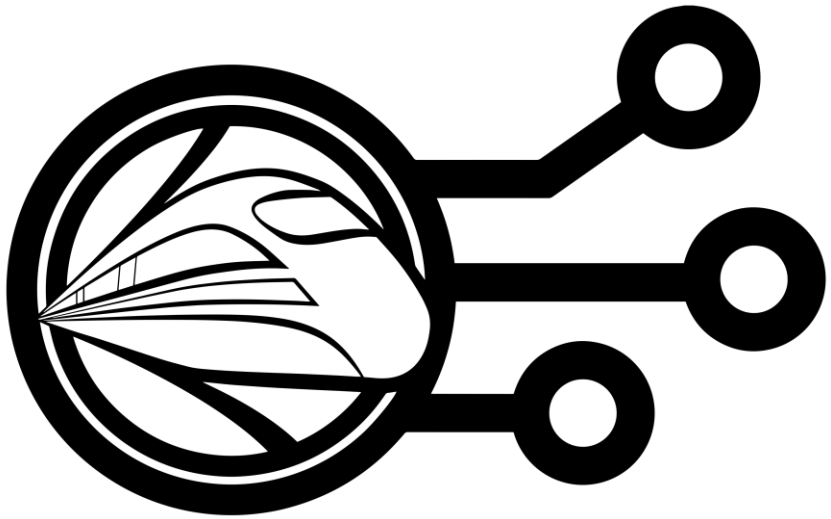


OCORA

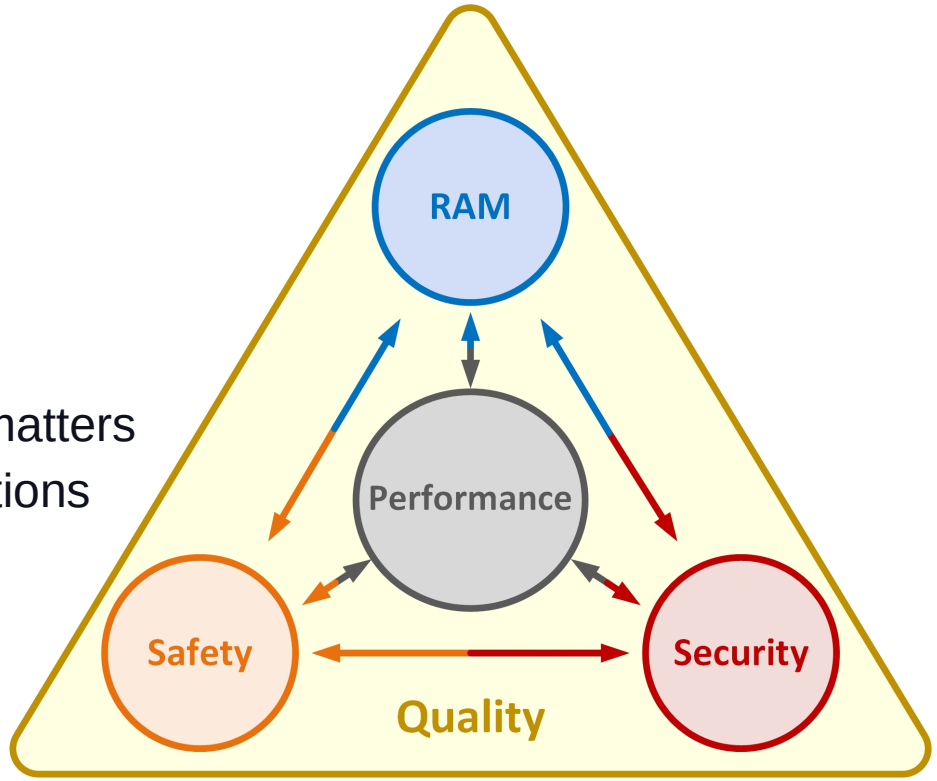
Quality, Performance, RAM, Safety & Security - (QPRAMSS) - Strategy

OCORA-TWS07-201 / v1.01 / 06.12.2022

Content

Content

The five (5) QPRAMSS-Domains



1. The Challenge to Meet

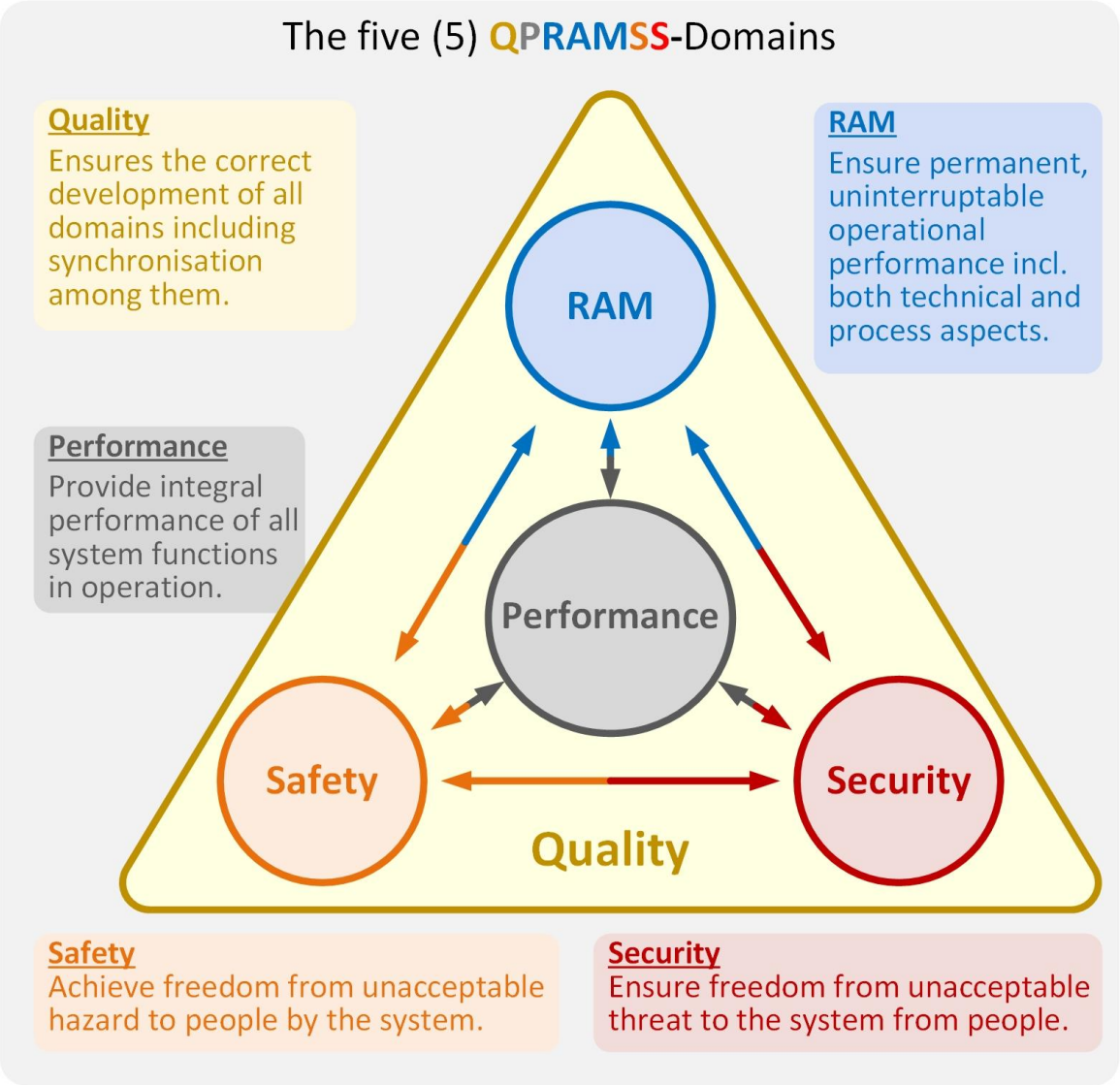
Challenge

Rail system must meet simultaneously demanding requirements regarding

1. Quality,
2. Performance (i.e. «system capabilities»),
3. RAM,
4. Safety, and
5. Security.

Question

How can these *conflicting requirements be met simultaneously* **without compromise**?



1. The Challenge to Meet

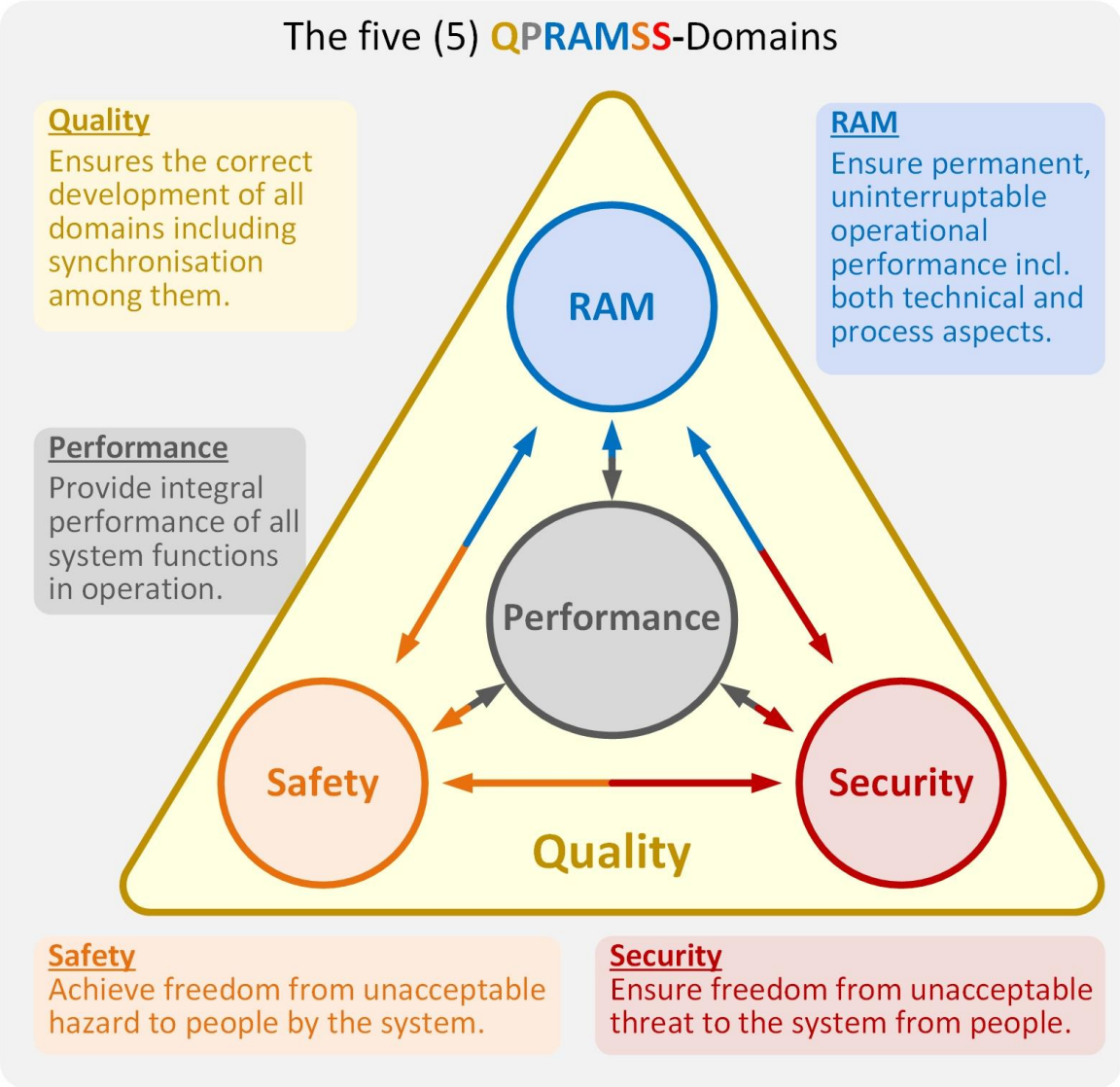
Question

How can these *conflicting requirements be met simultaneously* **without compromise**?

Facts

- **Railway Operators** are not keen to accept compromises on **Quality**!
- **Railway Operators** are not keen to accept compromises on **Availability**!
- **Railway Authorities** are not keen to accept compromises on **Safety**!

→ **Who approves (Cyber)-Security?**



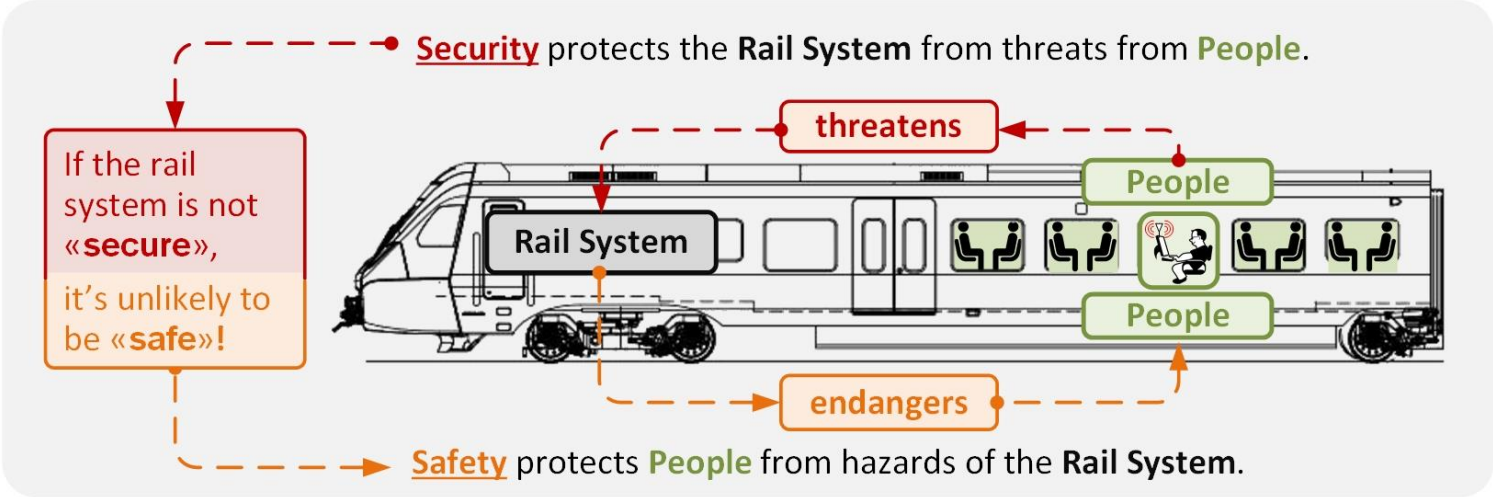
2. The Conflicts among the QPRAMSS domains are substantial

Safety vs. Security

- **Safety** protects **people** from hazards of the **rail system**.
- **Security** protects the **rail system** from threats from **people**.

If the **rail system** is not **secure**, it's unlikely to be **safe**!

→ **Is regulatory safety approval losing its credibility?**



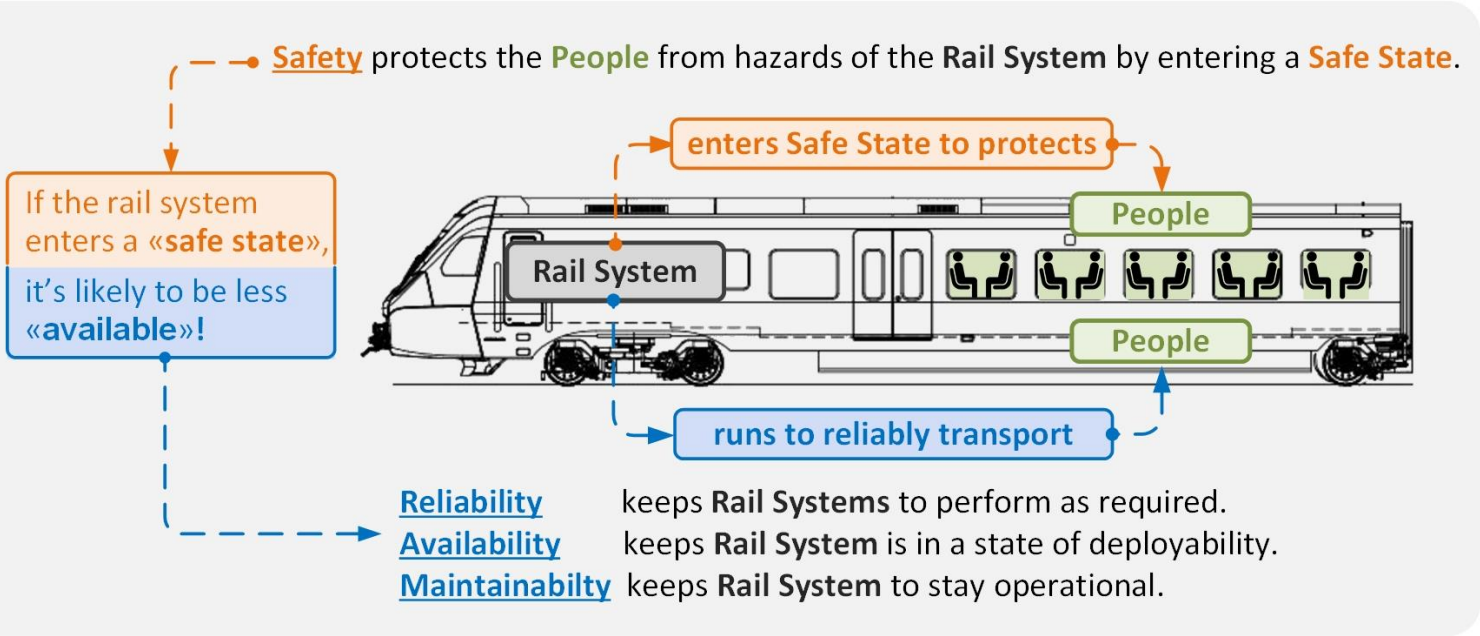
2. The Conflicts among the QPRAMSS domains are substantial

Safety vs. RAM

- **Safety** protects **people** from hazards of the **rail system** by entering a **safe state**.
- **RAM** is meant to run the **rail system** reliably to transport **people**.

If the **rail system** is in a **safe state**, it's likely to be less **reliable / available**!

→ **Is the railway operator prepared to accept this fact?**



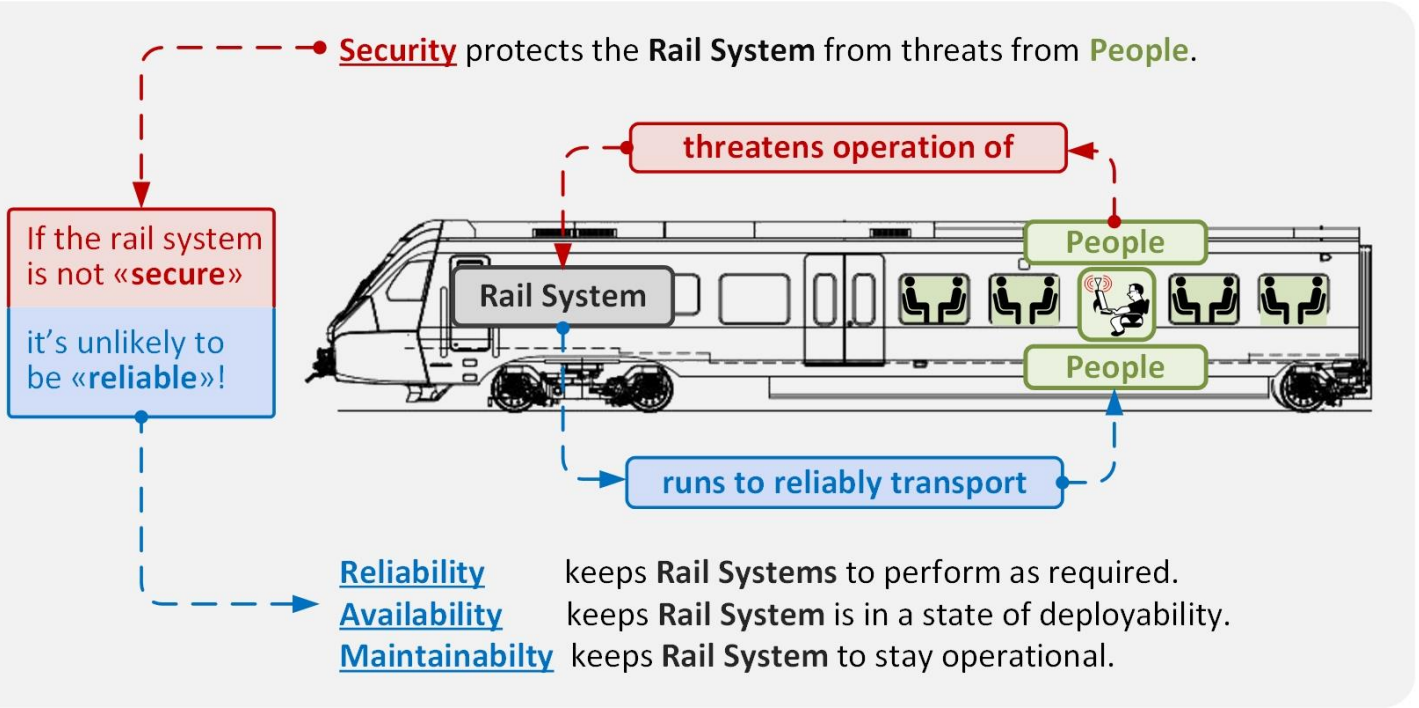
2. The Conflicts among the QPRAMSS domains are substantial

Security vs. RAM

- **Security** protects the rail system from threats from **people**.
- **RAM** is meant to run reliably to transport **people**.

If the rail system is not **secure**, it's unlikely to be **reliable**!

→ **What is more important**
- **RAM or**
- **Security?**



2. The Conflicts among the QPRAMSS domains are substantial

Human Factors matters!

- **Human Factors** protect the **rail system** from faults by **people**.
- **RAM** shall run **reliably** to transport **people**.

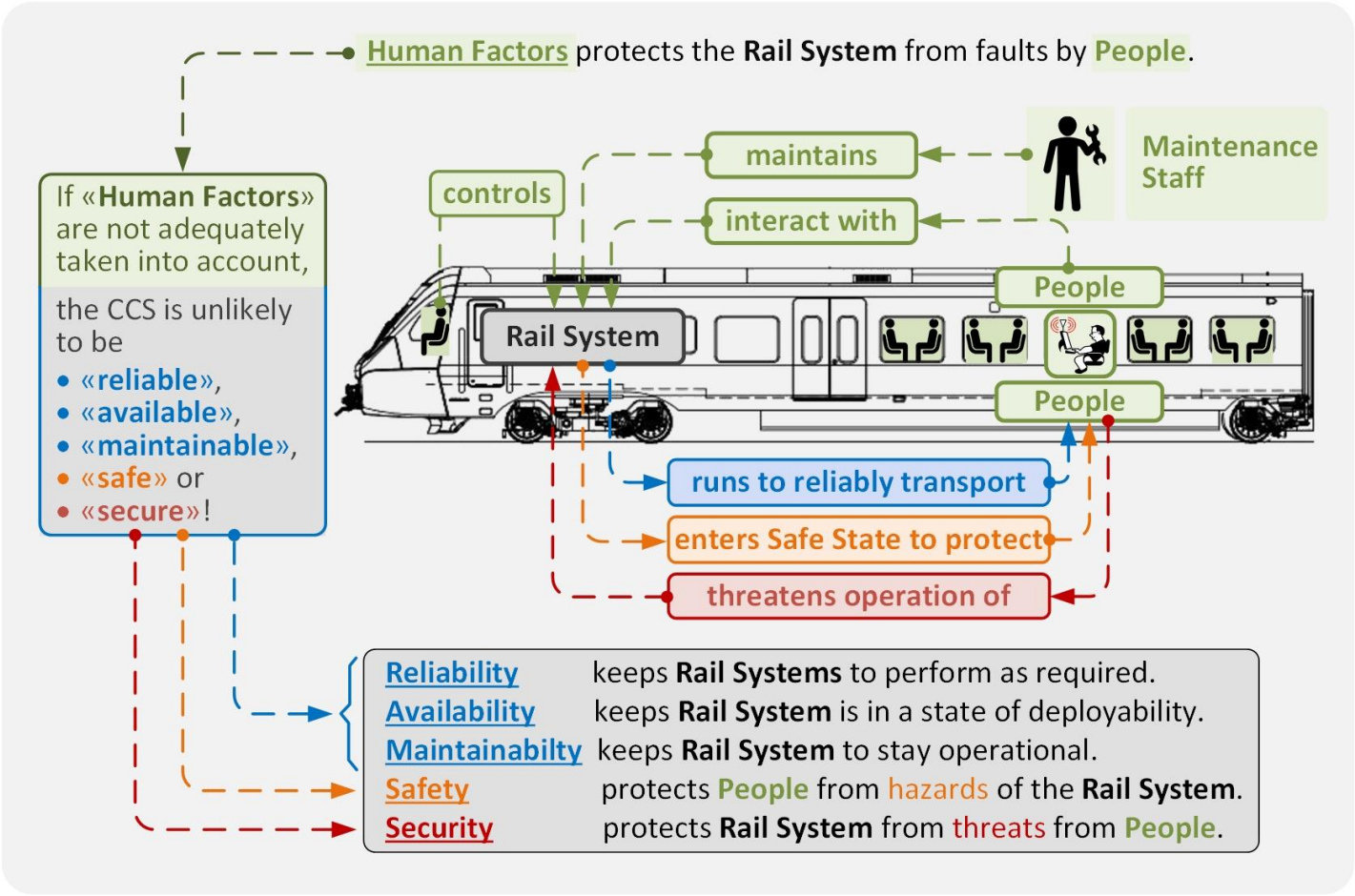
If **human factors** are not adequately considered, the **rail system** is likely to be

- less reliable,
- less available,
- less maintainable,
- less safe, and
- less secure!

→ **Are human factors adequately considered to ensure**

- **RAM,**
- **Safety, and**
- **Security**

of railway systems?



2. The Conflicts among the QPRAMSS domains are substantial

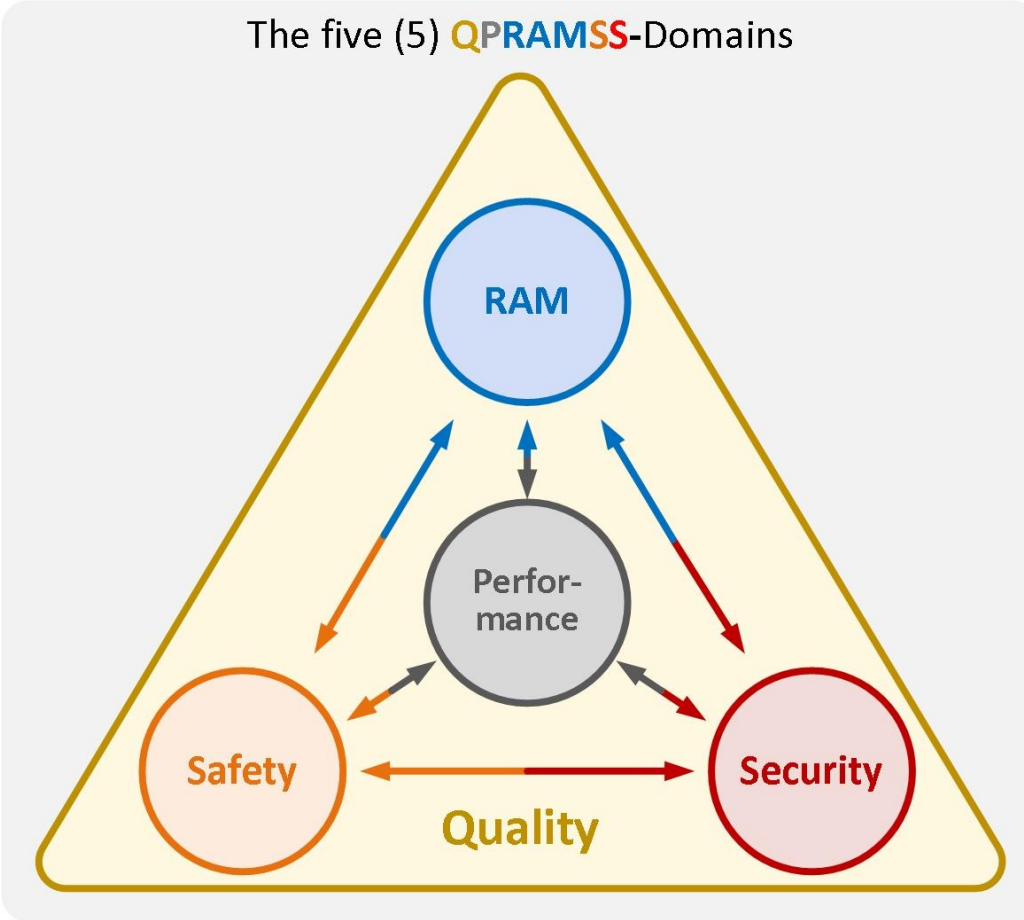
How to …

… **ensure completeness of the specifications** for all domains**?**

… **prevent conflicting requirements** among the domains**?**

… **master the complexity?**

… **guaranties consistent verifiability?**



3. How to solve the conflicts

3.1 Synchronisation among the QPRAMSS domains matters

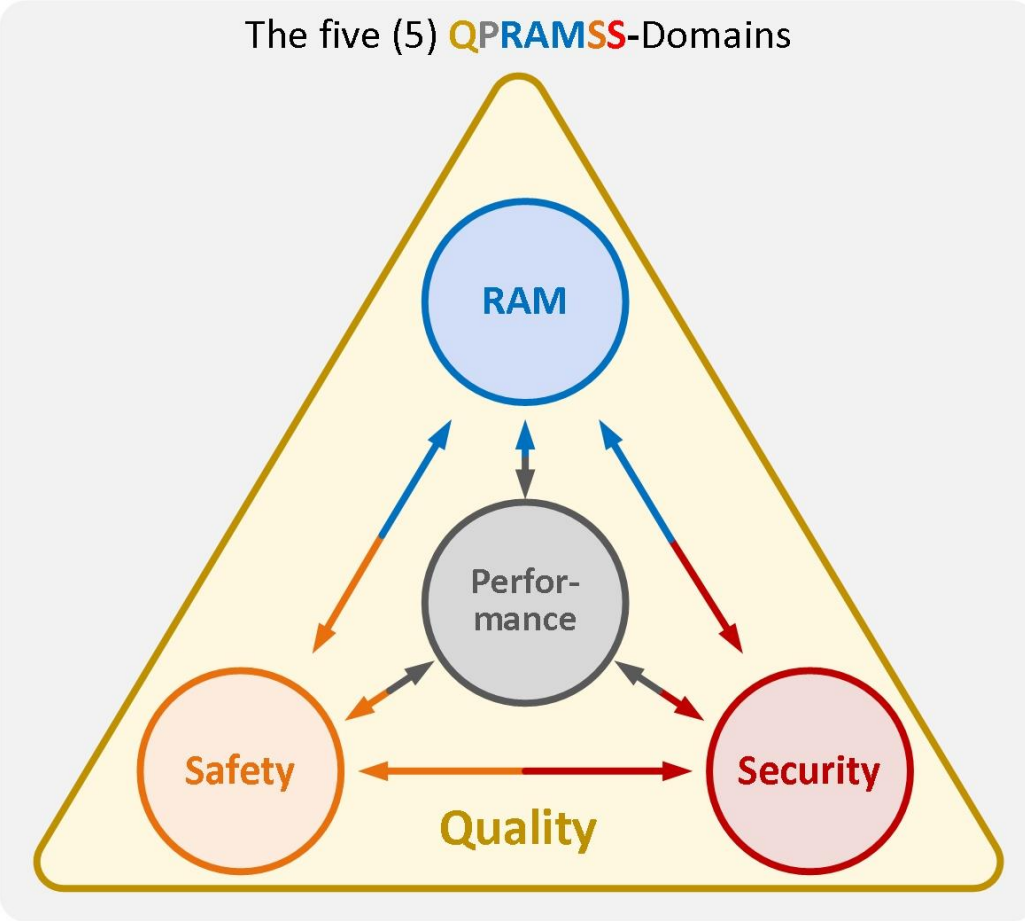
Content

3.1.1 Availability comes first!

3.1.2 Mastering QPRAMSS domains according to the railway standards

3.1.3 *Systematic top-down procedure* is key

3.1.4 *Synchronisation of activities* among all QPRAMSS domains is crucial



3. How to solve the conflicts

3.1 Synchronisation among the QPRAMSS domains matters

3.1.1 Availability comes first

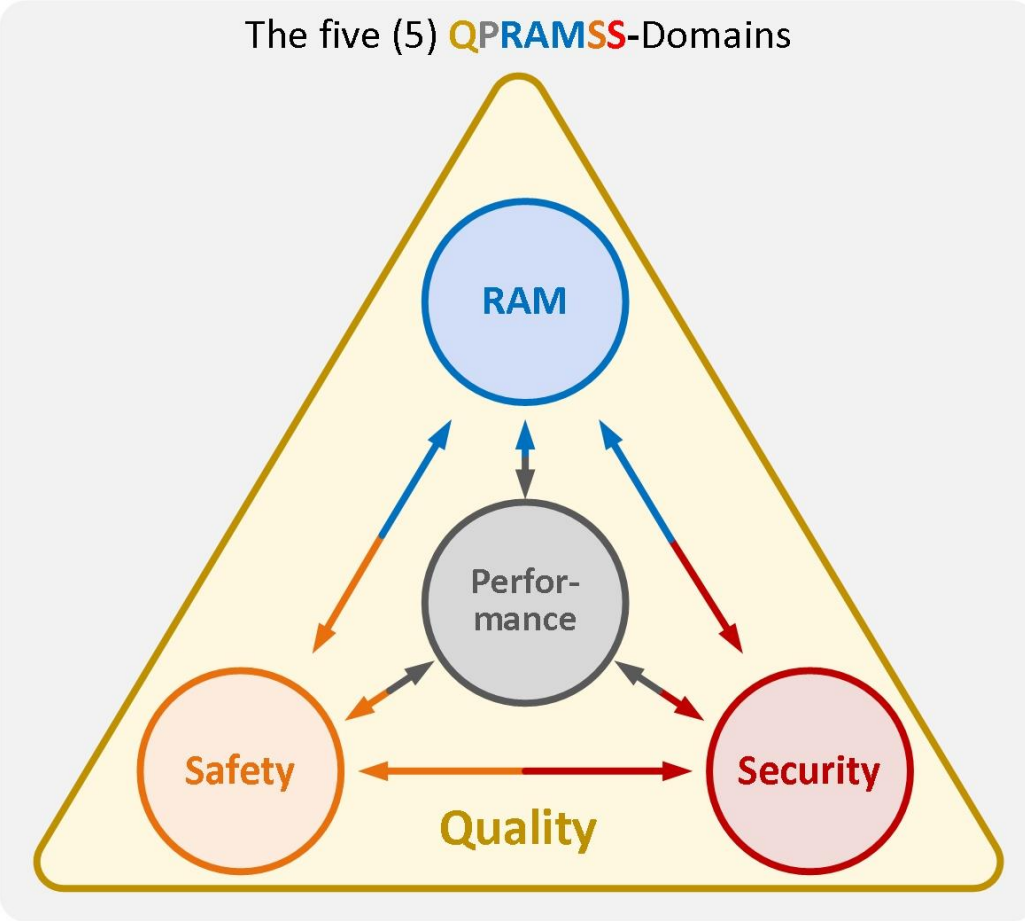
Guarantee

- **permanent** and
- **uninterruptable**

operational performance including

- **technical** and
- **process aspects**!

→ This prevents **cyber-security Denial of Service (DoS) attacks**!



3. How to solve the conflicts

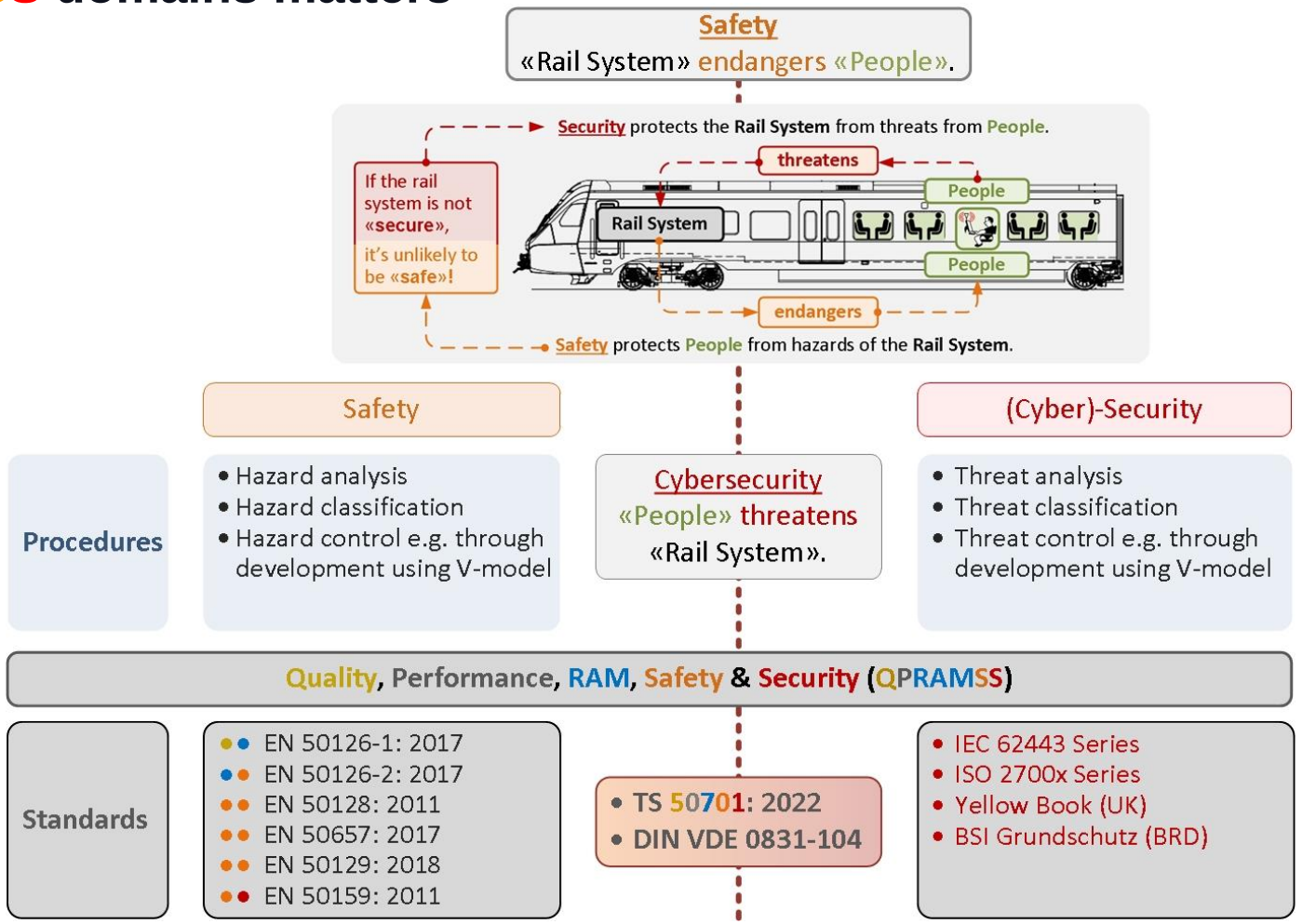
3.1 Synchronisation among the QPRAMSS domains matters

3.1.2 Mastering QPRAMSS domains according to the standards

Procedure specified in the TS 50701: 2022 coordinate the development activities among the domains

- Quality
- Performance («system capabilities»)
- RAM
- Safety (hazard analysis) and
- Security (threat analysis).

Mastering these five domains simultaneously is critical to the efficient development of railway applications at reasonable costs.



3. How to solve the conflicts

3.1 Synchronisation among the QPRAMSS domains matters

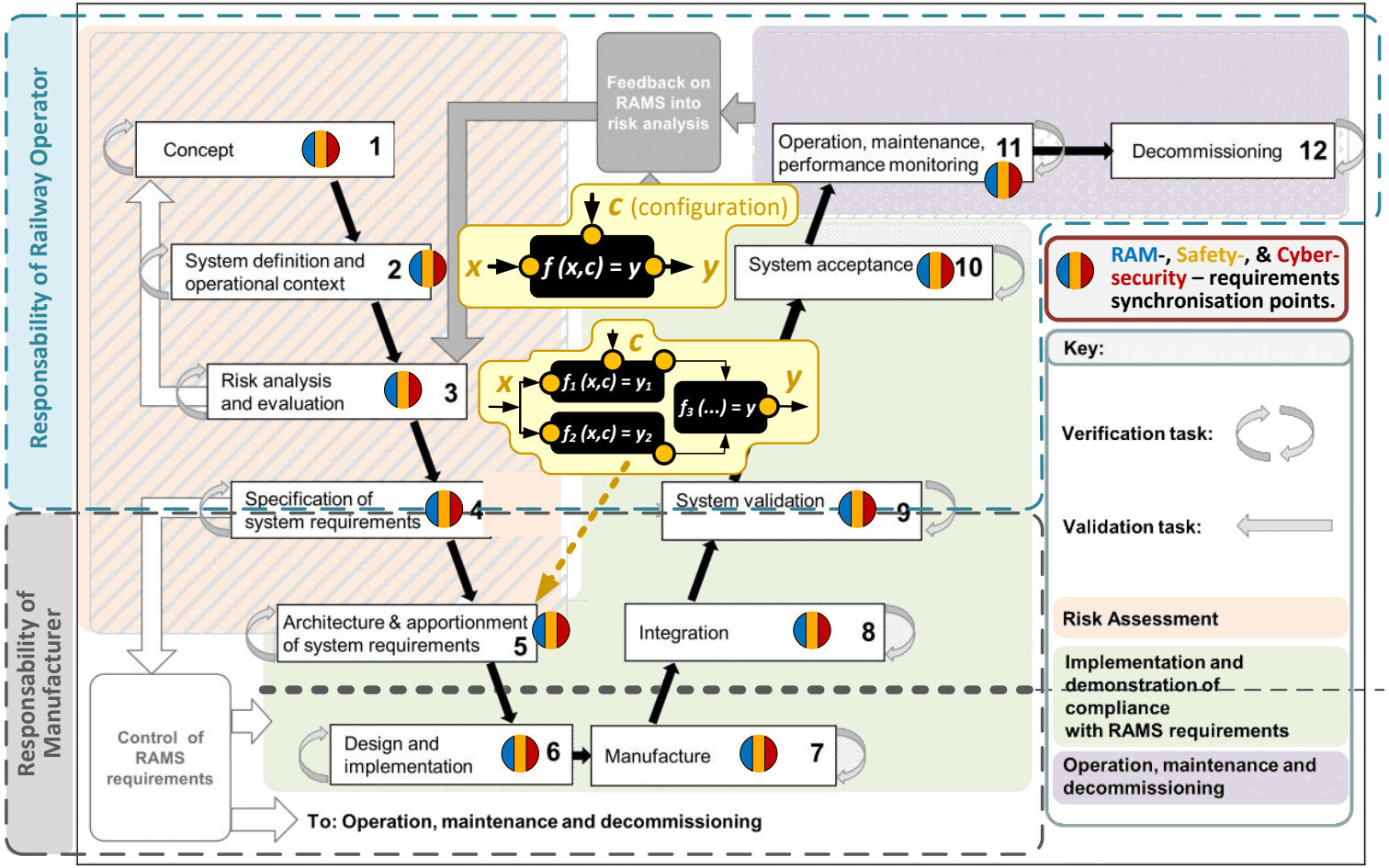
3.1.3 Systematic top-down procedure is key

This is at the core of the CENELEC standards

- EN 50126-1: 2017 (RAM & Safety),
- EN 50126-2: 2017 (RAM & Safety),
- EN 50128: 2011 (Safe SW, Signalling),
- EN 50657: 2017 (Safe SW, Roll.-Stock),
- EN 50129: 2018 (Safe HW, Safety Case),
- EN 50159: 2010 (Safe Communication), &
- TS 50701: 2022 (Security).

→ QPRAMS experiences are paramount.

→ Security experiences are required.



3. How to solve the conflicts

3.1 Synchronisation among the QPRAMSS domains matters

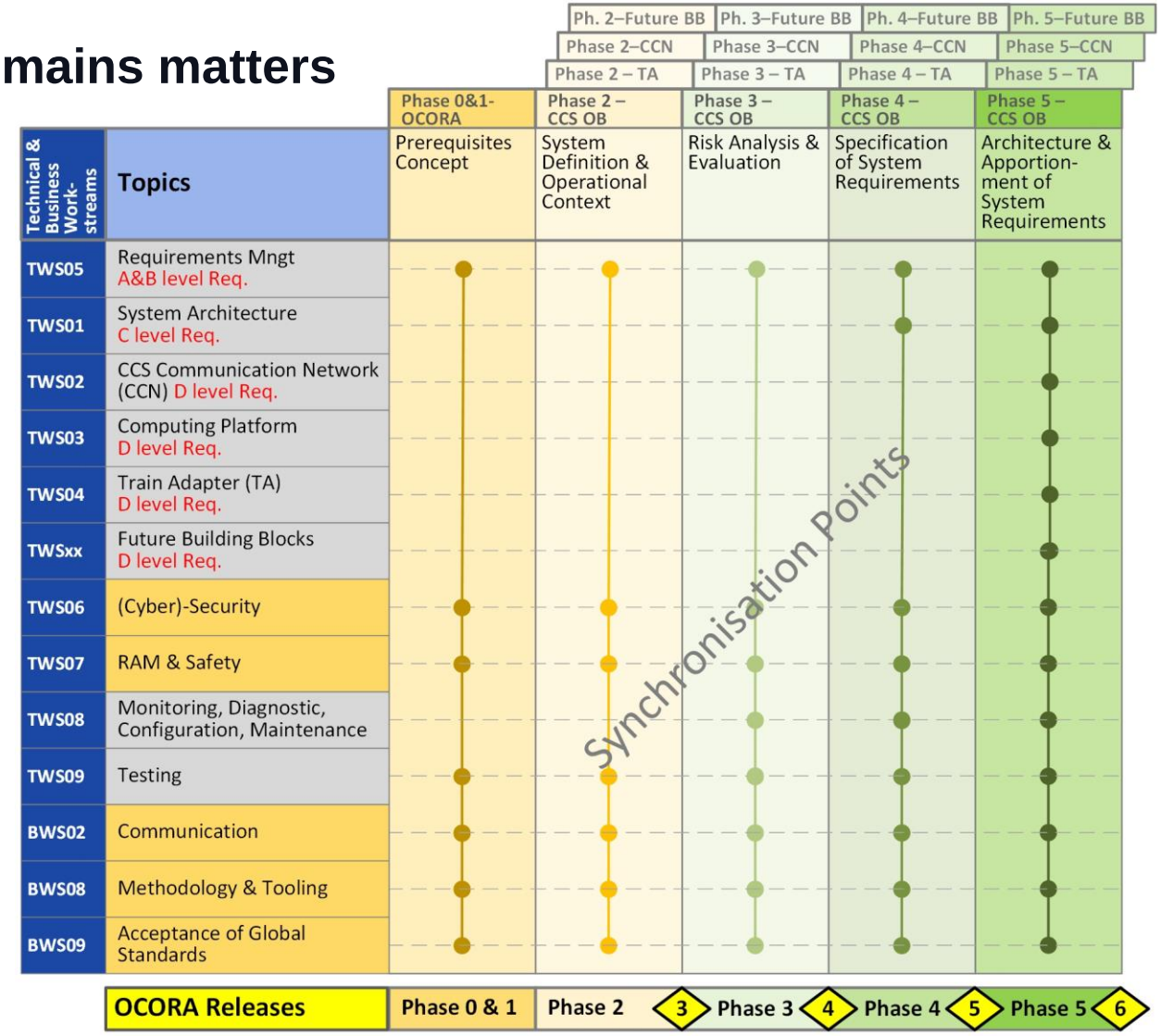
3.1.4 Synchronisation among all QPRAMSS domains is crucial

Synchronise results of all domains within each phase are essential to …

- ✓ ensures completeness of specifications.
- ✓ prevents conflicting requirements.
- ✓ masters complexity.
- ✓ guaranties consistent verifiability.

→ The provision of security requires permanent synchronisation of the QPRAMS- and the security-activities.

→ This is a Herculean task with the potential not to converge!



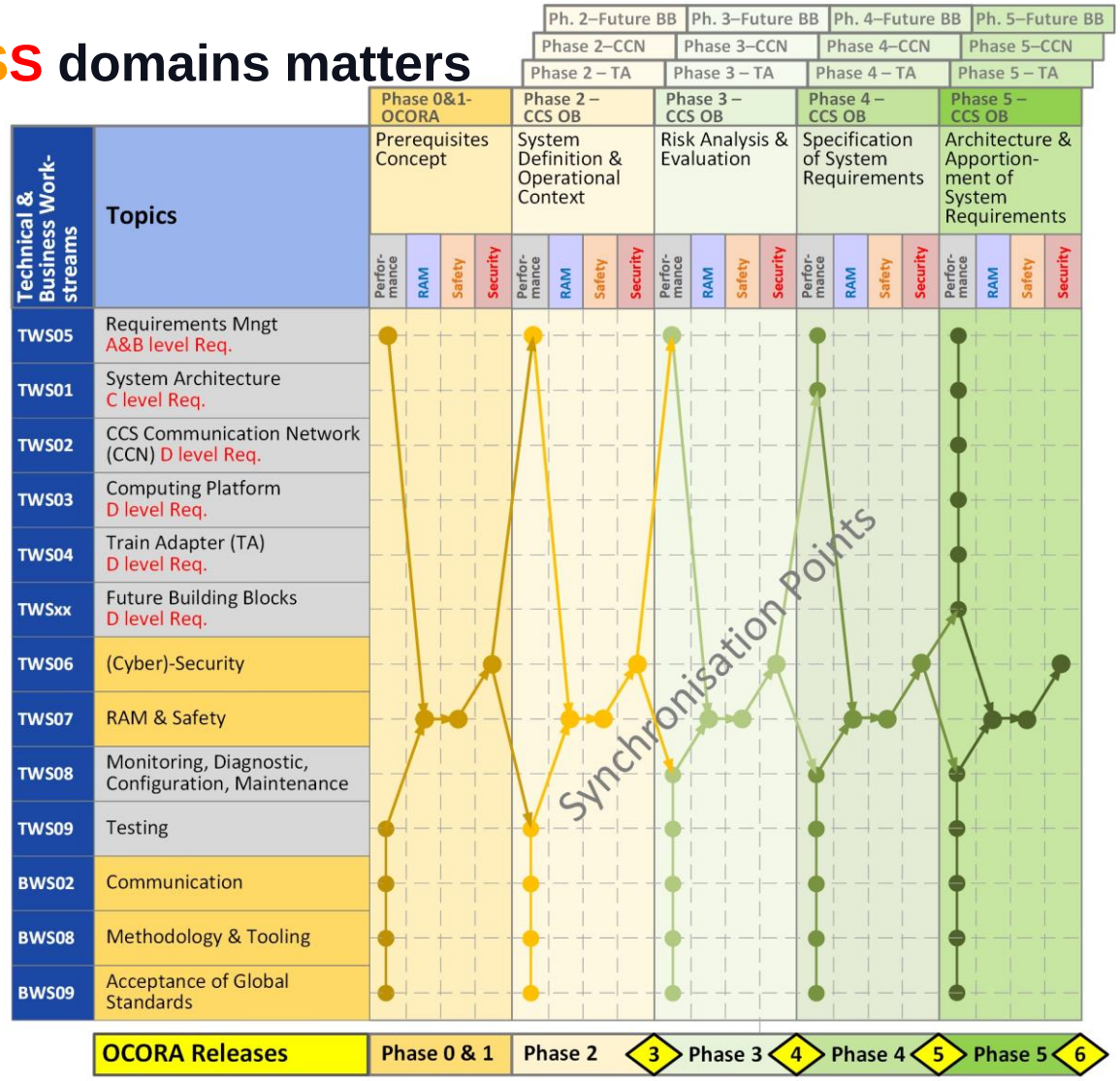
3. How to solve the conflicts

3.2 A common methodology among the QPRAMSS domains matters

There are risks that

- a lot of documents dealing with different aspects have to be reviewed – at least in phase 5 – by people of other TWSs and BWSs who might not have the specific experience and/or knowledge to detect possible conflicting requirements or objectives.
- if conflicts have been detected, there is a risk that no consensus without large compromises can be found.

→ As a solution, it is proposed to divide each phase into subsequent four sub-phases dedicated to **Performance**, **RAM**, **Safety** and **Security** in order to be fully synchronised.



3. How to solve the conflicts

3.3 Segregation of safety-related from non-safety-related functions

Content

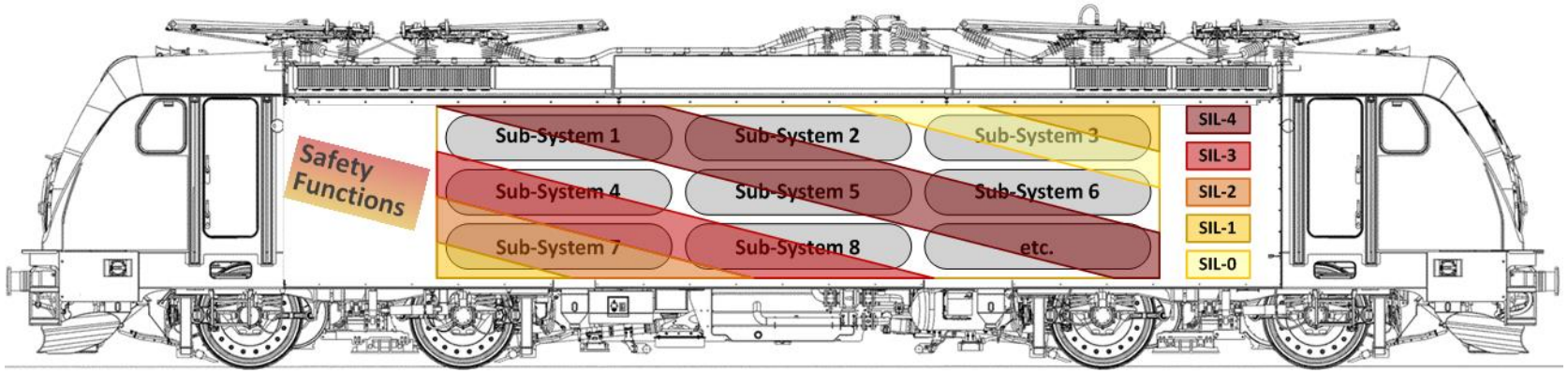
3. How to solve the conflicts

3.3 Segregation of safety-related from non-safety-related functions

3.3.1 Problem Statement: "Safety is likely to be part of every function"

Safety functions cannot be mapped 1:1 to the subsystem functions.

→ **All functions become safety-relevant.**



(from EN 50657: 2017)

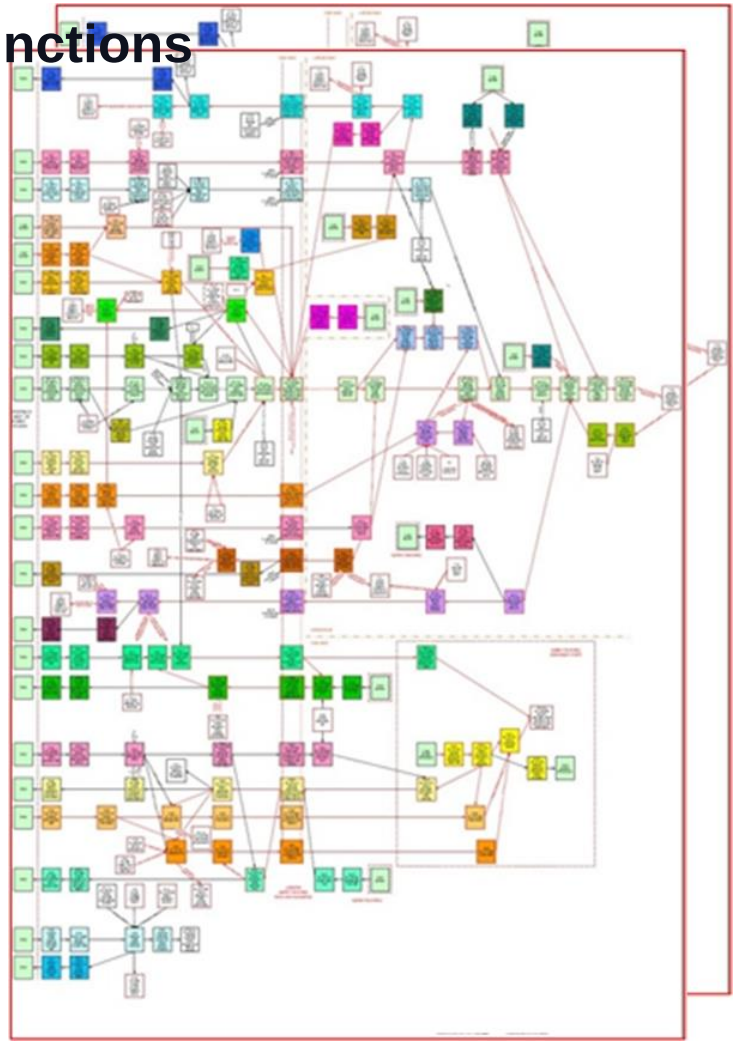
7.3.4.9 Where the software consists of components of different software integrity levels then all of the software components shall be treated as belonging to the highest of these levels unless there is evidence of sufficient segregation between higher software integrity level components and the lower software integrity level components. This evidence shall be recorded in the Software Architecture Specification.

3. How to solve the conflicts

3.3 Segregation of safety-related from non-safety-related functions

3.3.1 Problem Statement: "Safety is likely to be part of every function"

The functional chains of a TCMS form a three-dimensional spider web.



Modifications at one "place" inevitably have consequences at other "places".

Therefore, most modifications concern

1. safety-relevant functions.
2. require a new approval.

This costs a lot of time and money!

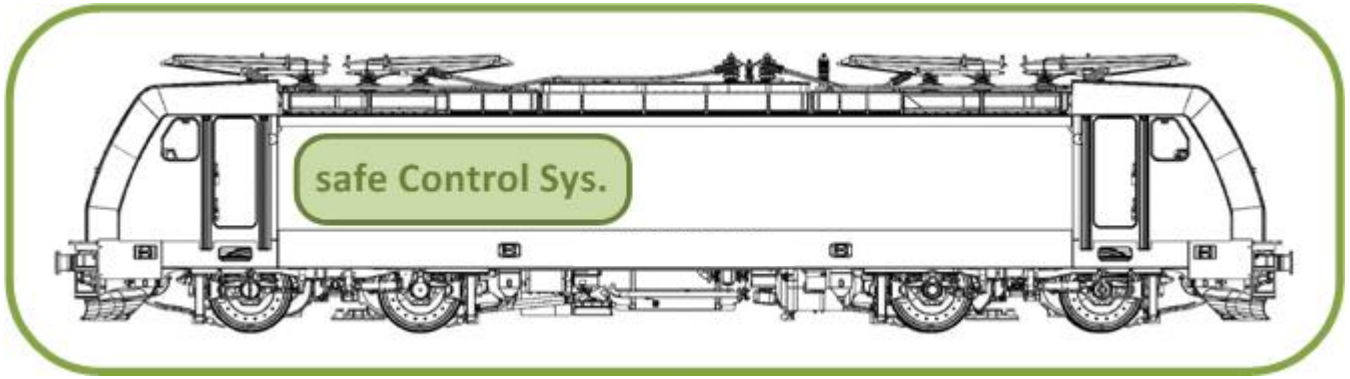
3. How to solve the conflicts

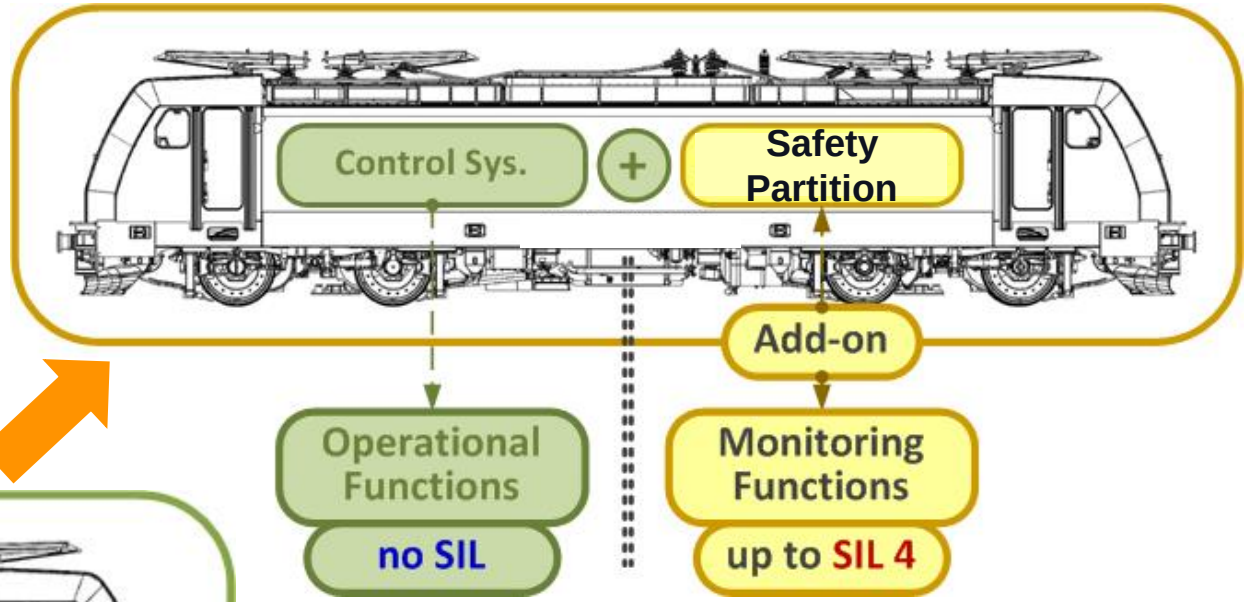
3.3 Segregation of safety-related from non-safety-related functions

3.3.2 The Solution: Segregation of safety-related from non-safety-related functions

Consider safety as an add-on component ...

- ... to flexibly address safety requirements of foreign countries (China, India, Brazil, ...)
- ... to keep pace with increasing European safety obligations.
- ... to master transition to new railway standards in Europe.





3. How to solve the conflicts

3.3 Segregation of safety-related from non-safety-related functions

3.3.2 The Solution: Segregation of safety-related from non-safety-related functions

D.72 Segregation

Aim

The aim is to sufficiently separate software components in order to prevent unwanted interactions or interference between them.

Description

Segregation is usually applied when a software is to implement safety-related functions of different software integrity levels. For software components of the same software integrity level no requirements on segregation are required. However, modifying one of those components would impact all the non-segregated components. This is why segregation may still be used for software components performing functions of the same software integrity level.

Separation between the safety-related functions of the different software integrity levels can be shown either by design techniques or by identifying measures which provide evidence that any violation of separation is safely controlled.

3. How to solve the conflicts

3.3 Segregation of safety-related from non-safety-related functions

3.3.3 The advantages are obvious

EN 50128: 2011, section 1.3

1.3 This European Standard is not relevant for software that has been identified as having no impact on safety, i.e. software of which failures cannot affect any identified safety functions.

EN 50129: 2017, section 5.3.1

Risk Assessment and Hazard Control processes, defined in EN 50126-1 and EN 50126-2, are always necessary in order to identify the appropriate degree of safety integrity for each particular function. This includes those cases where the analysis and assessment reveal that a function can be classified as not safety-related; once this conclusion has been reached, this safety standard ceases to be applicable.

→ **This saves considerable costs and effort.**

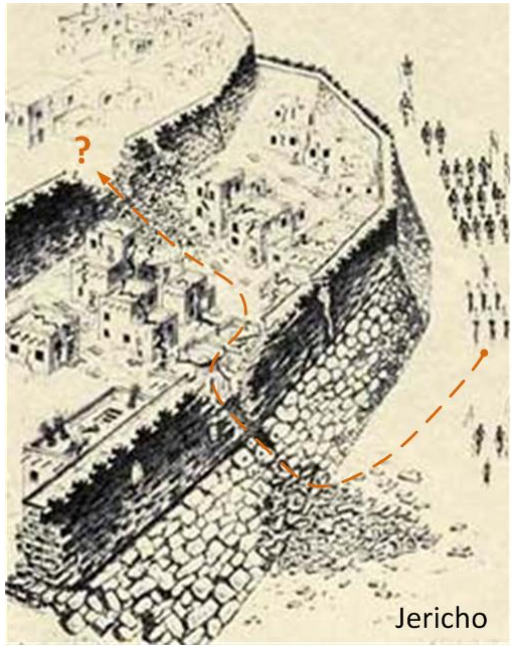
3. How to solve the conflicts

3.3 Segregation of safety-related from non-safety-related functions

3.3.4 Conclusion

- Segregation allows the separation of safety-related from non-safety-related functions such as security functions.
- Safety approvals remain permanently valid!
- Cybersecurity updates become independent of safety approvals.

→ **This solves the issue!**



- The European railway market is subject to strict regulations!
- The Europeans believe in the deterrent effect of these strict regulations!
- The concept of segregation of safety-related from non-safety-related functions provides an easy way to comply with the European's strict safety regulations – *also for non European suppliers* !
- **OCORA specification 547** states:

  "The OCORA reference architecture shall aim for a strict separation of functions with different quality attributes (safety, availability, performance)."

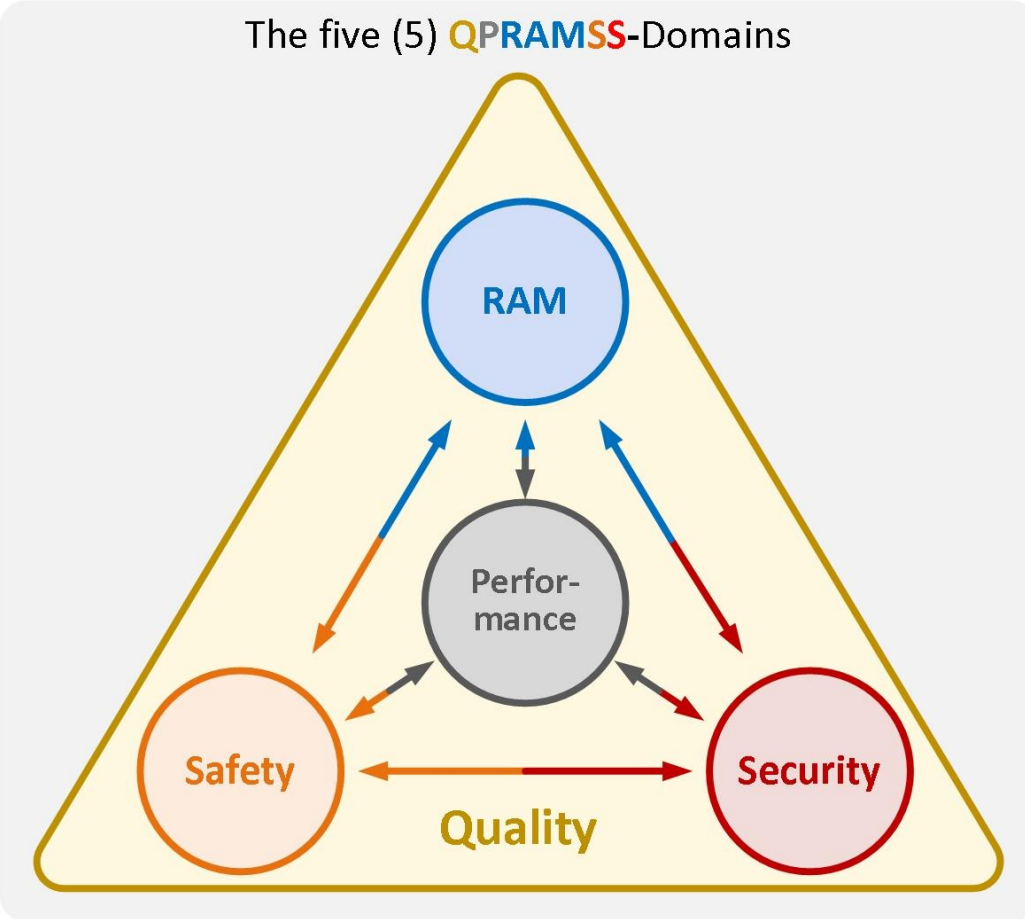
4. Summary

The conflicts among the QPRAMSS domains call for

1. a synchronisation of the QPRAMSS requirements within each development phase.
2. a common methodologies used by the QPRAMSS domains within each development phase.
3. the separation of
   - safety-related functions from
   - non-safety-related functions

   to prevent a spill-over of safety requirement onto functions related to

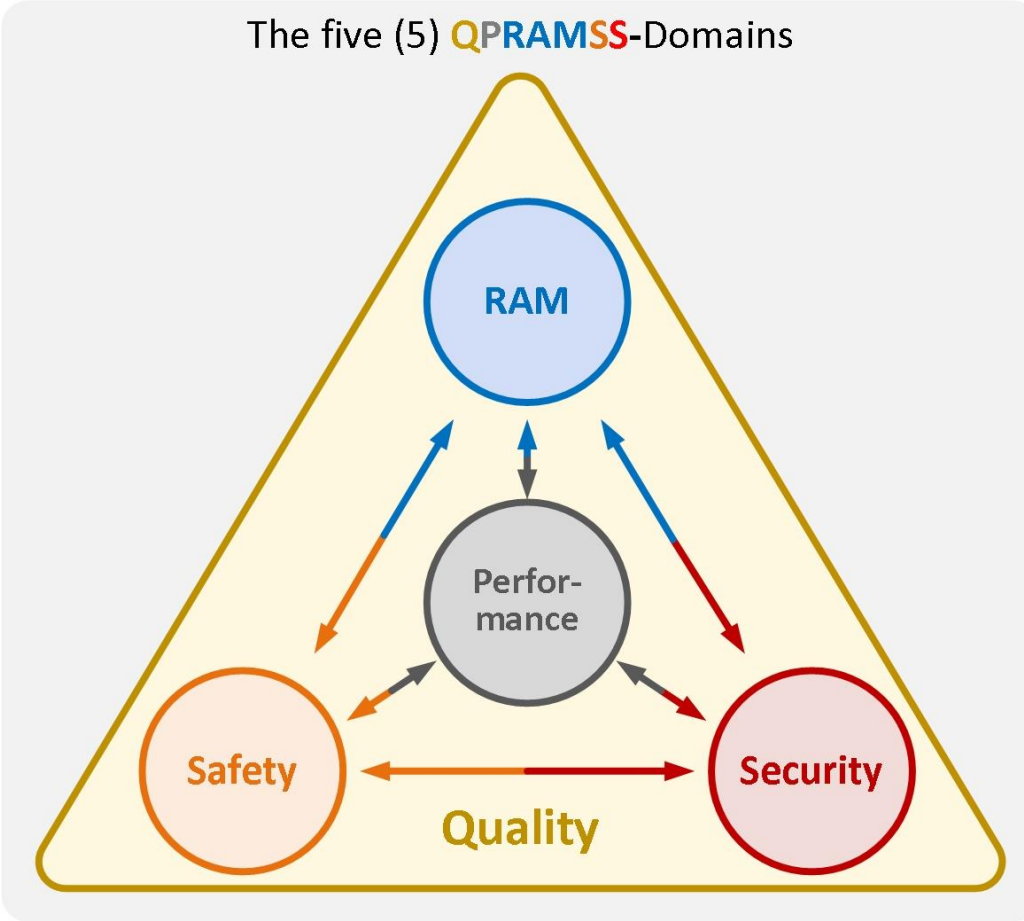
   → **RAM** including **Human Factors**,
   → **System Capabilities** and
   → **Security Functions**.



5. Follow-up

Q&A

- Do you support the QPRAMSS approach?
- Who will help with a strategy to let the other stakeholders support a common approach?



6. Annex – Examples illustrating the statements

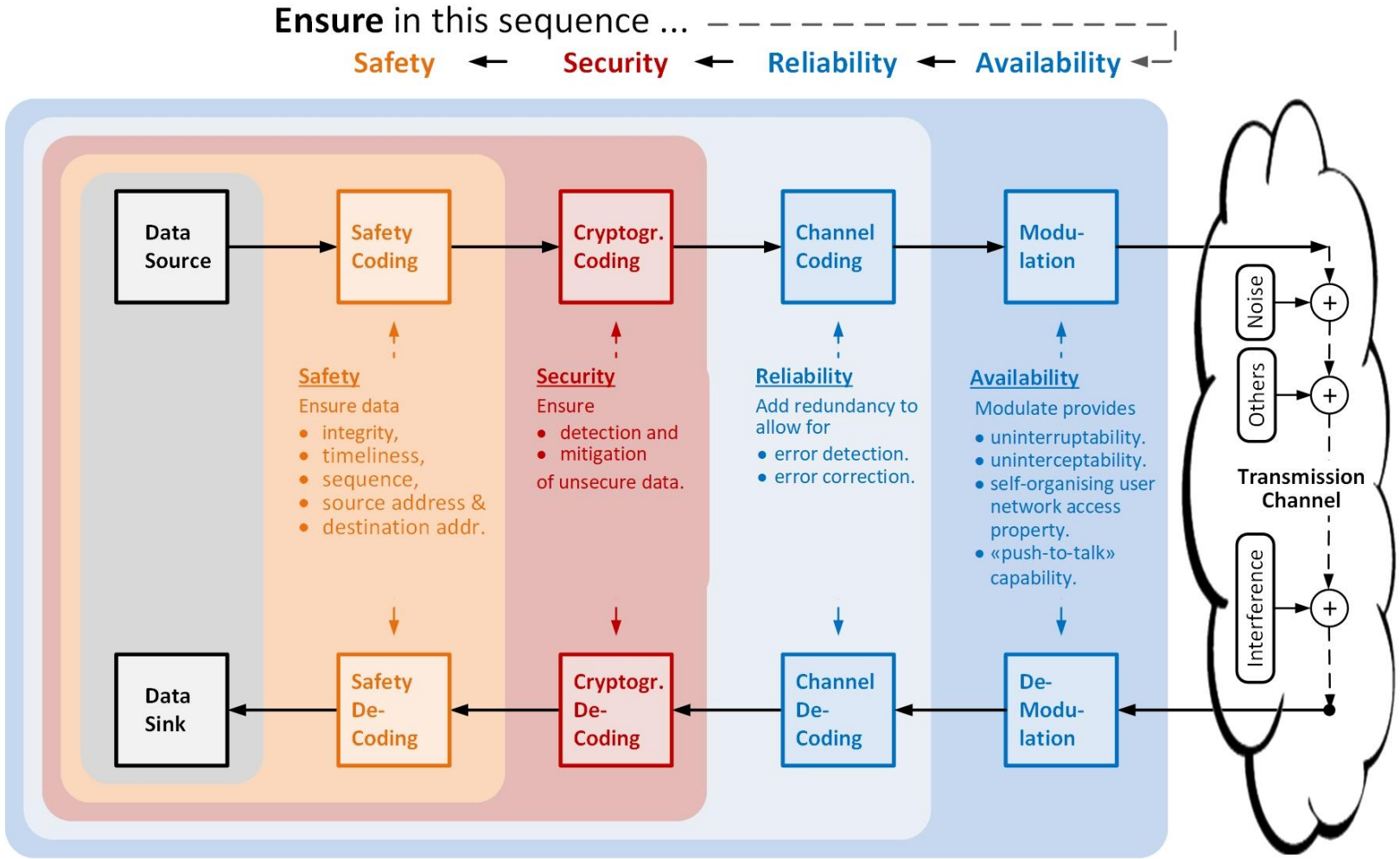
Annex

6. Annex – Examples illustrating the statements

6.1 Availability comes first!

Permanent **availability** requires

- **un-interruptability** (interference immunity).
- **un-interceptability**, (un-detectability).
- **self-organising** user network access property.
- **«push-to-talk»** capability.



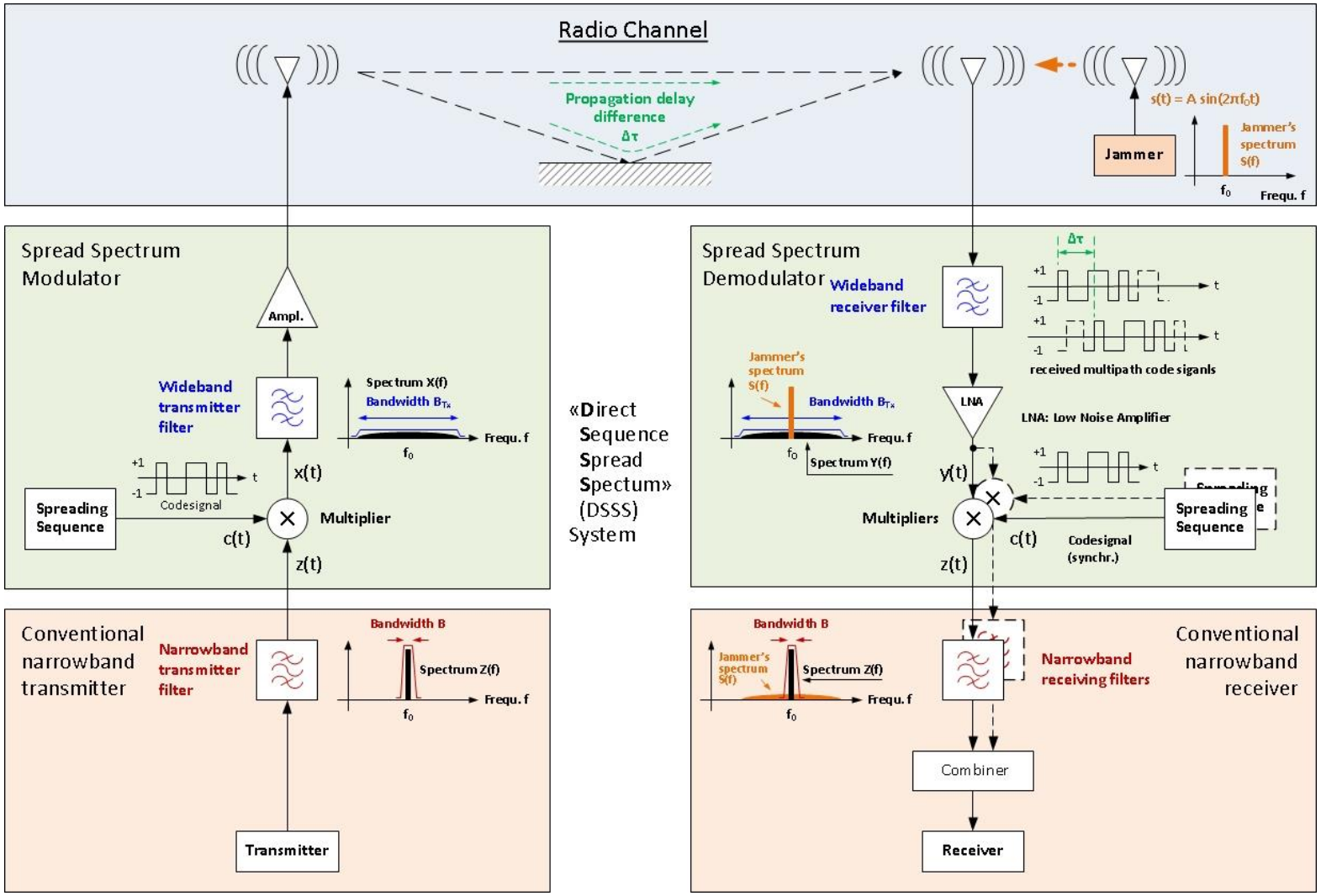
6. Annex – Examples illustrating the statements

6.1 Availability comes first!

Shannon-Hartley Theorem

C is the channel capacity, i.e. max. data rate of signal with power S a transmission channel of bandwidth B an additive white Gaussian Noise power N.

$$C = B \cdot ld\left(1 + \left(\frac{S}{N}\right)\right)$$

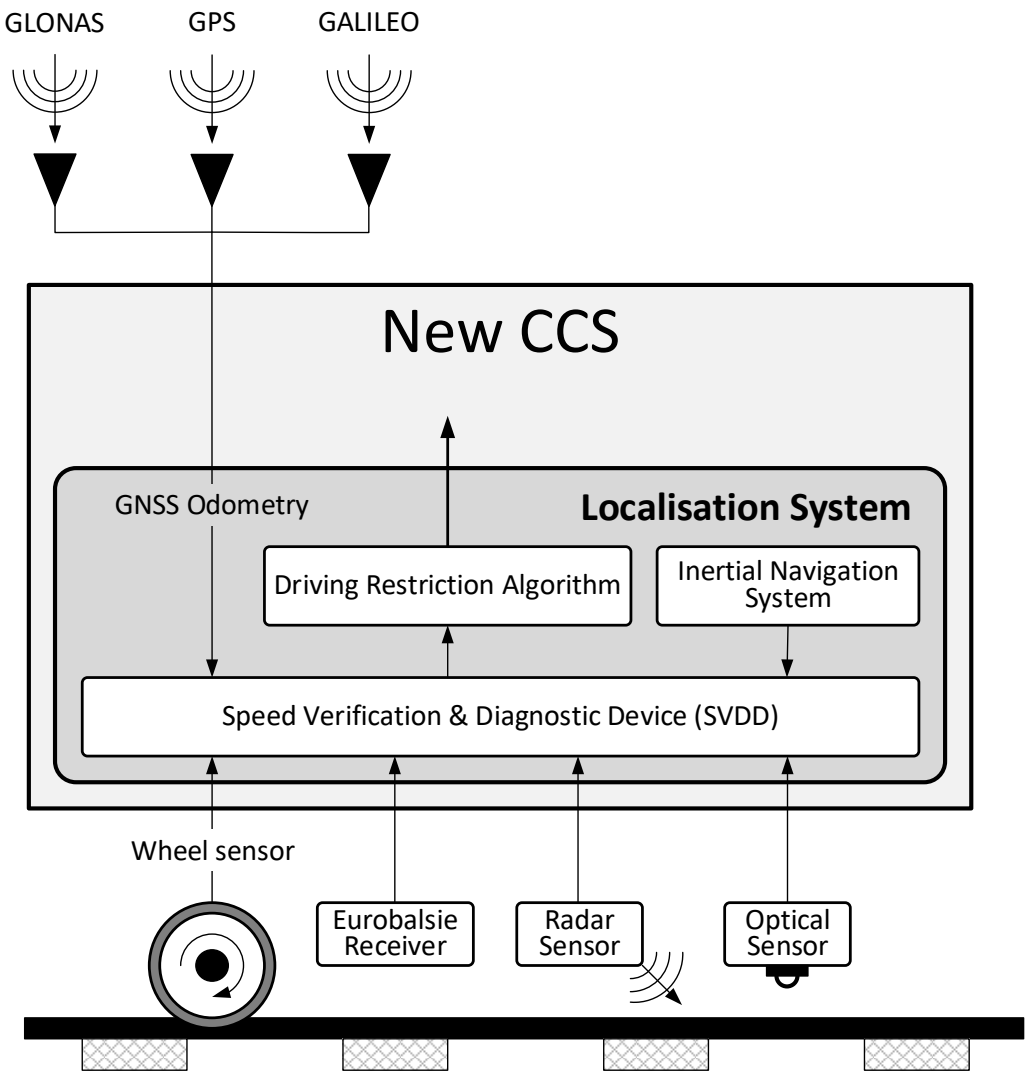


6. Annex – Examples illustrating the statements

6.2 Improved Availability through Smart Diagnostics

- "Speed Verification & Diagnostic Device" (SVDD) through checks of sensor signals plausibility.
- "Drive Restriction Algorithm" (DRA) indicates a safe drive restriction to the New CCS preventing vehicle standstill.

→ **This improves the availability of the localization system.**



6. Annex – Examples illustrating the statements

6.3 QPRAMSS approach coincides with Security approach

The procedures for the development of railway applications for the domains

- QPRAMS (EN 50126-1: 2012) &
- Security (TS 50701: 2022)

are the identical

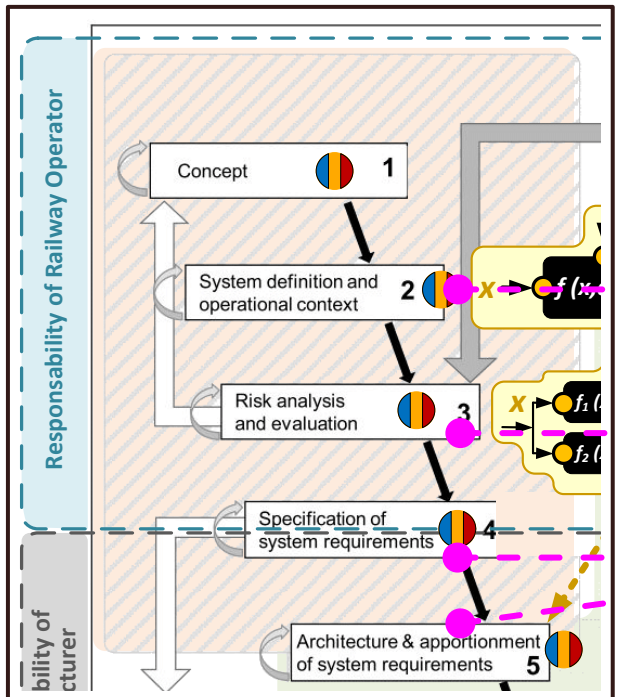


Figure 7 – The V-cycle representation (QPRAMS) (from **EN 50126-1: 2017**)

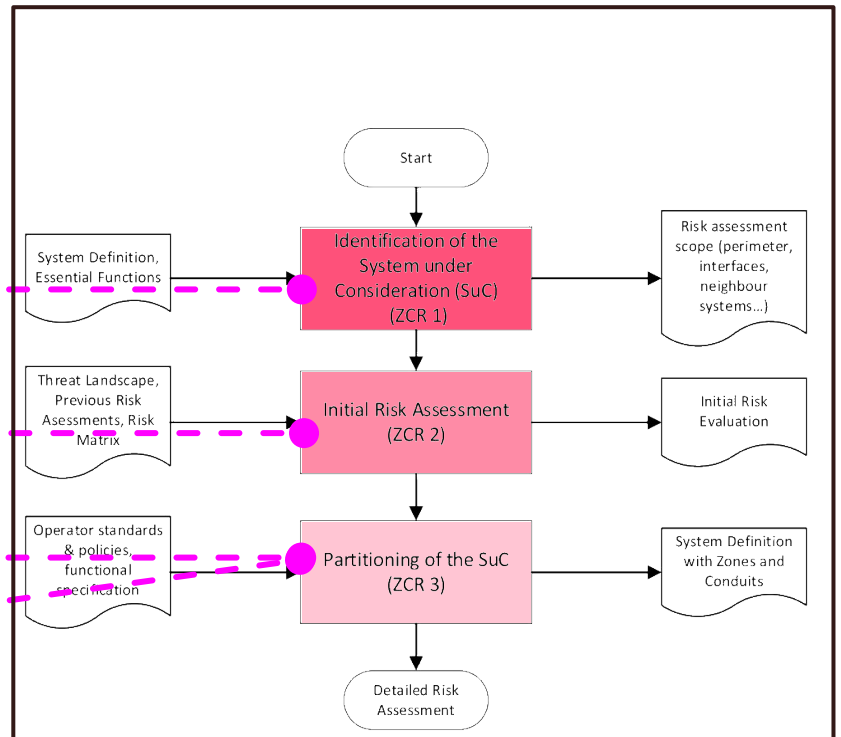


Figure 8 – Initial risk assessment flowchart for the Security Risk assessment (from **TS 50701: 2022**)

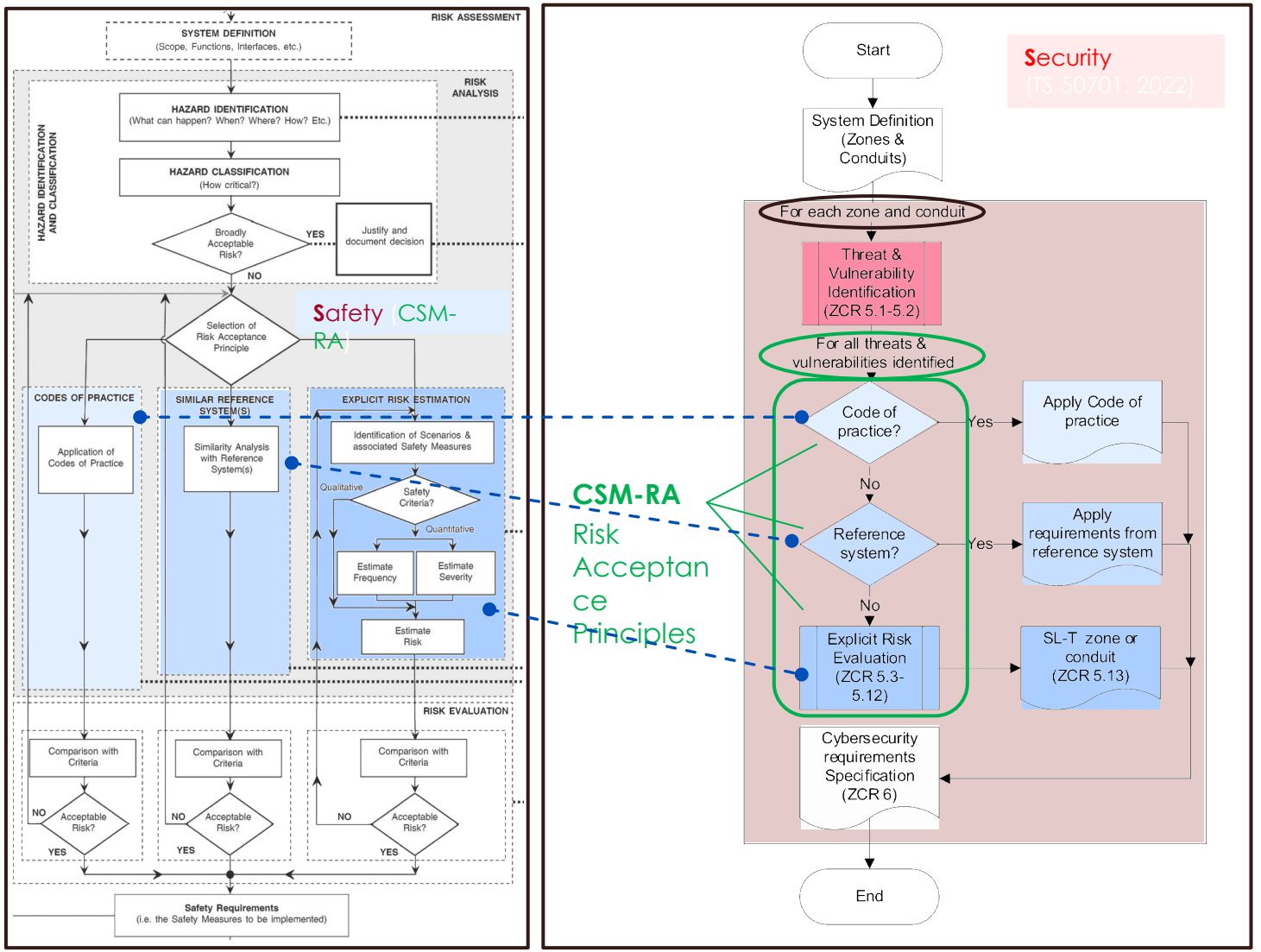
6. Annex – Examples illustrating the statements

6.3 QPRAMSS approach coincides with Security approach

The **Risk Acceptance Principles** for the development of railway applications for the domains

- Safety (CSM-RA: 2013) &
- Security (TS 50701: 2022)

are the identical!



6. Annex – Examples illustrating the statements

6.4 A common "methodology" among the QPRAMSS domains

Within OCORA, we need

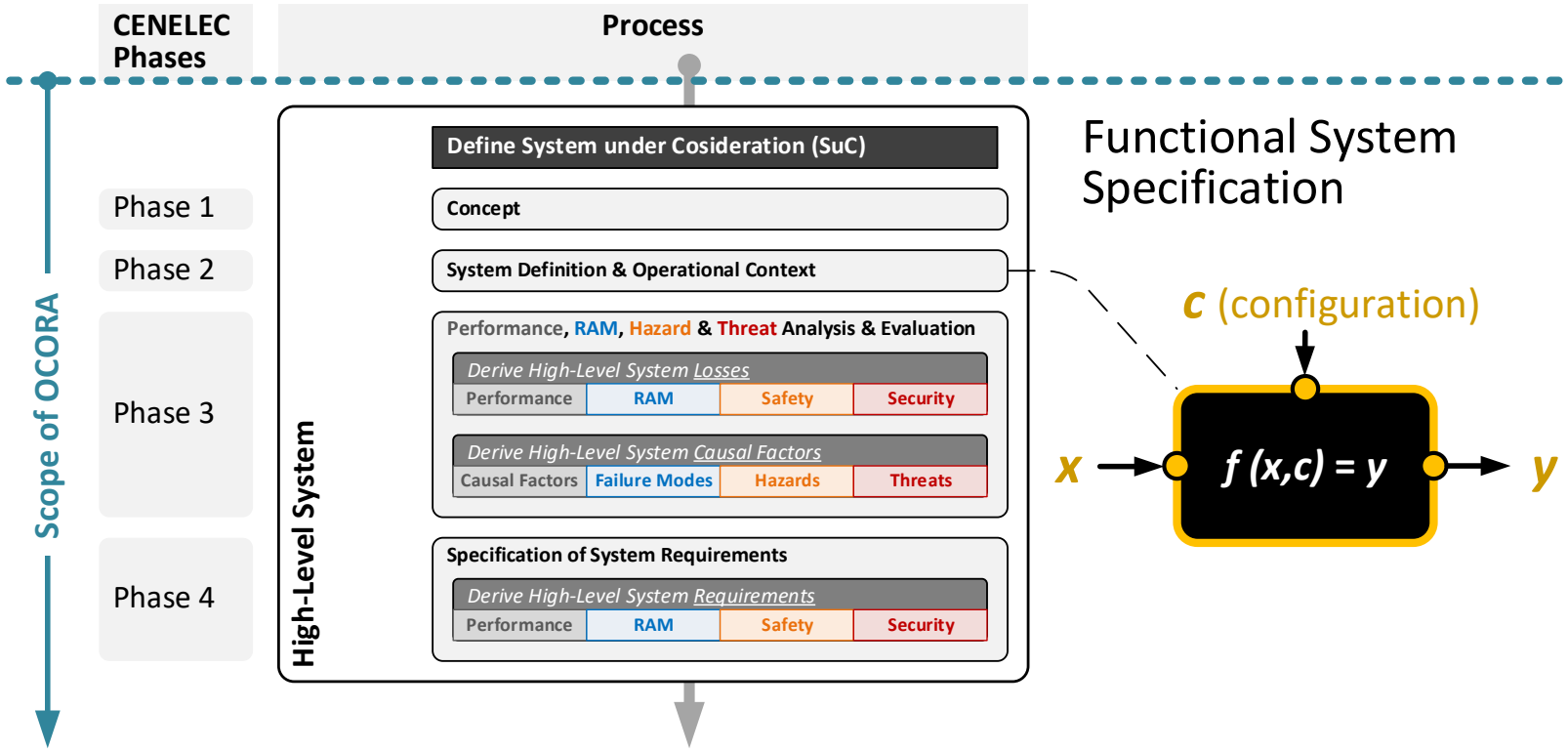
- *a top-down functional specification*

starting with

- a *functional system specification,*
- *where each domain uses its method(s) according to the relevant standards.*

→ **This must be adhered to by the domains.**

→ **How can we achieve this goal?**



6. Annex – Examples illustrating the statements

6.4 A common “methodology” among the QPRAMSS domains

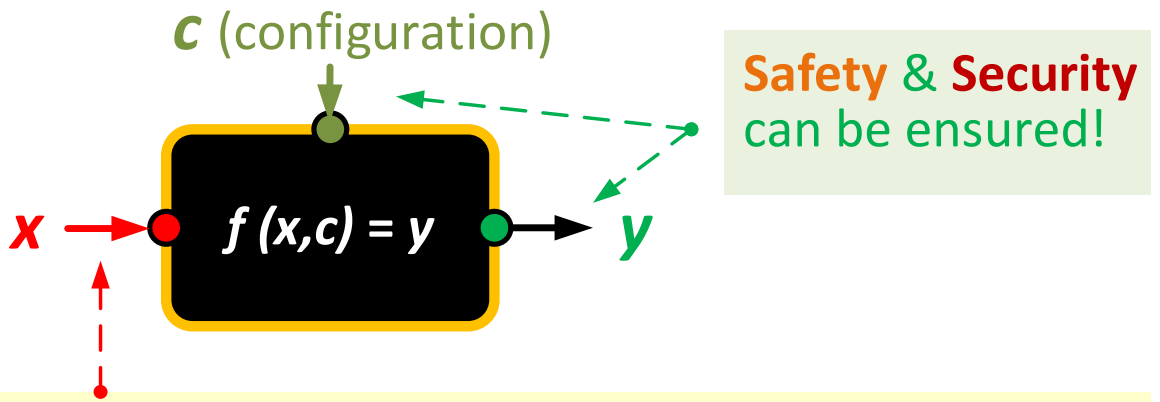
The system definition has it all.

How can we ensure Safety and Security for inputs to the “system” …

- … by Safety Related Application Conditions (SRACs)?
- … by Security Related Application Conditions (SecRACs)?

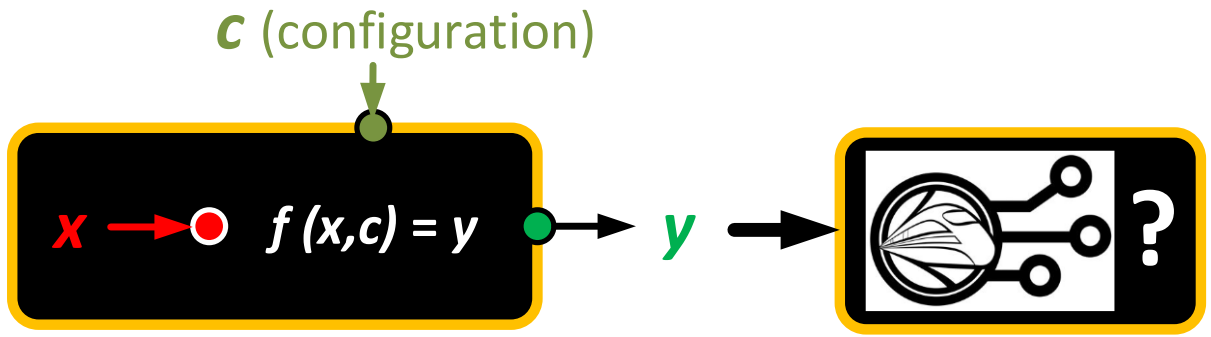
or

- does it imply that all inputs must be considered to be internal to the “system”?



How to ensure Safety & Security?
- with SRACs and SecRACs?

→ **Does this imply that port *x* will be internal?**



6. Annex – Examples illustrating the statements

6.5 Segregation of safety-related from non-safety-related functions

Overall documentary evidence

EUROPEAN STANDARD **EN 50129**

NORME EUROPÉENNE

EUROPÄISCHE NORM February 2003

ICS 93.100

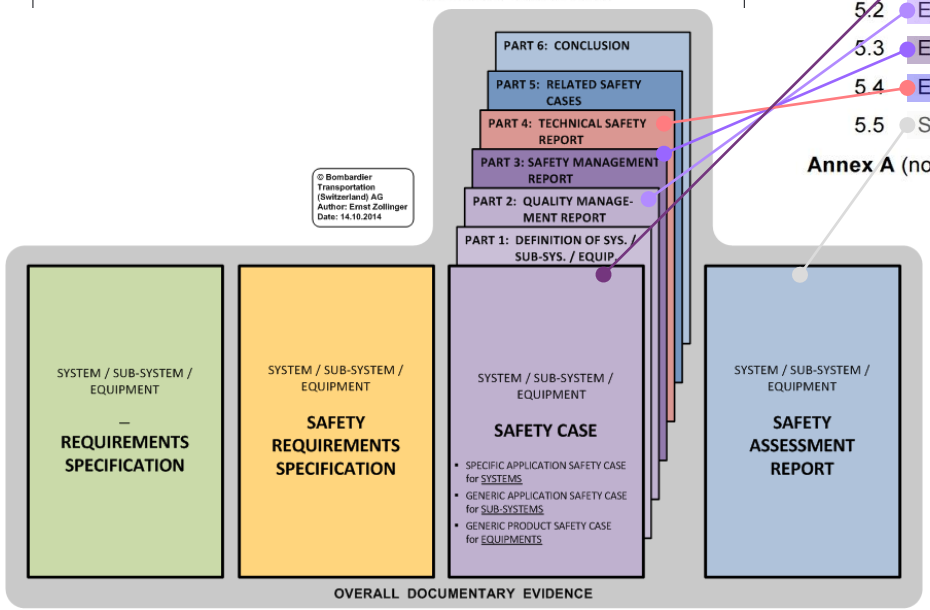
Supersedes ENV 50129:1998
Incorporating corrigendum May 2010

English version

Railway applications –
Communication, signalling and processing systems –
Safety related electronic systems for signalling

Applications ferroviaires –
Systèmes de signalisation,
de télécommunications et de traitement -
Systèmes électroniques de sécurité
pour la signalisation

Bahnanwendungen -
Telekommunikationstechnik,
Signaltechnik und
Datenverarbeitungssysteme -
Sicherheitsrelevante elektronische
Systeme für Signaltechnik



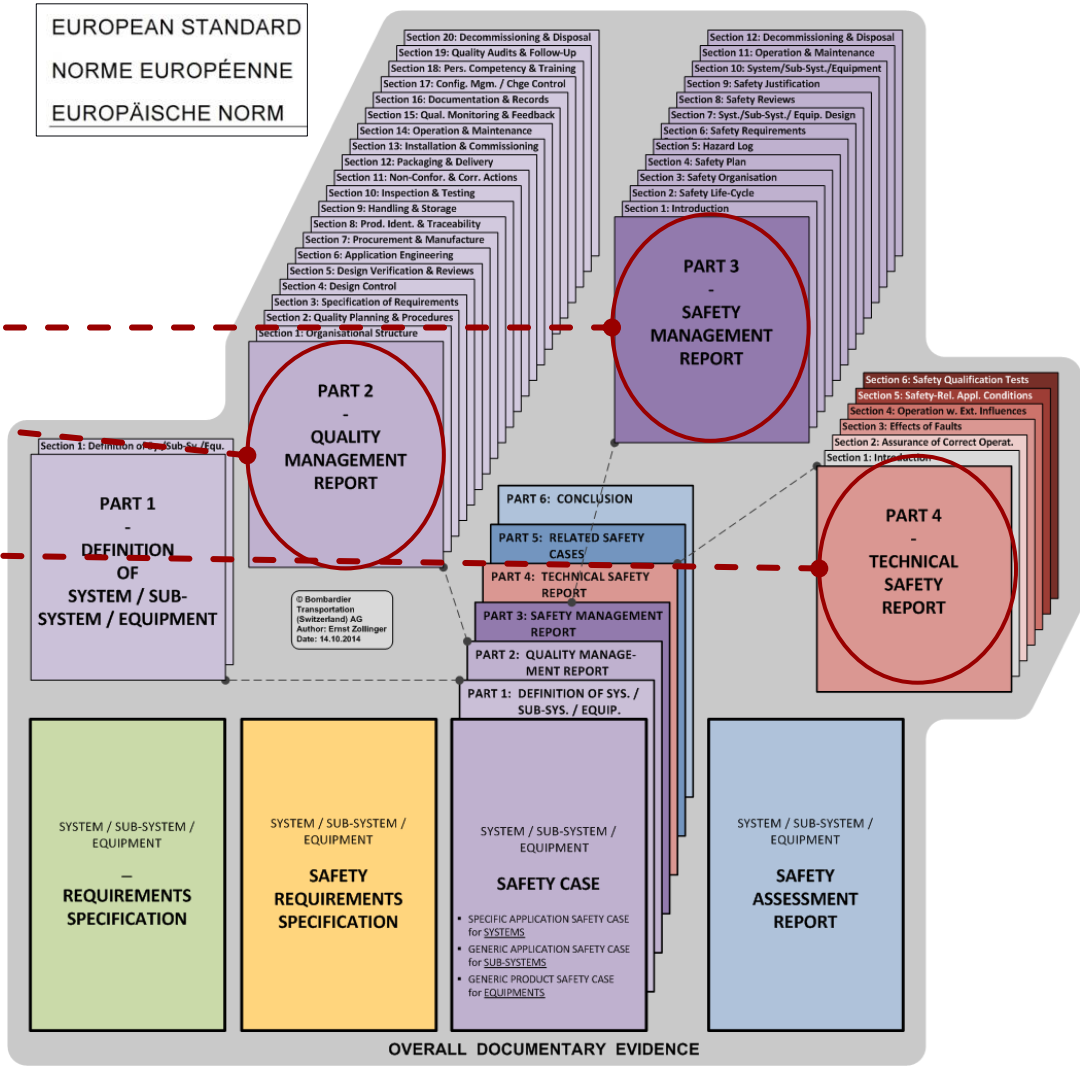
6. Annex – Examples illustrating the statements

6.5 Segregation of safety-related from non-safety-related functions

Safety Case contains …

- the Safety Management Report,
- the Quality Management Report, and
- the Technical Safety Report.

→ **The creation of this extensive documentation requires a considerable effort!**



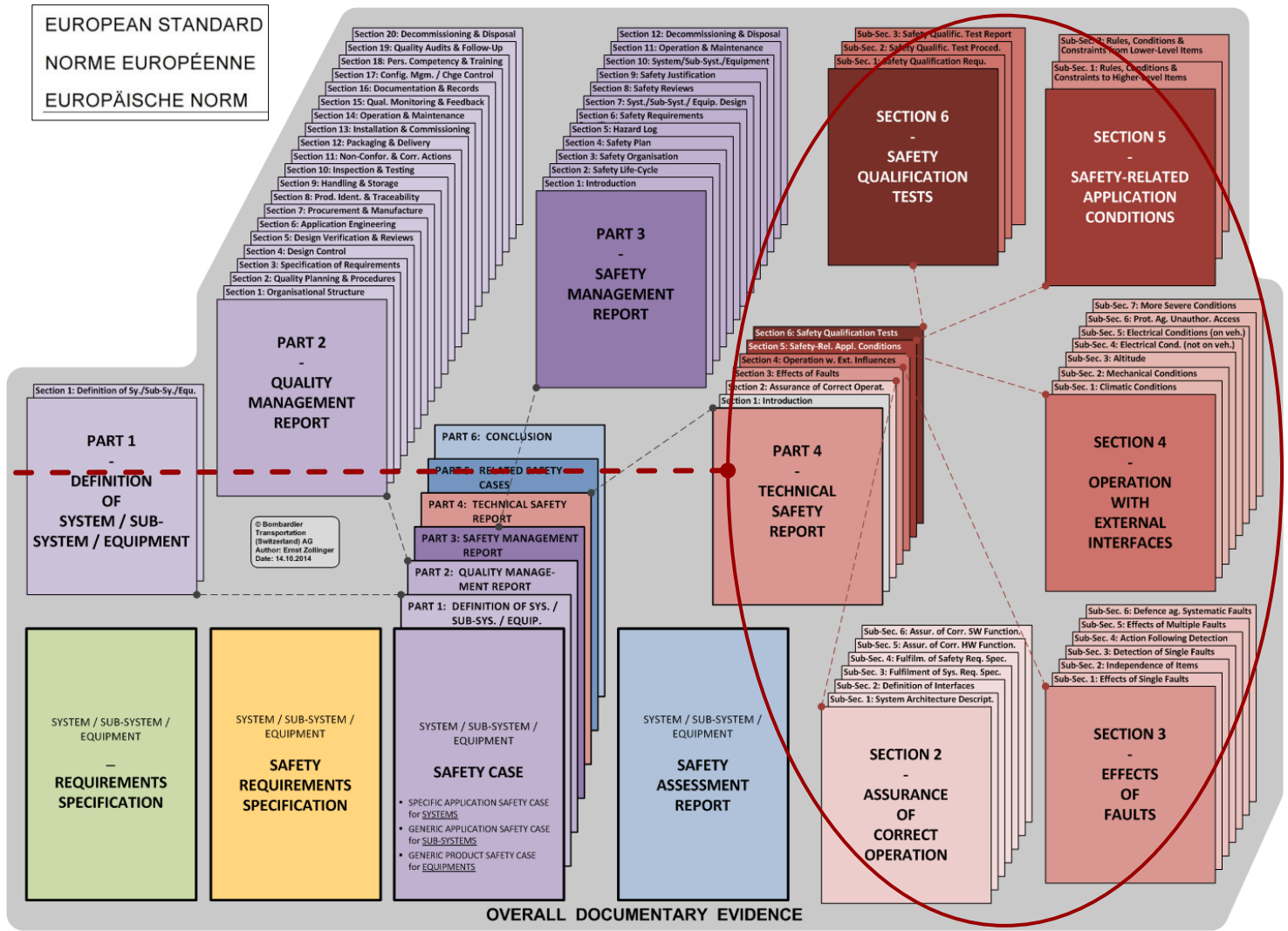
6. Annex – Examples illustrating the statements

6.5 Segregation of safety-related from non-safety-related functions

The "Technical Safety Report" is required for "each and every"

- Vehicle-release,
- SW-release,
- Sub-system, and
- Component

implementing or modifying at least one safety-relevant function.



6. Annex – Examples illustrating the statements

EUROPÄISCHE NORM **EN 50126-1**
EUROPEAN STANDARD
NORME EUROPÉENNE
Oktober 2017

ICS 29.280; 45.020 — Ersatz für EN 50126-1:1999

Deutsche Fassung

Bahnanwendungen –
Spezifikation und Nachweis von Zuverlässigkeit, Verfügbarkeit, Instandhaltbarkeit und Sicherheit (RAMS) –
Teil 1: Generischer RAMS-Prozess

EUROPÄISCHE NORM **EN 50126-2**
EUROPEAN STANDARD
NORME EUROPÉENNE
Oktober 2017

ICS 45.020 — Ersatz für CLC/TR 50126-2:2007

Deutsche Fassung

Bahnanwendungen –
Spezifikation und Nachweis von Zuverlässigkeit, Verfügbarkeit, Instandhaltbarkeit und Sicherheit (RAMS) –
Teil 2: Systembezogene Sicherheitsmethodik

6.5 Segregation of safety-related from non-safety-related functions

The European Railway Standards specify **"how"** a system should be designed to be **safe**.

→ They do not state **"what"** needs to be **safe**!

EUROPEAN STANDARD **EN 50128**
NORME EUROPÉENNE
EUROPÄISCHE NORM
June 2011

ICS 35.240.60; 45.020; 93.100 — Supersedes EN 50128:2001

English version

Railway applications -
Communication, signalling and processing systems -
Software for railway control and protection systems

EUROPEAN STANDARD **EN 50129**
NORME EUROPÉENNE
EUROPÄISCHE NORM
November 2018

ICS 93.100 — Supersedes CLC/TR 50451:2007, CLC/TR 50506-1:2007, CLC/TR 50506-2:2009, EN 50129:2003

English Version

Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signalling

6. Annex – Examples illustrating the statements

6.5 Segregation of safety-related from non-safety-related functions

Note

The Risk Assessment is the responsibility of the national Railway Duty Holder.

→ The Railway Duty Holder determine **"what"** must be safe!