



ELISA
Enabling **Linux** in
Safety Applications

WORKSHOP

Architectures for Linux in Railway Safety Applications

Florian Wühr

Senior Software Engineer, EMEA Field CTO Office, Red Hat

Dr. Daniel Weingärtner

Senior Software Engineer, EMEA Field CTO Office, Red Hat



What we'll discuss today

- Introducing ourselves
- Red Hat's role in the rail industry
- Safety applications in trackside and rolling stock appliances
- Certification hurdles
- Notable safety architectures for modern railway appliances
- Conclusion

Who we are



Florian Wühr, Senior Embedded Software Engineer, Red Hat's EMEA Field CTO Office

- 5 years safety-relevant communication technology at Deutsche Bahn
- 6+ years in Embedded Linux
- ~1,5 years experience in Functional Safety
- Working on GoA4 Autonomous Train Operation at Red Hat within funded research project “Automated Train”

Who we are



Dr. Daniel Weingaertner, Senior Software Engineer, Red Hat's EMEA Field CTO Office

- 13 years computer science professor at UFPR University (Brazil)
- 4 years automotive infotainment team lead
- ~1,5 years experience in Functional Safety
- Working on GoA4 Autonomous Train Operation at Red Hat within research project “Automated Train”

Red Hat's role in the rail industry

Red Hat's role in the rail industry

- Red Hat products currently not actively used in railway functional safety applications
- Active participation in governmental funded research project “Automated Train” researching GoA4 ATO
- Red Hat's role: Investigate manufacturer independent high performance computing platforms for autonomous driving
- And: Investigate transferability of Red Hat automotive safety solution (Red-Hat In-Vehicle Operating System) into railway domain

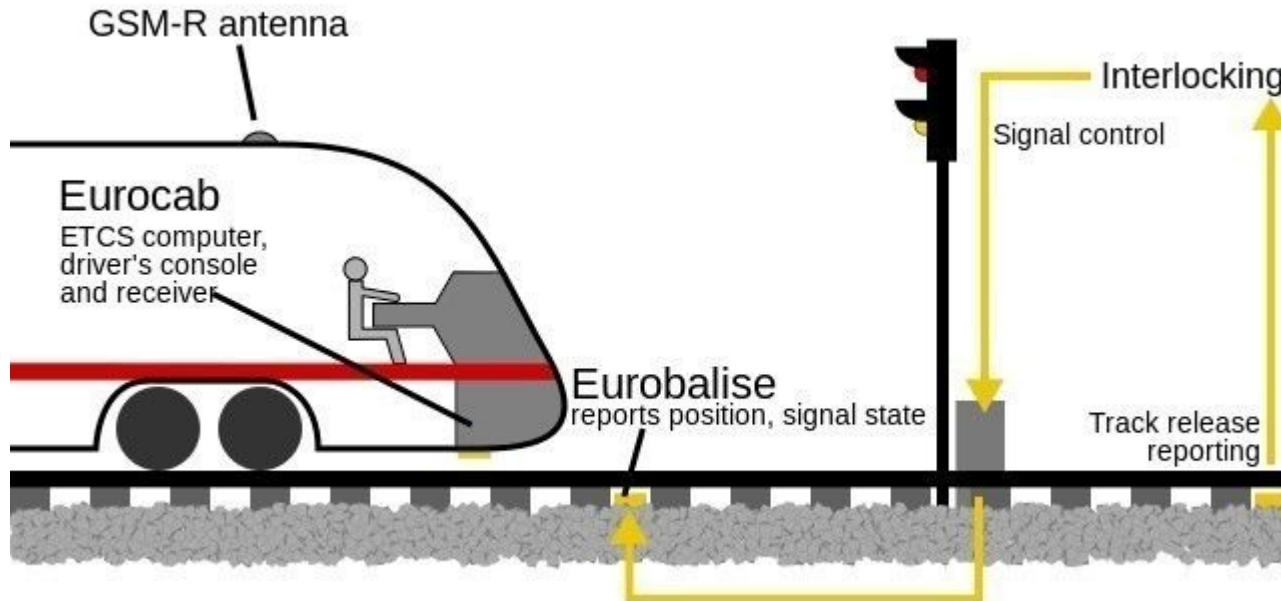
Glossary:

GoA4: Grade of Automation 4 (fully autonomous driving trains)

ATO: Automated Train Operation

Safety applications in trackside and rolling stock appliances

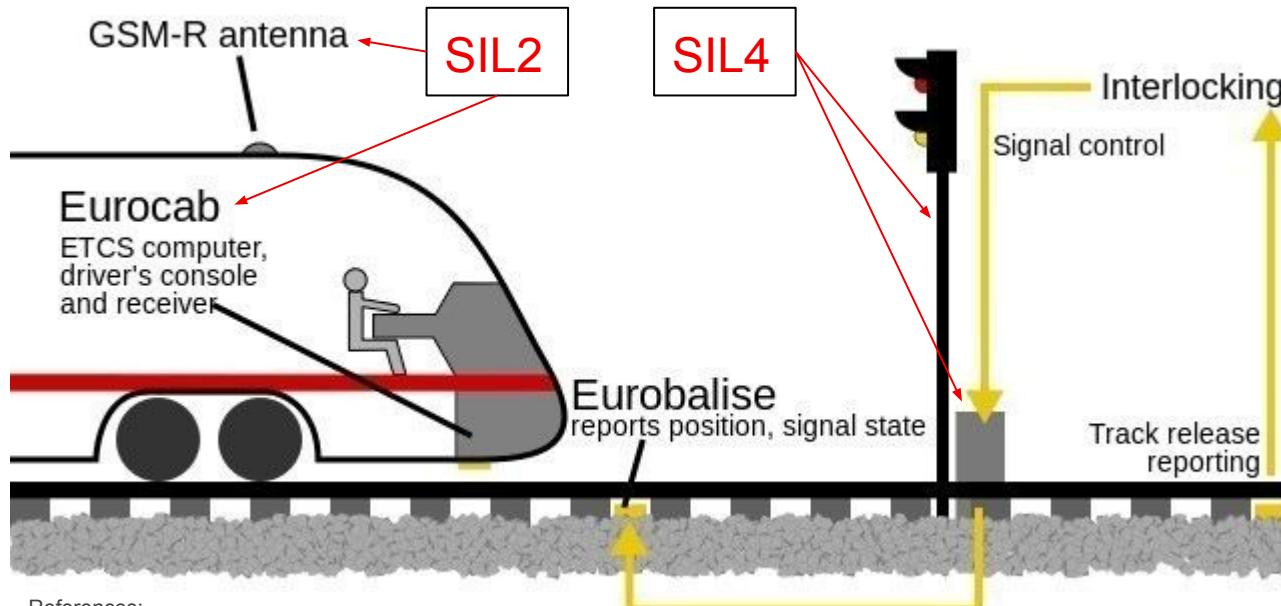
Trackside safety appliances (example ETCS)



Reference:

https://transport.ec.europa.eu/transport-modes/rail/ertms/what-ertms-and-how-does-it-work/etcs-levels-and-modes_en
(accessed on November 10th, 2025)

Trackside safety appliances (example ETCS)



References:

- https://transport.ec.europa.eu/transport-modes/rail/ertms/what-ertms-and-how-does-it-work/etcs-levels-and-modes_en
- <https://www.sensonics.com/en/blog/safety-integrity-levels-sil-explored--3247/#:~:text=SIL%20Examples.4%20%E2%80%93%20Mainline%20railway%20signalling%20systems>. (accessed on November 10th, 2025)

Handbuch Eisenbahninfrastruktur (German), Springer, 2025, p. 612

Rolling-stock safety appliances (example SiFa)



References:

<https://www.rundschau-online.de/wirtschaft/deutsche-bahn-gibt-neuen-ice-in-auftrag-371156>

(accessed on November 11th, 2025)

Handbuch Eisenbahninfrastruktur (German), Springer, 2025, p. 612

SIL2

Rolling-stock safety appliances (example ATO)



SIL2

References:

https://de.wikipedia.org/wiki/Automatic_Train_Operation#/media/Datei:ATO_over_ETCS_2016-09-21b.jpg,

<https://www.sensonic.com/en/blog/safety-integrity-levels-sil-explored--3247/#:~:text=SIL%20Examples.4%20%E2%80%93Mainline%20railway%20signalling%20systems>. (accessed on November 11th, 2025)

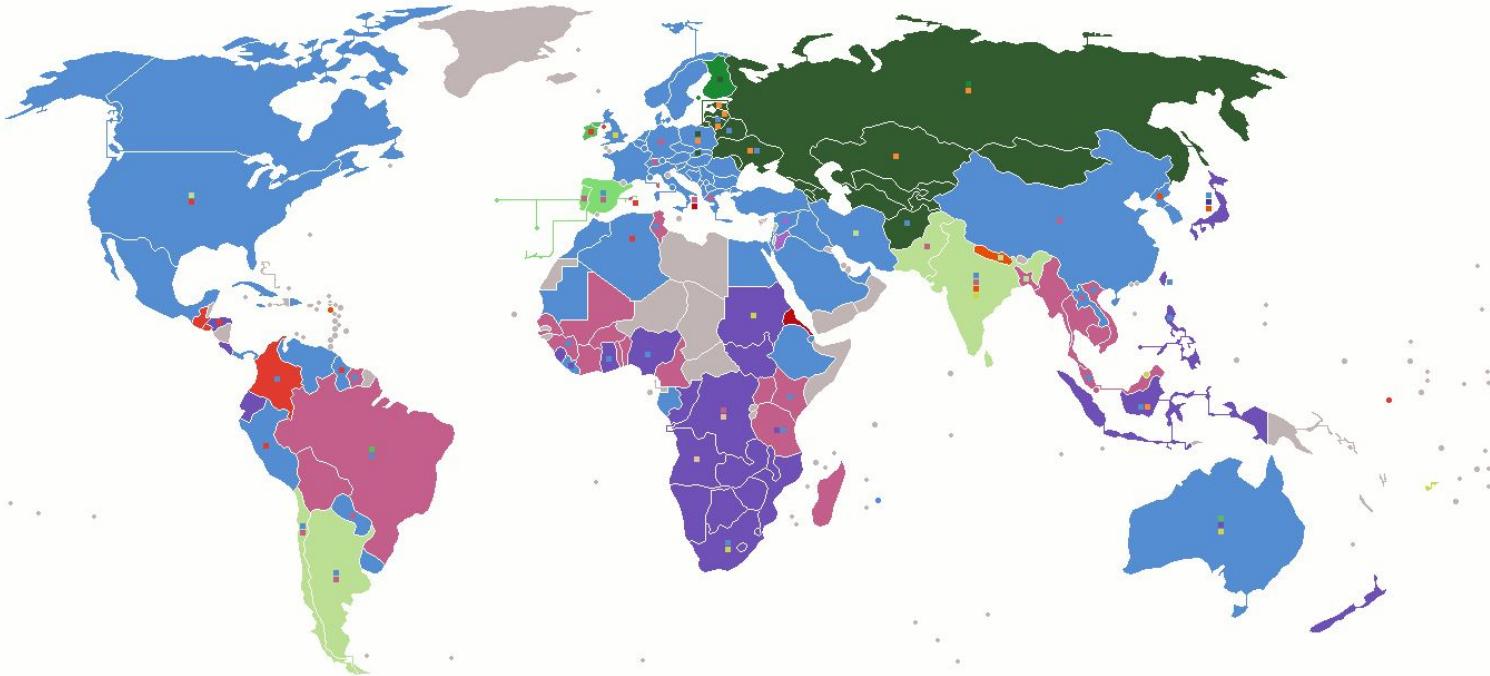
Certification hurdles

Historic development of railways

In history, railways have developed differently in different jurisdictions and historic empires which make it difficult to have interoperability. Even in some countries there are multiple systems in place:

Here are some examples:

- Track width: 1435mm (standard gauge), 1668mm (Iberian gauge, Spain, Portugal), 1520mm (russian broad gauge)



	1676	1668	1600	1524	1520	1435	1372	1067	1050	1000	950	914	762	750	610	600
ft in	5'6"	5'5.67"	5'3"	5'	4'11.8"	4'8.5"	4'6"	3'6"	3'5.3"	3'3.4"	3'1.4"	3'	2'6"	2'5.5"	2'	1'11.6"

Source:

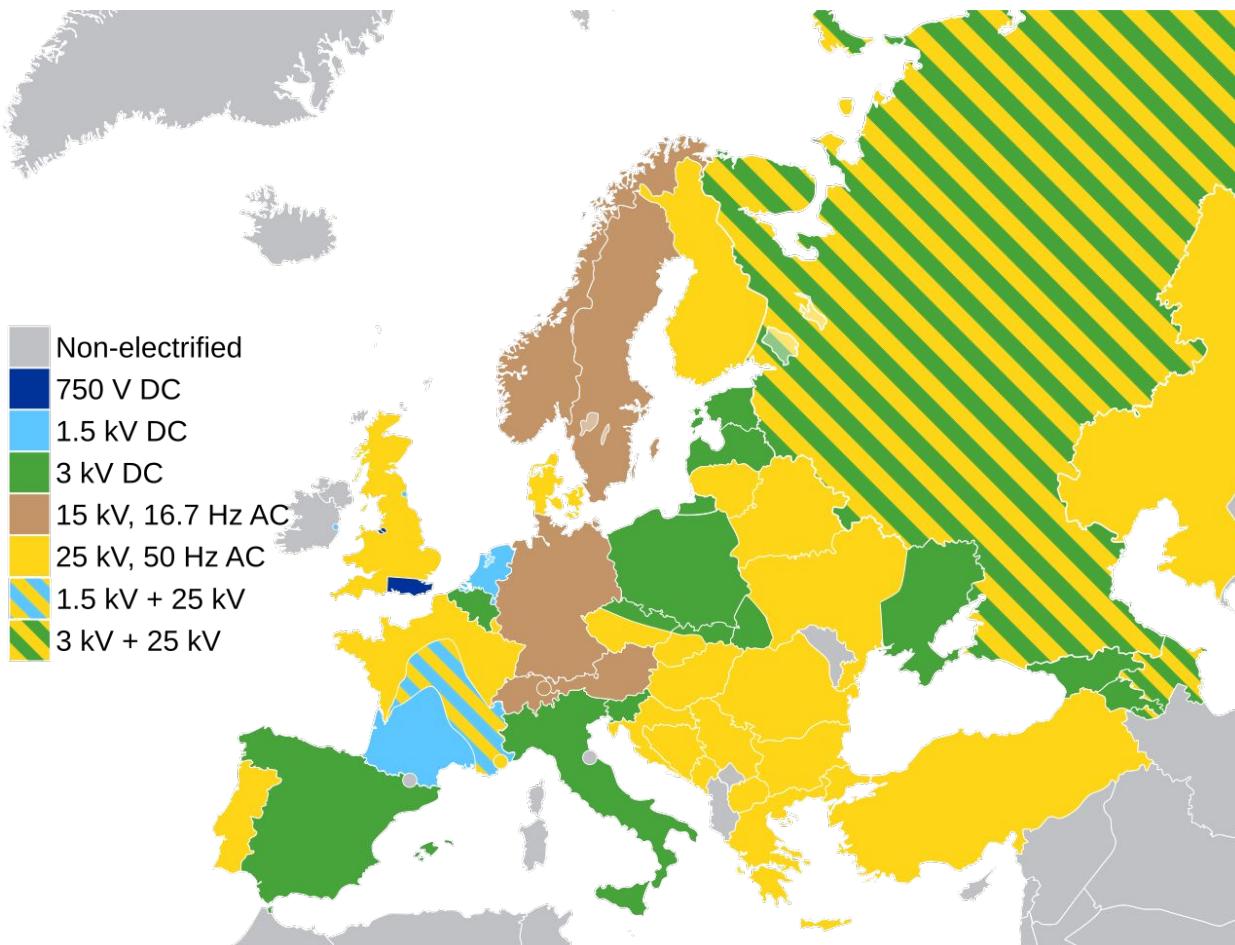
https://de.wikipedia.org/wiki/Normalspur#/media/Datei:Rail_gauge_world.png (accessed on October 16th, 2025)

Historic development of railways

In history, railways have developed differently in different jurisdictions and historic empires which make it difficult to have interoperability. Even in some countries there are multiple systems in place:

Here are some examples:

- Track width: 1435mm (standard gauge), 1668mm (Iberian gauge, Spain, Portugal), 1520mm (russian broad gauge)
- Electric power systems: 750 V DC, 1,5kV DC, 3kV DC, 15 kV 16,7 Hz AC, 25 kV 50 Hz AC, mixed systems, ...



Source:

https://en.wikipedia.org/wiki/Railway_electrification#/media/File:Europe_rail_electrification_en.svg (accessed on October 16th, 2025)

Historic development of railways (2)

- > Interoperability is harmed due to historic and national developments and regulation
- Railway components (both rolling stock and trackside) still need national certification in order to be admitted for legal use by law in European countries
- Therefore functional safety standards like EN 50128, EN 50129 and EN50716 become mandatory by law
- First steps into european certification made with rolling stock certification through the European Rail Agency (ERA) and TSI EU regulations in order to foster harmonization
- But trackside infrastructure still remains a national certification issue
- Other jurisdictions such as like the US or China and Japan do not enforce compliance to any of these standards, but also enforce certification of components before they can be used

Safety architectures for modern railway appliances

- [OCORA](#) publishes specifications of a Safe Computing Platform for future rail operations, based on requirements from the Railway undertakings.
- German Railways created the [Digitale Schiene Deutschland](#) initiative, fostering projects aimed at creating a vendor independent safety enabled compute platform.
 - [SIL4Datacenter](#), [SIL4Cloud](#)
- The European Union has been financing railway [research projects](#) through its [Europe's Rail body](#).
 - [FP2R2Dato](#) proposes a Modular Compute Platform

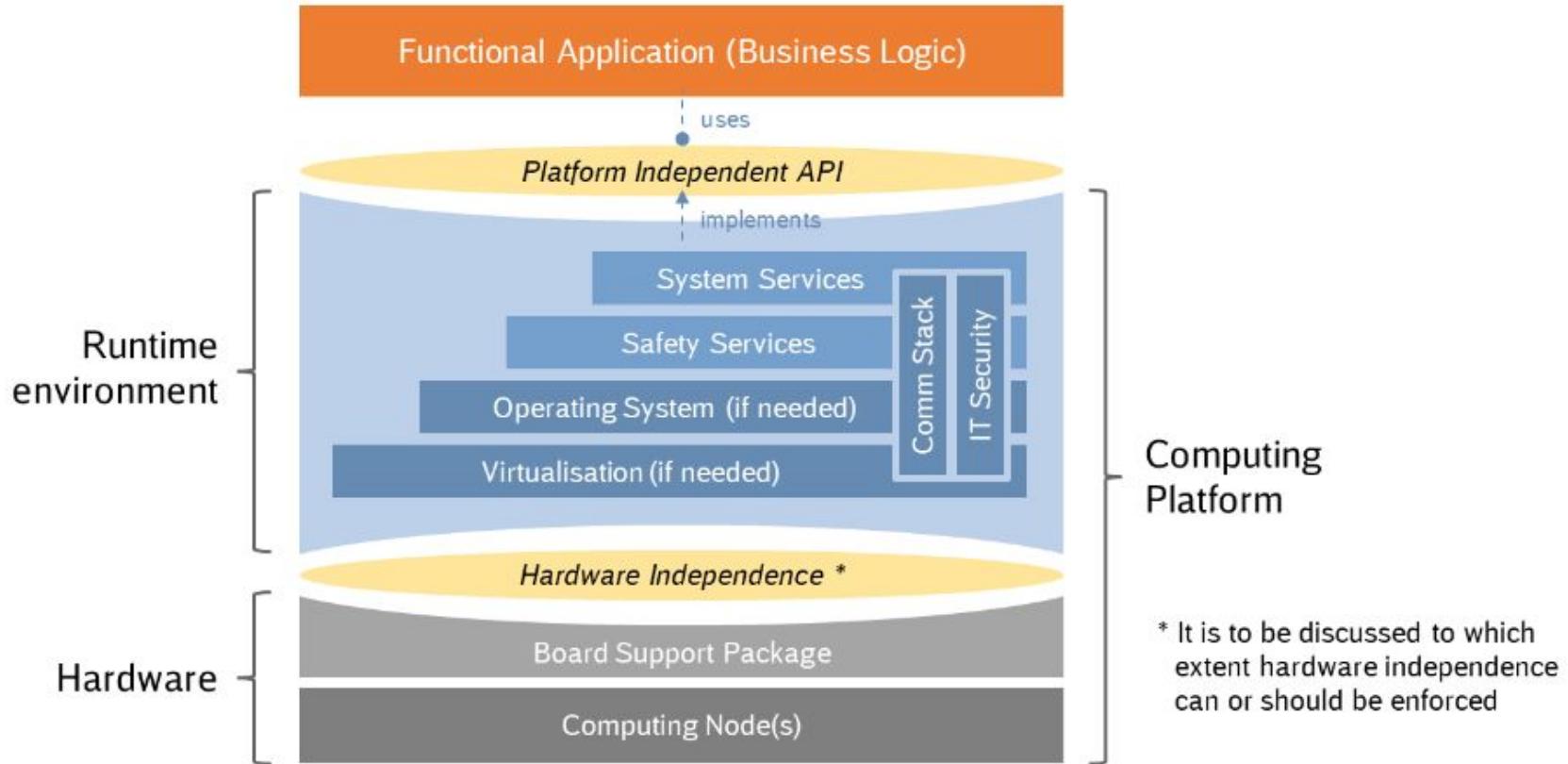
- Requirements for modern railway compute platforms:
 - Use COTS hardware and software
 - Allow for multi-tenant (multi-vendor) safety applications
 - Allow for mixed-criticality up to SIL4
 - System evolvability
 - Independence of the life cycle of components (e.g. HW, OS, libraries, applications)
 - Possibility of partial re-certifications

OCORA

Open CCS On-board Reference Architecture

CCS = Command, Control and Signaling

OCORA Generic Safe Computing Platform

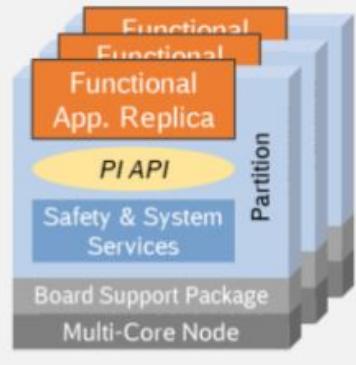


Source: [OCORA-TWS03-010_Computing-Platform-Whitepaper.pdf](#) (accessed in October/2025)

OCORA Onboard Safe Computing Platform

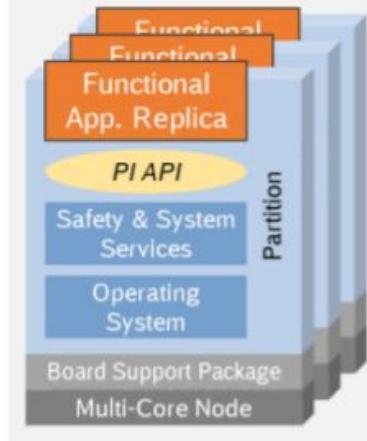
Option A:

Hardware without operating system



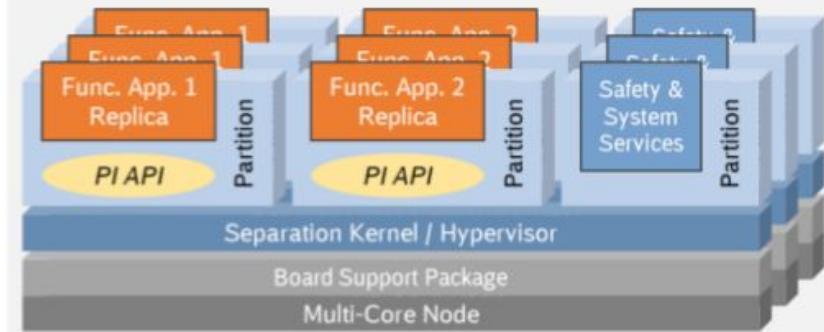
Option B:

Hardware with operating system



Option C:

Separation kernel / hypervisor based



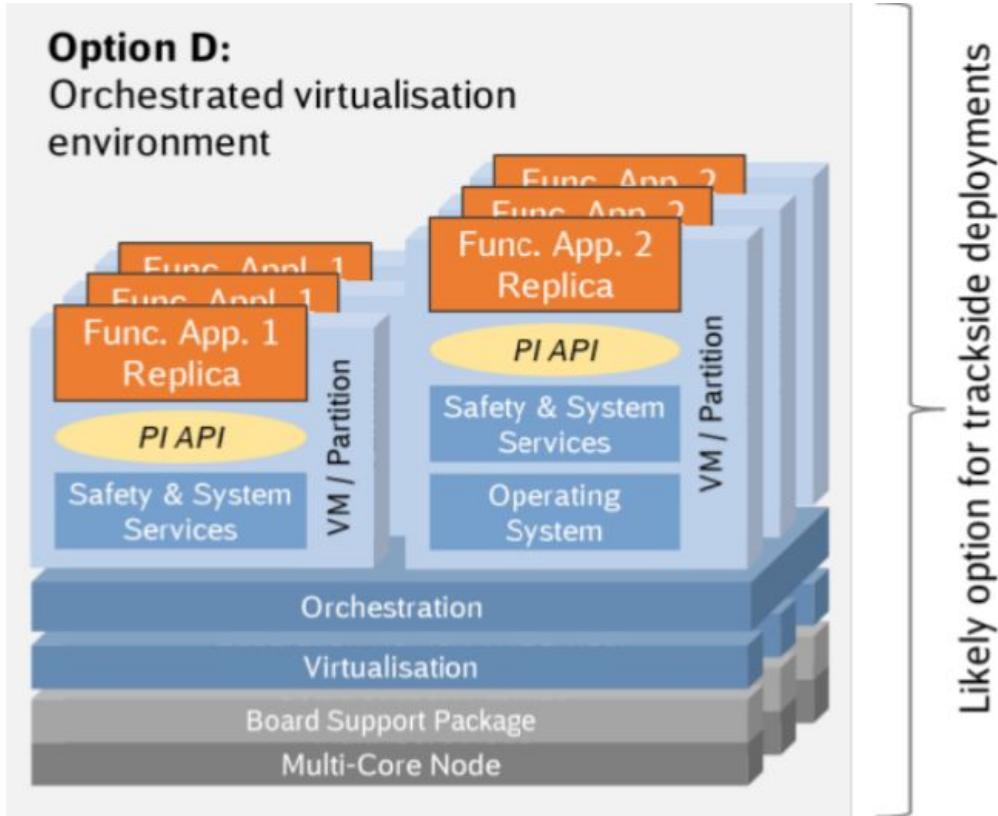
Likely options for onboard deployments

Source: [OCORA-TWS03-010_Computing-Platform-Whitepaper.pdf](#) (accessed in October/2025)

OCORA Trackside Safe Computing Platform

Option D:

Orchestrated virtualisation environment



- An **alternative approach** is to run applications with their own "**guest OS**" instead of compiling them against the Platform Interoperability (PI) API
 - Drawback:** critical aspects like memory management are handled by the guest OS, outside the control of the platform below the PI API.
 - More challenging for safety assessment and authorization

Source: [OCORA-TWS03-010_Computing-Platform-Whitepaper.pdf](#) (accessed in October/2025)

OCORA Modular Safety Assessment

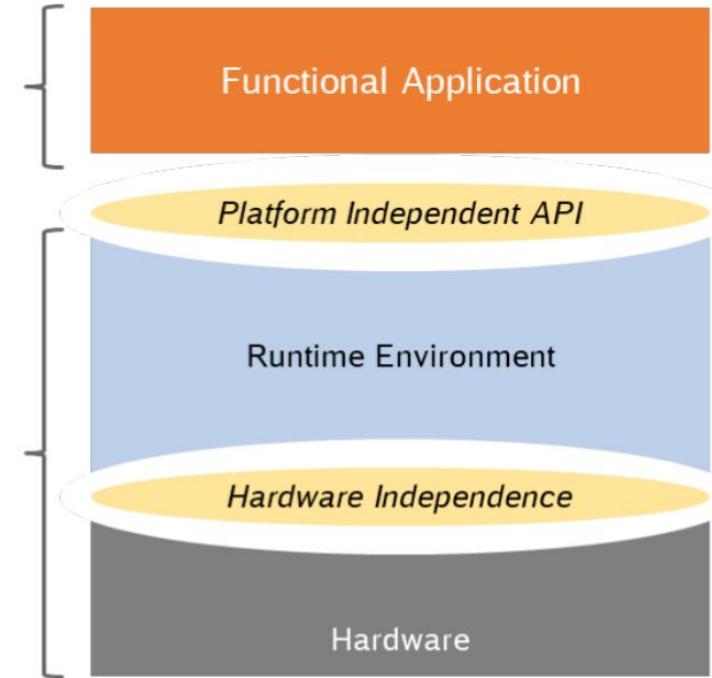
Subsystem integrator provides overall E2E certification and obtains authorisation

The subsystem integrator may be

- application vendor
- platform vendor
- 3rd party

Application vendor provides SW assessment and statement on SRAC compliance / handling

Platform vendor provides certification(s) for platform and (ideally simplified and standardised) SRACs imposed on applications



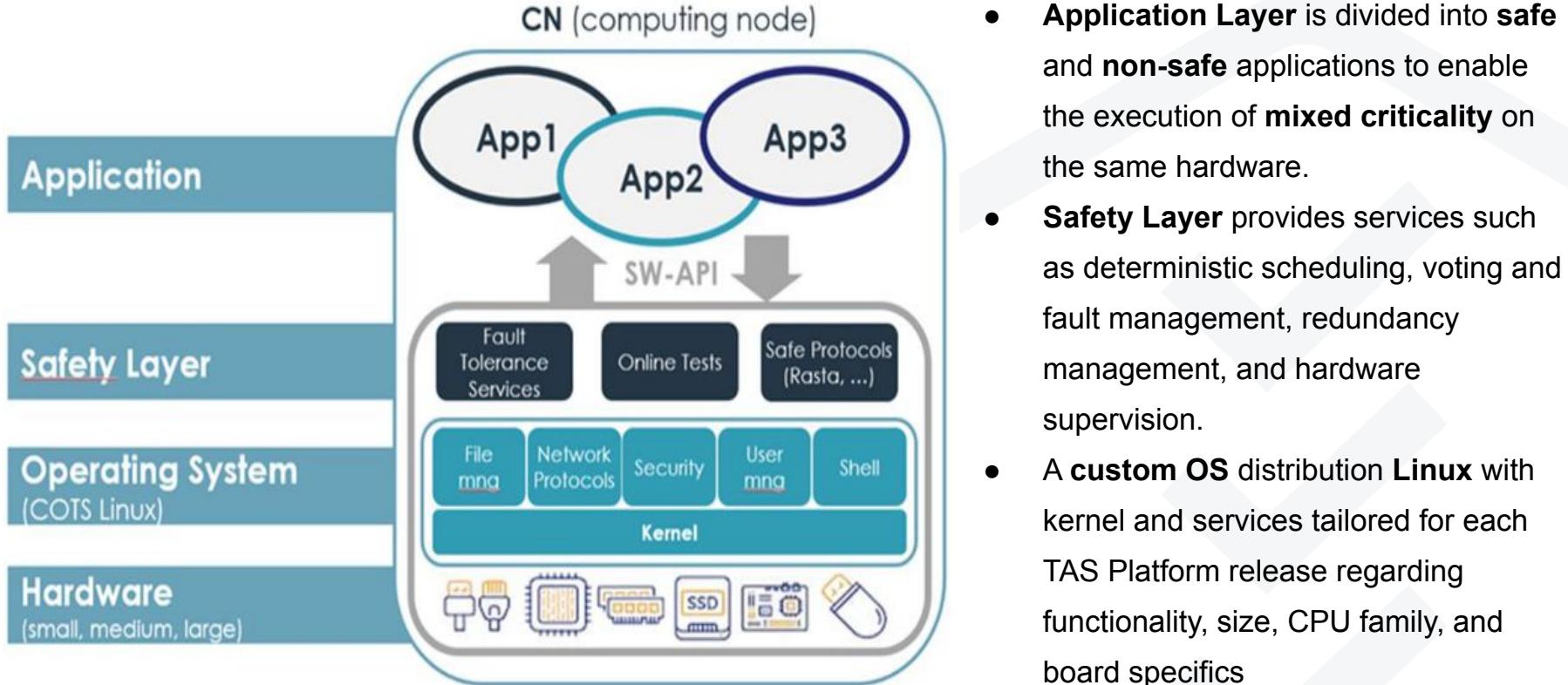
SRAC: Safety Related Application Conditions

Source: [OCORA-TWS03-010_Computing-Platform-Whitepaper.pdf](https://ocora-tws03-010_Computing-Platform-Whitepaper.pdf) (accessed in October/2025)

Digitale Schiene Deutschland

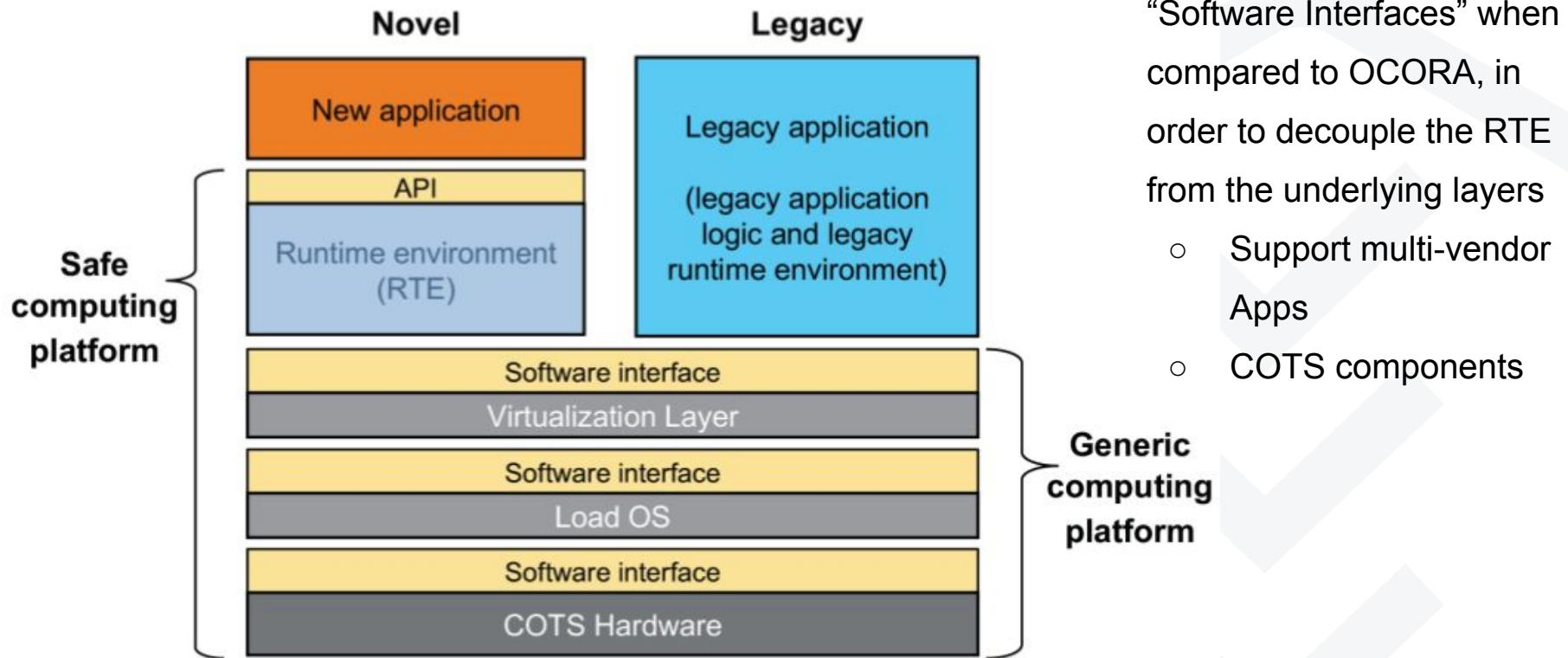
SIL4 Cloud and SIL4 Datacenter

SIL4Cloud architecture by Thales



Source: [Research Report SIL4 Cloud](#) (accessed on October/2025)

SIL4 Data Center architecture by Siemens

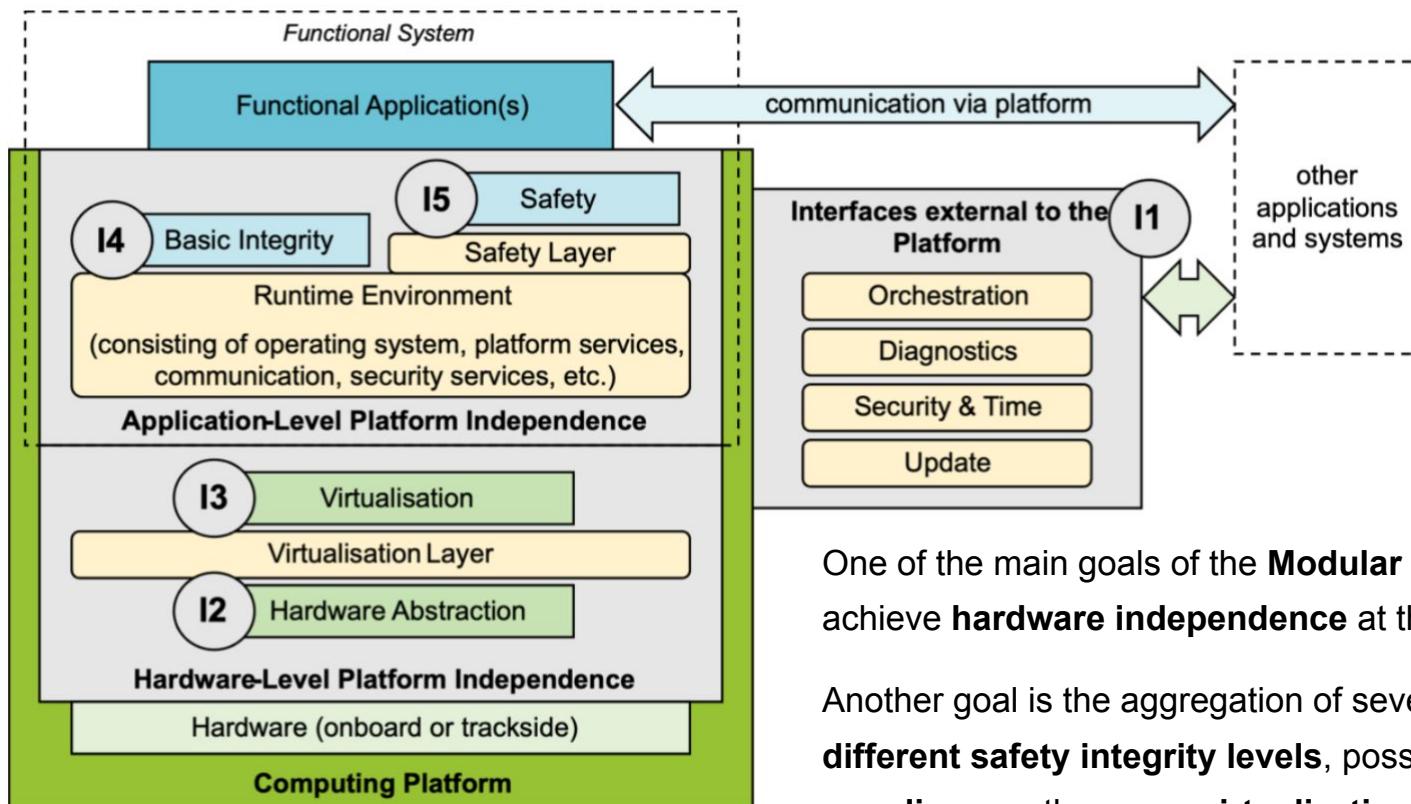


Source: [Research Report SIL4 Data Center](#) (accessed October/2025)

Europe's Rail Joint Undertaking

FP2-R2DATO Modular Compute Platform (2025)

Safety Architecture for FP2-R2DATO

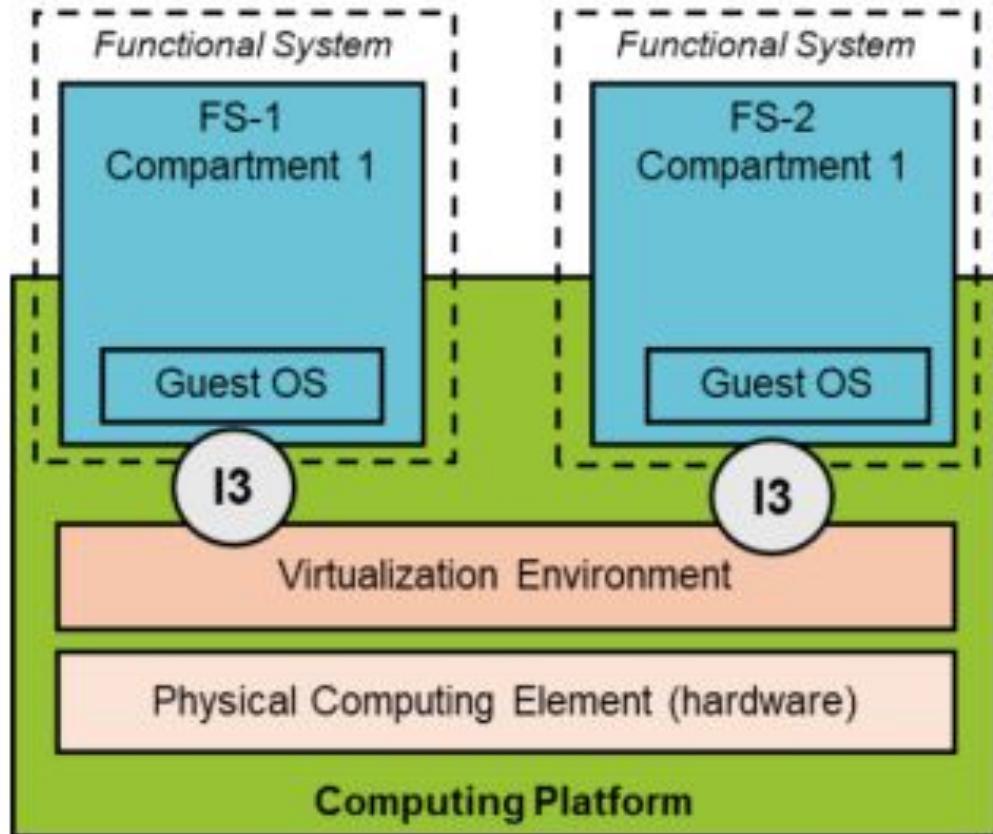


One of the main goals of the **Modular Compute Platform** is to achieve **hardware independence** at the interface I3.

Another goal is the aggregation of several FS compartments of **different safety integrity levels**, possibly provided by **different suppliers**, on the **same virtualization environment**.

Source: [D26.3 – Final Modular Platform requirements, architecture and specification](#) (accessed October/2025)

Safety Architecture for FP2-R2DATO

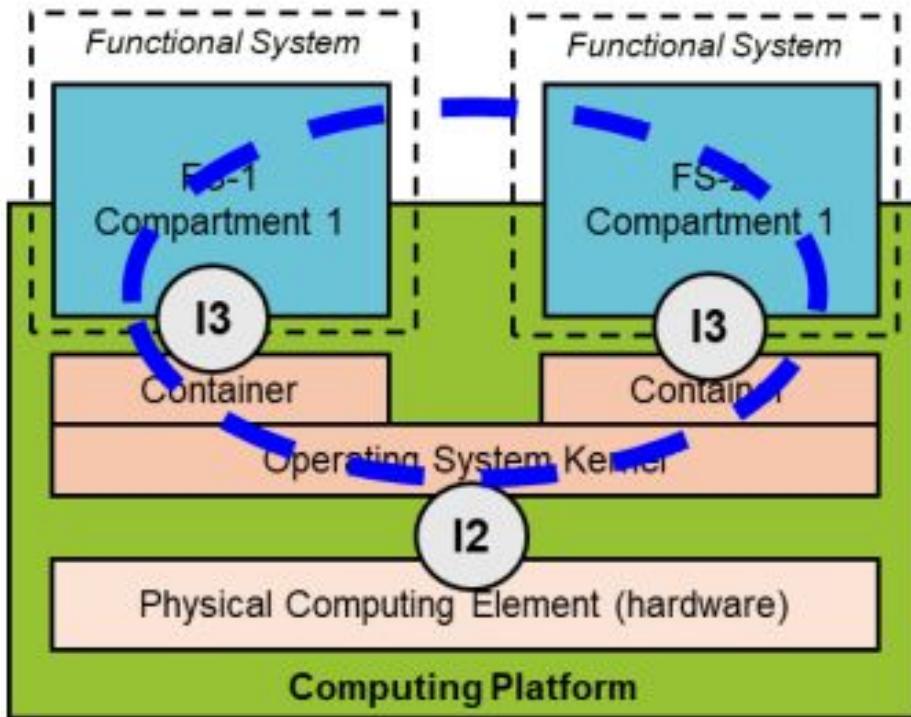


- **Basic architecture with Virtualization Environment (VE) and I3 interface**
 - I3 is not an interface in sense of "programming interface" but it's the definition of needed functionalities and features within the virtualization layer from view of the Functional Systems running above.

Source: [D26.3 – Final Modular Platform requirements, architecture and specification](#) (accessed October/2025)

Safety Architecture for FP2-R2DATO

overall performance testing

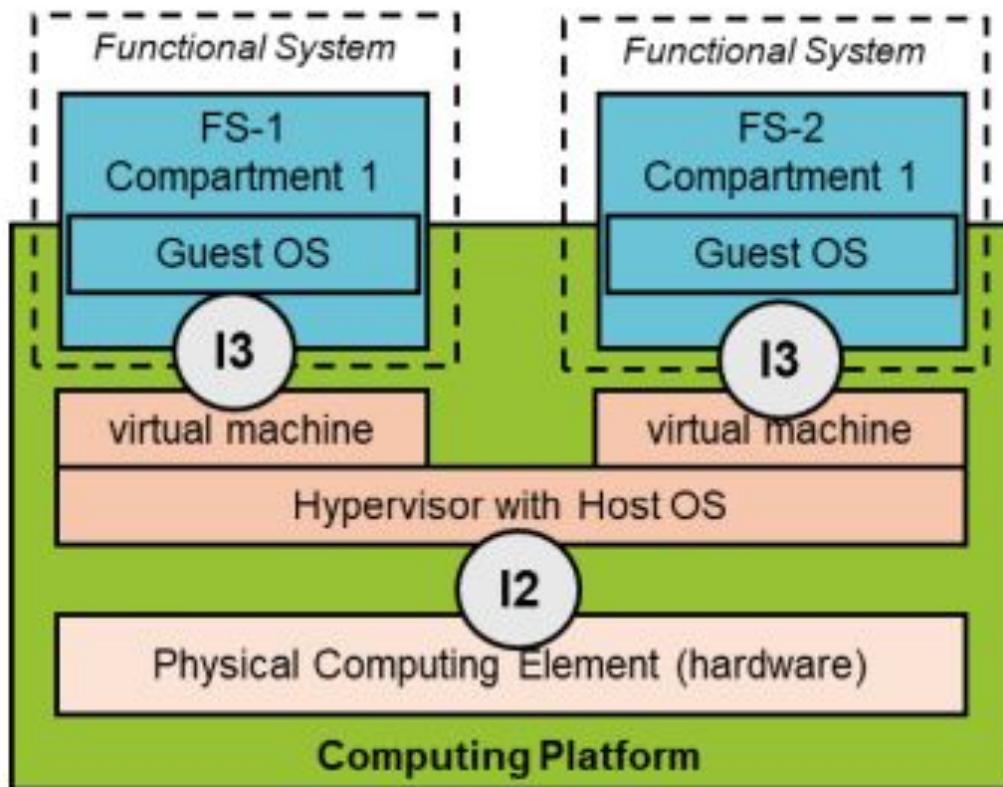


Container as Virtual Environment

- All FS must be based on the same OS
 - Only one OS has to be maintained
- Weak resource isolation.
 - Overall integration necessary, but potential for resource saving.
- Overall performance testing necessary
 - implying integration and qualification of containerised FS Compartments stays in the responsibility of one single vendor/ integrator.

Source: [D26.3 – Final Modular Platform requirements, architecture and specification](#) (accessed October/2025)

Safety Architecture for FP2-R2DATO

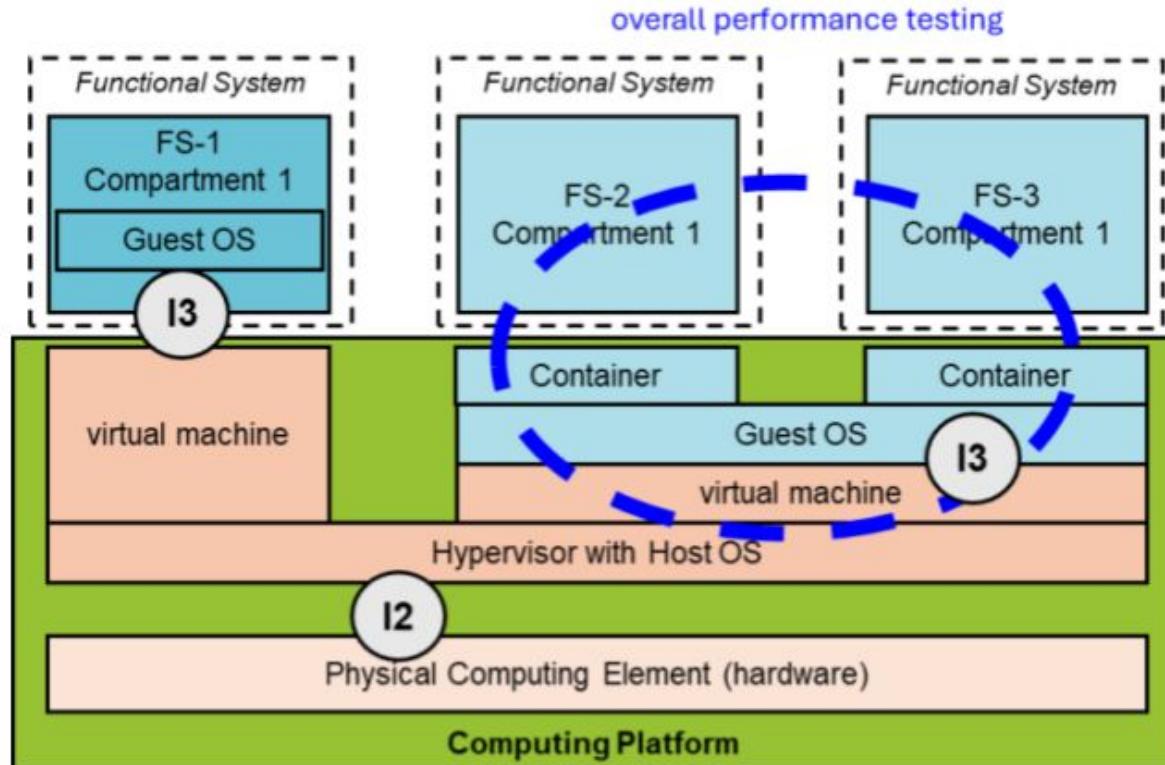


Hypervisor as Virtual Environment

- Each FS can be based on its own OS type
 - Increased maintenance effort
- Best available solution for the resource isolation
 - additional resource demands and a potential higher performance decrease in comparison to containers

Source: [D26.3 – Final Modular Platform requirements, architecture and specification](#) (accessed October/2025)

Safety Architecture for FP2-R2DATO



Hypervisor and Container as VE

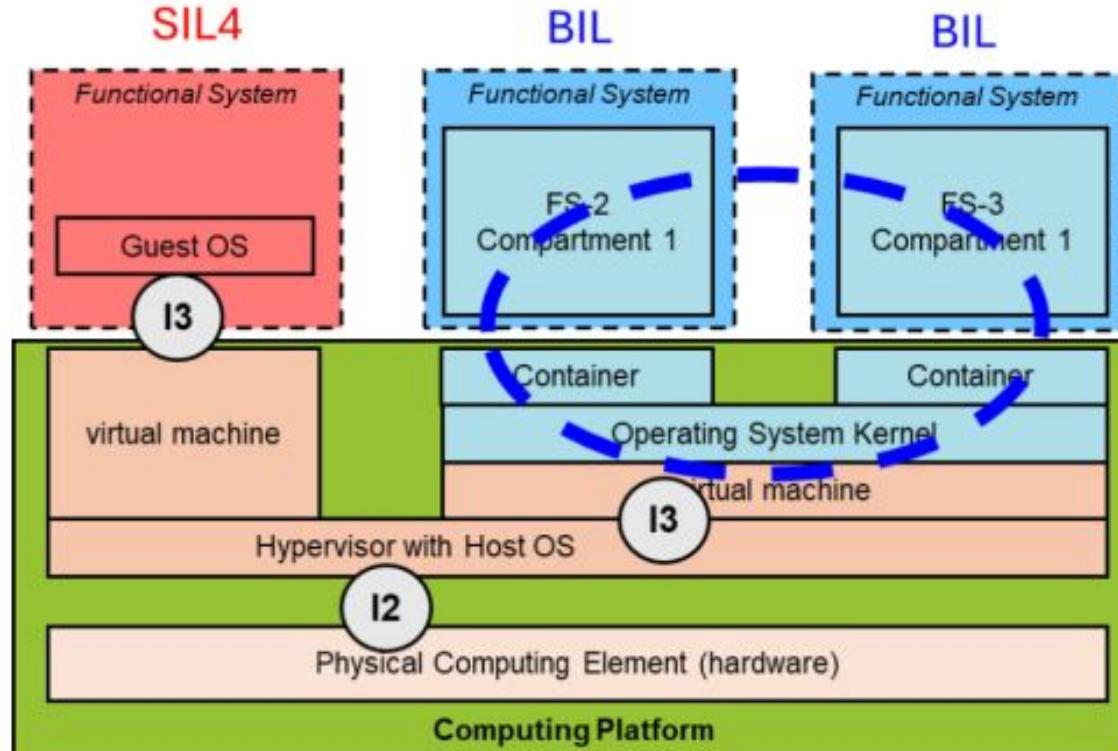
- Considers the usage of hypervisors, containers or a mixture of both.
- For safety-critical functions, a technical possibility is needed for functionalities running on the host OS to access the physical hardware.

Source: [D26.3 – Final Modular Platform requirements, architecture and specification](#) (accessed October/2025)

Containers x Hypervisors in FP2-R2DATO

Feature	Container Technology	Hypervisor Technology
OS Flexibility	No flexibility: All Functional Systems (FS) must use the same operating system type (e.g., Linux).	High flexibility: Each FS can use its own Guest OS type (e.g., Linux, Windows).
OS Maintenance	Advantage: Only one OS (the host kernel) needs to be maintained.	Disadvantage: Requires additional effort to maintain multiple Guest OS types.
Technical Dependency	Dependencies exist between the common OS kernel and the FS running above.	No direct technical dependencies to the FS compartments running above.
Isolation & Resource Usage	Weak isolation: Multiple FS share the same OS kernel resources.	Best isolation: Excellent resource isolation (cores, memory, communication) for aggregated FS compartments.
Resource Efficiency	Resource-saving approach, suitable for environments with limited hardware (like onboard rolling stock).	Higher resource demands and potentially a higher performance decrease compared to containers.
Integration/Testing	Overall performance testing is necessary. Integration/qualification is assumed to be the responsibility of a single vendor or integrator .	(<i>Not explicitly mentioned as a negative, but complexity is implied by isolation requirements</i>).

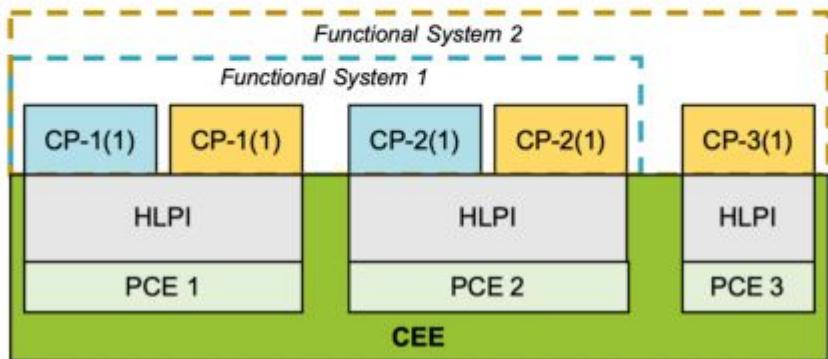
FP2-R2DATO Use Case for On-board Compute Platform



- **Independent maintenance** for different FS not required, as on-board SW updates occur only in maintenance
- **Limited resources** of on-board computing elements demand efficient usage/resource sharing
- Proposed architecture:
 - **Hypervisor** for SIL 4 FS
 - **Container** for Basic Integrity FS

Source: [D26.3 – Final Modular Platform requirements, architecture and specification](#) (accessed October/2025)

Redundancy and Diversity



Source: [D26.3 – Final Modular Platform requirements, architecture and specification](#) (accessed October/2025)

A more complicated – but hypothetical – setup employing two different FS, one **2oo2** and one **2oo3**, deployed on a pool of three Physical Compute Elements.

Q: What SIL level can containerized solution achieve?

Q: Are redundancy (MoN) with diversity (arm/i64) not something we should be considering more carefully for Linux?

Thank you!

Licensing of Workshop Results

All work created during the workshop is licensed under Creative Commons Attribution 4.0 International (CC-BY-4.0) [<https://creativecommons.org/licenses/by/4.0/>] by default, or under another suitable open-source license, e.g., GPL-2.0 for kernel code contributions.

You are free to:

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.



ELISA
Enabling Linux in
Safety Applications

WORKSHOP

