Virtualisierte Safety-Architekturen in der Leit- und Sicherungstechnik

The virtualisation of safety architectures in Control Command and Signalling

Zeeshan Ansar | Julian Wissmann

Virtualisierung, also die Entkopplung von Software und Hardware, verspricht geringere Kosten und einen schnellen Rollout. Die Virtualisierung Safety-relevanter LST-Funktionen (Leit- und Sicherungstechnik, LST) verlagert die Safety-Argumentation von physischer Trennung zu Plattformeigenschaften wie Isolation und Determinismus. Betrachtet werden zwei Architekturen: Safety-zertifizierte Virtualisierung mit Isolationsgarantien und COTS-Virtualisierung, ergänzt durch softwarebasierte Diagnose, Fehlererkennung und Native Hardware Access (NHA). Der Beitrag vergleicht beide, stellt Vor- und Nachteile dar und gibt Empfehlungen.

1 Einleitung

Die Entwicklung der LST im Eisenbahnsystem hin zu zentralisierten, IP-basierten Lösungen eröffnet die Möglichkeit, neue IT-Plattformarchitekturen basierend auf handelsüblichen Produkten (Commercial Off The Shelf – COTS) zu verwenden. Das ermöglicht eine Konsolidierung der Anwendungslandschaft und geht mit höherer Skalierbarkeit und größerer Flexibilität einher.
Kürzlich haben das Europe's Rail Joint Undertaking (ERJU) im Bereich „Computing Environment" [1] sowie das Projekt FP2 R2DATO [2] eine modulare Schichtenarchitektur mit definierten Schnittstellen entwi-

Virtualisation, the decoupling of software and hardware, promises reduced costs and faster rollouts. The virtualisation of safety-relevant Control Command and Signalling (CCS) functions shifts the safety argument away from physical separation to platform properties such as isolation and determinism. Two architectures are considered here: safety-certified virtualisation with isolation guarantees and COTS virtualisation, supplemented with software-based diagnostics, fault detection and native hardware access (NHA). This article compares both, presenting their advantages and disadvantages, and provides recommendations.

1 Introduction

The development of CCS in the railway system towards centralised, IP-based solutions has opened up the possibility of using new IT platform architectures based on commercial off-the-shelf (COTS) products. This enables the consolidation of the application landscape and comes with higher scalability and greater flexibility.
Recently, the "Computing Environment" domain [1] of Europe's Rail Joint Undertaking (ERJU) and the FP2 R2DATO project [2]

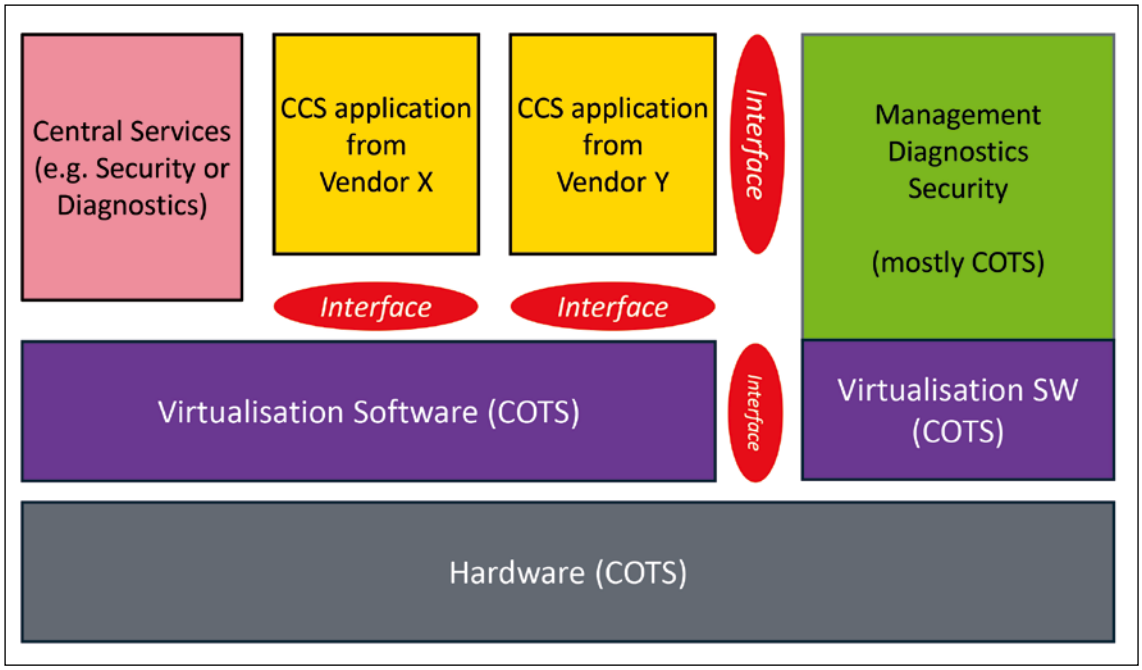

Bild 1: High-Level-Darstellung der geplanten Trennung von LST-Anwendungen und IT-Plattform über standardisierte Schnittstellen
Fig. 1: A high level depiction of the planned separation of CCS applications and the IT platform using standardised interfaces

ckelt, um LST-Funktionen von der zugrunde liegenden IT-Plattform zu entkoppeln. Eine solche auf Virtualisierung basierende LST-Architektur, wie sie in Bild 1 dargestellt ist, verspricht verringerte Investitions- und Betriebskosten, eine Vereinfachung der Umsetzung zentraler Dienste (z. B. IT-Security bezogene Dienste) sowie eine Verbesserung der Skalierbarkeit und Wartbarkeit des Systems.

Die dargestellte Architektur wird im Förderprojekt Cloud4Rail [3] im Kontext des europäischen Programmes IPCEI-CIS [4] erstmalig umgesetzt und erprobt und entspricht dem perspektivischen Zielbild der DB InfraGO für eine auf einer On-Premises-Cloud, d. h. einer im eigenen Rechenzentrum betrieben Cloud-Infrastruktur, basierenden LST [5].

Bei der Entkopplung von LST-Funktionen und IT-Plattform stellt sich die grundsätzliche Frage, wo die Safety in Bezug auf potenzielle Ausfälle von Hardware oder Software sowie für potenzielle Wechselwirkungen zwischen Anwendungen, die in der gleichen Virtualisierungsumgebung laufen, gewährleistet ist. In dieser Hinsicht sind zwei allgemeine Safety-Architekturen denkbar, die in Bild 2 dargestellt sind:

- Safety-Architektur A: Der Virtualisierungsschicht kann vertraut werden. Sie ist sicherheitszertifiziert und bietet Mechanismen zur Erkennung von und der Reaktion auf Hardwareausfälle, sodass Garantien für Isolation und Determinismus für die oben genannten LST-Anwendungen angeboten werden können.
- Safety-Architektur B: Die Virtualisierungsschicht ist nicht sicherheitszertifiziert. Daher sind Sicherheitsmechanismen innerhalb der LST-Anwendung (oder ein spezieller Safety-Layer oberhalb der Virtualisierung) erforderlich, um Ausfälle innerhalb der Hardware und der Virtualisierungssoftware zu erkennen und darauf zu reagieren.

Bei ERJU geht man im Allgemeinen davon aus, dass Safety-Architektur B verfolgt wird – unter anderem auch deshalb, weil die meisten LST-Anbieter bereits Safety-Layer in ihrem Produktportfolio haben, die mit nicht sicherer Hardware darunter umgehen können. Einige Anbieter im Bereich der IT- und OT-Lösungen schlagen jedoch auch Implementierungen vor, die der Safety-Architektur A entsprechen. Da die angestrebte Integration von LST-Anwendungen verschiedener Anbieter auf einer gemeinsamen Hardware- und Virtualisierungsplattform die Wahl einer gemeinsamen Safety-Architektur erfordert und es sich hierbei um eine grundlegende Entscheidung für die Zukunft der IT-Landschaft im Bahnbetrieb handelt, hat die DB InfraGO einen eingehenden Vergleich dieser beiden Safety-Architekturen durchgeführt.

have developed a modular layered architecture with defined interfaces to decouple CCS functions from the underlying IT platform. Such a virtualisation-based CCS architecture, as shown in fig. 1, promises reduced investment and operating costs, the simplified implementation of central services (e.g. IT security-related services), as well as the improved scalability and maintainability of the system.

The presented architecture is being implemented and tested for the first time in the Cloud4Rail project [3] within the context of the European IPCEI-CIS program [4] and corresponds to DB InfraGO's vision for a CCS based on an on-premises cloud, i.e. a cloud infrastructure operated at its own data centre [5].

When the CCS functions and IT platform are decoupled, a fundamental question arises as to where the safety is guaranteed with reference to any potential hardware or software failures, not to mention any potential interference among the applications running in the same virtualisation environment. Two general safety architectures are conceivable in this regard, as depicted in fig. 2:

- Safety Architecture A: the virtualisation layer can be trusted. It is safety-certified and provides mechanisms to detect and react to any hardware failures, thereby offering guarantees for isolation and determinism for the aforementioned CCS applications.
- Safety Architecture B: the virtualisation layer is not safety-certified. Therefore, safety mechanisms are required within the CCS application (or a dedicated safety layer above the virtualisation) to detect and react to any failures within the hardware and virtualisation software.

ERJU generally assumes that Safety Architecture B will be pursued, partly because most CCS suppliers already have safety layers in their product portfolios that can handle non-safe hardware underneath them. However, some suppliers in the IT and OT solutions field also propose implementations corresponding to Safety Architecture A. Given that the intended integration of CCS applications from different suppliers on a common hardware and virtualisation platform requires the choice of a common safety architecture and as this is a fundamental decision for the future of the IT landscape in rail operations, DB InfraGO

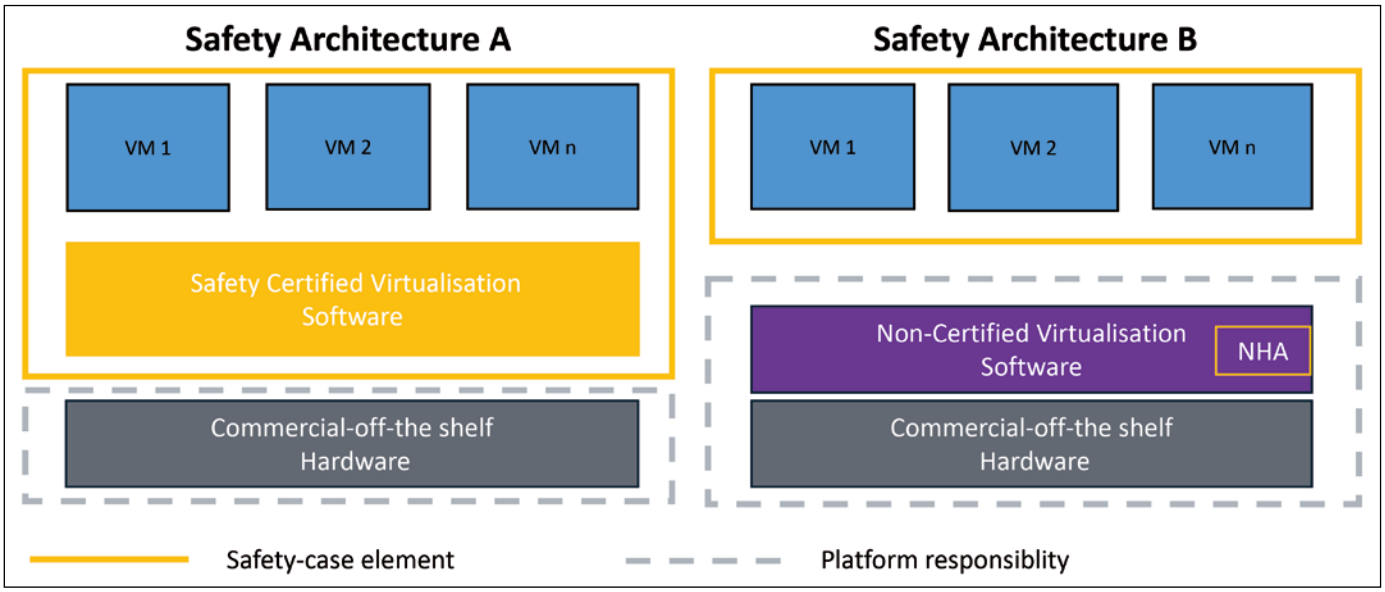

Bild 2: Übersicht der verglichenen Safety-Architekturen
Fig. 2: An overview of the compared safety architectures

In diesem Beitrag werden die beiden genannten Safety-Architekturen speziell im Hinblick auf die Eigenschaften Zuverlässigkeit, Verfügbarkeit, Wartbarkeit und Sicherheit (Reliability, Availability, Maintainability, Safety and Security – RAMSS) diskutiert und die sich daraus ergebenden Konsequenzen für den technischen Aufwand und die Zertifizierung in einen Kontext gestellt.

2 Architektonische Grundannahmen

Unabhängig von der gewählten Safety-Architektur wird davon ausgegangen, dass die folgenden Punkte zutreffen:

- Es wird COTS-Hardware verwendet, z. B. Multi-Core-CPU, optional ECC-Speicher zur Fehlerkorrektur sowie standardisierte Ein / Ausgabe-Systeme.
- Ein Hypervisor läuft direkt auf der Hardware (Bare-Metal-System) und stellt eine Hardware-Abstraktion für Gastbetriebssysteme und Anwendungen bereit, die in virtuellen Maschinen (VM) oder gleichwertigen, streng voneinander isolierten Ausführungsdomänen ausgeführt werden.
- Sicherheitskritische LST-Funktionen können mit nicht sicherheitsrelevanten Funktionen (z. B. Diagnosedienste, Wartungsdienste, Überwachung, Cybersecurity-Tools) koexistieren, wodurch eine gemischt-kritische Umgebung entsteht.
- RAMSS und Cybersicherheit werden als inhärent für den Entwurf betrachtet: Die Wahl der Plattform beeinflusst die Verfügbarkeit, die Wartbarkeit und die Security-Controls und wirkt sich daher auf den Safety-Case aus.

Im Folgenden werden die beiden verglichenen Safety-Architekturen detailliert beschrieben.

2.1 Safety-Architektur A: Sicherheitszertifizierte Virtualisierung (vertrauenswürdige Plattform)

Ansatz A verwendet vertrauenswürdige Virtualisierungssoftware, die für sicherheitskritische Systeme vorzertifiziert ist. Bei diesem architektonischen Ansatz ist Sicherheit ein inhärentes Prinzip, wobei die Verantwortung für die Sicherheit dezentralisiert und auf mehrere Komponenten verteilt ist, die eng integriert sind, um Sicherheitsmechanismen auf verschiedenen Ebenen zu implementieren.

Die sicherheitszertifizierte Virtualisierungssoftware wird in der Regel nach dem Kernprinzip eines Separation-Kernels entworfen, um die Ziele Minimalismus, Kontrolle und starke Isolierung zu gewährleisten. Das Separation-Kernel-Prinzip ermöglicht nicht nur die Trennung der Safety und Security von den Anwendungen, sondern auch eine domänenspezifische formale Verifizierung [6].

Die Konstruktionsprinzipien der sicherheitszertifizierten Virtualisierung gewährleisten eine garantierte räumliche und zeitliche Isolierung sowie integrierte Sicherheitsmechanismen auf Plattformebene. Zu den wichtigsten Merkmalen gehören:

- Garantierte räumliche und zeitliche Trennung: zertifizierte Mechanismen zur Störungsminderung wie formale Speichertrennung, Konfigurations- / Modellverifizierung, statische Ressourcenaussteuerung, begrenzte Ausführungszeiten und festes zyklisches Scheduling.
- Starke Fehlereindämmung: Ausfälle können sich aufgrund der erzwungenen Trennung nicht auf andere Partitionen ausbreiten.
- Eingebaute Plattformsicherheitsdienste: Zustandsüberwachung, Watchdog-Überwachung usw.

Der Entwicklungs- und Verifizierungsprozess für sicherheitszertifizierte Virtualisierungssoftware entspricht in der Regel der EN 50128 [7] oder ihrem Nachfolger EN 50716, einschließlich der Entwurfs-, Implementierungs- und Verifizierungsphasen. Die Softwarepakete enthalten eine qualifizierte Toolchain für die Konfiguration, Compiler und Debugger sowie eine Strukturabdeckungsanalyse, d. h. den Nachweis, welcher

has conducted an in-depth comparison of these two safety architectures.

This paper discusses both safety architectures specifically with regard to their reliability, availability, maintainability, safety and security (RAMSS) properties and places the resulting trade-offs pertaining to engineering effort and certification in context.

2 The architectural baseline and assumptions

The following points are assumed to apply regardless of the chosen safety architecture:

- COTS hardware is used, e.g. multi-core CPUs, optionally with Error Correction Code (ECC) memory for error correction and standardised input / output systems.
- A hypervisor runs directly on the hardware (a bare-metal system) and provides hardware abstraction for any guest operating systems and applications executed in virtual machines (VM) or equivalent, strictly isolated execution domains.
- Safety-critical CCS functions can coexist with non-safety functions (e.g. diagnostics, maintenance services, monitoring and cybersecurity tools), thus creating a mixed-criticality environment.
- RAMSS and cybersecurity are considered to be inherent to the design: the platform choices influence availability, maintainability and security controls and thus impact the safety case.

The two compared safety architectures are described in detail in the following.

2.1 Safety Architecture A: safety-certified virtualisation (a trusted platform)

Approach A uses trusted virtualisation software pre-certified for safety-critical systems. Safety is an inherent principle in this architectural approach, with the safety responsibility decentralised and distributed across multiple components that have been closely integrated to implement safety mechanisms on different layers.

The safety-certified virtualisation software is usually designed on the core principle of a separation kernel to ensure the objectives of minimalism, control and strong isolation. The separation kernel principle not only allows the separation of safety and security from the applications, but also enables domain specific formal verification to be feasible [5].

The design principles of safety-certified virtualisation ensure guaranteed spatial and temporal isolation, as well as integrated safety mechanisms at the platform level. Key features include:

- guaranteed spatial and temporal separation: certified interference-mitigation mechanisms such as formal memory separation, configuration / model verification, static resource allocation, bounded execution times and fixed cyclic scheduling.
- strong fault containment: failures cannot propagate to other partitions due to enforced separation.
- built-in platform safety services: health monitoring, watchdog supervision etc.

The development and verification process for safety-certified virtualisation software typically complies with EN 50128 [7] or its successor EN 50716, including the design, implementation and verification phases. The software packages include a qualified toolchain for configuration, compilers and debuggers, as well as structural coverage analysis, i.e. proof of which portion of the code and branches has been executed through tests. From the perspective of trackside migration, these techniques enable comprehensive requirement tracing and reduce the overall ef-

Anteil des Codes und der Verzweigungen durch Tests ausgeführt wird. Aus Sicht der streckenseitigen Migration ermöglichen diese Techniken eine umfassende Verfolgung der Anforderungen und verringern den Gesamtaufwand für die Zertifizierung von Sicherheitssystemen, wie im Automobilsektor nachgewiesen wurde [8].

Konsequenzen: Plattformkonfiguration und Änderungsmanagement werden sicherheitsrelevant. Aktualisierungen des Hypervisors, der VM-Konfiguration oder von Low-Level-Treibern erfordern in der Regel strenge Kontrollen und Regressionsnachweise. Darüber hinaus sind Anwendungen von Sicherheitsdiensten der Plattform abhängig, die spezifisches Verhalten von Anwendungen erzwingen können und möglicherweise ein Redesign bestehender Anwendungen erfordern.

2.2 Safety-Architektur B: Nicht zertifizierte Virtualisierung (nicht vertrauenswürdige Plattform) mit zentraler Safety-Umgebung

Die Safety-Architektur B verwendet nicht zertifizierte, universelle Virtualisierungssoftware, um eine hardwareunabhängige, parallele Ausführung von virtuellen Gastmaschinen zu ermöglichen. Bei dieser Architektur werden Sicherheitsmechanismen zentralisiert und durch eine einzige Komponente, die herstellerspezifische Safety-Umgebung, implementiert, die die funktional sichere Ausführung einer konkreten LST-Anwendung im Hinblick auf die IT-Plattform gewährleistet.

Bei marktüblichen Virtualisierungslösungen haben Leistung, Kosten und Skalierbarkeit in der Regel Vorrang vor deterministischem Verhalten und formal abgesicherter Isolierung. Daher gilt für diese Lösungen:

- Sie garantieren keine räumliche und zeitliche Isolierung.
- Sie bieten keine deterministischen Ausführungseigenschaften.
- Es existieren keine Zertifizierungsartefakte.

Das Fehlen dieser Funktionen kann zu einer schwächeren Fehlereindämmung als bei Safety-Architektur A führen und die Ausbreitung von Fehlern auf andere virtuelle Maschinen ermöglichen. Obwohl der fehlende, inhärente Architekturvorteil der zertifizierten Virtualisierung die Komplexität des Sicherheitsnachweises erhöht, ist die Verwendung einer nicht vertrauenswürdigen Virtualisierung in sicherheitskritischen Systemen durchaus eine Option, wenn der Safety-Layer oberhalb der Virtualisierung die Sicherheit des Gesamtsystems durch Mechanismen wie Composite Fail Sicherheit sowie Fehler- und Störungserkennung gewährleistet.

Eine wesentliche Voraussetzung für die Gewährleistung der Sicherheit oberhalb der Virtualisierungsschicht ist eine zusätzliche Komponente, die in ERJU [1, 2] als Native Hardware Access (NHA) bezeichnet wird. NHA ist eine nahe an der Virtualisierungsschicht implementierte Softwarekomponente, die sicherheitsrelevante Hardwareparameter kontrolliert und direkt an den darüberliegenden Safety-Layer liefert. Damit kompensiert NHA die fehlende Vertrauenswürdigkeit der Virtualisierungsschicht, verhindert unzulässige Zugriffe und ermöglicht den direkten, kontrollierten Zugriff auf ausgewählte kritische Hardwareparameter. Aufgrund dieser sicherheitsrelevanten Funktionalität ist NHA im Safety-Case zu berücksichtigen.

Konsequenzen: Nicht zertifizierte Virtualisierung kann Vorteile bieten wie eine größere Produkt- und Anbietervielfalt, einschließlich der Möglichkeit, etablierte Open-Source-Hypervisoren aus dem IT-Markt einzusetzen. Zudem bietet diese mehr Flexibilität, potenziell niedrigere Lizenz- und Betriebskosten sowie geringere Wartungsaufwände. Da die Sicherheitsargumentation im zentralen Safety-Layer und nicht in der Virtualisierungssoftware verankert ist, kann die zugrunde liegende Virtualisierungsplattform grundsätzlich mit geringerem sicherheitsrelevantem Anpassungsaufwand ausgetauscht werden, was die Herstellerbindung auf der Virtualisierungsebene reduziert. Diese Vorteile müssen jedoch gegen einen höheren Validierungs- und Verifizierungsaufwand auf der Anwendungs-

ebene abgewogen werden, da fehlende Plattformgarantien durch Architektur-, Test- und Betriebsbeschränkungen kompensiert werden müssen. Dies sollte jedoch für bestehende Anwendungen vernachlässigbar sein, da die Bündelung eines Safety-Layer mit der Anwendung heute bereits in LST-Produkten umgesetzt ist. Im Gegenzug kann die Safety-Architektur A für neue Anbieter von Anwendungen, die in den Bahnmarkt eintreten wollen, hilfreich sein, da sie keine komplexen zusammengesetzten Ausfallsicherheitsmechanismen von Grund auf implementieren müssen.

3 Vergleich

Die Safety-Architektur A nutzt vorzertifizierte Virtualisierungssoftware und bietet verteilte, mehrschichtige Safety-Mechanismen, die

fort required for the certification of safety systems, as demonstrated in the automotive sector [8].

Implications: platform configuration and change management become safety-relevant. Any updates to the hypervisor, VM configurations or low-level drivers typically require strict controls and regression evidence. Additionally, applications may also depend on platform safety services, which can induce specific application behaviours and may require the redesign of existing applications.

2.2 Safety Architecture B: non-certified virtualisation (an untrusted platform) with a central safety environment

Safety Architecture B uses non-certified, general-purpose virtualisation software to enable the hardware-independent, parallel execution of guest virtual machines. In this architecture, the safe-

Aspekt	Safety-Architektur A: Sicherheitzertifizierte Virtualisierung (vertrauenswürdige Plattform)	Safety-Architektur B: nicht zertifizierte Virtualisierung (nicht vertrauenswürdige Plattform) mit herstellerspezifischer Safety-Umgebung
Safety case	Dezentralisierte, verteilte Komponenten	Zentralisierte, einzelne Komponente
Garantierte Isolierung	Formale Trennung, Modellverifizierung und statische Ressourcenausweisung	Schwächere Isolierung, gemeinsame MMU-Verwaltung (Memory Management Unit), dynamische Ressourcenaussteuerung und keine formale Verifizierung
Deterministisches Verhalten	Begrenzte WCET (Worst Case Execution Time) und Timing möglich (z. B. zyklische Zeitpläne / begrenzte Latenzen)	Best-Effort-Verhalten möglich; unbegrenzte Latenzen müssen über Beschränkungen und Validierung kontrolliert werden
Zertifizierungsnachweis	Vorhandene Artefakte / Certification Support Packages; Wiederverwenden möglich	Keine Vorzertifizierung; projektspezifische Nachweiserstellung für COTS-Teile erforderlich
Produktverfügbarkeit	Produktanpassung im Hinblick auf spezifische Sicherheitsanforderungen erforderlich; bestehende Anwendungen müssen stark umgestaltet werden	Vorhandene COTS-Plattformen können verwendet werden; NHA-Implementierung erforderlich; bestehende Anwendungen können mit geringem Aufwand portiert werden
Hauptrisiken	Anbieterabhängigkeit / Lock-in; Plattformänderungen sind sicherheitsrelevant	Erhöhte Komplexität der Safety-Umgebung; potenzieller Single Point of Failure, wenn nicht redundant ausgelegt
Wartung	Stabil, wenn die Plattform unter eigener Kontrolle steht; Änderungen erfordern strengere Prozesse und spezialisierte Expertenressourcen	Flexibles Ökosystem; erfordert jedoch disziplinierte Konfiguration, Tests und kontrollierte Änderungen, um Regressionen zu vermeiden
Re-Autorisierungsaufwand	Höher, da die Re-Autorisierung von mehrschichtigen Komponenten und der Beteiligung mehrerer Anbieter abhängt und einen höheren Re-Integrationsaufwand erfordert	Einfach, da der Re-Autorisierungsaufwand von einer einzigen Komponente und einem einzigen Anbieter abhängt und weniger Re-Integrationsaufwand erfordert

Tab. 1: Vergleich der beschriebenen Safety-Architekturen

Aspect	Safety Architecture A: safety-certified virtualisation (trusted platform)	Safety Architecture B: non-certified virtualisation (untrusted platform) with vendor-specific safety environment
Safety case	De-centralised, distributed components	Centralised, single component
Guaranteed Isolation	Formal separation, model verification and static resource allocation	Weaker isolation, shared MMU (memory management unit) management, dynamic resource allocation and no formal verification
Deterministic Behaviour	Bounded WCET and timing feasible (e.g., cyclic schedules / bounded latencies)	Best-effort behaviour possible; unbounded latencies must be controlled via constraints and validation
Certification evidence	Pre-existing artefacts / certification support packages; reuse potential	No pre-certification; project-specific evidence creation required for COTS parts
Product availability	Product customisation required regarding specific safety requirements; Existing applications will require major redesign	Pre-existing COTS platforms can be used; NHA implementation required; Existing applications can be ported with low effort
Key Risks	Vendor dependency / lock-in; platform changes safety-relevant	Increased complexity of safety environment; potential single point of failure if not made redundant
Maintenance	Stable if platform is controlled; changes require stricter processes and specialised expert resources	Flexible ecosystem; but requires disciplined configuration, testing and controlled change to avoid regressions
Re-authorisation effort	Higher, as re-authorisation depends on multi-layer components and multiple vendor involvement	Simpler, as re-authorisation effort depends on a single component and vendor, requiring less re-integration effort

Tab. 1: A comparison of the described safety architectures

eine solide Grundlage für die Safety-Umgebung und sicherheitskritische Anwendungen bilden.
Im Gegensatz dazu verwendet die Safety-Architektur B nicht zertifizierte Virtualisierungssoftware, die keine Sicherheitsgarantien bieten kann, weshalb alle Safety-Mechanismen von einem Safety-Layer unter Verwendung des NHA gehandhabt werden müssen.
Beide Safety-Architekturen haben deutliche Vor- und Nachteile, die sich auf die Gesamtkosten des Systems, den Integrationsaufwand und die Zertifizierung auswirken. Tab. 1 fasst die zentralen Aspekte beider Architekturen zusammen.

4 Fazit

In diesem Beitrag wurden zwei Safety-Architekturen für die Implementierung eines sicherheitskritischen Systems beschrieben, die virtualisierte LST-Funktionen bis Safety Integrity Level (SIL) 4 beherrschen können.
In Anbetracht bestehender LST-Anwendungen (z. B. digitale Stellwerke (DSTW) und ETCS-Funktionen) erscheint die Safety-Architektur B vorteilhaft, da sie den Redesign-Aufwand für Anwendungsmigrationen reduziert. Dies ist insbesondere für streckenseitige Systeme von Bedeutung. Daher plant DB InfraGO zusammen mit Partnern aus der Industrie, Architektur B im Rahmen künftiger Initiativen und Technologiepiloten weiter zu verfolgen, z. B. im Zusammenhang mit der Virtualisierung von Stellwerksfunktionen auf einer On-Premises-Plattform. Dabei werden die Stellwerksfunktionen nicht als eigenständige Hardware-Bauart, sondern als Software-Instanzen auf einer gemeinsamen Virtualisierungsplattform ausgeführt, wie sie perspektivisch auch für DSTW-Lösungen genutzt werden kann.
Safety-Architektur A kann Vorteile für zugseitige Systeme und Anwendungen bieten, die strenge Echtzeitgarantien erfordern, wie z. B. TCMS (Train Control and Management System), das zentrale fahrzeugseitige Sicherheits- und Managementsystem. Sie kann auch für neuartige, sicherheitsrelevante Bahnanwendungen interessant sein, z. B. im Rahmen von Grade of Automation (GoA) 4, da sie dazu beitragen kann, die Markteintrittsbarriere für neue Player in der Anwendungsentwicklung zu senken.
Relevante Themen wie Isolation, deterministische Datenverteilung, Datenabgleich, Replikation und Voting sowie Ausfälle mit gemeinty mechanisms are centralised and implemented through a single component, a vendor-specific safety environment, that ensures the safe execution of a specific CCS application with regard to the IT platform.
Performance, cost and scalability typically take precedence over any deterministic behaviour and formally assured isolation in the case of market-standard virtualisation solutions. Therefore, these solutions:

- do not guarantee any spatial and temporal isolation.
- do not provide any deterministic execution properties.
- lack certification artefacts.

The absence of these features can result in weaker fault containment compared to Safety Architecture A and may allow any failures to propagate to other virtual machines. Even though the lack of the inherent architectural advantages of certified virtualisation increases the complexity of the safety case, the use of non-trusted virtualisation in safety-critical systems is indeed an option if the safety layer above the virtualisation ensures overall system safety using mechanisms such as composite fail-safety and fault and error detection.
A key prerequisite for ensuring safety using mechanisms above the virtualisation layer involves a concept referred to in ERJU [1, 2] as Native Hardware Access (NHA). The NHA is a software component implemented at or close to the virtualisation layer that controls any safety-relevant hardware parameters and provides them directly to the safety layer above. This compensates for the lack of trustworthiness in the virtualisation layer, prevents unauthorised access and enables controlled, direct access to selected critical hardware parameters. Due to its safety-relevant functionality, the NHA must be considered in the safety case.
Implications: non-certified virtualisation can offer advantages such as greater product and vendor diversity, including the option of using established open-source hypervisors from the IT market. It also provides more flexibility, potentially lower licensing and operating costs and reduced maintenance efforts. Since the safety argumentation is anchored in the central safety layer rather than in the virtualisation software, the underlying virtualisation platform can, in principle, be replaced with less safety-relevant adaptation effort, thereby reducing vendor lock-in at

samer Ursache werden auch im Projekt Cloud4Rail (C4R) [3, 4] im Rahmen des IPCEI-CIS-Programms untersucht [8], welches das Ziel verfolgt, eine erste Umsetzung einer Plattformarchitektur für sicherheitskritische Bahnanwendungen, basierend auf den Arbeiten in Europe's Rail [1, 2], vorzunehmen. In diesem Rahmen ist auch dieser Artikel entstanden. Die vollständigen C4R-Dokumente werden über das IPCEI-C4R-Portal der EU zur Verfügung gestellt (zum Zeitpunkt der Erstellung dieses Artikels noch nicht online). ■

AUTOREN | **AUTHORS**

Dr.-Ing. Zeeshan Ansar
Plattformarchitekt Technische Gesamtsystemarchitektur, Technologiegrundsätze und IT-Plattformen / *Platform Architect Overall Technical System Architecture, Technology Principles and IT Platforms*
DB InfraGO
Anschrift / *Address*: Adam-Riese-Straße 11-13, D-60327 Frankfurt am Main
E-Mail: zeeshan.z.ansar@deutschebahn.com

Julian Wissmann, B. Eng.
Plattformarchitekt Technische Gesamtsystemarchitektur, Technologiegrundsätze und IT-Plattformen / *Platform Architect Overall Technical System Architecture, Technology Principles and IT Platforms*
DB InfraGO
Anschrift / *Address*: EUREF-Campus 17, D-10829 Berlin
E-Mail: julian.wissmann@deutschebahn.com

LITERATUR | **LITERATURE**

[1] Thomas, M.; Sögütlü, B.; Marsch, P.; Guilhem, H.; Spindler, M.; Anowski, F.; Steffens, S.; Traeger, A.; Mottola, G. D.; Klapka, Š.: "SPT2-Computing Environment Deliverables. Recommendation on Interfaces to be standardized", Europe's Rail, 2024, abgerufen am 15.01.2026 18:00, https://rail-research.europa.eu/v1-release/
[2] ERJU Innovation Pillar FP2 R2DATO, Deliverable D26.3, "Final Modular Platform requirements, architecture and specification", 2025, abgerufen am 15.01.2026 18:00, https://rail-research.europa.eu/pages/fp2-r2dato/deliverables
[3] Marsch, P.; Heine, A.; Albert, W.: "Nächste Station Cloud: Wie Cloud4Rail den Plattform-Markt für Europas Bahnen mobilisiert", Eisenbahntechnische Rundschau 11/2024, abgerufen am 15.01.2026 18:00, https://digitale-schiene-deutschland.de/Downloads/2024-11-15_ETR_Naechste_Station_Cloud.pdf
[4] "IPCEI-CIS, Important Project of Common European Interest on Next Generation Cloud Infrastructure and Services", 8ra.com, abgerufen am 15.01.2026 18:00, https://www.8ra.com/ipcei-cis/
[5] Heine, A.; Marsch, P.; Rebstadt, J.: "Die Migration der Leit- und Sicherungstechnik in ein modernes IT- und Cloud-System", SIGNAL+DRAHT 11/2025
[6] Lozano, S.; Lugo, T.; Carretero, J.: "A Comprehensive Survey on the Use of Hypervisors in Sicherheit-Critical Systems", in IEEE Access, Band 11, S. 36244–36263, 2023, abgerufen am: 25.08.2025 und 15.01.2026 18:00, https://ieeexplore.ieee.org/document/10092745
[7] "CENELEC, EN 50128:2011 – Bahnanwendungen – Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme – Software für Eisenbahnsteuerungs- und Überwachungssysteme, Europäisches Komitee für elektrotechnische Normung", Brüssel, 2011
[8] Wagner, M.; Zabel, M.: "Cost-Benefit Analysis of Certified vs. Non-Certified Hypervisors in Automotive. Elektronik Automotive" 2021

the virtualisation level. However, these benefits must be weighed against the higher validation and verification efforts at the application level, as the missing platform guarantees must be compensated for by architectural, testing and operating constraints. This should be negligible for existing applications, as bundling a safety layer with the application is already a common architectural approach in today's CCS products. In return, Safety Architecture A can be helpful for new application suppliers wanting to enter the railway market, as they do not have to implement any complex composite fail-safety mechanisms from scratch.

3 The comparison

Safety Architecture A uses pre-certified virtualisation software and offers distributed, multi-layer safety mechanisms that provide a solid foundation for the safety environment and safety-critical applications.
In contrast, Safety Architecture B uses non-certified virtualisation software that cannot provide safety guarantees, so all the safety mechanisms must be handled by a safety layer using the NHA.
Both safety architectures have distinct advantages and disadvantages that affect the system's overall cost, integration effort and certification. Tab. 1 summarises the central aspects of both architectures.

4 Conclusion

This article has outlined two safety architectures for implementing a safety-critical system capable of hosting virtualised CCS functions up to Safety Integrity Level (SIL) 4.
When considering the existing CCS applications (e.g. digital interlockings and ETCS functions), Safety Architecture B appears to be advantageous as it reduces the redesign effort for application migrations. This is particularly relevant for trackside systems. Therefore, DB InfraGO, together with its industry partners, is planning to further pursue Safety Architecture B within the context of any future initiatives and technology pilots, such as the virtualisation of interlocking functions on an on-premises platform. Within this context, interlocking functions are executed not as standalone hardware units, but as software instances on a common virtualisation platform, which can also be used for DSTW solutions in the future.
Safety Architecture A may offer benefits for those onboard systems and applications that require strict real-time guarantees, such as TCMS, the central vehicle-side safety and management system. It may also be interesting for novel safety-relevant railway applications, e.g. within the context of GoA 4, as it can help lower the market entry barrier for new players in application development.
Relevant topics such as isolation, deterministic data distribution, data synchronisation, replication and voting, as well as common cause failures are also being investigated in the Cloud4Rail (C4R) project [3, 4] within the IPCEI-CIS program [8], which aims to implement a platform architecture for safety-critical railway applications based on the work in Europe's Rail [1, 2]. This article has been created as part of this project. The complete C4R documents will be made available via the EU IPCEI-C4R (not yet online as of the time of writing). ■