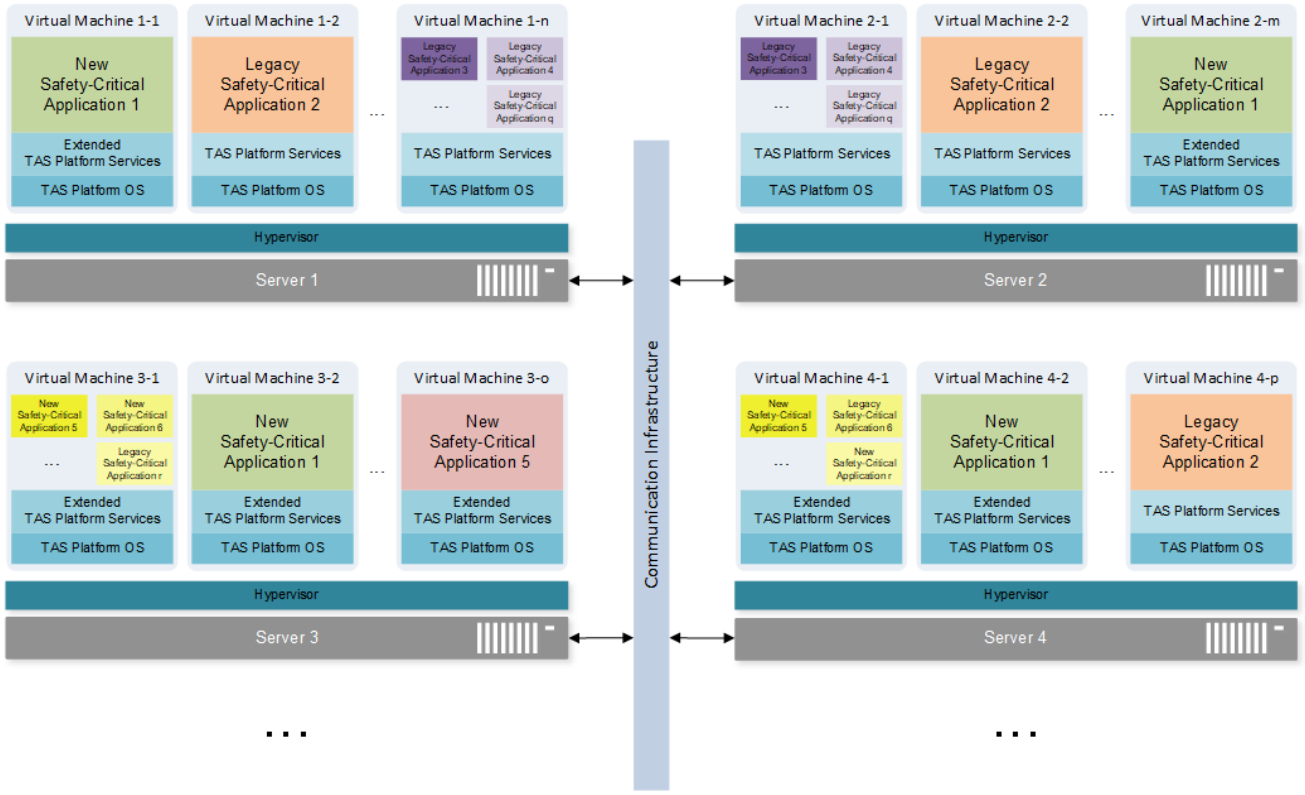

Figure 31: Example SIL4 Cloud setup based on TAS Platform

Figure 31 depicts an example setup showing how industrial servers and virtualization can be used to provide an execution environment for safe applications based on TAS Platform. It is feasible to host a non-safe application on the same hardware, along with a safety-critical application (up to SIL4), in separate VMs. It is also possible to host a non-safe application in the same virtual machine as the safety-critical application. In this case, however, it has a different validation strategy.

The virtual environment provided by the server hardware and the hypervisor is configured such that virtual machines are suitably isolated from each other, and are provided the required resources. A

fault can still trigger a misbehaviour in any case, hence the hypervisor must not have safety responsibility. This is quite different to many mixed-criticality solutions that have replication and voting in hardware (e.g., lock-step CPUs), where such faults cannot be detected.

The main benefit of using TAS Platform architecture is the flexible integration of a variety of applications on the same platform. In particular, legacy safety-critical applications currently running on embedded systems can be directly migrated to the SIL4 Cloud having servers and virtual machines infrastructure, while new safety-critical applications can simultaneously facilitate extended TAS Platform services enabled by the Cloud environment.

Mixed-criticality can be supported on two levels. The mixed-criticality approach of TAS Platform for applications and services on embedded boards can be directly reused in the virtual machines, while the virtualization itself enables mixed-criticality on hypervisor level by running software of different or no SIL level in virtual machines on the same server. The multi-application support already provided by TAS Platform can likewise be immediately applied in the cloud architecture.

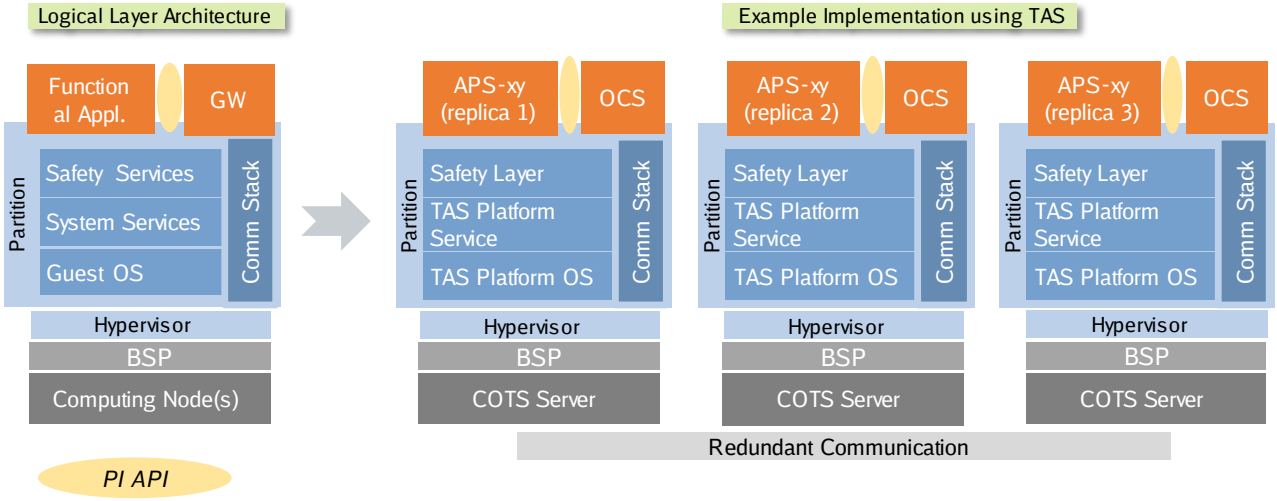

Figure 32: Example architecture of the SIL4 Cloud with replicas of safety-critical applications running on different servers (configuration 2oo3)

The mapping of the above architecture to OCORA is shown in Figure 32. Starting with the lowest layer:

- **Computing Node(s)**: on this level, multiple COTS servers are used to form CNs.
- **Board Support Package (BSP) and hypervisor**: Using a state-of-the-art OS with hypervisor virtualization functions provides virtual machines and communication to the layers above.
- **Guest OS**: The TAS Platform OS as guest OS running in the virtual machine, which also provides shared security services such as IAM, IDS or SIEM as per X2Rail specifications.
- **System Services**: The TAS Platform services provided on top of the TAS Platform OS.
- **Safety Services**: The TAS Platform safety layer providing the POSIX API to the application.
- **PI API**: The TAS Platform POSIX API.
- **Functional Application**: The applications running on top of TAS Platform shown as APS-xy.

Note that for the cluster/private Cloud scenario, there can be a vast pool of physical server hardware (e.g., multiple data centres consisting of dozens of servers each) that needs to be administered and coordinated. Hence, cluster management software (CMS) is necessary to efficiently administer the physical servers and control the VM deployment.

7 Architecture based on PikeOS (SYSGO)

This chapter discusses the certifiable real-time hypervisor PikeOS for safety-critical railway applications and its applicability to the Cloud environments and RCA/OCORA architecture. PikeOS combines RTOS, virtualisation platforms and the Eclipse based IDE for embedded systems. The PikeOS RTOS has been developed for safety and security-critical applications with certification needs. One of the key features of PikeOS is the capability to safely execute applications with different safety levels concurrently on the same platform. Figure 33 provides an overview of the SCP based on PikeOS hypervisor. The dark blue borders signify the strict boundaries of partitions that guarantee certain safety and security properties. The PikeOS distribution comprises multiple components that can be combined.

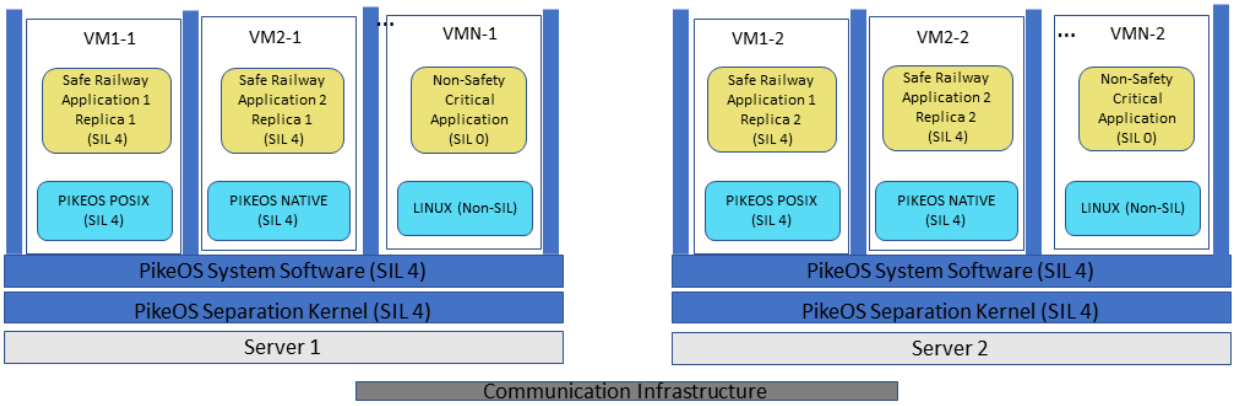

Figure 33: SIL4 Cloud based on PikeOS

7.1 Safe and Secure Virtualisation for SCP

The virtualization layer in SCP provides multiple virtual machines that execute Guest OS or bare-metal applications. Virtualization in the SCP provides the following advantages:

- Virtualization to reduce cost and development time: Building new Functional Applications software is extremely expensive and time consuming, so extending the useful lifetime of existing software is a critical component of the goal to save money and resources. An ideal life extension technique is the re-usage of entire existing subsystems. With Safe and Secure Virtualization (SSV), software systems can be retained from the original platform, and inserted into the new platform. SSV is now finding favour in the planning and development of next generation systems where safety and security are as important as economy and timeliness.
- Usage of legacy source code: Virtualization allows the re-usage of legacy code, which saves costs and effort in a complete re-design.

Tailoring the boot sequencing to achieve fast-boot requirements: In complex mixed-criticality systems, the hypervisor allows ordering the booting of different VMs such that VMs requiring fast boot times are booted before other VMs.

- Open-Source License Management: Software with restrictions due to the licensing model (e.g., open source, GPL) can be strictly separated from your own intellectual property. This clear model avoids ambiguity in case of legal issues and avoids the publication of your own source code.

At first glance, virtualisation and real-time seem to demand requirements that are contradictory to each other. Especially, when it comes to Functional Applications with mixed-criticality. The virtualisation approach proposed in SCP using a separation kernel-based hypervisor such as PikeOS makes it possible to bring the usual benefits of virtualisation (optimal use of hardware, legacy code reuse, concurrent execution of different types of applications, hardware obsolescence management) to mixed-critical systems with real-time constraints. This technology can even guarantee not only a strict compliance to real-time constraints but also the full integrity of data and safe and secure execution of applications at the highest level of corresponding standards.

This is possible through the implementation of specific multi-partitioning, that has been first developed according to the need of complex Avionics systems and their corresponding software architectures. The Integrated Modular Avionics (IMA) concept has been a great step forward in this direction and came with partitioning standards such as the ARINC 653 documentation. This standard defines virtualisation in terms of static resources (e.g., memory, I/O) as well as processor time. It also defines an API suitable for Avionics concepts. The Airbus A380 and the Boeing A787 are examples of the IMA concept success. Certifiable, mixed-critical systems for the railway domain can also be built using such partitioned architectures. The following section describes the partitioning architecture based on PikeOS, which is a certifiable separation kernel-based hypervisor.

7.2 Separation Kernel

The separation kernel (SK) is a special type of *operating system* whose primary functionality is to enforce the separation between partitions and information flow between partitions based on the security policy. The separation kernel guarantees separation and controlled information flow by enforcing the following policies:

- **Resource allocation policy**: This policy defines how the system resources such as CPU time and main memory are allocated to the partitions. If some resources are shared between partitions, this policy defines how the resource sharing shall be done such that the separation between partitions is enforced. For example, the separation kernel enforces spatial separation by allocating disjoint memory areas to the partitions and by controlling the memory accesses from the partitions. Similarly, temporal separation between partitions is enforced by executing partitions in separate, non-overlapping time windows.
- **Access control policy:** Access control policy specifies the access rights of objects under the control of the separation kernel. The implicit information flow between partitions is defined by this policy. For example, an access control policy might assign communication object C as writeable to partition A and readable to partition B, thereby creating an implicit information flow from A to B.
- **Information flow policy**: The information flow policy defines how the partitions shall exchange information. This includes rules based on the sender/receiver of a message and the message content. The separation kernel enforces information flow policy explicitly using the communication rights of partitions and implicitly by the access rights of objects. The separation kernel configuration also allows removing/limiting covert and side channels that might exist between partitions due the sharing of hardware resources such as caches, memory bus and memory controller.

7.3 Resource Partitioning

Resource sharing is a challenge for safety critical systems. If computing resources are shared, delays or even deadlocks can happen, if a resource is blocked by another application. Resource partitioning primarily targets the protection of resources, which shall exclusively be used by a dedicated application running in a partition. Once a resource is assigned to a partition, other partitions will not be aware of this resource.

PikeOS resource partitioning is achieved by static assignment of a computing resource such as memory, I/O and file devices, secure communication channels and cores to partition / virtual machines. PikeOS ensures that during runtime, an application has guaranteed access to the assigned resources and that the partitioned resources are not accessible from applications belonging to other partitions.

Resource partitioning is enforced by using the MMU to control access to the available resources. Each hardware device is represented by a physical address in order to access this device. Resource partitioning is realized by using the MMU to map a certain memory area into a partition’s virtual memory space, and to allow or deny read/write access to this memory space.

The configuration of the MMU is done statically at system start-up by the separation kernel and is not modifiable at run-time by the user applications, ensuring that an attacker or faulty user application cannot modify the resource configuration later on. In summary, the separation makes sure that errors occurring in one partition cannot propagate to another partition.

7.4 Time Partitioning for mixed-critical applications

PikeOS partitions may host real-time as well as non-real-time applications on the same hardware. These two types of applications have fundamentally different requirements with respect to scheduling:

- Real-time applications need to be able to make guarantees about the temporal behaviour of their processes. A necessary precondition for this is that the operating system itself knows the points in time when it will be able to use the processor and for how long it will be able to use it. In other words: the schedule, which defines the switching between partitions hosting real-time applications, has to be strictly a function of time.
- Non-real-time applications, on the other hand, work by a “best effort” principle, i.e., they try to do as much as possible, attempting to use all the computational resources they can get. Hence, their goal is to minimize processor idle time. For a VM scheduler hosting non-real-time application, this means that, whenever a VM is found to be idle, it should revoke the CPU from that idle VM and pass it to the next one, hoping that one will have useful work to do. The resulting VM schedule is obviously influenced to a large extent by the activities going on inside the VMs, so it is clearly not just a function of time.

The avionics standard ARINC 653 addresses the allocation of CPU time across different partitions. The approach described in this standard works on a fixed cycle time, within which the individual partitions are scheduled on the processor in a specified order for a guaranteed duration. This “time partitioning” method is suitable for real-time systems, but, as discussed above, it leads to poor system utilization.

Typical scheduling methods used by virtualization systems such as Xen and VMware, attempt to optimize processor utilization by revoking the CPU from VMs as soon as they become idle. But in doing so, they cannot guarantee deterministic timing to their guest operating systems.

The scheduler of PikeOS uses a combination of priority- and time-driven scheduling to join together these contradictory approaches. The time-driven scheduler is a mechanism to distribute the available CPU time amongst the PikeOS partitions. It can be defined as a first level scheduler, which quantifies the available time into time partitions (tp_i; i = 1 ... n). Depending on the application timing requirements, several time-partitions can be defined and grouped into a time frame (see Figure 34).

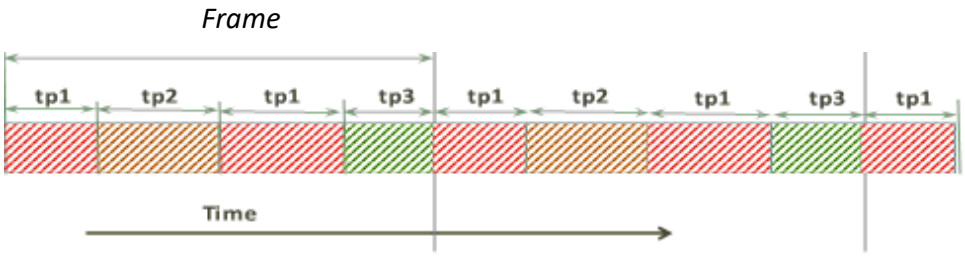

Figure 34: Time-partitioning frames

A time-partition may occur more than once in a timeframe and can have different durations. As soon as the time driven scheduler has completed a frame, the frame starts over again and again (see Figure 34). In contrast to the ARINC 653 standard, PikeOS scheduling uses a one-to-n assignment of resource partitions to time partitions. That is, one or more resource partitions can be assigned to one time-partition (see Figure 35).

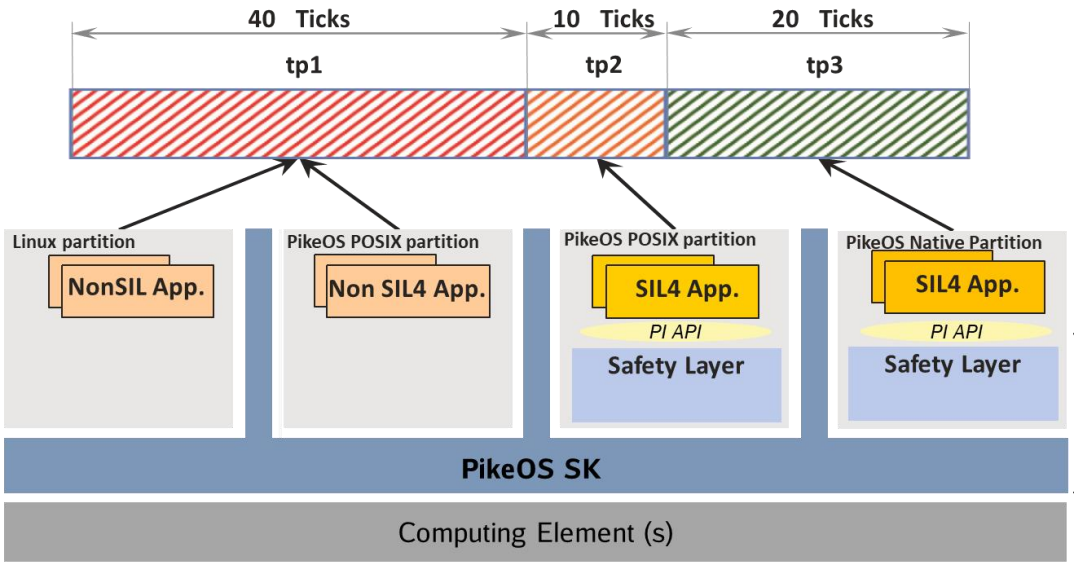

Figure 35: Scheduling of mixed-critical applications

In the PikeOS nomenclature, a user's Functional Applications are called processes. Processes can be started in the context of a task, where each task is characterized by its own address space and a set of schedulable entities (threads) bound to it. A PikeOS partition may host more than one task (other tasks and child tasks), so that within a partition address space additional task address spaces are defined.

By assigning one or multiple resource-partition(s) to a time-partition, threads are grouped into time-partitions, which are activated by the time-driven scheduler. However, in contrast to the ARINC 653 standard, one more time-partition exists, that is active at all times. This time-partition is referred to as the background partition, tp_0, whereas the currently active one of the time-switched partitions is called the foreground partition, tp_i (i = 1 ... n).

In addition to their time-partition, threads do have a priority attribute. Whenever both the foreground and background partitions have active threads, the thread to be executed is selected according to its priority. In Figure 35, a SIL4 Functional Application has a dedicated time-partition tp_2 and tp_3. In tp_2, a Functional Application acts as a voter. In case of a failure, the response-time of the fault-handler can be quite high, if the error happens right after the fault handler time-partition has ended. By assigning the fault handler to tp_0, the handler is starter immediately, as it has the highest priority among all active threads.

Clearly, such high priority threads must be considered as trusted code from the point of view of the threads that they can preempt. Therefore, it is important that the priority assignment functions implemented in the microkernel guarantee that no thread can incorrectly obtain rights to which it is not entitled. PikeOS achieves this by assigning a maximum priority to each thread. This value is set at thread creation time and is based on the maximum priority of the creating thread. It cannot be changed at any point during the thread's life cycle.

7.5 Interference and Threat mitigation

Mixed criticality applications integrated in multicore platforms can be mapped to the cores in different ways which can be leveraged by an attacker to perform covert channel and caches attacks. PikeOS uses several techniques to limit the interference and minimize the attacks [19]:

1. **Resource partitioning**: On platforms which provide multiple memory controllers, a dedicated memory controller is assigned to one partition to serve the memory access. PikeOS allows assigning a partition to a particular core.

2. **Time-partitioning:** The other approach is to execute one partition at a time. PikeOS time separation allows such separation.
3. **Cache flushing and sandwich partitioning:** A buffer partition called “sandwich partition” between the two partitions will be effective against covert channel and cache attacks (refer Chapter 5.5 for more information on cache attacks). By inserting such a sandwich time-partition, any timing variation on cache flushing operation based on the cache state will not be observable from the next partition. This can be implemented on COTS hardware and would require some tuning as it affects the performance depending on the frequency of switching.

7.6 System Software

PikeOS PSSW component initializes partitioning, inter-partition communication and health monitoring according to functions. During run-time, the system software component acts as a server providing the services such as partition and process management, file system and health monitoring to the applications executing inside the different resource partitions. The orchestrator component, discussed in the generic architecture, could be implemented at this layer as system extension.

7.7 Personalities

PikeOS has been developed in compliance with the standard EN 50128 and certified up to SIL4 level[11]. PikeOS ensures the safe execution of RTE, and applications developed on top of it, which makes it suitable for hosting safety-critical railway applications. PikeOS provides different application programming interfaces (API), RTE and guest OS to run inside the partitions, which are called personalities. These personalities define the interaction mechanism between the application and the lower software layers, i.e., the API. Since personalities are executed inside a partition, it is possible to run personalities which are not certified in parallel with safety-critical applications in other partitions. PikeOS provides PikeOS Native, industrial grade Linux, POSIX and ARINC 653 personalities, among others.

- **PikeOS Native**: PikeOS RTOS API, referred to as PikeOS native personality, is the basic and lightweight personality that directly exposes the PikeOS service interface to the application. Due to its small code base, it provides the best performance in terms of execution and memory usage.
- **PikeOS POSIX**: POSIX personality reproduces the Portable Operating System Interface (POSIX) standard, which is more or less strictly followed by most UNIX(-oid) operating systems. Due to its small size and low memory footprint, it is a safe and secure alternative to the Linux operating system in situations where massive feature support from the OS is needed. The PikeOS POSIX guest OS implements the Real-time Controller System Profile of the IEEE standard 1003.13-2003 (PSE52). Some additional real-time extensions and extensions for the PikeOS are also included.
- **PikeOS Linux**: PikeOS also supports full-featured OS as applications, which themselves host the intended “payload” software. ElinOS is a para-virtualized Linux distribution that runs on top of PikeOS hypervisor. It contains industrial grade drivers, connectivity stacks, real-time extensions, support for industrial hardware, a graphical development environment.

7.8 PikeOS Infrastructure:

PikeOS provides the following, additional elements for application development:

- **Integrated Development Environment (CODEO IDE):** CODEO is the IDE, which comes with PikeOS. It is based on Eclipse 4.6**.**

[11] PikeOS 5.1 is SIL4 certified

- **Cross-Development Kit (CDK)**: PikeOS CDK is based on the GNU toolchain. In addition, PikeOS specific configuration compiler, ROM image builder and validation tools are part of the CDK.
- **Debug Tools**: A GDB-based debugger, target monitor, and a trace tool are provided with PikeOS. In addition, support for third-party hardware debuggers is available.
- **System Service Providers**: File system services or network protocol services are provided as separate services. PikeOS provides various file systems: certifiable file system (CFS), YAFFS, or FAT file system. Network stacks are either provided through the personality (e.g., Linux or POSIX) or through the Certifiable IP Stack (CIP / ANIS).
- **Integrator Suite:** This component provides all the tools and materials needed to configure, build, run, debug, trace, and optimize a complete PikeOS system from scratch. Most importantly, the Integrator Suite is required to be able to integrate the partition images developed using specialized application suites.
- **Application Suite:** An application suite provides the tools and materials needed to develop software using the respective API, RTE or guest OS, within a PikeOS partition. The resulting partition images can be integrated into a pre-configured PikeOS system and executed, analysed and debugged.
- **Architecture Support Package:** The Architecture Support Package (ASP) provides the processor abstraction layer in the PikeOS kernel. The ASP implements processor context management, CPU exception model and address space and memory management. PikeOS supports x86, ARM and PowerPC, MIPS, RISC-V and Sparc CPU architectures.

7.9 Qualification of Separation Kernel (SK) based Hypervisor

In MILS architecture, the SK can be considered as a single point of failure. Thus, the SK shall provide assurance for the highest safety levels required for the system. This is described in the EN 50128 standard, which states that the sub-component that provides separation between components shall be certified at the same level as the highest assurance application. Thus, the SK used shall be certifiable to SIL4 for SCP that host SIL4 applications. Historically, the SK first had been certified for the avionics DO-178 standard, which forces the user to provide requirements for a V-model life cycle, starting from requirements at a high level, then going to low-level requirements and the implementation. On the other side of the V-model is the verification of low- and high-level requirements. For instance, this includes full test structural code coverage, robustness testing, and demonstrations that the compiled object code conforms to C code. Moreover, there is full requirement linkage between requirements, implementation and design artefacts. Additionally, artefacts have undergone independent review, are under version management, and tests are run on the target hardware configurations nightly (CI/CD). Later, from the avionics processes and artefacts, certification kits for IEC 61508, ISO 26262, EN 50128 and EN 50657 have been derived, leading to certifications as system element out of context (SEooC).

For security (Common Criteria certification), in view of the design, many safety artefacts could be reused. Threat agents, attacks, and security countermeasures have accordingly been formulated [2]. Testing was extended by fuzz testing. Analyses were extended by CVE analyses as well as vulnerability analyses (white box analysis by security evaluation laboratory). The use of these Common Criteria artefacts for system certification of IEC 62443 [20] has been demonstrated by SYSGO.

8 API between application and underlying platform

The goal of the API is to provide a standardised interface for safety-critical applications running on top of the SIL4 Cloud. The API abstracts the complex details of a safety-critical real-time RTE and allows application developers to just focus on the application (business) logic itself. This also eliminates the need to explicitly handle different redundancy architectures, fail-over scenarios, etc. within the functional applications, hence making functional application development simpler and reusable in various deployment scenarios of the SIL4 Cloud.

8.1 TAS Platform API

In this section, an overview of TAS platform API between Functional Applications and the underlying platform is presented. Important points about TAS API are as follows:

- Platform (TAS) contains all fault-tolerant mechanisms, and Functional Applications are supposed to implement only business logic
- Syntaxes and concepts are based on *POSIX*
- It is a *common API* for onboard and trackside systems (i.e., Functional Applications)

8.1.1 Functional Application needs

Safety-critical, real-time Functional Applications have stringent need for safe computation, safe communication, and guarantees for safe time and I/Os. Communication includes sending data to other communication partners (e.g., Functional Applications) and also data exchanges among parts of the Functional Application. Further, the access to time service is key for the Functional Application to integrate real-time behaviour. Additionally, the application may require accessing the state information of the safe computing environment itself for diagnosis purposes.

8.1.2 State-Of-The-Art: POSIX

Since its first version, the TAS Platform provides a POSIX-derived interface for safe applications. One of the major design goals was to build the API on the well-established POSIX interface and provide the safety and availability functions transparently to safety-critical applications running on top. This POSIX approach makes it easier for developers, since it leverages already established tooling and the knowledge base of developers and engineers with vast POSIX experience.

8.1.2.1 Programming model and restrictions

The TAS Platform programming model divides each application into TaskSets that together provide the application's functionality. These TaskSets are categorized into Model 1 TaskSets used for safety-critical computation, and Model 3 TaskSets used for non-safety-critical functionality.

- Model 1 TaskSets are transparently replicated between the individual CEs of the CNs and CGs (see Chapter 6) and can use only a very limited subset of the POSIX functionality.
- Model 3 TaskSets are only instantiated on one CE, but are able to access most of the POSIX functions.

TAS Platform provides the POSIX message queue interface to communicate between TaskSets. For Model 1 TaskSets, the messages sent via the queues are voted. This pattern enables the integration of different methods for communication or I/O to other systems, or other functionality into the application. The basic tasks of input, compute and output can then be split in different TaskSets as depicted in Figure 36.

Figure 36: TAS Platform TaskSet structuring example

Application execution consists of the three phases: start-up, operation, and shutdown, during which different parts of the API are permitted to be used.

- During **start-up**, Model 1 TaskSets have to allocate all necessary resources and prepare the operational phase. By calling certain functions, the Model 1 TaskSets declare to have finished by calling one of a defined set of functions which halts execution until all replicated instances have been started on all necessary CEs and all checks have been performed successfully.
- In the **operation** phase, the Model 1 TaskSets perform their safety-critical functions and use only a very limited set of the API functions. In particular, no resources must be allocated. I/O and external communication must be performed either via specific communication libraries, e.g., OCS, or in Model 3 TaskSets.
- During **shutdown**, application execution is prevented by the RTE and only diagnosis tasks are allowed to run.

8.1.2.2 Example POSIX function for Safe Communication (from TAS Platform)

Sending of a message via a queue is performed in Model 1 and Model 3 TaskSets by calling `mq_send` on the API:

```
res = mq_send (queue_dest, (char *) & msg, msg_size, prio);
```

Here the message `msg` with the length of `msg_size` is passed to the queue `queue_dest` with the priority `prio`. The RTE then adds the CENELEC EN 50159 communication mechanisms to this message and distributes and votes it for Model 1 TaskSets transparently. The receiving TaskSet can then retrieve the message with `msq_receive`, if voting and all other checks were successful.

8.2 PI API using PikeOS

PikeOS POSIX and Native API are suitable for developing the safe and real-time railway applications in Cloud environments. The PikeOS POSIX API interface can be extended and standardized for additional system and safety services such as application life-cycle management, tracing logging, access to IO devices, synchronization, safe communication, and hardware and software monitoring to provide Safe Computing Platform PI API [21].

8.3 Options for PI API in SIL4 Cloud context

RCA/OCORA Safe Computing Platform workstream specified and published a concept for a potential Platform Independent API, which also envisions to formulate a common API specification for trackside and onboard platforms [21]. This section evaluates the options, starting with the current state-of-the-art:

8.3.1 POSIX

TAS Platform shows that its POSIX-derived API is suitable for developing distributed, safety-critical, fault-tolerant, real-time applications.

8.3.2 Extended POSIX

Support of more dynamic and therefore scalable safe communication models such as publish-subscribe could be added to the existing API. However, certain assessment restrictions would apply to comply with the CENELEC standards[12] such as a bounded number of communication participants.

8.3.3 Other programming models

Other programming models such as the actor model[13] could also be envisioned for a PI API. Still, if the PI API is not derived from an established standard or practice, it would not benefit from the experience gained with that standard, and the risk of designing an API that is impactable for Functional Application development is higher.

8.4 Security

The PI API must support development of Functional Applications that allow adhering to security coding rules and prevent introduction of programming errors according to the Common Weakness Enumeration [22].

Other security measures such as encryption are best solved outside the PI API on OS level, hence not affecting the safety life cycle.

8.5 Virtual Machine Management and COTS Server Management

Virtual machine management should be external to the PI API specification, so as to not overload this interface. There are commercial products available that can manage multiple data centres. The important functions for safe operation are starting and stopping of virtual machines. Still, some API functions may be necessary to access the current state of the virtual machine deployment.

[12] Covers at least EN 50128 (software) and EN 50159 (communication). Mostly also some aspects of EN 50129 (Hardware) will have to be considered, but this is always a matter of detailed analysis.

[13] RCA /OCORA uses Actor Model to define various Generic SCP concepts

9 Development Process

Being a modern platform, SIL4 Cloud also demands modern mechanisms to manage the platform and applications in terms of tooling and techniques. This chapter highlights some of the basic aspects, that need to be elaborated in more depth subsequently.

9.1 Continuous Integration/ Continuous Development (CI/CD)

As SIL4 demands full traceability, consistent testing and revision management of the product is required. An efficient way to achieve these objectives is CI/CD, where automation methods are used to continuously integrate the system in the development phase itself e.g., for embedded system software running on the edge device, there is a nightly build, and any failures reported during building the software (e.g., compiler or linker errors) are reported to the developer. Moreover, a test suite is run over-night and failing tests are reported when new commits are made. This is done for several relevant hardware architectures generic software runs on, as well as project-specific software runs on. In our context, tool-wise there is a combination of an in-house test framework, git, and Jenkins.

9.2 Compositional design for testing, verification, and validation

Typical systems are compositional. High SIL levels require providing complete structural and test coverage for individual components [Vatrinet2019]. In situations where flexibility is needed, compositional certification can allow certifying systems incrementally, thus easing deployment and redeployment, as well as cooperation between different vendors. This requires that re-usable architectures and interfaces be defined. In our context, virtualization and the MILS architecture are well-understood architectures [Hohenegger2021].

9.3 Software updates

For modern systems, software updates for applications are key functionality – not only because of feature enhancements, but also because of changed configuration data, feature enhancement of the OS or basic libraries, or for security reasons. Technical means of download are available, well-known and tested.

High SIL levels pose inherent challenges for frequent SW updates, as changed parts within a system usually necessitate re-assessment of the whole system. Even if only minor flaws be fixed or security holes closed, these steps cannot be skipped easily.

On account of those pains, current systems need to ideally follow “compositional design”, as outlined in Section 9.2. By separating a system into components of the right size, the problem of assessment in case of updates can be dealt with. Additionally, carefully choosing the parts which need to be of higher SIL level and avoiding of mixing higher and lower/non-SIL parts helps to make download easier.

Another essential aspect of download is the way a switchover to a newer version is triggered. For systems of higher SIL levels, it needs to be avoided, that commands sent accidentally or by a single error, trigger a switchover. This means that there need to be specific measures in place to avoid such situations, e.g., a carefully designed protocol to avoid this problem.

All downloaded artefacts and the whole process should be secured according to prevailing security standards, to avoid inducing malicious code/applications using the download mechanisms. Ideally, this is done in combination with a whole secure and trusted boot-chain, including process whitelisting and other security mechanisms.

10 Homologation

For the SIL4 Cloud as a new type of system, which amongst other things takes over tasks of train control, train protection and signalling, a certification of the Federal Railway Authority (EBA, Eisenbahnbundesamt) will be necessary for its commissioning and use. This chapter begins by clarifying the process of certification, as well as the prerequisites and requirements for granting of the approval. In the second part, the necessary safety case and the assessment is dealt with in more detail. In particular, the challenges that have to be solved within this project in order to be able to prove the safe functioning of the system, are addressed.

10.1 Guidelines relevant for homologation

Table 6 lists relevant guidelines for the homologation process, along with English translations of the official German terms. To improve its readability, only the abbreviations shown in the table are used in the text below.

Table 6: Abbreviations of relevant guidelines

Abbreviation	German Title	English Translation
EIGV	**E**isenbahn**i**nbetriebnahme-**g**enehmigungs**v**erordnung	Railway Commissioning Approval Ordinance
Directive (EU) 2016/797	Richtlinie (EU) 2016/797 des europäischen Parlaments und des Rates vom 11. Mai 2016 über die Interoperabilität des Eisenbahnsystems in der Europäischen Union	Directive (EU) 2016/797 of the European Parliament and of the Council of 11 May 2016 on the interoperability of the rail system in the European Union
VV GIuV	**V**erwaltungs**v**orschrift für die **G**enehmigung zum **I**nverkehrbringen **u**nd **V**erwenden von sicherungstechnischen und elektrotechnischen Systemen und Komponenten	Administrative regulation on the authorization for placing on the market and use of safety-related and electrotechnical systems and components
Sector guideline	Sektorleitlinie für die Zulassungsbewertung von Signal-, Telekommunikations- und Elektrotechnischen Anlagen	Sector guideline for the approval assessment of signalling, telecommunication and electrotechnical systems
VV IBG Infrastruktur	**V**erwaltungs**v**orschrift zur Anwendung der Verordnung über die Erteilung von **I**n**b**etriebnahme**g**enehmigungen für das Eisenbahnsystem (EIGV) in Bezug auf die Teilsysteme **Infrastruktur**, Energie, streckenseitige Zugsteuerung, Zugsicherung und Signalgebung sowie für die übrige Eisenbahninfrastruktur	Administrative regulation on the application of the ordinance on the Issuing of authorizations to place the railway system in operation (Railway Commissioning Approval Ordinance – EIGV) with regard to infrastructure, energy, track-side control-command and signalling subsystems and other railway infrastructure
CSM-RA	Durchführungsverordnung (EU) Nr. 402/2013 der Kommission vom 30. April 2013 über die gemeinsame Sicherheitsmethode für die Evaluierung und Bewertung von Risiken und zur Aufhebung der Verordnung (EG) Nr. 352/2009	Commission implementing regulation (EU) No 402/2013 of 30 April 2013 on the common safety method for risk evaluation and assessment and repealing regulation (EC) No 352/2009.

The approval of new railway components in Germany is governed by the EIGV. This regulates the conditions for placing on the market and putting into service, components of the railway system in

accordance with Directive (EU) 2016/797. These conditions apply to the design, construction, placing in service, upgrading, renewal, operation, and maintenance of the parts of the rail system. Regardless of some exceptions, the first placing on the market of a vehicle or the first commissioning of a structural subsystem is subject to an authorization for placing on the market (**APOM**) or an authorization for placing in service (**APIS**) according to §9 EIGV. An authorization for commissioning is also required for renewal and upgrading, in case these are not among the maintenance measures as described in Annex 5 of the EIGV.

With the entry into force of the new national EIGV, a significant change has occurred in the commissioning and authorization of signalling, telecommunications and electrotechnical systems. With §27 EIGV, the possibility of granting a generic prior authorization for safety and electrotechnical systems has arisen, if these systems are to be used several times in installations that require a commissioning authorization (**IBG**). The examination of the application documents and the granting of this authorization for placing on the market and use (**GIuV**) by the EBA is regulated in the VV GIuV. The sector guideline sets out the approval assessment processes necessary for applying for a GIuV. Once a GIuV has been granted for the signalling or electrotechnical system under consideration, compliance with the requirements covered by the granted GIuV is not checked again when the IBG is granted. Thus, this possibility of prior authorization avoids multiple inspections of identical circumstances, which makes the authorization procedure more effective. Within the commissioning procedure of the system according to EIGV, only the application conditions specified in the GIuV or other applicable documents and the special project planning for the specific application must be checked. Accordingly, the permissible adaptations for specific applications must be specified in the GIuV.

The VV IBG Infrastruktur contains the procedures for applying for an authorization for initial operation, notification of upgrades or renewals and for the actual issuing of the authorization to place the specific application in operation. It specifies and explains the requirements of the EIGV and contains instructions for implementation. It describes the general conditions of the application and authorization procedure.

The graphic in Figure 37 provides an overview of the relevant guidelines and their interaction, as described. In the following chapters, the individual aspects will be discussed in more detail. The aspects are dealt with in chronological order (from left to right in the diagram).

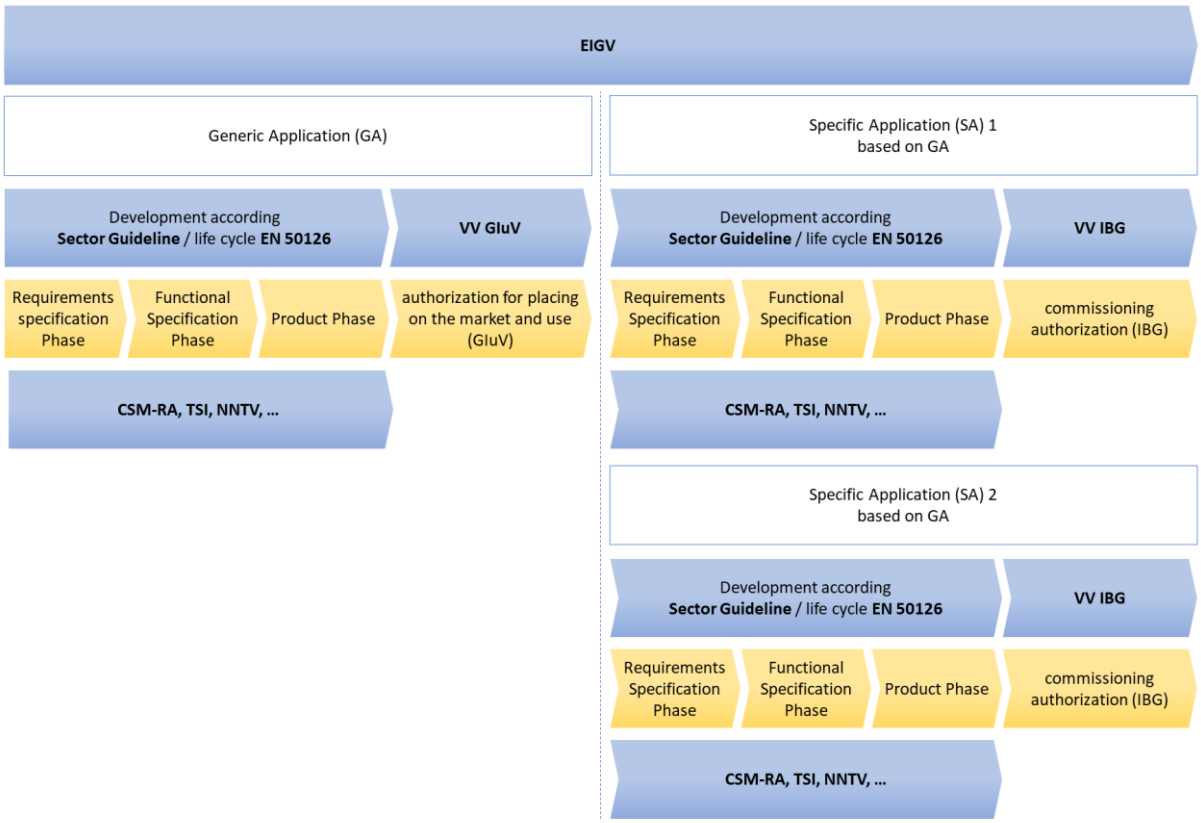

Figure 37: Overview of relevant guidelines and their interaction

10.2 Requirements for commissioning approval

In order to be granted a commissioning authorization (**IBG**), the component under consideration must meet the essential requirements, in particular for technical compatibility and safe integration. If these requirements are met, the Federal Railway Authority (**EBA**) grants the authorization. The EBA itself only carries out formal checks, and no content checks. These are carried out by notified bodies, designated bodies and independent assessment bodies recognized by the EBA. Thus, the EBA's decision is based entirely on the preceding inspection results of the inspection bodies and the inspection experts. The Notified Body (**NoBo**) is responsible for checking compliance with the Technical Specifications for Interoperability (**TSI**). The Designated Body (**DeBo**) checks compliance with all applicable notified technical rules, while an Independent Assessment Body (**AsBo**) carries out the assessment of a risk management procedure according to CSM-RA. Compliance with applicable non-notified technical rules is checked by certified inspectors (**PSV**) (German: **P**rüf**s**ach**v**erständiger) acknowledged by the EBA. As part of the commissioning authorization procedure, the EBA checks the application and the related documents for completeness and comprehensibility.

According to §16 EIGV, the following evidence must be provided to the EBA to prove compliance with the essential requirements, in particular technical compatibility, and safe integration:

1. *EC Declaration of conformity (GERMAN: EG-Prüferklärung) incl. technical dossier* i.e., the applicant makes a declaration that the concerned subsystem has been subjected to the respective test procedures and meets all relevant requirements (EU and national regulations). The declaration itself is part of the application form and does not have to be drawn up as a separate document. To confirm the conformity, the applicant commissions a conformity assessment body in advance, which carries out the test procedure accordingly. Depending on which regulations are being checked, the DeBo or the NoBo acts as the conformity assessment body. The NoBo checks and certifies that the concerned subsystem complies with the **TSI**s. The DeBo checks and certifies that the concerned subsystem complies with the relevant national notified technical regulations (**NNTR**). The TSIs contain requirements for the subsystems that

must be met in order to ensure a safe and technically compatible European rail system. Different TSIs are issued by the European Commission for different subsystems. As an example, the TSI INF for infrastructure or the TSI ZZS for train protection, control and signalling should be mentioned. Depending on the application and task, different or even several TSIs may be relevant in the context of the SIL4 Cloud. Here, it must be checked in each individual case, which requirements the respective application must meet. The NNTRs contain rules for compliance with the essential requirements. They are to be applied, if compliance with essential requirements is not covered by a relevant TSI or if the TSIs are not applicable by derogation. A technical dossier shall be attached to the EC declaration of conformity containing all the necessary technical characteristics of the subsystem and certificates of conformity of its interoperability components. This shall include any intermediate assessment reports and the design documents. The technical file shall also contain all the information regarding the conditions of use, maintenance, monitoring, control and servicing of the subsystem under consideration.

2. A declaration of the applicant that the essential requirements are met, and that technical compatibility and safe integration are ensured. The essential requirements contain general conditions and specific conditions concerning the subsystem. All requirements cover safety, reliability, health, environmental protection, technical compatibility and accessibility. The essential requirements are listed by subsystem in Annex III of Directive (EU) 2016/797 on the interoperability of the rail system within the European Union.
3. *a declaration of the applicant that all identified hazards and risks are controlled* i.e., the applicant makes a declaration that all identified hazards and associated risks are kept to an acceptable low level. For this purpose, an independent assessment body is commissioned to prepare a safety assessment report. The safety assessment report shall require the implementation of a risk management process in accordance with Article 5 of the Implementing Regulation (EU) No 402/2013. Unless a Technical Specification for Interoperability requires it to be carried out and any changes are deemed not to be significant, an own safety method may also be used, whose suitability is assessed by a certified inspector. The declaration itself is part of the application form and does not need to be prepared as a separate document:
4. a release of the reviewed planning
5. a confirmation of the usability of the safety-related or electrotechnical systems and of their components. According to the VV GIuV, usability and safety suitability are confirmed by the processes of the sector guideline for proving compliance with the requirements of EIGV. This means that if the processes described there are followed, this point has already been covered.
6. proof that the construction supervision has been carried out
7. the necessary acceptance tests.

In addition, an inspection certificate from a testing expert is required, in which compliance with the technical regulations is certified.

In the context of the SIL4 Cloud project, the extent to which compliance with all the above points is necessary must be examined with the EBA. In particular, Nr 6 relates to safe construction sites and thus to infrastructural buildings, rather than to control and signalling systems.

Safety and electrotechnical systems may be granted a GIuV, if they are to be used in several places in a compatible design and would require an IBG. The requirement for granting the GIuV is the fulfilment of number 1 to 3 of §16 of the EIGV listed above. In addition, an inspection certificate from a certified inspector is required, in which compliance with the technical regulations is certified. For more details, see Sections 10.3 and 10.4.

10.3 Approval assessment – phases according to product life cycle

The sector guideline describes the process of the assessment for approval. It describes the processes for proving compliance with the requirements according to EIGV, in particular technical compatibility and safe integration. Based on the life cycle of a product according to DIN EN 50126-1, a distinction is made in the approval assessment between the requirements specification phase, functional specification phase and product phase. At the end of the three phases, an application for authorization for placing on the market and use (**GIuV**) may be submitted. Even systems for which no application for a GIuV is made at the end, the processes of the sector guideline are also used as a technical regulation to ensure technical compatibility and safe integration. This provides a basis for decision-making for the subsequent authorization process for commissioning and granting an IBG.The named phases are briefly described below. For more detailed information or the treatment of special cases, please refer to the sector guideline.

Note: The following applies to all phases. In case of errors, new requirements or deviations from rules or regulations, a reiteration of one or several phases may become necessary. The respective process must then be run through again to the required extent.

10.3.1 Requirements specification phase

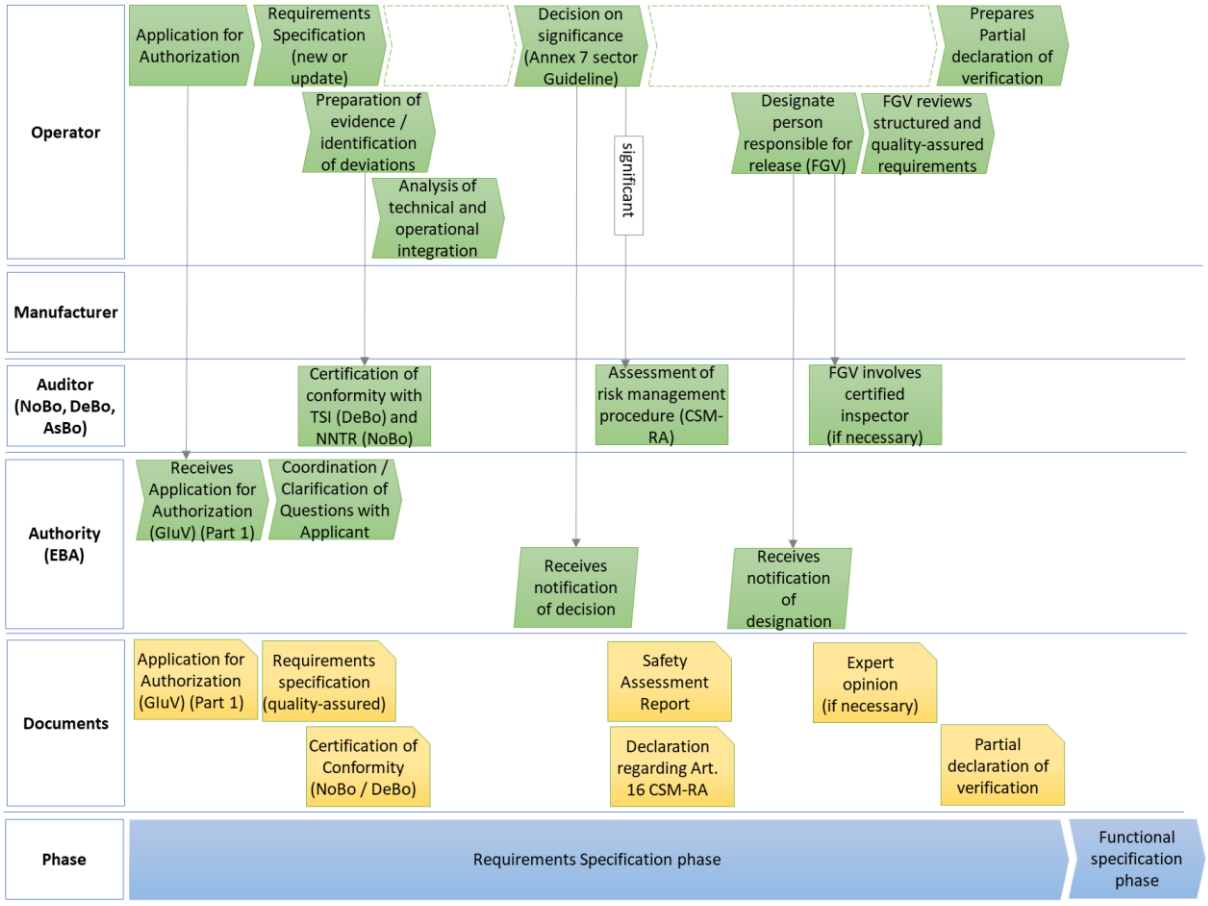

Figure 38: Requirement specification phase – based on Annex 1.1 sector guideline

Figure 38 shows an outline of the process of the requirements specification phase. The requirements specification is drawn up by the operator himself or with external support under the responsibility of the operator as part of the product development. The requirements specification must contain all requirements for the system or component to be evaluated.

The requirements in the specifications must be structured. i.e., new requirements must be identified, checked for significance according to CSM-RA and evaluated according to the risk management procedure. If changes are considered significant, the Independent Assessment Body needs to be commissioned not only with the evaluation of the risk management procedure, but also in detail with

regard to the examination of the content of the safety-relevant change, including the examination of the content of the corresponding evidence. The activities must be sufficiently documented.

For the assessment of the specifications, the operator also prepares evidence of compliance with the relevant legal requirements (accepted rules of technology, operational safety suitability). In this phase, the operator also carries out the operational and technical integration. This means that any necessary adjustments to the operational and technical regulations must be made on the basis of the requirements specifications, and the suitability of all interfaces to the system environment must be ensured.

The operator designates an authorized employee as the person responsible for release (**FGV**) (German: **F**rei**g**abe**v**erantwortlicher). The EBA shall be notified of the appointment of the FGV. The FGV has the task of evaluating the structured and quality-assured requirements. This assessment includes the examination of all requirements with regard to their correctness, consistency, comprehensibility and verifiability. The main focus is on the identification of new and safety-related requirements, as well as the fulfilment of the relevant rules and technical regulations and the documented assessment of significance. The assessment also includes consideration of the operational and technical system integration. Depending on the qualification of the FGV, he or she may have to involve a certified inspector for partial tasks. The certified inspector prepares an expert opinion on the result based on his inspection. If requirements or deviations from rules and regulations are classified as significant, an Independent Assessment Body must also be involved in the assessment of the risk management procedure according to CSM-RA. The FGV draws up a partial declaration of verification (German: Teil-Prüferklärung) in accordance with Annex 13 of the sector guideline as the conclusion of the requirements specifications phase.

10.3.2 Functional specification phase

Figure 39 shows an outline of the process of the functional specification phase. The functional specification contains all intended technical implementations of the requirements from the specifications, as well as, if applicable, further requirements that are necessary to fulfil the specifications. The requirements specification and all applicable documents of the previous phase serve as input. The manufacturer is responsible for drawing up the functional specification. The manufacturer also provides evidence of compliance with the requirements specification, the legal requirements and the accepted rules of technology, as well as of the implementation of all requirements and conditions imposed by the operator. He also confirms the operational safety suitability from the manufacturer's point of view. The requirements in the functional specification are to be structured according to the activities as in the requirements specification phase.

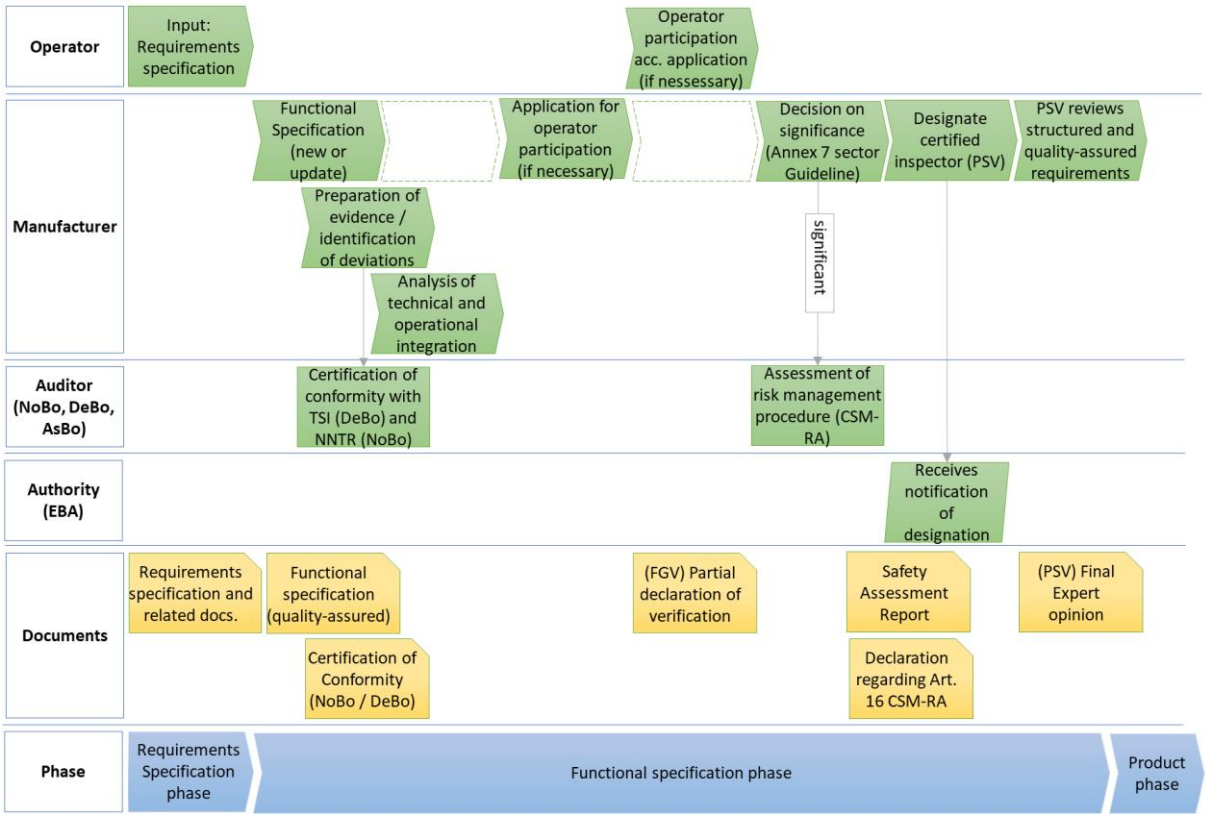

Figure 39: Functional specification phase – based on Annex 1.2 sector guideline

In accordance with Annex 9 of the sector guideline, it must be checked whether operator participation is necessary in the requirements specification phase. This is not necessary, for example, if there is a change or new development of a generic product according to the definition of EN50129. If operator participation is necessary, the operator appoints a FGV (see requirements specification phase), who acts as the interface between operator and manufacturer. The FGV creates a partial declaration of verification in which he confirms that safe integration is guaranteed from an operational point of view. Any associated additional requirements or conditions for the later product are documented in the partial declaration of verification. If new requirements or deviations from the rules arise, these must again be examined for significance and, if necessary, the risk management procedure must be evaluated by an Independent Assessment Body. If the changes are not significant, the person responsible for the release, or a certified inspector, prepares evidence of the manufacturer's own safety method.

The manufacturer commissions a FGV or a certified inspector with the testing of the following aspects. New requirements must be checked for suitability, consistency, comprehensibility and completeness in accordance with the requirements specification phase. The responsible person confirms, that the requirements of the previous phase are correctly and completely fulfilled by the regulations of this functional specification phase. Compliance with all relevant rules, technical regulations and European and national laws must also be checked by the FGV. Finally, the technical and, if applicable, operational system integration must also be evaluated.

The final result of the functional specification phase is an inspection report or expert opinion by the certified inspector. All partial results generated in this phase are included in the report. This creates a final document that summarizes all the statements from all the testing and assessment bodies involved and, if applicable, refers to all other documents that contain the corresponding content.

Note: It is possible that even in the functional specification phase, the focus of activities lies with the operator. In this case, the manufacturer first carries out his activities up to the preparation of the expert opinion of the PSV. And after that, the FGV at the operator evaluates the results and, on this

basis, draws up his partial declaration of verification. Thus, this partial declaration of verification by FGV represents the final result in the functional specification phase.

The manufacturer prepares a report on the preparation and approval assessment of the functional specifications, that consists of a sufficient process documentation about this phase including any assessments and evidence that may have been created by any independent assessment body. This report and the functional specifications serve as the input for the next product phase.

10.3.3 Product phases

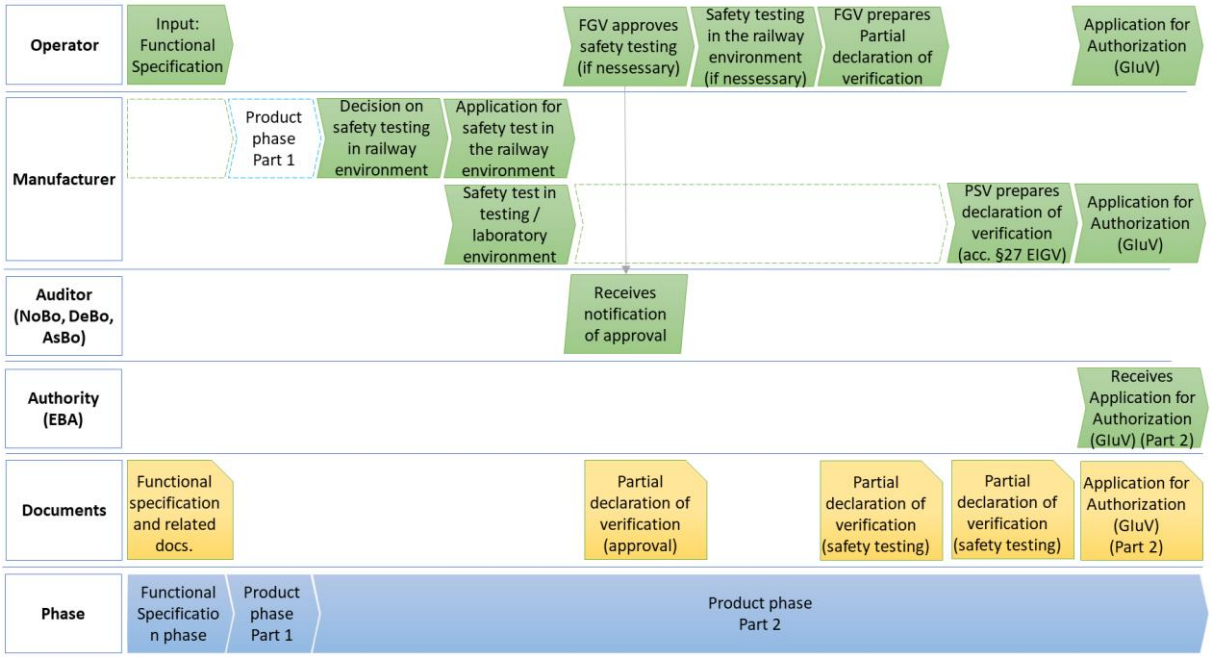

Figure 40: Product phases – based on Annex 1.3 sector guideline

Figure 40 shows the second part of the process of the product phase. For reasons of simplicity, Part 1 has been omitted in the graphic. It does not differ from the functional specification phase and can therefore be taken from Figure 39. In the beginning, the processes are very similar. The manufacturer creates the product according to the specifications of the approved functional specification. He then produces evidence of compliance with all relevant legal requirements and the accepted rules of technology. Furthermore, the manufacturer provides evidence of compliance with and implementation of all requirements and conditions from the documents generated in the functional specification phase. The technical integration and operational safety suitability from the manufacturer's point of view must also be proven. In case of ambiguities, the operator is consulted. In accordance with Annex 9 of the sector guideline, it is checked whether operator participation is necessary in the product phase. If this is the case, the procedure is the same as for the functional specifications phase. As in the other phases, new requirements or deviations from regulations are to be checked for significance in accordance with Annex 7 of the sector guideline, and dealt with accordingly.

From here on, the process in this phase begins to differ. In accordance with Annex 6 and Annex 9 of the sector guideline, the manufacturer checks whether a safety test in the railway environment is necessary as part of the safety verification in accordance with EN 50129:2019. The necessary procedure in the event that safety testing is required is also regulated in Annex 6.

The manufacturer commissions a certified inspector to evaluate the evidence of fulfilment of all requirements within the scope of product realization. Depending on the type of product, the sector guideline describes different test contents. In the following, the contents of the test and evaluation of the certified inspector are described, if the product is a generic application. The other product types will not be discussed further here, as they will probably not be relevant for this project, as the Cloud platform that enables the execution of different safety-relevant applications can be seen as a generic application here.

The appointed certified inspector confirms the suitability of the product for the intended purpose and checks compliance with all relevant rules and technical regulations. In the case of deviations from the regulations, he checks the admissibility and correctness of the evidence. He confirms the completeness, comprehensibility and suitability of the documents submitted by the manufacturer and that these prove the suitability, safety and conformity of the product with laws and standards. This evaluation also includes all documents necessary for the use of the product. The documents must be sufficient, correct in content and plausible. He also confirms that all requirements for the product from the specifications have been implemented correctly and completely, and that the requirements of the technical system integration have been fulfilled. If necessary, he also confirms the positive result of the operational system integration. If new requirements have been added in the product phase, the appointed certified inspector also checks the existence of a documented significance assessment. The identification of the necessity of an operator participation in the product phase, and a safety test in the railway operational environment, is also part of the examination. If a necessity exists, the test report is also checked and the consideration of the results of the operator involvement is confirmed on the basis of the partial test declaration of the FGV. The appointed certified inspector draws up a final report in which he includes all results of the other inspection bodies and independent assessment bodies. This creates a final document that summarizes all statements from all participating testing and assessment bodies or, if applicable, refers to all other documents that contain corresponding content. On the basis of the final expert opinion, the certified inspector draws up a test certificate for the subsequent application for a GIuV by the manufacturer or operator. The applicant also prepares the partial data sheet in accordance with Annex 15 of the sector guideline. The correctness of the partial data sheet must be confirmed in the inspection certificate of the certified inspector.

10.4 Application for authorization for placing on the market and use

The application for an authorization for placing on the market and use (**GIuV**) is split into two parts. In Part I, the subject of the application is first described and classified in the existing subsystems (e.g., energy, CCS, infrastructure, etc.) so that the corresponding regulations to be applied can be listed. If necessary, new or modified technologies are also described here and the underlying regulations and standards are listed. Furthermore, information on the applicant is transmitted to the EBA. The manufacturer of the product or the operator is entitled to apply. Part I of the application should be submitted as early as possible. On this basis, coordination with the Federal Railway Authority on the possible granting of an authorization for placing on the market and use, and the clarification of questions for later evidence, can be carried out at an early stage, and thus the necessary reliability in planning can be achieved. Part II is the actual application for the GIuV, which is submitted at the end of the product development phase. It contains the necessary declarations and evidence of EIGV § 16 paragraph 1, sentence 3, numbers 1 to 3 (see also §27 EIGV), that are listed in chapter 10.2. Part II and the documents referred to therein are not submitted by the applicant, until the evidence documents to be submitted have been completed, and on this basis a decision can be taken by the Federal Railway Authority on the granting of the license.

10.5 Commissioning approval procedure

The VV IBG Infrastructure distinguishes between two different application scenarios, depending on the status of development of the system under consideration. However, the scenarios only differ in minor details within the scope of the notification or the application, but not in the examination for the granting of a commissioning authorization (**IBG**). In the first scenario, the first commissioning of a component of the railway system is considered. This is understood as the new construction of a line between two nodes that were not previously connected by railway infrastructure. The second scenario considers the upgrading and renewal of components of the railway system. Measures of renewal and upgrading require an IBG if they fulfil any of the criteria of Annex 4 EIGV. In contrast, maintenance work does not have to be notified to the EBA. Annex 5 EIGV shows which measures are

considered as replacement in the course of maintenance measures. In this case, notification to the EBA is only required if the repair is part of an upgrade or renewal.

The notification of the upgrade or renewal of a component of the railway system must be submitted to the EBA at least 10 weeks before the start of construction. However, it is recommended that the notification be made as early as possible. Particularly in the case of measures on the CCS subsystem with ETCS or GSM-R, more time should be considered, as the approval of the ERA in accordance with Art. 19 of the Interoperability Directive 2016/797 can only be applied for once the need for an IBG has been established. For STE installations, the form from Annex 2.1b of the VV IBG Infrastructure is used for notification. For this purpose, the documents required in §21 (2) VV IBG and Annex 6 EIGV must be provided, and corresponding documents must be submitted. After receiving the notification, the EBA decides on the basis of the notification whether an IBG is necessary for the measure notified. If the decision is positive, the notification shall be deemed to be an application for authorization to place in service. The application is checked for completeness and testability. The application is deemed to be ready for inspection, if the documents to be submitted are the final version of the documents. This can always be assumed if the documents are not recognizably marked as a draft. Furthermore, the documents submitted must be dated and signed, or bear an equivalent electronic signature, and be written in German. The result shall be communicated to the applicant within 4 weeks. In the event of objections or additional requirements, the applicant shall be granted a period of time, in which the missing documents are to be submitted.

After checking for completeness and testability, the EBA shall examine the submitted declarations, documents and evidence for comprehensibility. The application documents must be coherent in terms of content. A substantive examination by the EBA does not take place. As also described in Section 10.2, the verification of whether the evidence for the individual licensing requirements meets the applicable requirements is carried out by various inspection bodies (Nobo, DeBo, AsBo) and inspection experts. A decision by the EBA takes place within 4 months, provided that the documents are free of defects. In case of justified doubts, the EBA may demand additional tests for the granting of the IBG. If the applicable requirements of §16 EIGV are fulfilled and the described tests have been carried out, the EBA shall issue the IBG. Conditions and other restrictive provisions from the test documents, certificates, etc. must be summarized and evaluated by the FGV and, if necessary, compensated for by suitable alternative measures. These conditions can be summarily or individually included in the IBG as ancillary provisions within the meaning of § 8 EIGV. This applies in particular, if the conditions are necessary for the fulfilment of the essential requirements, or for the safety of railway operations.

10.6 Generic and specific application – safety case

According to §27 EIGV, a safety-related or electrotechnical system or its components may be granted an GIuV if they are to be used in conformity at several locations, in accordance with the CENELEC standard EN 50129, which distinguishes between generic and specific safety cases. The CENELEC standard further distinguishes between generic products and generic applications. A generic product can initially be used in different classes of applications. A generic application is a system that is only suitable for one class of applications. In the case of a specific application, the system is intended for a very concrete (specific) application. For instance, in a level crossing, the generic product is represented by a computer that can reliably process input variables into output variables in terms of signalling technology. The generic application for this is then a software that can process messages of occupancy, close barriers, transmit the closure, and raise the barriers again after they have been cleared. In this case, the specific application is represented by a concrete level crossing, for which dependencies on surrounding and adjoining subsystems must also be considered.

In the context of the SIL4 Cloud project, the Cloud Platform that enables the execution of different safety-relevant applications can be seen as a generic product. The individual applications, which in turn take on concrete tasks, are then to be treated as specific applications.

The TAS Platform (see Chapter 6) is also a generic product, on which various products are built as special applications. This is shown in Figure 41 within the context of Main Line Rail or Urban Mobility.

The development of a product according to the CENELEC standard EN50129 ensures that the product is free of systematic faults. Furthermore, the safety verification demonstrates that random faults are controlled and do not lead to dangerous situations. In the context of the project, there are some challenges that need to be solved for the successful safety case. These challenges are discussed in more detail in Section 10.7.

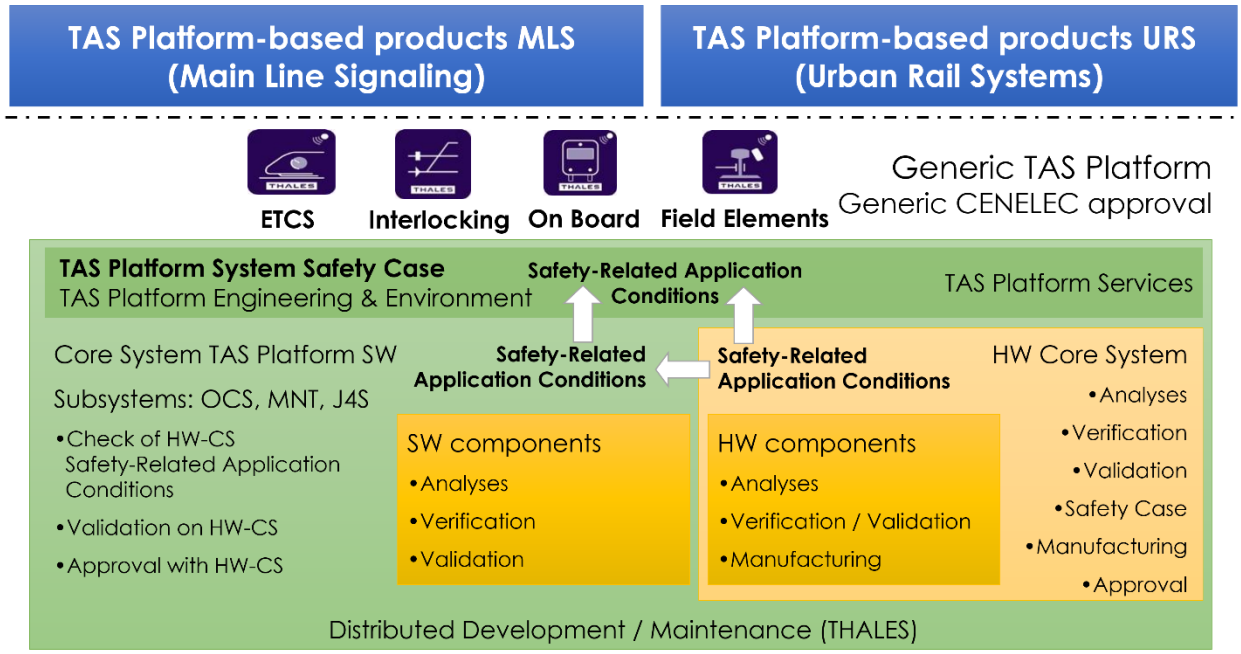

Figure 41: TAS Platform generic safety case

10.7 Safety-related challenges of a SIL4 capable Cloud Platform

The objective of a safety case is to provide evidence for:

- Efficient quality management,
- Efficient safety management,
- The achievement of functional and technical safety.

Especially the third point in this list can be challenging for a SIL4 capable Cloud Platform. Reasons for this are provided in the following sections.

10.7.1 Dynamism

Today's public/private Cloud Infrastructures are characterized by their highly elastic nature, such as:

- New Functional Application(s) can be started or stopped anytime.
- Resources allocated to Functional Application(s) can be resized (e.g., extended or reduced), and Functional Application(s) can be migrated from one machine to another.
- New hardware (in form of machines/servers) can be added or removed anytime.
- Cloud/data centre infrastructure provider overprovisions resources to optimize the utilization of hardware resources.
- This all happens without shutting down any running Functional Application(s).

This dynamism, however, contradicts principles that are recommended or requested by safety standards. For instance, in DIN EN 50128 Table A12, the omission of dynamic objects or dynamic variables is highly recommended (HR). This however is the computational foundation to implement dynamism.

Another challenge is that each variation of the hardware and/or Functional Application setup can be understood as a separate configuration. DIN EN 50128 requires that each SW-configuration must be specified, documented and tested.

In order to handle the contradiction of dynamic behaviour with the safety standard, it is important to show that the developers were aware of all possible failures that may rise from it, and that effective measures have been applied.

- This can be done by identifying all risks or failures that are introduced by utilizing dynamic behaviour. E.g., the use of dynamic memory opens the risk of memory leaks or so-called use-after-free scenarios, where undefined data is accessed. The latter may lead to undefined behaviour of the system and yield erroneous results.
- When all possible failures are identified, safety measures shall be introduced into the system to mitigate all unsafe failures. For instance, the addition of a new Functional Application to the system may only be safe, if it is ensured that this process may never interfere with any other running Functional Application on the system (see 10.7.2). However, finding measures (and arguments) to safely implement overprovisioning or the reduction of resources may be challenging, or even impossible, on a generic Cloud Platform.
- Finally, a verification must be conducted to show that each measure is effective and each of the identified failures can be handled properly.

In order to argue that any configuration of the system is safe without explicitly testing it, the key arguments are:

- To show that each new configuration does not interfere with the previous configuration
- To provide automatic self-tests for each new configuration. For instance, when a new replica is introduced, it is ensured that this is working correctly, and is an exact copy of its original application, before the new configuration is accepted to be safe and taken as a reference for the next change.
- To apply extensive testing upfront that tests several scenarios and corner cases of different configurations.

  With this method, features like replication, migration, addition of communication channels or partners or other dynamic changes in a server environment can become possible. However, each new safe application that is introduced to the cloud will have to be tested upfront in order to ensure that it was implemented and set up correctly for the Cloud Platform. A safe manipulation of available resources may also be hard to justify, since this may affect execution time in an unpredictable way.

10.7.2 Freedom from interference – safety view

Freedom from interference is a key concept for achieving functional safety in the Cloud. It is required for enabling reconfiguration of the Cloud, and it is even more important for enabling replication, and eventually achieving functional safety. However, several aspects in a cloud environment may break the freedom from interference. To name a few:

- An unsafe operating system or hypervisor may suffer from systematic failures that affect multiple replicas, or it may simply behave in a non-safe way. For instance, by collocating replicas of the same application on a single core or RAM-Module.
- Network traffic uses best effort protocols and may suffer from congestion caused by high traffic applications, which in turn will interfere with message runtimes of other applications.
- Multiple replicas may run machine clusters with a central power supply.
- Incidents may affect a whole data centre, as seen in the OVH Data Centre Fire or the Amazon Web Services (AWS) outage in 2021.

As the Cloud Platform heavily relies on replication techniques to achieve functional safety, evidence must be provided that each application instance (i.e., replica) is free from interference with other replicas and applications, or that any remaining interference is safe. This can be done by conducting a dedicated analysis that is based on sophisticated methods like FTA or HAZOP. Also, measures can be introduced that bring the system in a safe state in case of any significant interference.

10.7.3 Asynchronous network/ Time synchronization

Traditional Cloud environments heavily rely on fast but asynchronous networks. It means that messages can be delayed indefinably (although this is seldom a problem). However, this lack of real-time guarantee is a problem for safe operation, which heavily relies on predefined error detection and fault handling time intervals. This even becomes more challenging due to the fact that each replica shall run on its own machine with its own clock. To provide the argument that the predefined fault detection and fault handling time intervals are met, a reliable/safe time synchronization can be implemented or provided by a dedicated Cloud service.

10.7.4 Handling systematic failures in underlying non-SIL system-software

A key argument required in the safety case is the evidence for achieving sufficiently small mean time before-failure (MTBF) rates. For hardware that was developed according to a safety standard, the MTBF can usually be provided, even if the MTBF does not reach the desired numbers in first place. Software measures like replication or self-tests can be applied to improve the MTBF. However, Cloud infrastructures heavily rely on non-SIL, COTS hardware. This leads to the following challenges:

- MTBF numbers are not available for the hardware.
- MTBF is provided, but is not trustworthy, since no reliable development process has been applied. Please note that if this is to be applied, the “similar machine” most likely won’t be available for purchase right away, since at least two new processor generations would be rolled out between gathering the data, analysis and the publication.

It should also be noted that electronic components used for computing boards/hardware today are likewise not produced according to a reliable development process, and that their FIT rates are used for calculating the MTBF.

To cope with that in the safety case, the MTBF of the COTS hardware can be pessimistically estimated by considering scientific publications and literature that provide MTBF for similar machines. Replication schemes can then be used as an argument to raise the MTBF to the required values.

11 Migration of legacy systems

Migration in railway systems has many different aspects. In this chapter, some thoughts on how to migrate existing/legacy systems toward SIL4 Cloud infrastructure are listed. For this, the most essential systems currently in the field are addressed, which are Interlockings and RBCs.

11.1 Interlockings

The reason why current and legacy Interlocking systems were deployed in a relatively small area is the fact, that those systems always controlled all the field-elements from a central place ("relay-room") via copper cables and electric signals. To keep environmental-influence such as electro-magnetic disturbances or effects from lightning-strikes in an acceptable range, this room could not be far away beyond roughly 7 km. As it is not feasible to replace such installations in total, this kind of technology will be in the field for the next 25 years or more from their date of provisioning.

Systems deployed today usually go one step further. They put field-element control in a very close perimeter of the field-element itself (10-100m). These so-called object-controllers are then controlled via a standardized (IP) data connection and some power. This way, they are much more suitable for being controlled from a location-independent SIL4 Cloud, as data transmission can be realized with COTS devices. The term used for this concept is "Digital Interlocking", and it is already running in field.

In each of the above cases, the logical core of the Interlocking would be "transferred" into the SIL4 Cloud under the control of some SCP. The methodology, how this can be done technically, depends on the architecture of the Interlocking. If the SW architecture already relies on standardized APIs with separation of safety, the step into that Cloud would not be that big. If a totally different SW architecture was chosen, other technical options need to be considered. In any case, it needs to be decided case-by-case and cannot be answered in a generic way for all existing systems.

In every case, the transferred SW needs to be re-tested and re-assessed – as would be the case, if just the HW-board would be changed. Additionally, the benefit is not very big, given the fact, that for most already deployed solutions, the physical room (relay room) will still be necessary, as replacing legacy field-element control by object-controllers would be even more expensive.

Another, possibly more suitable alternative would be to simply change the software of a legacy Interlocking in a way, that it behaves like a collection of RCA compatible field elements on the interface. This way, legacy systems can be brought into a new system relatively smoothly, without the necessity of transferring old architectures to modern SIL4 Clouds.

Advantages and disadvantages of these possibilities need to be evaluated and rated with the help of the infrastructure managers.

11.2 Radio Block Centre (RBC)

As the name implies, RBCs are already centralized. The central part of current RBCs can be migrated relatively easily and smoothly on the SIL4 Cloud, assuming that these systems are not very old (around 10 years), and therefore already adhere to quite a modern SW architecture.

The nature of the RBC is to communicate with trains over the air interface. For this communication, some specific technology and hardware is used. This hardware, of course, still needs to exist and needs to be accessible by the RBC core network. As those systems are not very old, communication is already based on modern protocols such as TCP/IP and Ethernet and therefore, connecting to a SIL4 Cloud is not expected to be demanding.

As for Interlockings, this one-time-effort of transferring the existing system onto a SIL4 Cloud is a step which would incur costs, and introduce the necessity of re-assessment.

Another possibility would be to switch directly to modern, Cloud-based “by design” RBC, and simply replace the existing ones. This procedure would be much easier in comparison to that for Interlockings, because the RBCs are already centralized and the number of deployed RBCs is much lower than that of Interlockings.

Before a well-founded decision can be taken, these alternatives need to be assessed in greater depth, also considering economical aspects.

TAS Platform is based on standard linux. This means, that a big number of legacy SW can run on it or can be ported with low effort. Furthermore, if non-SIL and other OS (like windows) it could be instantiated in a side-by VM and simply use a communication channel easily.

PikeOS allows to execute existing software within a virtual machine, called resource partition. It does not matter whether legacy application used to run on the bare metal or had been using an operating system so hosting legacy applications is not so challenging.

12 Evaluation of high-level objectives

This chapter assesses the architecture and concepts of the SIL4 cloud against the high-level objectives of RCA/OCORA Safe Computing Platform (see Section 2.1 and Annex A).

12.1 Meet safety and real-time requirements of CCS (and similar) railway applications

Recap from RCA/OCORA White Paper: The platform shall meet safety requirements of applications up to safety and integrity level (SIL) 4, e.g., acc. to EN 50126, EN 50128 and EN 50129, and support applications with real-time characteristics (e.g., overall processing cycles in the order of 10-100ms).

As the platform is intended to host software with SIL4 functionalities, meeting CENELEC EN standards is clearly a must requirement, and it is achievable. In the distributed system, real-time demands become more important since the correctness of output depends not only on the logic result, but also on the timing, at which the results are produced. Platform and applications, both must support real-time characteristics to implement deterministic time behaviour of the applications. To analyse timing demands of a platform, following definitions are required:

Functional Application's end-to-end reaction time is the response time required by a business logic under given conditions. As the subsystem's specification (involving multiple applications) will be defined by railways, RCA and OCORA, the maximum reaction time must be specified. As the current requirements of interlockings (IXL) are already below 500ms end-to-end, including EN 50159 protected communication, end-to-end reaction time could be considered as below 500ms.

Thales: The TAS Platform approach, which is in field usage since over 20 years, has proven to suit the task of building fault-tolerant, safety-critical, real-time systems with highest safety requirements. The TAS Platform is used for all safety-relevant products within Thales Main Line Systems and was recently extended to be usable for the cluster/server approach as well.

For safety, the TAS Platform is assessed as a generic product for SIL4 usage according to CENELEC EN 50129, EN 50128, EN 50159.

For security, the TAS Platform is assessed as an enabling component for SL3 according to IEC 62443 4-1 and IEC 62443 4-2.

SYSGO: For building a safe, hard-real time system, the underlying OS and platform APIs shall have statically known worst case execution time (WCET). PikeOS is a certified RTOS with APIs characterized with WCET.

12.2 Minimise total cost of ownership

Recap from RCA/OCORA White Paper: The platform shall minimize the total cost of ownership, i.e., the overall life-cycle cost. Here we could consider various direct (under regular situations) costs incurred by the SCP/SIL4 cloud/data centre.

Performance of SIL4 Cloud plays the key role in optimising cost. E.g., a highly available and reliable platform has minimum downtime, hence higher rail operations.

Thales: For fulfilling this task, three essential capabilities are key: Flexibility, Modularity and Standardization. All those capabilities are well combined in the TAS Platform, leading to unquestioned advantages for all the products. TAS Platform is flexible in terms of extensibility with additional functionality, including features coming from "the community", and scalability, from the single-board low-performance computer up to many-CPU server boards. It is standardized in terms of API and other OS-centric concepts. Finally, it is modular in terms of software architecture and safety assessment, where a very clear and understandable ruleset, to be fulfilled for getting applications safe, exists.

SYSGO: The modularity and flexibility of PikeOS allows fast adaptation to new hardware or software and brings predictability to life cycle cost. Fast transition from development to deployment, reducing time-to-market and total cost of ownership.

12.3 Avoid vendor locking/vendor independence.

Recap from RCA/OCORA White Paper: Different vendors shall be able to provide Functional Applications, computing platforms and development tools, respectively, without a vendor lock-in. It shall be possible to purchase hardware directly from different vendors throughout the lifespan of the software. The platform shall build on existing HW/SW solutions, stimulating competition among vendors and allowing them to shine with their specific expertise and distinctive solution features.

Vendor independence can be achieved by the following:

- Decoupling Functional Application from underlying platform by implementing PI API. For details, p*lease refer to Chapter 8.*
- Independence from computing hardware: it is possible to support different computing hardware with different processor architecture in SIL4 Cloud. A key requirement would be a qualification process for the type of HW. Once the HW variant is qualified, an exchange can be done without assessment, such as for maintenance activities.

Thales: TAS Platform adheres to the independence principles from the very beginning. As CPU complexity gets higher, hardware independency strategies will be the only possibility to use modern hardware in highest-safety environments. As the API of TAS Platform already acts as an abstract interface to different types of boards, with more than 10 types currently actively used, this feasibility-proof was already done.

SYSGO: PikeOS's standardized RTE interfaces allows to integrate the SCP from different vendors.

12.4 Respect diverse lifecycles of business logic, run-time environment and hardware

Recap from RCA/OCORA White Paper: The platform shall be partitioned with respect to the different life-cycles of business logic, run-time environment and hardware. The platform shall support fully independent life-cycle handling, i.e., with minimal dependencies.

Thales: TAS Platform implements the different life-cycles by adhering to key qualities. Those are backward-compatibility, modularity and stable interfaces. From a technical point of view, these qualities are supported with specific remote-update modules, usable for OS, application and configuration data. Furthermore, a holistic method of supervision brings the hardware independence demanded, wherever needed.

SYSGO: Our products such as PikeOS are maintained for the customer-defined product life-cycle. Maintenance is always long-term-based, can be extended to full customer device configuration (3rd party drivers, BSPs, etc.).

12.5 Open market to new players

Recap from RCA/OCORA White Paper: The platform shall open the market to new, non-rail-oriented software and tooling companies. They shall be able to become involved in Functional Application development without providing their own platform safety mechanisms (e.g., related to safe communication, fault tolerance implementation, etc.).

Thales: The clear interfaces and API used for TAS Platform makes this task feasible. In fact, hardware boards already come from five different vendors, and applications span from small object-controllers, over classical Interlockings and RBCs, up to Onboard Systems, and from internal Thales research-and-development centres for Main Line Systems, Urban Rail Systems, up to joint-ventures using TAS Platform for various purposes.

SYSGO: SCP is modular and allows integration of components from different vendors on all system levels including hardware, operating system, RTEs, middleware, application, and tools.

12.6 Industrial readiness

Recap from RCA/OCORA White Paper: It shall be possible to procure a platform as off-the-shelf solution supported by an open and dynamic market. The solutions shall be mature (e.g., reliability proven in field) and backed by effective acceptance and integrated logistical support (e.g., maintenance service, tooling, availability of spare parts).

Thales: As TAS Platform already serves the base for all Main Line System products, industrial readiness is long given and proven. The step to open TAS Platform for the benefit of third-party companies under the name “TransVital” is currently under way.

SYSGO: Reuse of existing code and, where applicable, certification artefacts means that market readiness is usually fast. For the same reasons, new functions can also be integrated quickly, economically and functionally safe with an RTOS, and in particular, a hypervisor-based RTOS.

12.7 Migratable and portable business logic

Recap from RCA/OCORA White Paper: The business logic is considered a significant system asset, being the component with the longest lifetime. It must hence be portable to different computing platform evolutions. We here further differentiate: • Migratability for legacy applications: It should be decently easy to migrate legacy applications to the new platform; • Portability for new applications: Applications running on the platform should be portable to any other vendor’s or evolved version of the platform.

Thales: TAS Platform uses a stable API. This makes it possible to still run products with an age of more than twenty years on current TAS Platform versions. Backward-compatibility ensures that mature applications can still be run on current versions. Extensions and additions were, and will be, done in a careful and compatible way. The hardware boards are designed to be pin-compatible towards the hardware-layer and function-compatible towards the OS without critical breaks. This will be easier for server concepts, as the hardware layer is abstracted and backplane pinnings are not of relevance.

SYSGO: Extreme flexibility provides independence from suppliers in the choice of hardware and software: PikeOS supports a broad range of hardware architectures and software interfaces. Easily adapt to new requirements and incorporate legacy technology.

12.8 System evolvability

Recap from RCA/OCORA White Paper: The platform shall be open to extensions (in the sense of additional system services that are added over time, e.g., related to FRMCS). Adding new functionalities shall be possible with minimal to no changes to existing applications (though these may naturally not be able to leverage the new functionalities).

Thales: Evolution always means to either translate new functions to currently used APIs or to introduce new APIs (and functions) in parallel to the already existing ones. These steps were done several times within the life cycle of the TAS Platform. System evolvability is proven to be possible, feasible and useful.

SYSGO: The modularity and flexibility of PikeOS allows fast adaptation to new hardware and software. With each release of the API, a clear list of deprecated/removed API calls shall be provided, with a clear way on how to migrate from Version X to Version X+1.

12.9 Facilitation of application development

Recap from RCA/OCORA White Paper: The platform shall use an open, well-documented application model and programming interface, facilitating that third parties develop applications.

Thales: TAS Platform comes with a certification according to EN 50129. This means, that the whole system including OS, HW and necessary tooling is covered. The applications only need to prove adherence to the rules (programming model / API, safe application conditions, secure application conditions) in order to get safety- and security certification, if needed. This way, the application engineers can focus fully on the business logic, as all base functionality is provided and taken care of by the TAS Platform.

SYSGO: PikeOS is pre-certified according to EN 50128 and EN 50657 SIL4 and enables embedded software engineers to build outstanding embedded railway software fulfilling strong safety demands.

12.10 Modularity

Recap from RCA/OCORA White Paper: The platform shall allow for a modular safety certification process, using pre-certified components leading to a dramatically simplified and shortened full system certification process. An evolution or update of the platform shall not require a new E2E homologation of application and platform, as detailed in Section 8 of RCA/OCORA White Paper.

Modularity of SIL4 Cloud could be discussed/divided as follows:

- Decouple the processes of certification of applications and qualification of server hardware. The basis is a reference computing platform with well-known properties. A server qualification suite is used to demonstrate the suitability of the server for use within the computing platform.
- Like today, novel applications can be tested on a reference computing platform for certification and later be integrated on specific platform realization (which is similar/based on reference computing platform).
- The test setup for certification, where the applications are evaluated together with the RTE, shall be certified.

Thales: As described in earlier statements, TAS Platform fully follows a modular approach, for functionality and for assessment. Furthermore, it already supports the possibility of running more than one independent application on the same TAS Platform instance, irrespective of the safety-level of those applications. And TAS Platform can also run a virtual machine of e.g., another TAS Platform for specific purposes, as needed by the user/application.

SYSGO: PikeOS provides a modular system architecture allowing various applications to run simultaneously on a single hardware. A safe and efficient integration in safety-critical systems is achieved via virtualization technology.

12.11 Encapsulated, transparent fault tolerance mechanism

Recap from RCA/OCORA White Paper: The platform shall transparently encapsulate the safety and fault tolerance mechanisms. Vendors may offer different (new) approaches to safety and fault tolerance as they become available on the market – solution agnostic and future-proof.

Thales: TAS Platform comes with a dedicated safety layer. This layer is encapsulated in a way, that the application engineer does not need to change the API, if the safety layer needs to be used. A safe application has to follow well-defined, documented and trained rules (i.e. different programming model in our nomenclature). Those rules and the knowledge, how to use them, are the most essential part of TAS Platform.

SYSGO: The PikeOS isolation mechanisms (spatial/temporal), and internal health monitoring system provided, support transparent fault tolerance and isolation for safety critical applications.

12.12 Scalability

Recap from RCA/OCORA White Paper: The platform shall be highly scalable, i.e., it should by design be able to support an arbitrary number of applications and arbitrary number of compute nodes.

Thales: TAS Platform is used for different types of application. This starts with very low-performance single object controllers, over small, medium or large Interlockings, RBCs up to very ruggedized onboard systems. Furthermore, all potentially virtual systems (like Interlockings, RBCs and future RCA-Safety-Instances) can already be executed on a server/cluster in a safe (CENELEC) way. As server hardware is completely independent from the platform, computing power can be dynamically assigned as well. Of course, this kind of functionality needs to be supported also by the (future) applications.

SYSGO: Flexibility and scalability are the most prominent needs for future train control SCP. Scalability shall ensure that, whatever complexity, performance, safety and security demands the final use-case may require, the SCP platform doesn't have to be changed. PikeOS meets the requirement of being pre-certified up to the highest assurance level according to EN 50128 is self-evident.

12.13 Flexible usage of compute resources

Recap from RCA/OCORA White Paper: The platform shall enable a flexible mapping of business logic to compute resources (e.g., such that the platform can be expanded while applications are running, and that business logic can be re-mapped when compute nodes fail). It shall be able to leverage advances in computing technology (i.e., when better compute nodes are available, it shall be possible to assign more instances of business logic to the compute nodes).

Thales: TAS Platform fully abstracts hardware form application. This includes all computing resources like CPU, memory, devices and so on. In principle you could, for example, run several applications on one CPU or one application on several CPUs, only dependent upon the configuration and the needs of the application. Generic support for dynamic allocation is available, but the application architecture must be suitable to match this requirement.

SYSGO: PikeOS allows applications to have access to different set of resources (CPU time, memory, IO devices) depending on the operation mode.

12.14 Flexible hosting architecture

Recap from RCA/OCORA White Paper: Centralisation: The platform shall allow to centralise applications physically in a safe data centre to simplify life-cycle management, reduce TCO by means of simplified, optimised operations, and benefit from increased availability and optimised resource usage.

Thales: TAS Platform's principle works on each type of server hardware. Furthermore, it is, by definition, independent from the used hypervisor, adding even more flexibility in this sector. Of course, both, the server HW and the hypervisor, need to meet basic qualities and requirements.

SYSGO: PikeOS modularity, virtualization and integrated safety brings predictability to life-cycle costs.

12.15 Support for running multiple applications (also with different SIL levels) on one physical platform

Recap from RCA/OCORA White Paper: It shall be possible to run multiple applications, possibly with different SIL levels, on a single physical platform to reduce cost, space, power dissipation, etc., and simplify certification, maintenance, system evolution, etc.

We can divide this objective into the following points:

- Using a partition/VM-based approach, it is possible to host multiple applications on one physical platform. [here is more than one SIL application]
- Using applications of different SIL level on one VM/physical machine needs to be treated within the safety- case of the application. Of course, the OS below must support/allow it.

Hosting applications with different SIL rating require following consideration at platform side:

- Each application needs to be investigated regarding its safety need and be assessed accordingly. Again, the underlying operating system needs to support mixing different SIL levels, otherwise a basic SRAC would be violated and the assessment would fail.

Thales: TAS Platform supports mixed-criticality applications by using the feature MAPS – Multi Application Support – within one virtual machine or several virtual machines on server or embedded board configuration. This way, the user has all flexibility to design a system according to his needs.

SYSGO: PikeOS MILS separation kernel provides strongly separated containers for applications with different SIL levels to co-exist on a single hardware. In this way, only the really needed SILx activities are carried over for the specific applications, avoiding the need to manage each application at the same (top) level of safety-criticalness.

12.16 Life-cycle management capabilities

Recap from RCA/OCORA White Paper: The platform shall provide automated mechanisms related to software lifecycle and configuration management, diagnostics, etc. This may also be expanded to software development automation, e.g., taking orientation in SPEM.

Thales: TAS Platform supports an addon named “MNT”. This package contains specific maintenance- and diagnostic features, as well as remote update for the application, OS and configuration. It also assures security for these kinds of operations.

SYSGO: The contradiction between rapidly changing electronics and long life-cycle of railway systems requires an intelligent and extensible system architecture that honours older, legacy components, while being open to easily adding components based on new architectures. System services such as application life-cycle management is part of PikeOS system services.

12.17 Meet Security requirements

Description: Recap from RCA/OCORA White Paper: The platform meets the Security standards, guidelines, policies defined and established by respective railway organisations.

Thales: TAS Platform has a security certification according to IEC 62443 with security level 3 (SL3) and maturity level 3 (ML3). It provides Security Application Conditions and a Security Handbook towards the application to make developing a certified application as easy and straight-forward as possible. The TAS Platform also follows the TS 50701.

SYSGO: SCP based on PikeOS provides the highest security level for safe and secure railway applications as PikeOS (RTOS and Hypervisor) is certified to Common Criteria (CC) at evaluation assurance level (EAL) 3, which is a widely accepted security standard for COTS OS used in railway [23].

13 Conclusion and next steps

With multiple industry partners, this project is first of its kind on Safe Computing Platform topic which is focused on trackside use cases. Overall, the project outcomes are positive and very beneficial for SIL4 Cloud target picture. For a very good and productive collaboration, the SIL4 Cloud collaboration team would like to thank every person involved during the project.

The project has touched and addressed several dimensions of SIL4 Cloud platform which is planned to be setup for future rail applications at DB. Through this project, involved partners were able to discuss future computing platform requirements coming from safety-critical functional applications and its hosting in a private cloud infrastructure. Involved industry partners from different domains have brainstormed, analysed and provided feedback on requirements and proposed novel architecture to be implemented and tested in the near future.

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.

To standardise any platform approach, safety analysis is crucial, and it was carried out in this project to some extent. At the same time, there was also a need to understand the functional applications in a more detailed way. Hence, it was concluded that going forward, a concrete functional application as an example would be required. Since CCS systems are categorised as critical infrastructure (KRITIS), cyber security analysis was carried out based on IEC 62443 and it is believed that these identified requirements and recommendation on cyber security could be quite useful, which could be further validated in SIL4 Cloud PoCs. Communication protocol topic was also one of the key topics in the project, which is very much driven by standardised interfaces involving multiple stakeholders. As it is targeted towards CCS domain, the topic also brings challenges such as timing-constraints and inherent support for safety towards Functional applications. Technical design and platform architecture for SIL4 Cloud were discussed based on two platform solutions (i.e., TAS Platform and PikeOS) with quite helpful insight and recommendations from partners.

Homologation topic was also covered, and key challenges were identified along with possible ways to handle them e.g., to handle the contradiction of dynamic configuration of system with the safety standard, it is important to show that the developers were aware of all possible failures that may rise from it, and that effective measures have been applied. This can be done by identifying all risks or failures that are introduced by dynamic behaviour of the system.

Moving forward, platform architecture can be refined further, and the involved partners could address open-questions with help of some concrete proof-of-concept (PoC) projects. These PoC projects could help to analyse and compare different technologies towards fulfilment of non-functional requirements such as 1) Flexibility 2) Scalability 3) Maintainability and 4) Availability. A real platform design can be realised based on this research and particular example application which could later be analysed for integration aspects. For better comparison two validation example applications can be chosen.

As stated in the beginning, participated partners in the research have different approaches however have combined view and support to the vision of SIL4 Cloud which is an important supporting block for digital rail. Overall, there is very positive feedback on the architecture, and it can be realised by addressing open points sooner than later. With these points, we are looking forward for next steps in realisation and implementation of SIL4 Cloud for railways.

14 References

[1] Digitale Schiene Deutschland, [Online]. Available: https://digitale-schiene-deutschland.de/en.

[2] RCA/ OCORA, “Safe Computing Platform White Paper,” Jun 2021. [Online]. Available: https://github.com/OCORA-Public/Publication/blob/master/06_OCORA%20R2/OCORA-TWS03-010_Computing-Platform-Whitepaper.pdf. [Accessed Sep 2022].

[3] “RCA,” [Online]. Available: https://www.eulynx.eu/index.php/news .

[4] “EUG,” [Online]. Available: https://www.eulynx.eu.

[5] “EULYNX,” [Online]. Available: https://www.eulynx.eu.

[6] “OCORA,” [Online]. Available: https://github.com/OCORA-Public/Publication .

[7] “Carbon emission figures,” [Online]. Available: https://bahnindustrie.info/de/themen/gesamt/detail/digitale-schiene-deutschland.

[8] D. Schneider, “Conditional safety certification for open adaptive systems (Dissertation),” 2014. [Online]. Available: http://publica.fraunhofer.de/dokumente/N-283653.html.

[9] H. Kantz, S. Resch and C. Scherrer, “Communication in train control,” in *Industrial Communication Technology Handbook*, CRC Press, 2017, pp. 64-1.

[10] CENELEC, *EN50159: Railway applications - Communication, signalling and processing systems - Safety-related communication in transmission systems,* 2010.

[11] OPC Foundation, “OPC UA Specification Part 15 - Safety,” Oct. 31, 2019.

[12] OPC Foundation, “OPC UA for Field Level Communications - A Theory of Operation,” Nov. 2020.

[13] D. H. Woo and H.-H. S. Lee, “Analyzing Performance Vulnerability due to Resource Denial-of-Service Attack on Chip Multiprocessors,” [Online]. Available: https://www.eecg.utoronto.ca/~moshovos/CMPMSI07/DongHyukWoo-DoS.pdf.

[14] Intel, “Introduction to Cache Allocation Technology in the Intel® Xeon® Processor E5 v4 Family,” [Online]. Available: https://www.intel.com/content/www/us/en/developer/articles/technical/introduction-to-cache-allocation-technology.html?wapkw=cat.

[15] “Spectre Attacks: Exploiting Speculative Execution,” [Online]. Available: https://spectreattack.com/spectre.pdf.

[16] A. D. Sinnhofer, W. Raschke, C. Steger and et al., “Evaluation paradigm selection according to Common Criteria for an incremental product development,” in *International Workshop on MILS: Architecture and Assurance for Secure Systems*, 2015.

[17] S. Dupont, G. Ginis, M. Malacario and et al., “Incremental Common Criteria Certification Processes using DevSecOps Practices,” IEEE, 2021. [Online]. Available: https://www.computer.org/csdl/proceedings-article/euros&pw/2021/999900a012/1y63lsv7pf2.

[18] R. Hametner, P. Tummeltshammer, S. Resch and W. Wernhart, “Cloud architecture for SIL4 railway applications,” *SIGNAL+DRAHT,* no. 3, 2022.

[19] “SAFety and secURity by dEsign for interconnected mixed-critical cyber-physical systems D3.3: Integrity Methodology,” [Online]. Available: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5bb482b4f&appId=PPGMS .

[20] H. Blasum, M. Brotz, F. Golatowski and et al., “Security Certification of Cyber Physical Systems for Critical Infrastructure based on the Compositional MILS Architecture,” Presented at the 47th IECON, 2021.

[21] RCA/ OCORA, [Online]. Available: https://github.com/OCORA-Public/Publication/blob/master/06_OCORA%20R2/OCORA-TWS03-030_SCP_Specification_of_the_PI_API_between_Application_and_Platform.pdf.

[22] [Online]. Available: https://cwe.mitre.org/.

[23] M. Heinrich, J. Vieten, T. Arul and S. Katzenbeisser, “Security Analysis of the RaSTA Safety Protocol,” *IEEE International Conference on Intelligence and Security Informatics,* pp. 199-204, 2018.

[24] P. Drahos, E. Kucera, O. Haffner and I. Klimo, “Trends in industrial communication and OPC UA,” *Cybernetics & Informatics (K&I), Lazy pod Makytou,* pp. 1-5, 01 2018.

[25] CENELEC, *DIN EN 50128: Railway applications - Telecommunications technology, signaling technology and data processing systems - Software for railway control and monitoring systems.*

[26] CENELEC, *DIN EN 50126-1: Railway applications - Specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 1: Generic RAMS process.*

[27] CENELEC, *DIN EN 50126-2: Railway applications - Specification and verification of reliability, availability, maintainability and safety (RAMS) - Part 2: System related safety methodology.*

[28] CENELEC, *DIN EN 50129: Telecommunications technology, signaling technology and data processing systems - safety-related electronic systems for signaling technology.*

[29] X2RAIL-1 D8.7, “X2Rail-1 Deliverable D8.7 “Application of the Security Assessment,” 2021.

[30] X2RAIL, “X2Rail-3 Deliverable D8.2-2 “Generic cybersecurity architecture and shared security services”,” 2021.

[31] X2RAIL, “X2Rail-3 Deliverable D8.2-3b “Protection Profile - Trackside components”,” 2021.

[32] G. Gala, S. Resch and G. Fohler, “RT-Cloud: Virtualization Technologies and Cloud Computing for Railway Use-Case,” in *IEEE 24th International Symposium on Real-Time Distributed Computing (ISORC)*, 2021.

[33] TüvSüd, “ZERTIFIKATSDATENBANK,” [Online]. Available: https://www.tuvsud.com/de-de/dienstleistungen/produktpruefung-und-produktzertifizierung/zertifikatsdatenbank.

[34] B. Rother, F. Golatowski, Z. Ansar and et al., “Analysis of Safety-Critical Communication Protocols for On-Premise SIL4 Cloud in Railways,” in *4th International Conference Reliability, Safety, and Security of Railway Systems (RSSRail 2022)*, Paris, France,, Juni 2022.

15 Terms

ARINC	Aeronautical Radio Incorporated
AUTOSAR	AUTomotive Open System ARchitecture
BKP	Backup
CCS	Command and Control Systems
CENELEC	Comité Européen de Normalisation Électrotechnique
COTS	Commercially-off-the-shelf
CPU	Central processing unit
CTMS	Capacity and Traffic Management System
DoS	Denial of Service
E2E	End to end
EAL	Evaluation Assurance Level
ETCS	European Train Control System
EULYNX	EULYNX is an European initiative by 13 Infrastructure Managers to standardise interfaces and elements of the signalling systems
GoA4	Grade of Automation 4
I/O	Input and Output
IAM	Identity and access management
IDE	Integrated development environment
IDS	Intrusion detection/continuous security monitoring
IXL	Interlocking
KPI	Key performance indicator
KRITIS	Critical Infrastructures
LOG	LOGging service in X2RAIL
MIB	Management Information Base
MooN	M-out-of-N
MTBF	Mean Time Between Failures
OCORA	Open CCS Onboard Reference Architecture
OPC UA	Open Platform Communications – Unified Architecture
PI API	Platform Independent Application Programming Interface
PKI	Public Key Infrastructure
POSIX	Portable Operating System Interface
RaSTA	Rail Safe Transport Application
RBC	Radio Block Centre
RCA	Reference CCS Architecture
RTE	Runtime Environment

RTOS	Real-time operating systems
SAHARA	Safe, Highly Available and Redundant
SCI	Standard Communication Interface
SIEM	Security Information and Event Management
SIL	Safety Integrity Level
SL-T	Target security level
SNMP	Simple Network Management Protocol
SRAC	Safety -related application conditions (as per EN 50129: 2018)
SW	Software
VM	Virtual machine

16 Annex A

RCA/OCORA High-level Objectives

No.	Objective	Applicability to	
		Onboard	Trackside
1	**Meet safety and real-time requirements of CCS (and similar) railway applications**. The platform shall meet safety requirements of applications up to safety and integrity level (SIL) 4, e.g., acc. to EN 50126, EN 50128 and EN 50129, and support applications with real-time characteristics (e.g., overall processing cycles in the order of 10-100ms).	✓	✓
2	**Respect diverse lifecycles of business logic, runtime environment and hardware**. The platform shall be partitioned with respect to the different lifecycles of business logic, runtime environment and hardware. The platform shall support fully independent life-cycle handling, i.e., with minimal dependencies.	✓	✓
3	**Open market to new players**. The platform shall open the market to new, non-rail-oriented software and tooling companies. They shall be able to become involved in functional application development without providing their own platform safety mechanisms (e.g., related to safe communication, fault tolerance implementation, etc.).	✓	✓
4	**Minimise total cost of ownership**. The platform shall minimise the total cost of ownership, i.e., the overall life-cycle cost.	✓	✓
5	**Vendor independence**. Different vendors shall be able to provide functional applications, computing platforms and development tools, respectively, without a vendor lock-in. It shall be possible to purchase hardware directly from different vendors throughout the lifespan of the software. The platform shall build on existing HW/SW solutions, stimulating competition among vendors and allowing them to shine with their specific expertise and distinctive solution features.	(✓) * with possible limitations in HW choice, at least short-term, due to highly integrated solutions	✓
6	**Industrial readiness**. It shall be possible to procure a platform as off-the-shelf solution supported by an open and dynamic market. The solutions shall be mature (e.g., reliability proven in field) and backed by effective acceptance and integrated logistical support (e.g., maintenance service, tooling, availability of spare parts).	✓	✓
7	**Migratable and portable business logic**. The business logic is considered a significant system asset, being the component with the longest lifetime. It must hence be portable to different computing platform evolutions. We here further differentiate: • **Migratability for legacy applications**: It should be decently easy to migrate legacy applications to the new platform; • **Portability for new applications**: Applications running on the platform should be portable to any other vendor's or evolved version of the platform.	✓	✓
8	**System evolvability**. The platform shall be open to extensions (in the sense of additional system services that are added over time, e.g., re-	✓	✓

No.	Objective	Applicability to	
		Onboard	Trackside
	lated to FRMCS). Adding new functionalities shall be possible with minimal to no changes to existing applications (though these may naturally not be able to leverage the new functionalities).		
9	**Facilitation of application development**. The platform shall use an open, well-documented application model and programming interface, facilitating that third parties develop applications.	✓	✓
10	**Modularity**. The platform shall allow for a modular safety certification process, using pre-certified components leading to a dramatically simplified and shortened full system certification process. An evolution or update of the platform shall not require a new E2E homologation of application and platform, as detailed in Section 8 of the paper	✓	✓
11	**Encapsulated, transparent fault tolerance mechanism**. The platform shall transparently encapsulate the safety and fault tolerance mechanisms. Vendors may offer different (new) approaches to safety and fault tolerance as they become available on the market - solution agnostic and future-proof.	✓	✓
12	**Scalability**. The platform shall be highly scalable, i.e., it should by design be able to support an arbitrary number of applications and arbitrary number of compute nodes.	(✓)	✓
13	**Flexible usage of compute resources.** The platform shall enable a flexible mapping of business logic to compute resources (e.g., such that the platform can be expanded while applications are running, and that business logic can be re-mapped when compute nodes fail). It shall be able to leverage advances in computing technology (i.e., when better compute nodes are available, it shall be possible to assign more instances of business logic to the compute nodes).	✓	✓
14	**Centralisation.** The platform shall allow to centralise applications physically in a safe data centre to simplify life-cycle management, reduce TCO by means of simplified, optimised operations, and benefit from increased availability and optimised resource usage.		✓
15	**Support for running multiple applications (also with different SIL levels) on one physical platform.** It shall be possible to run multiple applications, possibly with different SIL levels, on a single physical platform to reduce cost, space, power dissipation, etc., and simplify certification, maintenance, system evolution, etc.	✓	✓ (though need less pronounced as for onboard)
16	**Life-cycle management capabilities.** The platform shall provide automated mechanisms related to software lifecycle and configuration management, diagnostics, etc. This may also be expanded to software development automation, e.g., taking orientation in SPEM.	✓	✓

-------END OF THE REPORT-------