RCA

Reference CCS Architecture

An initiative of the ERTMS users group and the EULYNX consortium

RCA– Architecture Overview

Document id: RCA.Doc.2

Version: Beta.1

Date: 26.8.2019

Table of contents

1. Introduction

1.1. Context

RCA (= reference CCS[1] architecture) is an initiative by the members of EUG[2] and EULYNX to define a harmonized architecture for the future railway CCS, with the main goal to substantially increase the performance / TCO[3] ratio of CCS in comparison with today's implementations.

The reasons and main background for the RCA initiative are explained in the RCA white paper, accessible here: https://ertms.be/workgroups/ccs_architecture and here https://www.eulynx.eu/index.php/home2/37-reference-ccs-architecture-white-paper.

The background for RCA includes (from the white paper):

- in 1989, the European Union, together with the railway organizations, decided to develop a standard European System for Automatic Train Protection, the European Rail Traffic Management System (ERTMS);
- today, the specifications of ERTMS are mature enough to start the large-scale implementation of ERTMS;
- both members of EUG and EULYNX believe that it is the right moment to try to define a common, simple reference CCS architecture to support the step from installed base to ERTMS and to increase the capacity of the existing network, improve the deployment speed and reduce life cycle costs for CCS.

The scope of the RCA work can be described by "command, control & signalling" or "everything needed to safely control movements and handle restrictions on the track". The architecture described by RCA focuses on the IMs perspective, while acknowledging that the command & control loops include the vehicles. For more detail, see the "Scope"-chapter in this document.

RCA in a nutshell:

- is founded on radio-based ERTMS/ETCS cab-signalling;
- defines standardized, evolvable interfaces for all major components of the future railway CCS;
- defines a clean target architecture without legacy systems, while providing a migration path[4];
- brings in new technology and ensures that technological progress from other sectors reaches the railways;
- RCA uses the existing ERTMS standards for the interfaces to the vehicle according to the TSI CCS;
- RCA integrates and builds on the standardization work of EULYNX.

1.2. Purpose of the document

This document should help:

- achieving the next level of understanding and commitment by the EUG and EULYNX members concerning RCA;
- organizing the next steps in the RCA development process (the RCA process is described separately), involving several working groups;
- continuing the discussion, allowing feedback, providing guidance with other stakeholders (industry, regulators, owners).

History:

- RCA alpha was published in February 2019 and generated valuable feedback from railways, industry, regulators, owners and the institutions.

---

[1] CCS = command, control & signalling

[2] EUG = ERTMS Users Group

[3] TCO = total cost of ownership i.e. including initial procurement and lifecycle costs, as well as direct and indirect costs.

[4] the migration path will, of course, include legacy systems in most cases.

- RCA Beta, published in August 2019, mainly deals with corrections as a result of misunderstandings and frequently asked questions. RCA Beta is released in the form of an updated set of documents from RCA Alpha and with a few additional “Beta chapters” on topics which had generated a lot of interest.

1.3. Structure of this document

2.	Purpose of the architecture	Why an architecture for CCS is needed?
3.	System scope	What topics are inside RCA, what topics are outside, but have to be considered?
4.	Description of RCA deliverables	What will the RCA initiative provide?
5.	High-level principles and design guidelines	Which architectural “logic” has been applied in decomposing the system?
6.	RCA architecture	What is the result of the system decomposition: interfaces, components, technical interfaces?
7.	Cross-cutting concerns, non-functional requirements	How are aspects impacting several interfaces / components handled?
8.	Overview RCA usage scenarios	How can RCA be used? What differences / variances in application of RCA are foreseen?

1.4. Other relevant documents

- **RCA white paper**: the rationale for starting RCA (published in august 2018): available here https://ertms.be/workgroups/ccs_architecture and here https://www.eulynx.eu/index.php/home2/37-reference-ccs-architecture-white-paper.
- **RCA FAQ** (frequently asked questions): a short summary of important question regarding RCA.
- **RCA Alpha Process Overview**: how the RCA group works to prepare, maintain and bring RCA to market (to be published along with this paper).
- Recommended read: Command and Control 4.0 by Josef Doppelbauer (ERA): https://www.era.europa.eu/sites/default/files/library/docs/command_and_control_en.pdf
- Additional chapters on specific topics (separate documents, part of the RCA Beta release):
    - **Modular Safety**
    - **Platform Independence**
    - **Capacity effects**
    - **Architectural approach and “Systems-of-systems” view**

2. Purpose of the architecture

RCA provides an architecture based on harmonized requirements. By definition, an architecture describes the decomposition of a system into components and how these components are linked with each other. The main purpose of RCA is to define the interfaces of important (procurable) components of a CCS. RCA therefore focuses on well-defined interface specifications.

2.1. Overarching goals supported by RCA

The RCA enables IMs to modernize their systems by going beyond the currently available ETCS L2 implementations. According to the RCA white paper the goals of RCA are:

- "Low LCC": The RCA shall allow the implementation and operation of a trackside CCS with a low number of standardized system components (simplified architecture), with the ability to procure these standardized system components in a competitive market with automated processes (this requires integrated data management);
- "A single modular framework": The RCA shall specify the generic design of trackside CCS with ability to select configurations within the modular framework to facilitate migration strategies and adaptations to specific business challenges;
- "Migratability": Low cost solutions for interfacing to existing systems in the environment around the RCA, protecting existing investments;
- "Adaptability": The RCA shall be a generic design, allowing different levels of requirements concerning safety, costs / LCC, availability, performance and other non-functional requirements by configurational parameters or component selection (plug & play). Software / hardware adaptions of the individual components are to be avoided[5];
- "Safe Investment": The interface quality of the RCA (in general Form, Fit, Functional Interface Specifications - FFFIS) shall aim to avoid incompatibilities to future developments by using well prepared mechanisms for upwards and downwards compatibility. Interfaces shall be defined on a formalized basis (and with real and early prepared reference systems for testing) for all communication layers to avoid vendor specific deviations.

According to a brief survey[6] conducted among some participants of the RCA group, additional goals (such as higher capacity, increased safety, higher reliability, faster implementation speed) are also important for at least some modernization projects. We will therefore enlarge the list of goals supported by RCA. In addition to the "wished-for" goals it is clear, that an architecture like RCA will have to address issues such as cyber security, performance and others which will be treated in the category of non-functional requirements.

2.2. Targets for the RCA itself

The RCA supports the above-mentioned goals by providing the following enablers (excerpt from the white paper):

- Significant reductions in the number and types of trackside assets;
- Optimization of operational processes[7];
- Implementation of the game changers[8];
- Standardized and automated data preparation for open engineering and testing;

[5] This refers to IM-specific adaptions (variance). The RCA will support different target configurations (e.g. using or not using a component or using a fully vs a semi-automated implementation of a component).

[6] We plan to make that survey broader, as part of the RCA Beta feedback process.

[7] See also chapter 4.7 Relation of RCA to "harmonized operational processes".

[8] The ERTMS game changers include new technology such as ATO, satellite-based train-localization or a 4G / 5G communication infrastructure.

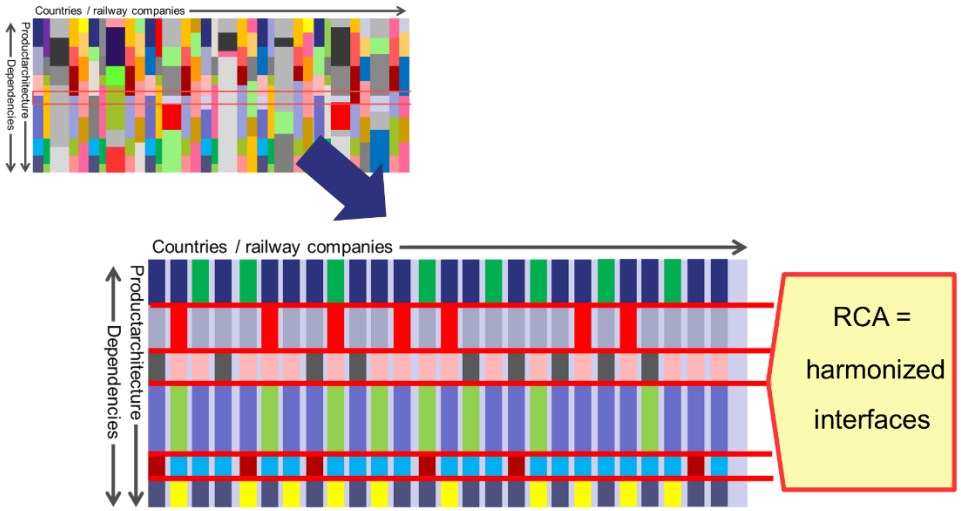
- Cost effective implementation and maintenance features for every interface and component (e.g. simple integration safety case by an automated impact analysis, remote updateability);
- Competitive procurement based on exchangeability of components;
- Open specification, standardized interfaces and use of mature industry standards;
- Components allowing independent industrial developments and deployments in a whole infrastructure or fleet ('components of the shelf')[9].

According to a brief survey[10] conducted among some participants of the RCA group, additional targets, such as "simple integration of / into safety case" are also important and will be incorporated.

2.3. How RCA will be applied by railways and by industry

The purpose of the RCA is, that the architectural requirements concerning the main interfaces of the CCS architecture published in tenders are harmonized in detail and used in every tender in the same way. This includes all layers in the interfaces including the communication between applications, security and safety protocols, transport layers, carriers and standardized interfaces between applications and operating systems or hardware representation layers.

For this, the RCA defines a coherent set of references to specifications, that make up the CCS architecture for the future railway system. These specifications may be coming from the TSI, from research projects, from industry projects, from railway projects or from other types of cooperation or bodies. Some of the important conditions for becoming an RCA-accepted interface specification are, that the interface is openly usable, well-defined, stable and has mechanisms for upwards as well as downwards compatibility. The components between these RCA interfaces are considered as "black boxes" coming from the product market. The way of specifying an RCA interface will allow use of basic functionalities (always available, always the same), or to use additional functionalities if they can be handled on both sides (runtime negotiation) and are accepted for the RCA architecture. This assures compatibility on one side but also allows product developers to generate USPs (unique selling propositions) for their products. Railways (IMs) committed to RCA will use the RCA architecture and its specifications in the planning and procurement ("tender templates") of their modernization projects. Suppliers can use RCA to guide their product development roadmaps. The following diagram shows the basic idea, going from heterogenous architecture (and, as a consequence, components) to an overall CCS architecture harmonized by important interface specifications.



Note: The RCA development process is described in the document "RCA Alpha Process Overview".

[9] This includes supporting independent lifecycles of components, such that single components can be updated or replaced.

[10] We plan to make that survey broader, as part of the RCA Beta feedback process.

2.4. Why is RCA useful?

RCA is important for the future of the railway system because:

- by using shared specifications, we create a more attractive system / component market with larger scales, we reduce the integration effort and lifecycle complexity;
- by designing a system with built-in upgradability, we allow the railways to integrate innovations faster;
- this increased performance- / TCO-ratio will help increase the competitiveness of the railway sector and, in the longer run, the market share of rail.

This is beneficial for the railway customers, IMs, RUs and suppliers.

2.5. Role of existing standards / architectures

RCA will use already elaborated standards:

- ERTMS: RCA is based on the existing ERTMS architecture and re-uses the TSIs. To fully use RCA, some change requests to the TSI will be necessary, which will be submitted in due form and will go through the official change process (in preparation);
- FRMCS: RCA is based on the availability of the successor to GSM-R;
- EULYNX: RCA re-uses the EULYNX-defined interfaces (and object controllers) for track-side objects. RCA will also re-use other EULYNX elements (such as modelling standards, and other concepts);

RCA has to take into account the existing standards / procedures for designing, building, integrating systems in a railway environment (CENELEC, etc.).

2.6. Business case

Concrete (project-related) business cases are the responsibility of individual railways. This document provides a short overview over the “mechanics” of an RCA business case i.e. how solution elements of the RCA contribute to a valuable business outcome. On request a simple business case template can be made available.

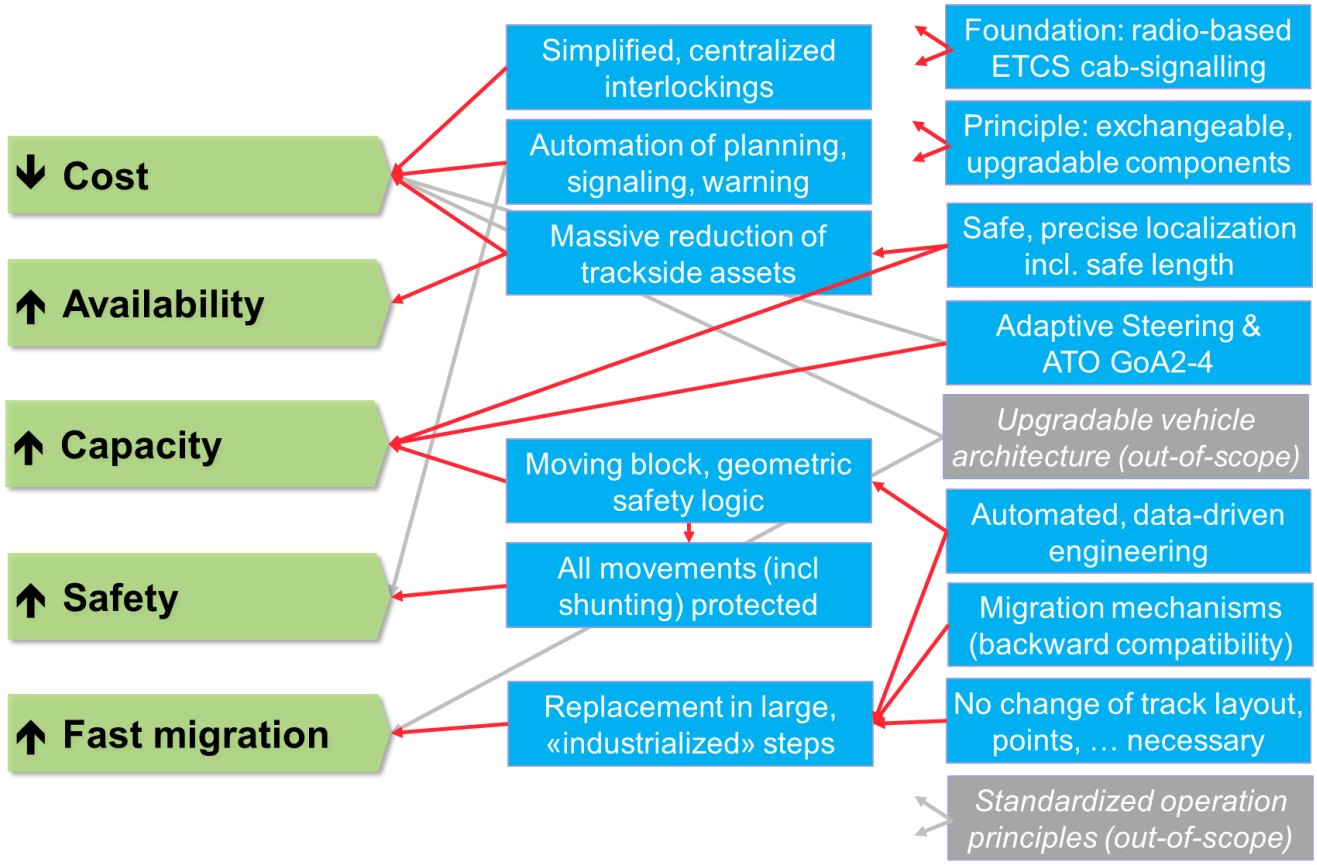
Many of these solution elements or effects are based on known mechanisms such as moving block, ATO etc. RCA does not pretend to invent these mechanisms “on top”, but provides a coherent, upgradable architecture to fully use those mechanisms.

2.6.1. Business case aspects lead to architectural requirements

The following picture shows the main aspects of the business case of the future railway system from the RCA perspective. They lead to special architectural requirements for the RCA. Lifecycle cost dominate the total cost of ownership of a digitized landscape of applications. Therefore, the architectural requirements get a much higher importance in the tender.

The diagram shows 5 important goal dimensions for the business case: cost, availability, capacity, safety and fast migration. The diagram also shows, how the features of the RCA contribute to the goals. The features are described throughout this document. Not all IM will weigh the importance of these goals identically, RCA allows targeting differing priorities.

For readability: red arrows show substantial contributions, grey arrows show 2nd order contributions.



Important elements of the “business case mechanics” include (not exhaustive):

- Fast migration may be an important goal because of obsolescence problems and / or to achieve the business case quickly and / or to reduce complex “patchwork” situations. Fast migration is driven by several migration mechanisms in the architecture and by relying on industrial processes (important role of data preparation) for the migration;
- Cost reduction is driven by the reduction of track-side assets and the increased automation of tasks such as planning, dispatching or driving (for ATO GoA3 / 4). [Planning (long-term and short-term) systems are out-of-scope for RCA];
- Availability is driven through reduction of track-side assets;
- Capacity is driven by moving block, higher localization resolution, geometric safety logic and more precise control of vehicles through ATO;
- Safety (reducing collision risks) is driven through offering full protection for shunting and by including other “objects” such as track workers or heavy machines in the system.

See also the document “RCA Beta – Chapter on capacity effects” for more details.

2.6.2. Business case examples

A simplified business case calculator is available on request. Several companies have performed an analysis but not disclosed their results yet. SBB has made public some important aspects of their case, as shown in the following table.

RCA Goal category	Example SBB (program smartrail 4.0)
Cost reduction	450 Mio CHF /year, through ▪ reduction of 70% of trackside assets ▪ increased automation for activities such as engineering, planning, dispatching and warning (some of this functionality is TMS related and out-of-scope for RCA) The underlying network has a length of 3'200 km, 10'700 trains a day, train density of 157 trains/km/day.
Increase capacity	Increase of 15-30% to be used for increased traffic volumes while avoiding / reducing investments in tracks / switches.
Increase availability / reliability / operational stability	Increase of punctuality through • 50% reduction of track-side device failures (through asset reduction) • time-table stabilization (through increased capacity)
Reduce collisions / Increase safety	Increase safety for shunting and construction site through localization and "full supervision" all the time.
Energy savings / ecological footprint	Savings through avoided braking. For SBB included in the cost reduction case. In other countries this may also be expressed as CO2 reductions.
Speed of rollout	Industrialized rollout: Accelerate commissioning and putting into operation by 50%. Necessary to handle upcoming replacement peaks. Also contributes to cost reduction (above).
Customer information	Precise customer information based on more accurate information on trains position and speed. This is available with a ETCS L2 implementation. Not explicitly part of SBB's business case.
Reduces journey times	Acceleration effects are supposed to be low. Fresh buffer times are invested in stability of the network. Not explicitly part of SBB's business case.
On-demand traffic	Made possible by higher automation on all levels. Driven by automated planning / scheduling in the TMS (out-of-scope for RCA). Not explicitly part of SBB's business case.

3. System scope

A first, very broad scope statement for an RCA-based system is:

> ***Everything (processes, systems, people) needed to safely and efficiently control movements and restrictions on the tracks***.

Out-of-scope of RCA are the following functions / systems:

- the long-term planning, the booking and financial book-keeping of slots for the RUs;
- the short-term planning;
- the planning of resources like people, trains or construction equipment;
- customer information (customers of the RUs);
- trackside assets like point machines, train detection devices, etc.;
- the technical implementation of CCS equipment on the vehicles, see also 8.2.3 “Vehicle migration / OCORA”; however the functional CCS behaviour of the vehicles (interoperability) is in-scope for RCA;
- other devices such as warning devices for track workers;
- braking curves are an important driver for safety and capacity, RCA does not work on this topic but welcomes initiatives to optimize braking curves;
- implementation of communication or computing infrastructure (e.g. datacentre design), except the interfaces of an CCS application to its runtime environment.

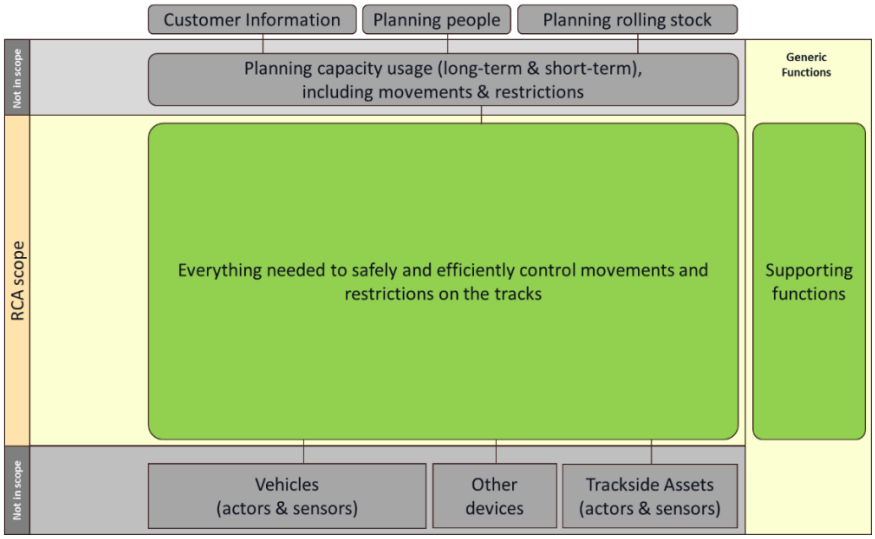
Reasons for excluding functions / systems:

- systems not “belonging” to the IM-role (such as CCS equipment on trains) have been excluded from RCA, because they cannot be directly influenced by the IMs. At the same time, it is clear, that a complete solution must include the vehicles;
- existing (trackside) devices have been excluded, as RCA is supposed to be able to use them “as-is” (as long as they are needed);
- the “upper-level” of the TMS (including long/short-term production plan) has been excluded, since the automation goals are quite different among IMs and this part serves a “hub” into the different existing information systems of each IM; however, the execution of the production plan is in-scope for RCA.

While the functions / systems corresponding to these topics are out of scope, interfaces from RCA to these functions / systems are in scope. In particular, the interface to short-term planning and dispatching (the operational plan) plays an important role in RCA.

The following diagram shows a simplified system scope. A more detailed system scope, including interfaces to the surrounding functions / systems is described in the chapter “High-level architecture”.

Note: The current version of the architecture covers ATO GoA2 but is not yet complete with respect to ATO GoA3 / 4.



4. Description of RCA Deliverables

This chapter explains the targeted architectural description for RCA. In RCA Alpha, only a small subset of this description is available. A more complete description will evolve in the coming iterations of RCA.

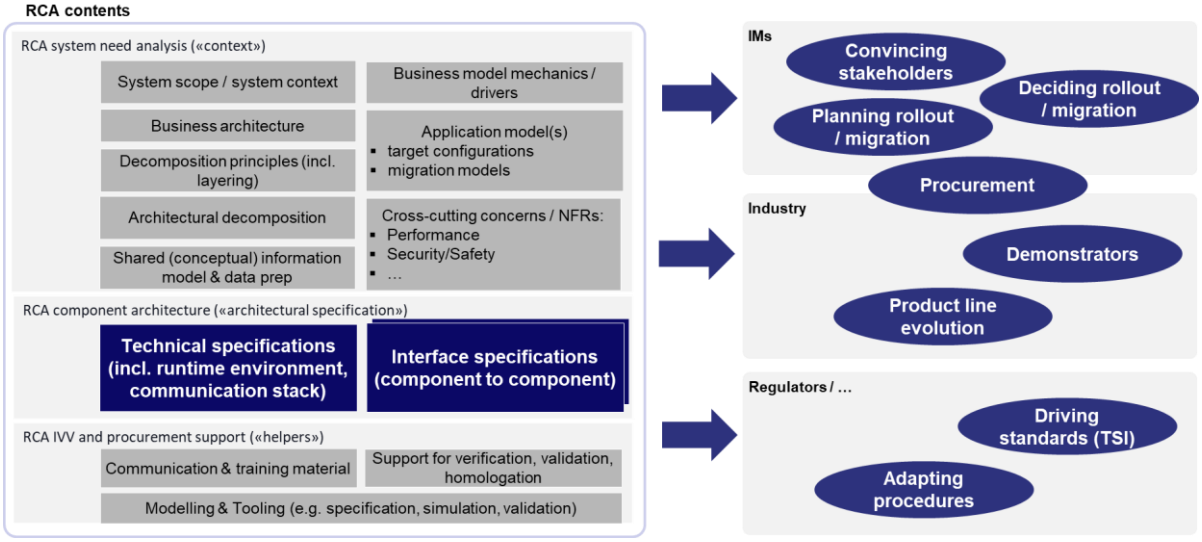
Distinguishing RCA from an RCA-based system

RCA is not a complete system architecture but provides a reference i.e. a blueprint for building / procuring a concrete CCS for a specific IM. When we talk about goals, features, etc. of RCA we often mean the goals, features, etc. of an actual system implemented based on RCA. When we want to make this explicit, we use the expression “RCA-based” system.

See also the document “RCA Beta – Chapter on architectural approach and system-of-systems perspective” for more details.

4.1. Overview Deliverables

The following diagram shows the planned deliverables for coming RCA releases. The deliverables include formalized, technical content such as the specifications for important interfaces, but also supporting material to help apply RCA.



Explanations for RCA contents (left-hand side):

- the core of RCA are the technical specifications and the interface specifications of the RCA components;
- the ”system needs” describe the context and requirements with respect to the RCA core and is considered supporting material to explain and maintain the RCA core;
- IVV (Independent Verification & Validation) and procurement support facilitates implementing an RCA-based system. These elements have not been started for RCA Alpha.

Explanations of how RCA contents are to be used (right-hand side):

- the RCA must be known and be judged as promising by IMs who want to build RCA-based systems, by suppliers taking up RCA specification in their product portfolios and by regulators, who will be asked to adapt existing standards / rules.

4.2. Purpose of the RCA interface architecture

The interface architecture specifies well-defined interfaces for all components in the scope of RCA. For these components all interfaces need to be well-defined by RCA to ensure exchangeability & upgradability. In addition to the interfaces defined between components, we also define interfaces to the runtime environment and to the communication stack (see technical architecture).

For components outside the RCA scope, but having an important interaction with RCA, we specify well-defined interfaces for the interaction only.

An IM can use these specifications to build or procure exchangeable & upgradable components. An IM is free to procure components independently or to group them (see procurement option).

4.3. Purpose of the RCA technical architecture

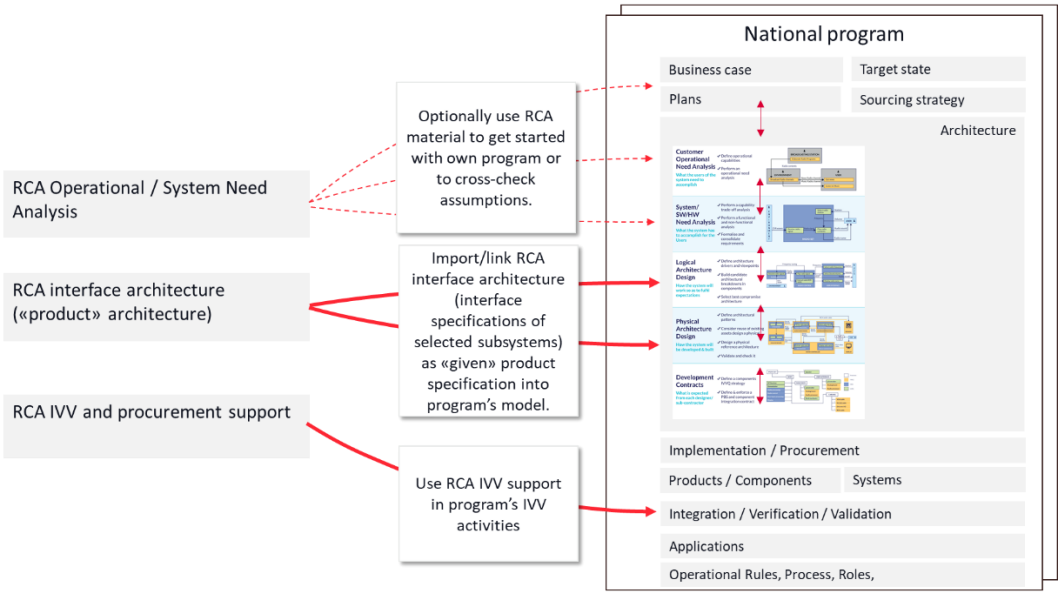
The components of the interface architecture can be thought of as “boxes”. In RCA the components are however not necessarily “physical boxes”. Adopting successful architectural thinking from the IT domain, we use the following decomposition:

1. The input- / output-behaviour (“function”) of a component (mostly implemented by software).
2. The platform or runtime environment providing
   a) fundamental resources such as computation or storage to the “function” and
   b) mechanism for common concerns like fault tolerance, application management, persistence (defined by APIs and implemented by software and (maybe virtualized) hardware).
3. The communication stack providing the communication channel. On one level this is the “virtual” channel from the point of view of the different (functional) components, on another level, this includes the “physical” channel between the platform the functions have been deployed to. The stack is defined by APIs and protocols; implemented by soft- and hardware, over several layers.

The runtime environment(s) and the communication stack(s) define the technical architecture of RCA.

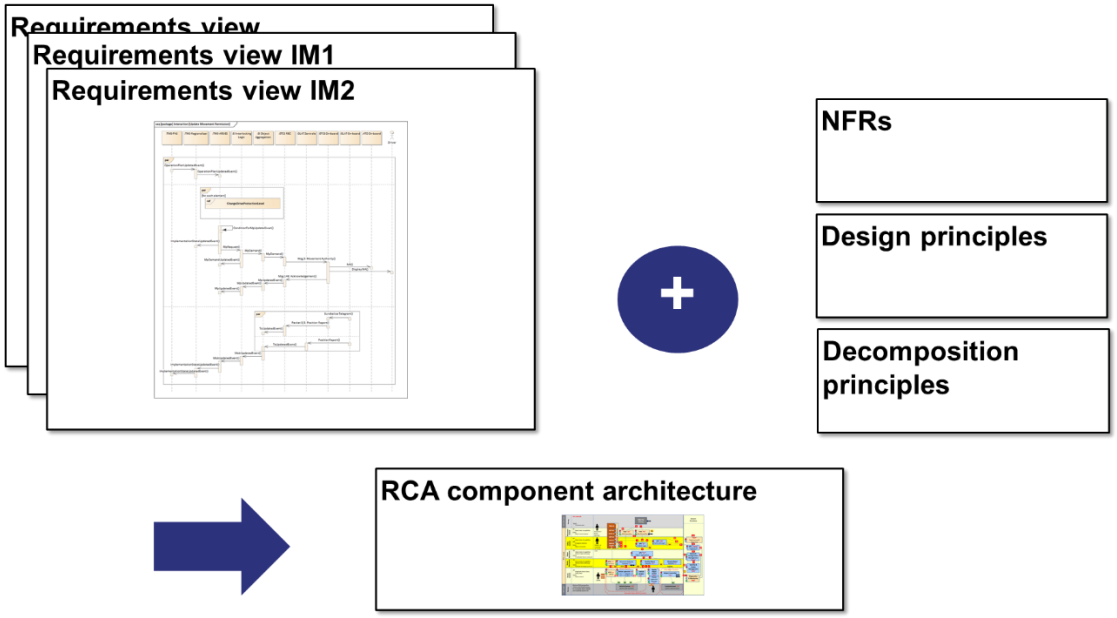
4.4. How IMs will apply the RCA Deliverables

The following diagram shows, that RCA provides (only) a part of the necessary content for a modernization. Many aspects of a modernization program (justification, business need, migration plans, procurement strategy, rollout planning, etc.) fully remain with each IM, with supporting material from RCA.



4.5. Where does the “harmonized” RCA component architecture come from?

The harmonized RCA is the result of applying functional and non-functional requirements of the participating IMs to the RCA interface architecture and checking for completeness / fit. RCA therefore contributes to the harmonization of on-going programs of different IMs.



This "RCA interface overview" does not show a pure functional structure since the defined decomposition principles (see chapter 5.4) are already applied. They combine aspects of the functional-, system-, lifecycle- and technical- architectures, to support the scoping of real products (HW, MW, SW) that make sense as standard products for the railways. The decomposition rules follow different architectural quality attributes, e.g. exchangeability and independence, difference of lifespan or lifecycle type, isolation of safety cases (“modular safety”), no combination of functionality that is not always used together, etc.

In general, the split, that leads to the "RCA components", describes interfaces at positions in the CCS automation pyramid[11] where different products shall be able to work together on a standardized basis. But these interfaces can also be used inside of a product or product cluster. This may be a requirement in a tender to have the flexibility for reactions on unexpected lifecycle events (e.g. discontinued products, strong price hikes) to change the supplier or product type only for RCA components in the later lifecycle. How many RCA interfaces are used is decided by the individual railways when they design their lifecycle strategy for their CCS architecture, their migration programs and their tenders.

An "RCA interface" defines all aspects and layers of communication and transaction of the interface, and the behaviour of products on both sides concerning those transactions. This interface can be defined by a set of interface specifications like for example for ERTMS or EULYNX. The RCA interface specification can be an existing or a new specification. Because of the quite large RCA scope and different sources of the specifications, the ontologies and specification methods will not be the same in the beginning. The RCA group will work on the translations and continuous harmonization.
If a component has only RCA interfaces for its external communication, it is completely specified. If a component has also other interfaces, it may not be fully exchangeable.

The different business processes, operational processes and non-functional requirements of the participating railways are used as “test cases” for the RCA architecture. There is not one single defined set that is the basis for the RCA architecture (only best practices will be selected). Since operational processes are in future only defined in the TMS (traffic management system), which is mostly out of RCA scope, the RCA architecture shall

[11] https://en.wikipedia.org/wiki/IEC_62264

be "process independent" and useable as standard architecture in many different implementations to support various operational processes.

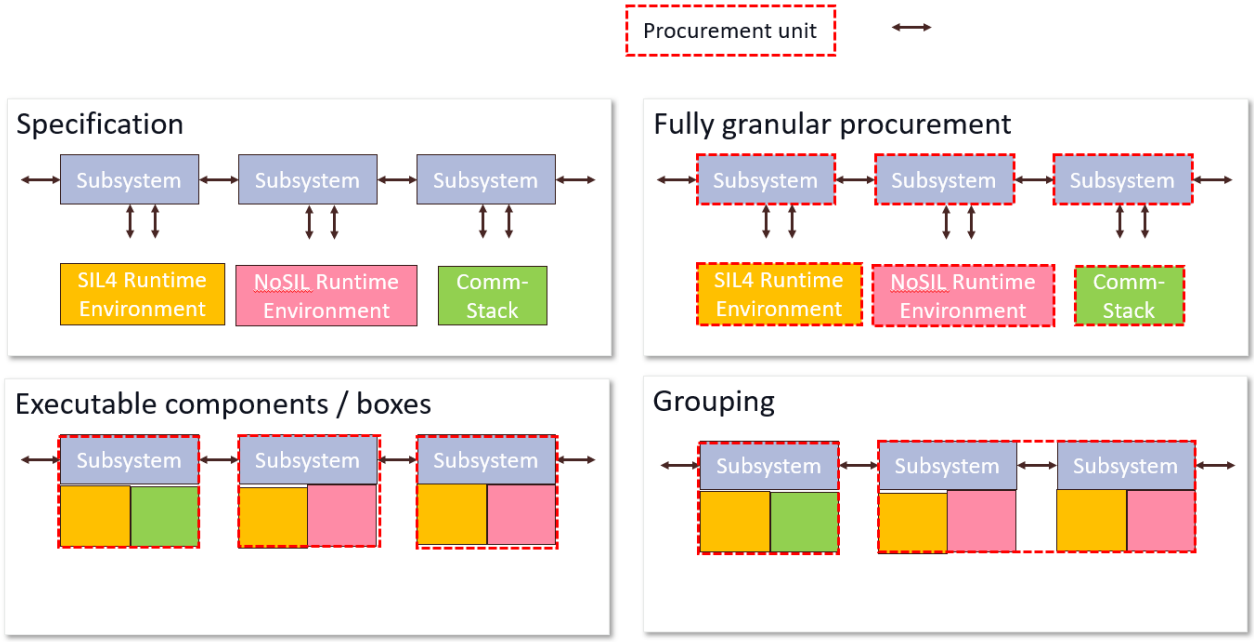
The RCA does not specify the functions or level of automation of RCA blocks, only the external communication is defined. RCA blocks may be implemented as a completely automated system or as a system operated by a person. Every RCA interface is always implemented electronically. Only the so called ui-interfaces are different: They define the functions for mandatory user interfaces an RCA block shall provide.

4.6. How the interface architecture is linked to procurement options

By way of example, the following diagram shows that IMs can choose different procurement strategies for an RCA-based system, these procurement strategies lead to different integration needs.

Granularity of procuring RCA components:

- Upper left: The RCA specification defines interfaces for the RCA components. These interfaces include “functional” interactions with other components and interactions with technical building blocks, including runtime environments and communication stacks;
- Upper right: an IM can choose to procure in a fine-granular way, isolating all RCA components in its procurements;
- Lower left and right: an IM can choose to bundle RCA components in larger procurement units. Whether the internal interfaces are part of the procurement requirements is up to the IM.



Implications of fine- vs coarse-granular procurement:

- With fine-granular procurement the task of integration becomes more important. Integration can be performed by an IM or can be tasked out to a third party;
- When procuring bundled components, the IMs should make the “internal” interfaces between RCA components a part of their requirements to a) ensure future exchangeability and to b) help drive harmonized specifications;
- An IM may procure in an even more fine-granular way, by splitting up an RCA component. In that case, the IM must define their own specifications for the new interfaces. These interfaces may be candidates for inclusion in RCA.

4.7. Relation of RCA to “harmonized operational processes”

Full harmonization of operational processes is not in scope of the RCA. When changing to a new architecture for the future railway system in large migration programs, some harmonization of operational processes becomes possible (compare to standardized DMI, etc.).

RCA has excluded full harmonization of operational processes from its scope, because the processes have too many dependencies such as national laws; national affordability of the same safety, performance or availability targets; processes are “programmed” into hundreds of products and thousands of installed systems; processes are part of thousands of safety cases, etc.

Migrating to the RCA will support the harmonization of operational processes. But the way of harmonization even in this case needs certain steps and depends on the initial situation of the IM and the structure and duration of the migration.

5. High-level principles and design guidelines

5.1. General architectural principles for “RCA-based” systems

The following table shows an overview of the currently identified principles. Some of these principles need more work to be applicable by the RCA group or the Working Groups.

Principle	Rationale	Consequences
Target state requirements and migration requirements have equal priority in RCA = RCA design for a “pure” future state and includes necessary mechanisms for migration	ERTMS/ETCS deployment lessons learned - importance of considering operations and degraded mode, being stuck with a supplier specific solution due to lack of common interfaces - importance of agreeing a target state and functional needs and then considering migration. Starting from current state and working forward leads to incremental development which may not provide optimum solution and likely to lead to national flavour of solutions.	- Architecture supports several migration (and sourcing) strategies. - IM must define their requirements for a migration towards RCA into the RCA process.
In RCA every function is only designed once (no large set of functional alternatives in the target state)	In a target state with “pure” ETCS “L3+” there is no objective reason for functional diversity. Diversity has led to poor performance / cost and poor innovation.	- Harmonization of requirements has to take place. - If needed, variability specified through configuration / parametrization. - A certain degree of variability can be achieved by using different configurations of RCA components. - Note: this does not require total harmonization of operational processes.
Minimum trackside assets possible	Trackside assets exist in higher numbers, in harsher conditions and are more difficult to get at. Reducing them reduces LC-cost / function and increases reliability (environment, MTTR).	- Safe and reliable alternatives have to be developed (e.g. localization, train integrity moved from trackside to vehicle).
Modularity: Exchangeability of components	Key to ensuring keeping LC-cost down and ensuring evolvability.	- Requires careful interface specifications. Requires interfacing techniques for up- and downward compatibility e.g. capability-based protocols.
Scalable to different needs (e.g. achieving safety levels by choosing the right (amount of) devices)	- Ensure broad applicability of RCA-based systems.	- Need to consider several target configurations.

Principle	Rationale	Consequences
Functions are allocated to Software. Software is separated from Hardware	- In many cases, SW has better quality- / price-ratio. - SW is easier to evolve.	- The SW / HW interface has to be specified[12]. - Procedures for updating SW securely are needed.
Communication (carrier, lower-level protocols) is exchangeable	Key to ensuring keeping LC-cost down and ensuring evolvability.	- Functional and communication aspects are separated in SW. - Communication is properly layered.
Slim SIL 4 components: No combined implementation of business processes and safety functions	Development on higher SIL levels is disproportionally more expensive. Safety related functions kept to a minimum.	- Architecture needs to provide the corresponding interfaces. These interfaces may be critical from an NFR-point of view.
High ability to automate functions ("transactional completeness")	The architecture must allow (not necessarily enforce) full automation to support corresponding goals by implementers.	- The functional architecture must not rely on "miracle" functions but describe functional blocks, that can (at least in principle) be automated[13].
Interfaces are upward and downward compatible	Evolvability (see also modularity)	- Profile- / capability-based interfaces
Modular safety: Ability to isolate safety cases and homologation	Evolvability	- Design of components (interfaces) must allow isolation of safety cases. - Early evaluation of "modular safety" concepts with assessors necessary
"Core" systems and their interfaces are "process-agnostic"	Ensure broad applicability of RCA-based systems.	- E.g. Business processes only implemented in the TMS, other layers are process independent and focus on the basic "physics" of rail control and command.
"Cutting edge" technology is properly isolated i.e. integratable, but not mandatory	The speed of technological maturing is difficult to predict. The architecture must be open for foreseeable evolutions while allowing implementations "here and now".	- E.g. Flexibility concerning the connection and mixture of localization devices

[12] In some cases, an integrated stack may be procured, at the cost of having lock-in between SW and HW.

[13] RCA focuses on interfaces. To provide context for the interfaces, components/blocks/subsystems and behavior will have to be described to some extent.

5.2. What needs to be specified to “sufficiently” define a component?

A building block (component) is normally described by the following aspects:

- External interfaces: Input, output, transactions, communication layers, non-functional requirements (NFR, e.g. RAMS);
- Internal state(s);
- Internal events;
- Internal memory and data storage;
- Internal operations and algorithms.

For a freely usable architecture only the external interfaces are relevant. They describe the complete behaviour of the function at its external interfaces. Behaviour has to be qualified by NFR (e.g. availability, hazard scope or hazard rate of a behaviour). If all interfaces of a block are specified this way the functional requirements for the block are complete from the perspective of the RCA[14].

Need for FFFIS

Many building blocks of an RCA are SW-based and don’t need a “physical incarnation” and therefore don’t need full FFFIS. For the SW-based blocks the “physical” connection is sufficiently specified by their interface to the runtime environment and to the communication stack (see also “technical architecture”).

Other important qualities for RCA interfaces include:

- Providing mechanisms for upward- and downward-compatibility;
- Being independent from the communication stack;
- Being free of IPR that may be an obstacle to the exchangeability or upgradability of blocks / components (the products or component implementations may, of course, still be protected by IPR).

5.3. Variability management

Variability is due to the fact, that not all features of an RCA-based system will be needed by all IMs, with the same functionality and / or at the same time. Variability introduces complexity, increases cost, and may reduce overall performance. An important principle is: RCA specifies a toolbox of components. Necessary differences among IMs should be expressed as:

- a difference how the components are assembled or if they are used;
- a difference how the components are used (manual process);
- a difference of how much redundancy is applied;
- a company-specific add-on (automated process) outside of the RCA scope;
- the parametrization of components.

To express these conditions in one sentence: If an interface exists in a CCS architecture (RCA scope), it shall be an RCA interface, everything else can be freely designed. The design process / requirements model of RCA must include explicit links to requirements of IMs, leading to a need of variability (e.g. operational rules).

5.4. Rules for splitting functions into components

These rules are rough guidelines helping to think about the functional break-down.

Reason to split	Explanation
Different NFR (non-functional requirements like safety, RAM or security)	NFRs drive the needs for the SW development process and for the deployment configuration. Packing functions with very different requirements together may lead to waste / over-design.

[14] While this is true, once a high-quality specification exists, the process of defining such a specification must include building models / prototypes for the internal working of the components to be able to “test” the specification and the resulting end-to-end behaviour of the system.

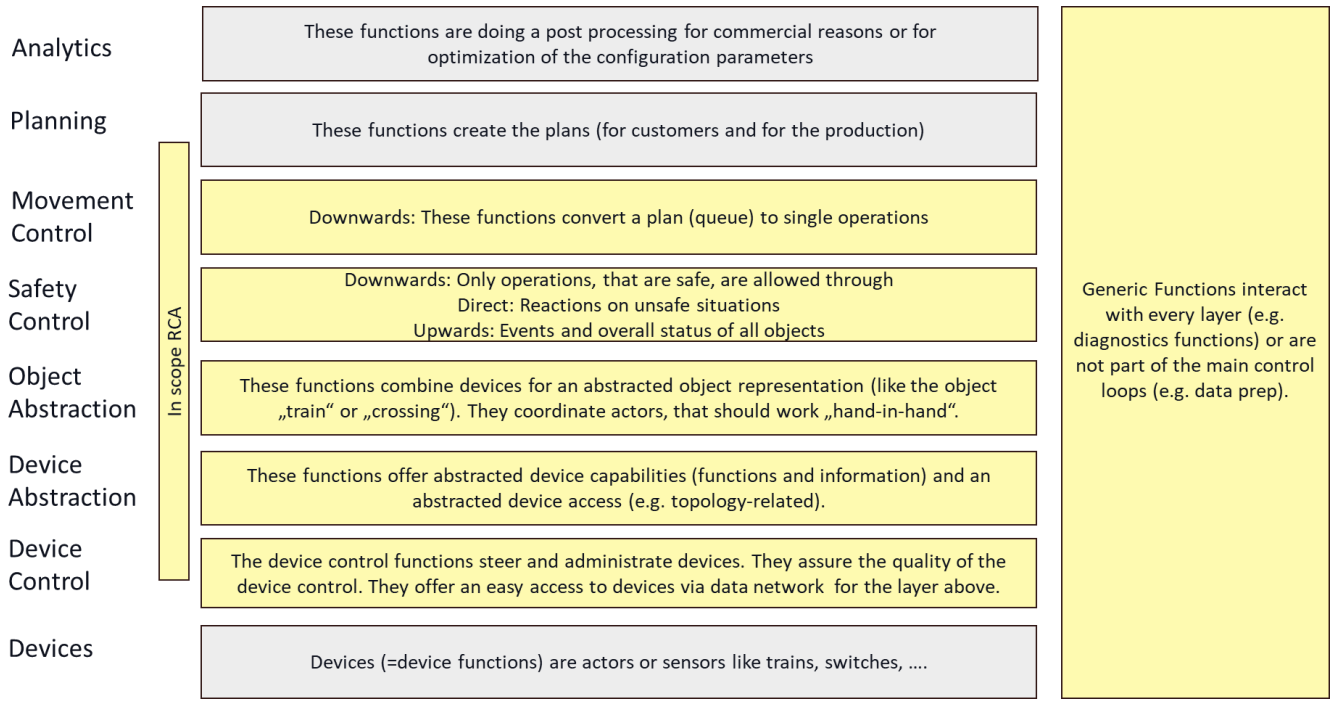
Reason to split	Explanation
The right split creates simple interfaces and self-sufficient systems	The classic modularity principle: high cohesion / low coupling.
Independent lifecycle • Independent installation timing or installation process • Independent change • Independent lifespan	Separated functions may help upgradability (e.g. need for retesting).
Usage only in a part of all installations	Possibility not to deploy a function at all.
Necessary hardware topology (e.g. regional)	Available devices (computing power, storage capacity) and / or communication access to the devices and / or realms of responsibility (e.g. in a vehicle) can make this necessary.
Different markets or supplier types	“Unusual” bundles may lead to difficult / inefficient procurements.
Smaller components make it easier for more vendors	Lower market entrance hurdles. BUT: additional integration work.
Split of homologations and safety cases	Reduce procedural effort.
Split the work / Division of labour	Ability to distribute / parallelize work.
Re-use potential	Possibility to re-use important existing work.

5.5. Layering principle for the architecture

The RCA is divided into architectural “layers”. Layers play an important role in structuring architectures and have successfully been used in computer architecture, communication architecture, etc.

In RCA, every layer has special types of blocks, and may have special design rules for interfaces (generic for all blocks in the layer) and especially special “abstraction levels”. Example: On the device control layer a function will know about the type of hardware (switch, crossing, TDS) it controls. On the safety control layer “objects” are only known by their abstract hardware-independent capabilities (e.g. “trafficability on a node-edge-model”).

RCA components are assigned to exactly one layer.



The following table describes the layers in more detail:

Layer name	**Interface upwards**	**Interface downwards**	**Type of functions on the layer**	**Layer characteristics**
Planning		Operation Plan	These functions create the plans (for customers and for the production)	
Movement Control	"**Execution Status" =** Progress of the plan execution "**Operational Status**" = States and actual capabilities of all **objects** = **trackside assets** ("**TA**" like tracks, switches, crossings….), all **moveable objects** ("**MOB**" like trains, Persons, obstacles).	**Object control requests (OCR)** (update movement permission, control object, change object status, etc.)	These functions implement the operation plan by issuing single object control requests when the condition regarding the current operational status are met These OCR can for example change a switch position or update a movement permission.	"Process independent". No implicit assumptions for process rules on this layer. Every process- or company-specific rule should be implemented on a higher architecture layer (planning). The operation plan (and delivered static descriptions for process rules), that is converted to device requests shall fully describe the process, that must be executed - also in degraded modes. Layers from here downwards are "real-time" layers. The functional design has to take the system implementation into account (high workload, some hundred thousand events per second for a small network, so efficient simple functions have to be designed, decision processes shall be fast). This layer is very important for the migration. The functional interfaces to legacy systems must be designed and shall be taken into account in the functions for the target architecture. All interfaces down from this level are asynchronous (cut up functional sequences, queueing, locking, necessary for scalability). All functions of this layer shall be designed for a maximum autarchy duration with stepwise degrading modes, when the planning system is offline. This and higher layers shall support the most simple functionality in the lower safety

Layer name	Interface upwards	Interface downwards	Type of functions on the layer	Layer characteristics
				layers by relieving lower layers from functionality, that is not safety critical. Lower layers shall only have "check and driver" functionality.
Safety Control	**Operational Status**	**Object** control commands for the control of MOB/TA	These functions check requests from upper layers or users: Do they lead to a safe state of the production? If yes, then they are executed. They also check events and overall status of all objects and invoke emergency reactions for unsafe situations.	On this layer no hardware-, asset-, asset-layout-, track-layout, train-type-, track-usage-condition- or train-capability-specific requirements shall exist. All these attributes are described as abstract object attributes that shall fit together for a safety check function. Functions shall be implemented that allow a generic safety case for a specific parameter- and ruleset, that is set for a whole network. The parameter- and rule- sets shall allow to define different safety targets and principles without ever changing the implemented system.
Object Abstraction	Addressable abstracted **objects** with functions and attributes	Actor coordination: Coordinated **device** control commands for the control of MOB/TA	These functions combine devices for an abstracted object representation. They coordinate devices (actors) for the execution of object control commands, that should work „hand-in-hand“.	Devices are handled on this layer in a generic way. They are described only by their generic attributes and capabilities (functions, methods), not as hardware models or hardware types. Aggregating devices to objects (e.g. to trains or train-components) is done by a rule-interpreter for configurable rules, where each has its own verification (extendability). An object is defined by one or more attributes (like ID, start-position, end-position, length, type, etc.) and devices deliver one or more of these attributes together with an object ID, to whom they belong. The execution of object control commands can influence more than one device inside of the moveable object and in parallel on the track.

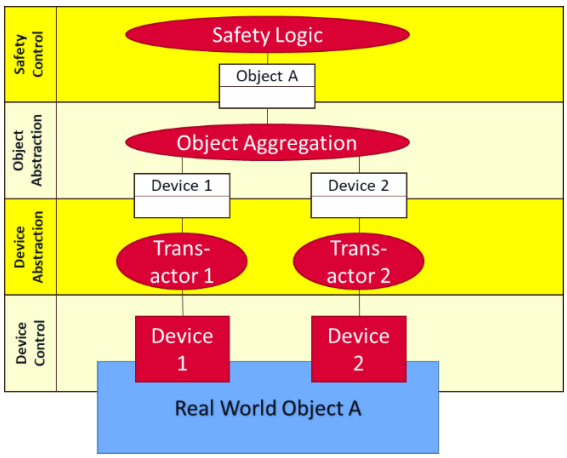
Layer name	Interface upwards	Interface downwards	Type of functions on the layer	Layer characteristics
Device Abstraction	Abstract device capabilities. Abstract device addressing.	System specific protocols	These functions offer abstracted device capabilities (functions and information) and an abstracted device access (e.g. topology-related).	The downward interfaces (and on the layer below) shall implement the "modular safety" strategy which reduces the effort to integrate new pre-certified devices to a minimum. This shall be achieved by a small and stateless protocol (small capability protocol modules) and fully testable behaviours on both sides of the interfaces. This layer shall allow to treat different device models as the same generic device-type.
Device Control	(safe) device status	(safe) Device management and control	The device control functions steer and administrate devices. They assure the quality of the device control. They offer an easy access to devices via data network for the layer above.	
Generic functions	Information for Management systems	remote management control or prepared static data	Generic functions interact with every layer (e.g. diagnostics) or are not part of the main control loops (e.g. data prep).	This layer contains cross-cutting-concern functions and aspect-concern-functions. The design of these functions shall be done as a flexible service layer.
Devices	System specific interfaces		Devices (=device functions) are actors or sensors like in trains, switches, ….	

5.6. Concepts of Objects and Devices

To understand some of the architectural decisions in RCA, it is helpful to discuss the role of “Objects” and “Devices”.

The diagram shows that:

- Objects are monitoring and managing 1..* devices;
- “Object Aggregation” combines multiple device information to a single object representation of the “real world object”.



Examples include:

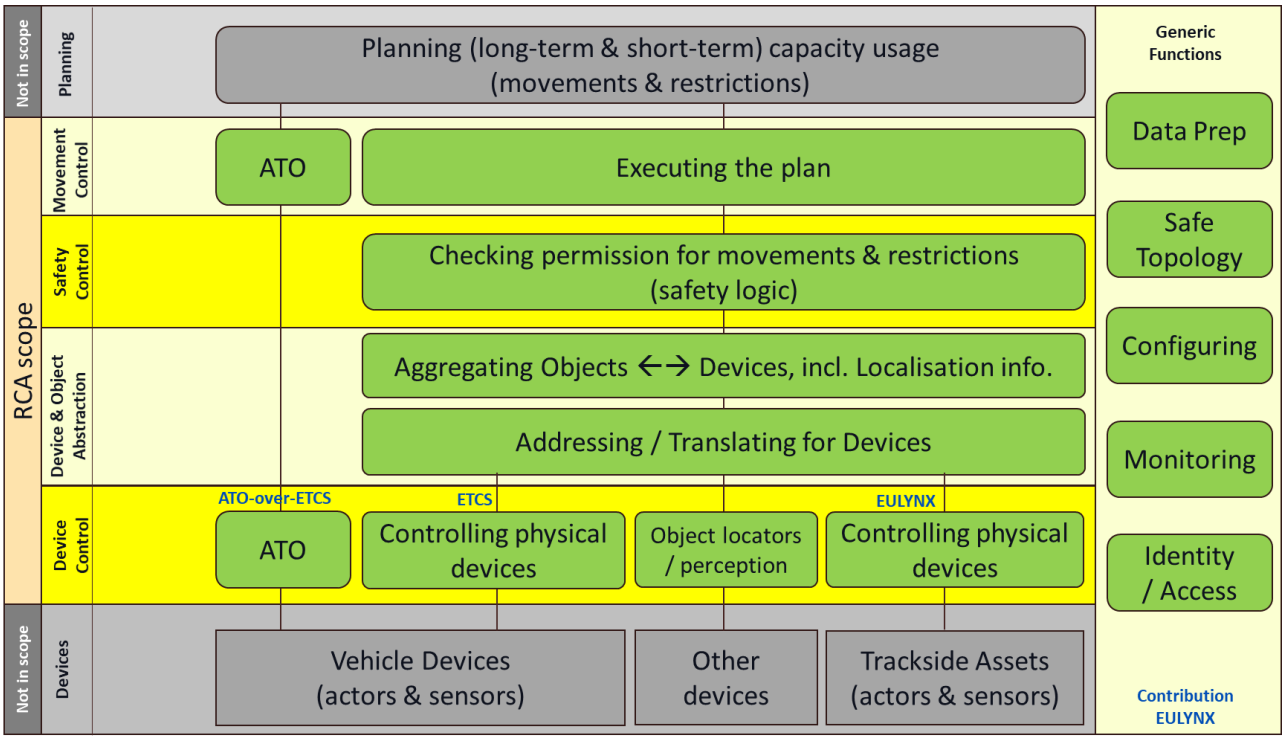
- Train with ETCS-OBU at the front and a localization tag at the rear and / or using trackside TDS;
- Movement permission to the ETCS-OBU and / or to a trackside signal (e.g. border signal);
- Level crossing with separate device for intersection scanning or warning lights;
- New devices in the future.

Object Aggregation is an important function, because:

- Object Aggregation provides to upper layer an **abstract view** of controlled objects
- How objects are connected with devices is hidden from the upper layers
  Advantage:
  - Changes for new device mixes only affect OA, but not upper layers
    → Similar to Hardware-Abstraction Layer in Operating System;
  - Object Aggregation can support many different mixes of devices;
  - Migration from old to new can be encapsulated in Object Aggregation.
- The problem of aggregation is independent of the problem of the safety logic.
  Advantages:
  - Good “separation of concerns” (→ Reducing complexity of each problem);
  - Independent lifecycle of independent changing logic;
- This follows the proven pattern of an “layered architecture” and an “automatization pyramid”, where **each layer solves a specific problem**.

5.7. Allocation of functions to layers

The following diagram shows the allocation of (simplified) functions to the layers of the RCA. For each function there will be one or more RCA components (based on the defined decomposition principles). The RCA components are shown in chapter 6.



6. RCA architecture

For RCA Beta the architecture has been refined in several steps:

- scope (chapter 3);
- design principles and decomposition principles (chapter 5);
- layer architecture (chapter 5).

In this chapter we show two main results which provide the foundation for the future specification work:

- RCA interface architecture (components to components);
- RCA technical architecture (components to runtime environment and communication stack).

For RCA Beta this is expressed as drawings and texts in this document and does not qualify as a "specification". In the future the RCA will follow a MBSE[15]-approach and will choose a more formalized and tool-supported notation.

Note:

- The current version of the architecture covers ATO GoA2 but is not yet complete with respect to ATO GoA3 / 4.
- See also the document "RCA Beta – Chapter on Architectural approach and systems-of-systems perspective" for more details.
- See also the document "RCA Beta – Platform independence" for more details.

6.1. RCA interface architecture

6.1.1. Overview diagram

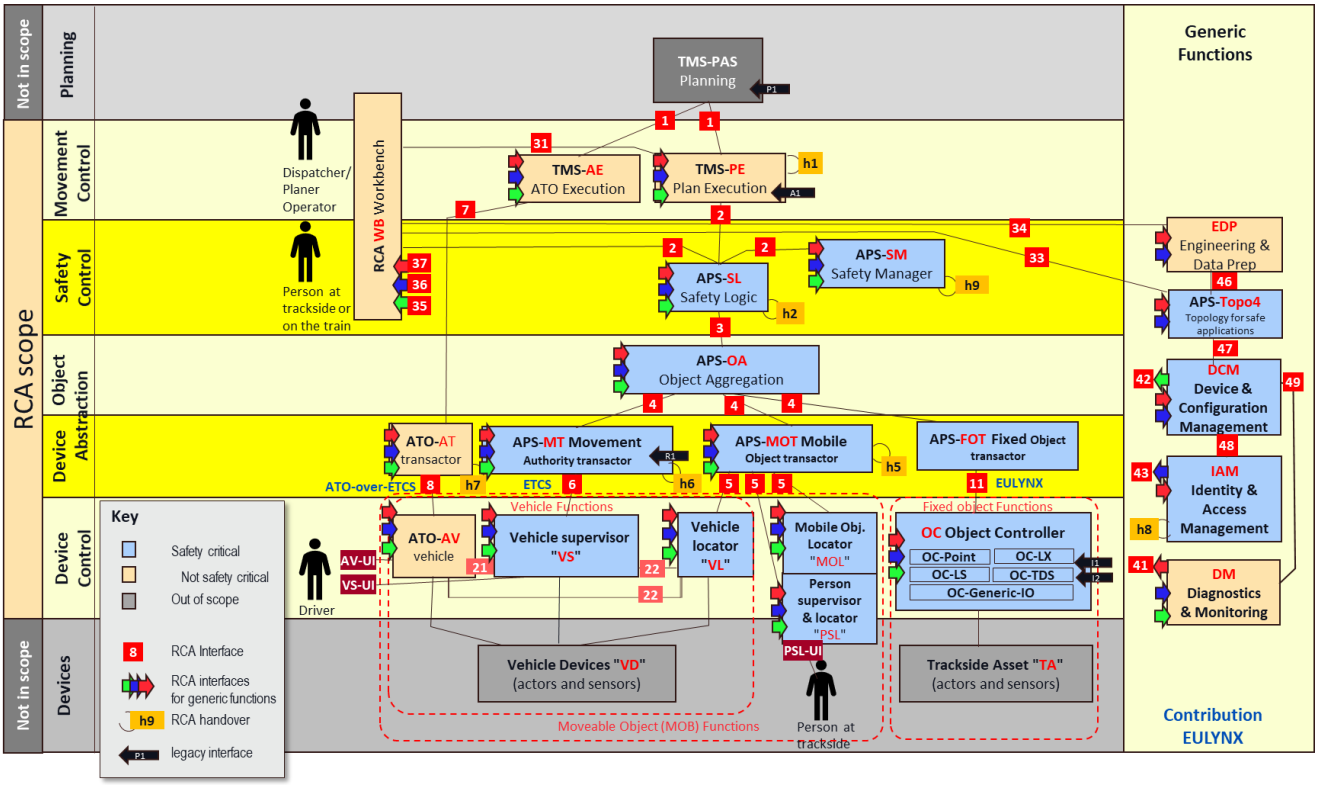
This is the core of the RCA and describes the main components and especially their interfaces. These components and interfaces are defining the primary working structure of the RCA. Every interface that is marked with a red number shall be defined by an existing or new specification. The specifications overall shall fulfil the requirements of the end2end processes. The specified components are the candidates for being units of procurement and thus "product-candidates". Different IMs may choose to use different granularities for procuring these components (see discussion above).

This architecture is derived by applying the decomposition principles and the layering principles to high-level functional architecture.

The following diagram shows:

- In the background, the RCA layer model;
- RCA components (orange boxes = non-safe components, blue boxes = safe components, grey boxes = out-of-scope components);
- RCA interfaces (red boxes, orange boxes, red, green, blue, black arrows);
- In the centre of the safety relevant components, we have the APS ("advanced protection system") with its 3 components SL, SM, and OA. The APS can be thought to replace (the core of) the current interlockings;
- ERTMS/ETCS and EULYNX are sets of specifications, that are re-used for RCA. "Contribution EULYNX" means, that some work of EULYNX will be re-used for the generic functions;
- Note: interfaces with users are only summarily shown and need to be detailed at a later stage.

[15] MBSE = Model-Based Systems Engineering. EULYNX applies MBSE.



6.1.2. Types / groups of interfaces

The following table shows how the interfaces have been grouped, according to their characteristics

IDs	Type of interfaces
1-19	Control and Safety for multiple objects
h*	Handover interface (for multiple instances)
*-UI	Mandatory user interfaces
21-29	Command, Control, and Safety for one vehicle
31-39	API and interface "RCA Function"<>"Workbench"
41-49	Interfaces for cross cutting concerns and aspect concerns
P, A, I, R	Interfaces to legacy CCS e.g. during migration

6.1.3. List of components

The following list provides a short description per component, the ID corresponds to the red abbreviations in the overview diagram.

ID	Name	Description
TMS-PE	TMS (Traffic Management System)[16] Plan Execution and Control	The PE generates the requests to the Safety Logic (SL) at the right point in time to execute the operation plan. According to the progress it reports the execution status of the operation plan back to planning system. The execution status describes the parts of the plan that are already executed and the parts of the plan that are allocated (e.g. when movement permission is already set). Near-time optimization is done in the planning system. PE controls movements inside a given bandwidth of operation performance. Includes all traditional (non-safe) functions of interlockings and control systems, that are shifted out of APS.

[16] The TMS prefix indicates that these components are attached to the TMS. Depending on the concrete configuration they may be or not part of the TMS.

ID	Name	Description
TMS-AE	TMS[16] ATO Execution	The AE generates from the operation plan the required information (ATO plan) for AT, inside a given bandwidth of operation performance. Optimizes energy consumption and overall traffic flow as a trade-off[17].
APS-SL	APS (Advanced Protection System) Safety Logic	SL decides if a PE request is granted or rejected depending on the evaluated risk (rule-based). The request can ask for a state change of a trackside asset, the creation / modification / removal of a movement permission or set / unset a "danger area". For the decision, SL stores the state of the trackside assets, the movement permissions, the position of moveable objects (e.g. trains), the current danger areas and the topological data. See also chapter 6.5 “Principles of the safety logic” (includes a rationale for splitting the safety logic in SL and SM)
APS-SM	APS Safety Manager	SM continuously monitors the state of the system, such that it can recognize patterns that are identifying hazardous situations. It will trigger one or several reactions (e.g. emergency stop of a moveable object, reduction of the speed, extend movement permission) to prevent or minimize the danger. In addition, it also recognizes situations that require the warning of a moveable object (e.g. personnel at trackside). Danger pattern recognition, danger patterns as well as emergency reactions are configurable. See also chapter 6.5 “Principles of the safety logic”.
WB	RCA Workbench	WB isolates the functional logic and user processes. Its function is to present a process specific frontend to different types of user-roles that can even change in certain events. User interface elements are registered to WB statically and dynamically depending on the actual registered functions and devices. The user process management of the WB invokes different sets of user interface elements depending on the process situation. User interface elements and WB are functions that optimize the input- and output efficiency as much as possible and that offer collaborative frontend functions as well as synchronized input and output on multiple device. WB can handle safe and unsafe user interface elements.
APS-OA	APS Object Aggregation	The OA combines the information received from one or multiple devices to one consolidated object representation that is provided to SL. That consolidated object representation contains the state of the moveable object (e.g. trains) like position and extent (length) as well as the state of the trackside assets. In the other communication direction, it dispatches information from the SL to one or several devices. This information includes the Movement Permissions, the state request for the trackside asset and warning messages for personnel at trackside and engine driver. Aggregation rules are configurable and have an isolated homologation. The OA function therefore is a rule-interpreter together with a set of rules. The OA "collects" device information for an object, that does not need to arrive at the same time. The rules shall describe the correct reaction on timing hazards.
ATO-AT	ATO Transactor	AT implements the communication with all registered ATO vehicles and provides the standardized interface.
APS-MT	APS Movement Authority Transactor	MT communicates with the registered ETCS capable vehicles. Among others it translates the movement permissions to ETCS Movement Authorities and send them to the vehicle. Only radio-based ETCS is supported. In the other direction it will receive the train position reports from the vehicle and forward them to OA.
APS-MOT	APS Mobile Object Transactor	MOT manages the different kinds of mobile devices, that are locatable and optionally can be warned. It provides information to the mobile devices, which they need to localize themselves. The MOT processes the received localization information such that it can be forwarded to the OA. It also forwards warning information to the mobile devices.

[17] The current SUBSET-125/126 defines the interface in such a way, that timing-points are sent to the vehicle. How the optimization responsibilities are allocated between AE and vehicle has to be defined (on-going in GoA2 S2R).

ID	Name	Description
APS-FOT	APS Fixed Object Transactor	FOT communicates with all the relevant Object Controllers. It translates the abstract commands of the OA to asset specific commands when fitting to its own capabilities. In the other direction, it translates the asset specific status of the OC to an abstract status for the OA along the trackside asset's capabilities.
ATO-AV	ATO Vehicle	The AV operates the vehicle automatically and optimizes the speed such that it reaches given points at a given time as received from the AT.
VS	Vehicle Supervisor	VS is based on existing ETCS on-board with some change requests (CR). Necessary changes will rely on established CR processes. The VS displays to the "Driver" (if existing) the current allowed movement authority by using cab signalling. It also supervises the speed and ensures that the train does not violate its movement authority. Further it will send the current position as a "train position report" to the MT. This supervision has the following functional aspects (which can be combined or used partially): 1. "ETCS": Functions of ETCS cab signalling as defined today with necessary adaptions e.g. for FRMCS / multi carrier com, ATO interfaces, full moving block, and mixed localization configurations 2. "Full supervised shunting": Supervision with efficient management of low speed movements with less train information (just track occupation), multiple moves in different directions, and some local safety responsibilities 3. "Supervision in degraded modes": Fall-back functionality that allows to stay operable on a high level in degraded modes 4. "AMS: Autonomous movement supervision": Diversely implemented fall-back functionality that provides a basic safety with the minimal use of other functions (e.g. only train2train coordination and direct access to OC or trackside assets). AMS could also be a completely isolated function. For simplicity, the 4 functions have been described together. According to the decomposition principles of RCA, the 4 functions could be 4 independent components. The functions 2-4 are extensions to today's ETCS functionality and will need the TSI CR to make VS an interoperable component. This component is only partially in scope for RCA, see chapter "Scope".
VL	Vehicle Locator	VL uses mobile localization technology to safely and reliably provide position and speed information of the train. It may emulate a location balise to the ETCS functions. In addition, it provides the actual position to the vehicle function, and over a direct interface to the MOT. The VL sends the full virtual track occupation of the train or only a part of it (only the front or rear position). This implies a new equipment requirement for vehicles. For safe length different options are possible (train integrity or second localization unit at the other end). This component can use different localization technology, from today's "balise + odometry" to additional sensor inputs such as GNSS or inertial measurement units. This component is only partially in scope for RCA, see chapter "Scope".
PSL	Person Supervisor & Locator	PSL can either be set up to block tracks or to warn or authorize the person or a group of persons (different forms of implementation / hardware solutions) in accordance with configurable parameters if another moveable object / vehicle approaches. Can be a tag, a trackworker safety system or an app on a tablet that interacts with the person. A PSL integrates typically a MOL-function. It is shown as a separate block here, since it may be procured as separated product.
MOL	Mobile Object Locator	MOL sends its current location to the MOT. It can be used to locate any type of object on or near of the track. Can be used for multiple use cases like tagging an obstacle, a crane, a train-end, a wagon or coach, a door that swings on the track, a person, etc.
OC	Object Controller	OC monitors and controls one or multiple trackside assets (TA). The OC can be either in the interlocking room or in the field directly at the trackside asset.

ID	Name	Description
		For each type of trackside asset (Point, Light Signal, Train Detection System, Level Crossing, Generic IO) there is a type of OC. The OC specifications are based on EULYNX.
DM	Diagnostics & Monitoring	DM collects monitoring and diagnostics information from all systems like central systems, trackside assets or the vehicles. The information is on one side used to derive the capacity limitation and an estimated duration of the capacity limitation that is used in TMS to reschedule Capacity Plan. On the other side the information is forwarded to a monitoring system of the IM, which triggers the corrective maintenance actions.
EDP	Engineering / Data Preparation	EDP provides the configuration data for the APS and MOT. It highly automates the process of capturing and validating the data.
APS-Topo4	APS Safe Topology System	APS-Topo4 provides correct topology and topography data for SIL4 applications by combining information from different sources, which also includes the acquisition of data by mobile measurement devices in the field. Note: additional (non-safe) topology data may be needed in the TMS. The architecture allows an export of the safe data to be used in other systems, but the non-safe topology systems are out-of-scope for RCA.
DCM	Device & Configuration Management	DCM is used to register, setup, and manipulate devices. This includes updating the configuration data and the software version. Safety criticality: DCM is safety critical in so far, that part of the configuration is safety critical. Not the whole DCM needs to be on highest safety levels.
IAM	Identity & Access Management	The Identity & Access Management authenticates and authorizes users and technical systems and grants or denies access to the system. Therefore, it will need to store the credentials to authenticate the entities. Supports the implementation of an ISO27001 / IEC 62443 compatible architecture.
TMS-PAS	TMS Planning	Represents the functionality preparing the long-term and short-term production plan. The component itself is out of scope for RCA, but there is an RCA interface to this component.
VD	Vehicle Devices	Represents the devices (sensors and actors) on the vehicle (such as engine, brakes, …). These devices as well as the interfaces to them are out of scope for RCA.
TA	Trackside Asset	Represents the devices (sensors and actors) at the trackside (such as points, train detection devices, …). These devices as well as the interfaces to them are out of scope for RCA.

6.1.4. List of component-to-component interfaces

The following list describes the identified interfaces and indicates “candidate interface definitions”. These are definitions which will be included 1:1 in RCA or may need some adaptions, depending on the specific interface. This will be detailed later.

ID	Between	Description

1	TMS-PAS ↔ TMS-PE
	
	TMS-PAS ↔ TMS-AE
		This interface provides the operation plan from the planning part to the control part and gives the current execution status back to the planning level. It includes the following information:
		Downstream:
		• The current version of the operation plan for each planned capacity object includes:
		  ○ In the case of a capacity reservation (Train Run, Shunting Movement, Stabling):
		  ▪ The track-precise path defined for the capacity reservation
		  ▪ The order in which the different capacity reservations are allowed to use each track
		  ▪ Time constraints for departure, arrival or pass-through at certain points in the track network.
		  ▪ Relations between capacity reservation for interconnections, usage of vehicles and personnel.

ID	Between	Description
		▪ The optimized speed profile. o In the case of a planned Capacity Limitation (e.g. planned maintenance work) ▪ The affected area on the topology ▪ The start and end time of the limitation. ▪ Details about the limitation like allowed speed. ▪ The order relative to the track usage of the Capacity Reservation, such that a capacity limitation is not activated before the preceding Capacity Reservation have used the track. Upstream: • The execution status for each capacity object. The status is not only provided for the Capacity Object planned in the Operation Plan but also for unplanned Capacity Object (e.g. unavailable track due to a failure). • This includes updates about actions taken by APS-SM.
2	TMS-PE ↔ APS-SL APS-SL ↔ APS-SM APS-SL ↔ WB	This interface allows that the non-safety critical block requests state changes from the APS-SL and monitors the APS-SL. It includes the following main information: Downstream: • Request required allocation state of the elements in a route (e.g. trackside asset) • Request Movement Permission for a Moveable Object (e.g. train) • Request Danger Area • Request Warning[18] Upstream: • Provides the current allocation state (updates) of the elements (e.g. Trackside Asset) • Provides the state of the Moveable Objects (e.g. trains), position, and extent • Provides Danger Area • Updates about actions taken by APS-SM. **Candidate interface definition**: Adaption of EULYNX SCI-CC Note: description of interfaces logic to SM and WB will be described in a later phase.
3	APS-SL ↔ APS-OA	Interface between an SL and the outside world that it controls. It includes the following information: Downstream: • Requests the required allocation state of the elements in a route (e.g. Trackside Asset) • Grant Movement Permissions to the Moveable Objects (e.g. trains) • Warn Moveable Objects (e.g. Personnel at trackside) Upstream: • Provides the current allocation state (updates) of the elements in a route (e.g. Trackside Asset). • Provides the position and the extent (length) of all the Moveable Objects (e.g. trains).
4	APS-OA ↔ many	This interface is a single device-oriented interface, which can provide or consume only part of the control or monitor information. It includes the following information: Downstream: • Requests the required allocation state of the elements in a route (e.g. Trackside Asset) • Grant Movement Permissions directly to the Moveable Objects (e.g. trains) or indirectly via a trackside signal. • Warn a Moveable Objects (e.g. Personnel at Trackside) Upstream:

[18] How can a non-safety critical system such as TMS-PE request warnings and danger areas from APS-SL in interface 2? For APS-SL a movement permission can only be extended, when the waring has been requested (analogous to setting a point inside a movement permission). The fact that a warning is necessary is established with a (safe) user interaction directly with APS-SL.

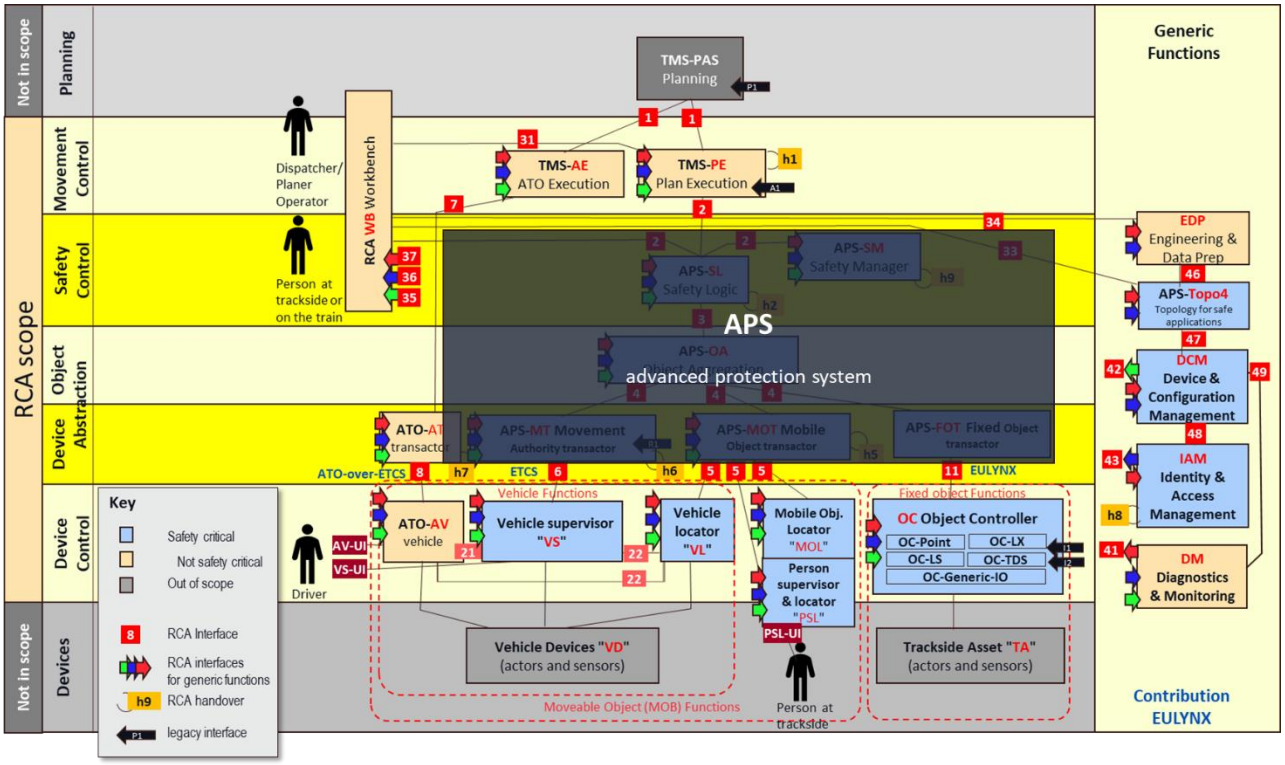
ID	Between	Description
		• Provides the current allocation state (updates) of the elements in a route (e.g. Trackside Asset). • Provides information about the position and extent (length) of a Moveable Object. The information can already be assigned to a Moveable Object or be just location based without an assignment to a moveable object (e.g. occupancy of a track).
5	APS-MOT ↔ many	This interface is used to communicate with the (safe) mobile devices. It includes the following information: • Management of the devices • Provides information to the Device, which it needs to localize itself. • Position of the Device • Requests to warn the Moveable Object
6	APS-MT ↔ VS	This interface is the existing ERTMS interface (ETCS trackside-ETCS OBU) with additional functions that are necessary for the RCA. Needed change requests will be handled using established CR processes. An example for such a CR would be inclusion of more train data from the vehicle "upwards" e.g. the current brake capabilities (for lambda and gamma trains). **Candidate interface definition**: ERTMS SUBSET-026 System Requirements Specification, ongoing work at S2R, EUG, UIC.
7	TMS-AE ↔ ATO-AT	This interface is connecting the specific ATO transactor to the TMS. **Candidate interface definition**: ATO over ETCS SUBSET-131 ATO-TS / TMS Interface Specification (X2Rail-1-131).
8	ATO-AT ↔ ATO-AV	This interface connects the ATO transactor to the ATO vehicle function, that controls the vehicle devices. **Candidate interface definition**: ATO over ETCS SUBSET-126 ATO-OB / ATO-TS Interface Specification.
11	APS-FOT ↔ OC	This interface connects the Advanced Protection System to the different types of trackside assets by using an OC (Object Controller) according to EULYNX specifications.
h1	TMS-PE ↔ TMS-PE	The PE handover interface is used between two TMS-PE to exchange information about each other's areas and to pass a Moveable Object from one region to the next.
h2	APS-SL ↔ APS-SL	The SL Handover Interface is used to pass a Moveable Object from one APS-SL to the next (adjacent APS-SL). Therefore, it must be possible to request a Movement Permission that start in one instance of SL and ends in another SL. The two instances can be from two different IMs or the same. **Candidate interface definition**: Basis EULYNX SCI-ILS.
h5	APS-MOT ↔ APS-MOT	The MOT Handover Interface is used to pass a Mobile Object from one APS-MOT to the next.
h6	APS-MT ↔ APS-MT	The MT Handover Interface is mainly the ERTMS interface to hand over a vehicle from one APS-MT to the next APS-MT. **Candidate interface definition**: ERTMS SUBSET-039 FIS for the RBC / RBC Handover; ERTMS SUBSET-98/129/26 RBC-RBC Safe Communication Interface.
h7	ATO-AT ↔ ATO-AT	The ATO Handover interface is used to handover a vehicle from one ATO trackside to another ATO trackside. **Candidate interface definition:** ATO over ETCS SUBSET-132 ATO-TS / ATO-TS Interface Specification.
h8	IAM ↔ IAM	The IAM-IAM interface allows to find out the communication parameters for RCA Blocks (like APS-Topo4 or DCM) in other networks.
A1	TMS-PE ↔ PE legacy	The PE legacy interface allows to coordinate the TMS-PE with a legacy PE, that controls a neighbouring region. Related to h1.
R1	APS-MT ↔ legacy RBC	Connect an APS-MT to an ETCS RBC using the ETCS RBC-RBC protocol. **Candidate interface definition**: ERTMS SUBSET-039 FIS for the RBC / RBC Handover; ERTMS SUBSET-98/129/26 RBC-RBC Safe Communication Interface.
I1	OC ↔ legacy IXL	This interface is used for switching one OC to be controlled by two different IXL (for large segment commissioning). Also called the "Y-switch".

ID	Between	Description
I2	OC ↔ block	The Block Interface is used at the system border (adjacent interlocking) to enter and leave Moveable Objects. This electrical interface would be addressed over a new OC of type “Block”. A design alternative is currently evaluated, which would remove this interface and move it to the h2 interface.
21	ATO-AV ↔ VS	The ATO-AV ↔VS Interface is used between the two device controllers ATO-AV and VS to coordinate their parallel vehicle control. It includes the following information transfers: ATO Status ("AD Mode request", "ATO Engaged"), ETCS Train Data, Dynamic ETCS Data (e.g. "EB is requested", "Positioning Information", "MA Information", "Speed Information"), ETCS supervision information. **Candidate interface definition**: ERTMS SUBSET-130.
22	VL ↔ VS VL ↔ ATO-AV	The Vehicle Localization Interface is an interface to forward the localization information computed by Vehicle Locator to Vehicle Supervision and ATO Vehicle. It transports: position, speed and acceleration with confidence intervals.
31	TMS-PE ↔ WB	The operation plan, the operation status and all object control requests are part of this interface / API. This will also be a mobile UI that provides the user interaction for the Personnel at Trackside including but not limited to entering requests (e.g. request a Shunting Movement) or display current information about next capacity usages.
33	APS-Topo4 ↔ WB	This API / interface provides safe input / output functions for the "APS-Topo4".
34	EDP ↔ WB	This API / interface provides rich input / output functions for the Engineering / Data Preparation system “EDP”.
35	DCM ↔ WB	This API / interface provides input / output functions for the Device & Configuration Management system “DCM”.
36	IAM ↔ WB	This API / interface provides safe input / output functions to edit the identity and access register.
37	DM ↔ WB	This API / interface provides rich input / output functions to monitor and analyse diagnostic data.
41	DM ↔ many	The Diagnostics Interface is used between Diagnostics & Monitoring and the monitored blocks. **Candidate interface definition**: EULYNX SDI
42	DCM ↔ many	The Device and Configuration Management Interface is used between Device & Configuration Management and the managed blocks. **Candidate interface definition**: evolution of EULYNX SMI
43	IAM ↔ many	The Identity & Access Management Interface provides services for authenticate and authorize human user and technical systems.
46	EDP ↔ APS-Topo4	The EDP ↔ APS-Topo4 Interface is used to provide the needed acquisition of data to APS-Topo4 and to return the validated data back to APS-Topo4.
47	APS-Topo4 ↔ DCM	The APS-Topo4 ↔ DCM interface is used to synchronize the device references.
48	DCM ↔ IAM	The DCM-IAM interface is used to synchronize device capability rights
49	DM ↔ DCM	The DM-DCM interface is used to synchronize the device status
not numbered	OC ↔ TA	The physical interface from different type of OCs to different type of trackside-assets (e.g. 4 wires for a point machine). Out-of-scope for RCA.
not numbered	many ↔ VD	The interfaces from on-board components such as ATO, VS, VL to the vehicle actors, and sensors are out-of-scope for RCA.

Table 1: List of component-to-component interfaces

6.1.5. Where is the interlocking?

In RCA the functionality of today’s interlocking and RBC has been allocated to several well-defined components with clear responsibilities. The following picture shows the architecture, when the components with the “APS”-prefix are shown as an overall “APS” (Advanced Protection System).



6.2. RCA technical architecture

The RCA component architecture takes into account technical considerations (such as dealing with objects and devices) but focuses mostly on functional or logical aspects i.e. without considering where and how the components actually run. In this chapter we outline the technical aspects needed to deploy and run an RCA-based system.

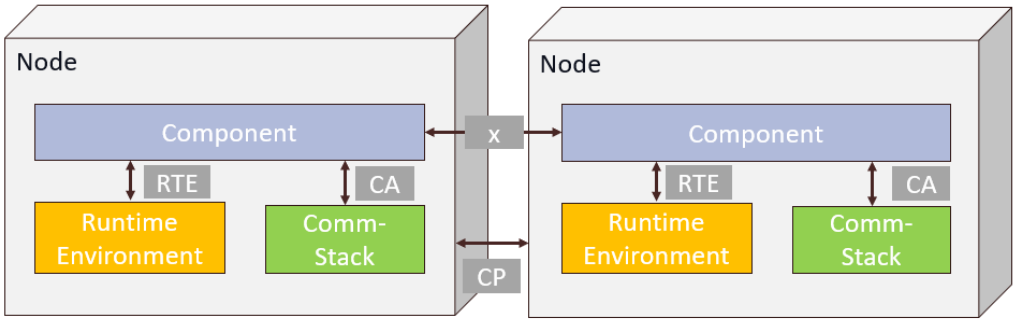
We divide the technical architecture into 3 perspectives:

- **Deployment architecture**: what computation, what data will reside on what kind of resource nodes at which locations connected with which communication infrastructure;
- **Runtime environment**: architecture of each node: HW, OS, SW;
- **Communication architecture**: how exchange of information (on different levels e.g. between components and between nodes) is implemented (as a stack).

We use the term “node” to denote a physical resource to which functionality in form of a component can be mapped. The following diagram shows:

- RCA components (as described in the interface architecture) are deployed on nodes. Not shown: several components can be deployed to the same node;
- The nodes include a “Runtime Environment” and a “Comm-Stack”. The interfaces towards the components are standardized (as “RTE and “CA” (“Comm-API”)); see the document “RCA Beta – Platform independence” for more details;
- Communication between nodes happens physically over a set of communication protocols (“CP”). Logically the components have specific interfaces as described in the interface architecture;
- Note: for safety-relevant functionality mechanisms such as voting, guaranteed delivery, etc. are necessary. We expect some of these mechanisms to be provided in the runtime environment and / or Comm-Stack. This normally entails that the component must obey specific requirements while interacting with the underlying systems;
- Note: important non-functional requirements such as performance and availability are strongly driven by the features and overall architecture of the nodes;

- Note: it is likely, that there will be two or more types of runtime environments, depending on required safety level or other non-functional characteristics;
- Not shown: in RCA the nodes may be virtualized or even be “running in the cloud”;
- Not shown: several components may share the same node.



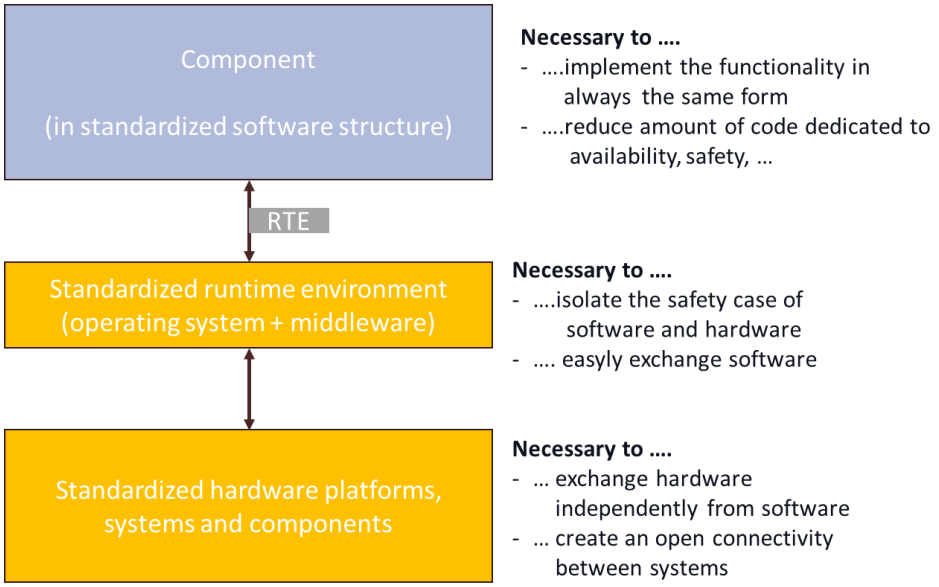
6.2.1. Deployment architecture

The specific deployment architecture is not specified by RCA, except the technical implementation of functional interfaces and the standard runtime environment for applications. The concrete deployment architecture depends on local decisions by the implementing IM. In chapter 8.1 “Overview of possible target configurations” some examples of deployment architectures will be given.

We strive for a standardized runtime environment for each RCA component. This means that an RCA component development can focus on functional aspects, with most technical and some non-functional aspects being delegated to the runtime environment.

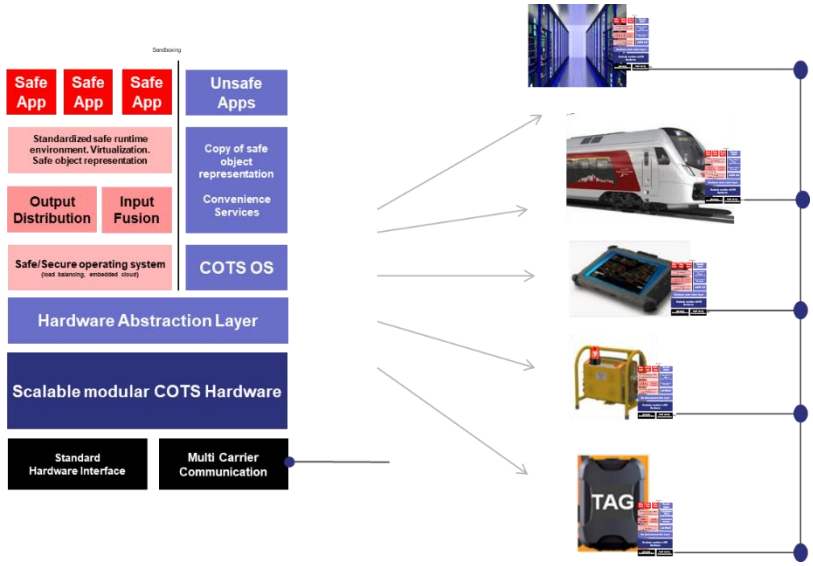
6.2.2. Runtime environment

The following diagram shows the important principle of separation of functional SW (the components), the OS (with common middleware such as safety functions) and the hardware. These elements are loosely coupled over well-defined interfaces and allow a) the components to be separately procured from the computing infrastructure and to b) share the computing infrastructure among several components.



An even more ambitious example is given in the following diagram (target for the program smartrail 4.0):

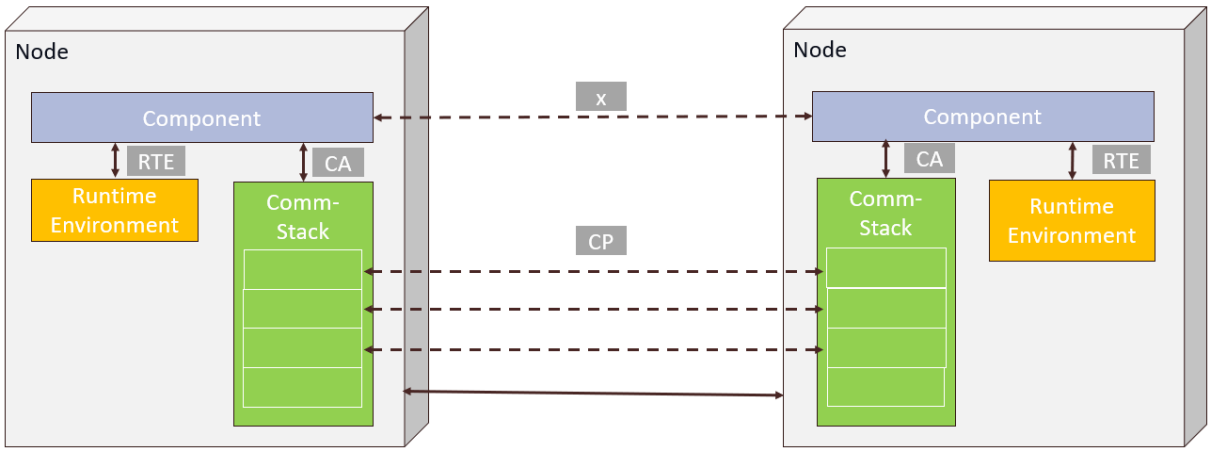
- Modularized platform, distinguishing safe and unsafe components (“apps”);
- Using the same platform on all devices (from data centre to small personal devices).



6.2.3. Communication architecture

RCA specifies communication over the “whole stack” i.e. not only on the “logical” level between components but all the way down to physical details to allow the exchangeability of logical components, but also of physical nodes (“boxes”). This includes integratability between nodes from different suppliers.

The following diagram shows the principle of an RCA communication stack linking logical components and physical nodes:



The interfaces RTE, CA and CP are described above. The interface “x” stands for a specific interface between 2 components, as described in 6.1.4 “List of component-to-component interfaces”. The “x” stands for the “application-level”, abstracted communication between 2 components, physically the communication happens over the “Comm”-Stack.

For the RCA communication stack, we use the following principles:

- Upper layers are independent of lower layers (this includes “bearer independence”);
- Ideally one stack can be used for all components / nodes; at the same time, it is possible to use more than one stack in an RCA-based system;
- The stack includes functions such as transport of packets, sessions, security layer, safety layer, application protocols (including protocols for versioning, upward / downward compatibility);
- The stack is to be based on (a selection of) existing and open protocols (to ensure long-term support).

- When components are co-located on the same node, simplifications at the lower levels are possible.

The concrete architecture needs to be developed.

6.3. Relation to other architectures: ERTMS, EULYNX

6.3.1. ERTMS

RCA is based on the architecture of ERTMS/ETCS L3 and re-uses existing ERTMS/ETCS specifications / protocols. Major building blocks include: possibility for inclusion of new localization methods, a geometric safety logic, fully supervised shunting, ATO GoA2-GoA4, FRMCS, and new options for fall-back systems. Some of these are already known as “game-changers”.

6.3.2. EULYNX

Relation to EULYNX on a formal level:

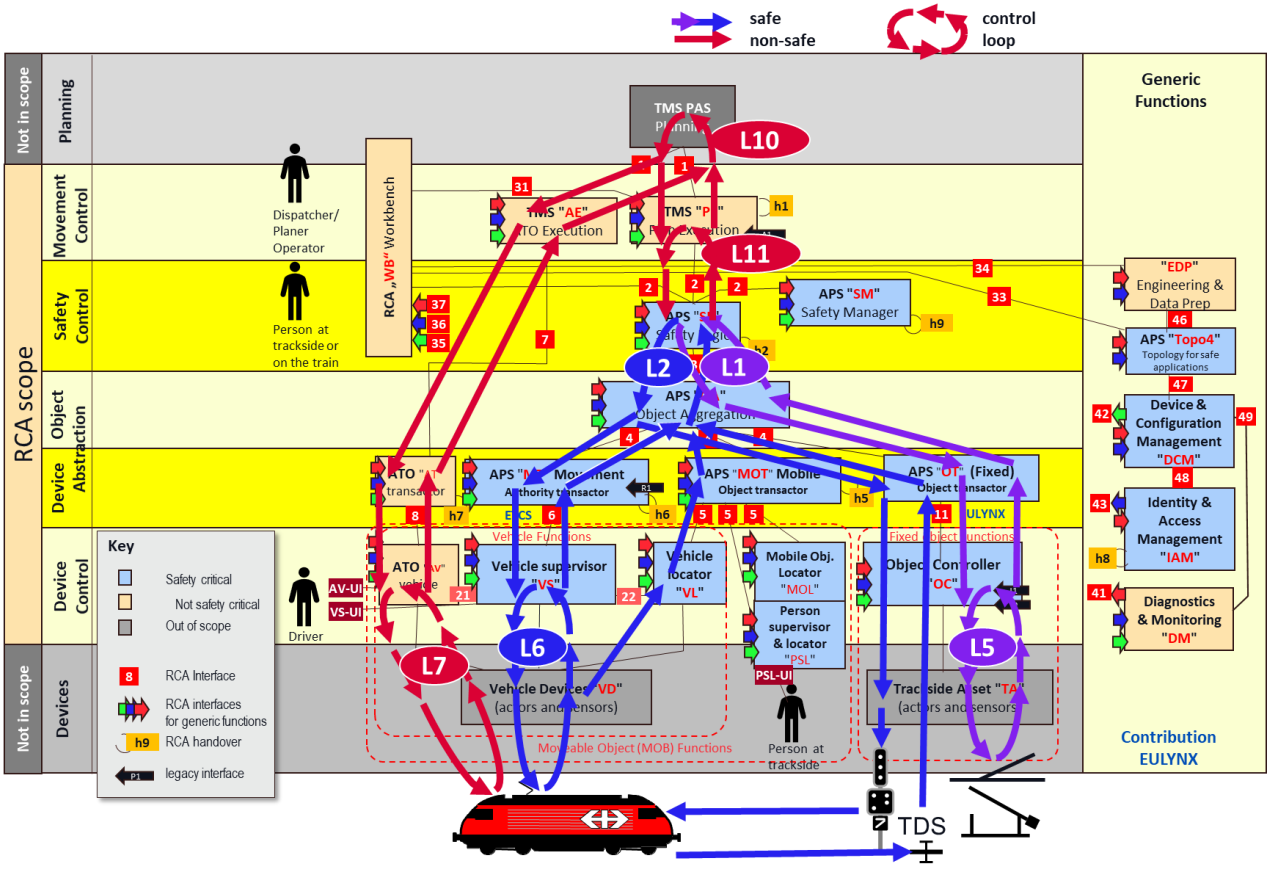
- Whenever feasible, RCA re-uses concepts, processes, techniques developed for EULYNX;
- Functional, non-functional, and implementation architecture are used similarly;
- Physical architecture of EULYNX is a part of the technical architecture of RCA.

Relation to EULYNX on content level:

- RCA integrates EULYNX interface definitions for trackside assets into the overall RCA;
- EULYNX object controllers will be part of RCA-based systems.

6.4. Important control loops

The control loops are the functional “highways” of a CCS. They help illustrate the interfaces and how they are used. The control loops do not specify any additional behaviour. They tend to be performance-critical and to drive non-functional requirements. The following diagram shows important control loops in RCA:



The following table provides a short description of these control loops:

Id	Description
L1	Interplay between: • APS-SL prescribes state: trackside device in a certain allocation state; • "Real world" (as seen by the device controllers): has effective allocation state, which must comply with prescribed state. Downward: Demanded allocation state. Upward: Effective allocation state.
L2	Interplay between: • APS-SL prescribes state: Grant specific Movement Permission to a Moveable Object; • "Real world" (as seen by the device controllers): effective location, which must comply with prescribed state. Downward: Movement permission (permission to change location). Upward: Effective location of the Moveable Object in the real world. Note: There are multiple paths for detecting the location and granting the Movement Permission.
L5-L7	Local control loops, not directly in scope of RCA.
L10	Interplay between: • TMS-PAS, where an operation plan is established (and frequently updated / optimized); • TMS-AE and TMS-PE executing the plan as faithfully as possible and updating TMS-PAS with "real-world" information by providing an execution status.
L11	Interplay between: • TMS-PE, decomposing the operation plan from PAS into single requests for changing trackside device allocation state (L1) and for granting movement permission (L2), issued when all the preconditions regarding "device allocation state" and "location of the moveable objects" are met; • APS-SL, ensuring TMS-PE requests are safely executable and controlling their execution, updating TMS-PE with the "effective device allocation state" and the "location of the moveable objects". Downward: Requests for changing trackside device allocation state (L1) and for granting movement permission (L2). Upward: "effective device allocation state" and "location of the moveable objects".

Note: The signals shown in this diagram may be needed at the border (border to another supervision system or to non-supervised area) of an RCA-based system.

6.5. Principles of the safety logic

In RCA different implementations of the safety logic can be chosen, as long as the interfaces specifications are still respected. However, the interfaces are based on a few basic principles outlined below.

Note: the safety logic is in RCA implemented in the components "SL" (Safety Logic) and "SM" (Safety Manager). This naming may lead to confusion and may be revised in the future.

Basic functions of the safety logic:

Following is a list of all main functions including a short description:

- The safety logic in APS-SL verifies if the requested movement (e.g. train movement) or object state change (e.g. move the points) is safe. This means that the APS-SL checks that there will not be a collision and that the object is not occupied by another movement or reservation;
- The safety logic APS-SL is the only component allowed to demand a state change of an object (e.g. level crossing). After the safety check of the requested state change (sent e.g. by the TMS) the APS-SL demands the state change from the corresponding object via the object controller;
- The operating state is the "heart" of the whole system. All current states (e.g. moving objects) and demanded state changes are stored in it. The object aggregation deduces the state of the real objects from the information it collects from all the objects (e.g. trains, points, etc.) and stores these states in the operating state;
- The safety logic in the APS-SM monitors all operations on the network through the operating state. If the APS-SM detects a violation of the safety rules, it triggers a measure to minimize the risk (e.g. emergency brake, warning). Following are two examples, there will be other risk minimizing measures:
    - The APS-SM may demand emergency stops if a violation of a safety rule is detected (e.g. train is detected outside the limits of permission);
    - The APS-SM may demand a warning if an object violates a safety rule (e.g. worker is too close to a train movement);
- The safety logic (in APS-SL and APS-SM) explicitly ensures only safety of movements but does not consider any operational implications (e.g. does this movement lead to a congestion). This deliberate design decision enforces a strict separation of concerns which provides several benefits, like separate lifecycles of the business logic and the safety logic, thus enabling fast, innovative and cheap improvements of the business logic and reduced development and maintenance costs for the safety logic.

Basic concepts of the safety logic:

Utilization Permission	A utilization permission is a permission to utilize a geometric area of the network topology under defined utilization conditions. There are two types, Movement Permission and Danger Area / Usage restriction area.
Movement Permission	A movement permission is an authorization to move in a specific direction for a specific distance according to a given speed profile. This includes data on track-conditions as known today in ERTMS/ETCS. The movement permission is requested by the traffic management system and verified by the APS-SL. After verification the movement authority is sent to the moving object (e.g. Movement Authority in ETCS). The moving object must stay inside its movement permission at all times. Movement permissions may overlap under certain conditions (e.g. joining).
Danger Area / Usage restriction area	It is possible to set a danger area over a certain part of topology (e.g. track segment). A danger area request is typically submitted due to an exceptional situation (e.g. landslide, maintenance work, etc.). This request may be submitted by the traffic management system but also by the safety manager APS-SM (watch dog). A danger area and a movement permission may overlap under certain conditions (e.g. construction vehicle must enter in a construction site).
Utilization Condition	A utilization condition defines how a certain geometric area may be used (e.g. maximum speed, allowed driving direction, allowed train type).
Safe Distance	The safe distance (in time and space) between two consecutive utilization permissions is needed for safety reasons. To ensure these safe distances, Risk Buffers will be set at the boundary of the utilization permissions (e.g. movement permission).

Rationale for splitting safety logic into APS-SL and APS-SM:

See also the component descriptions in chapter 6.1.3 “List of components”.

- APS-SL: The main purpose of the APS-SL is to react to requests from the plan execution by either granting them (if safe) and relaying commands to vehicles or trackside objects, or rejecting, based on the safety logic and the knowledge of the operating state. This implementation is expected to be almost universal (except for parameters such as overlap).
- APS-SM: the main purpose of the SM is to observe the operating state and to react to situations becoming unsafe due to external effects (e.g. point lost position, vehicle no longer localizable, etc.). Here we expect to use a rule engine, with different rules enabled in different IMs)

So, the main reasons for separation are:

- differing activation model: request / reply vs continuous observation
- differing degree of variance among IMs[19]

It is not the role of the APS-SM to do technical monitoring of the operation of the APS-SL, or to be a diverse implementation of the APS-SL. Both APS-SM and APS-SL must attain their SIL4 rating with built-in mechanisms of their own.

[19] This is a hypothesis for the moment, which will be confirmed or invalidated during the next design steps.

7. Cross-cutting concerns, non-functional requirements

7.1. Non-functional requirements and how to handle them in RCA

Non-functional requirements dominate complexity and cost of an architecture for systems where safety and availability are critical. For RCA the following points are essential:

- Missed critical NFRs may invalidate architectural choices;
- Critical NFRs have to be sensibly propagated and / or allocated to the components and interfaces;
- NFRs may lead to the need of additional “technical” functions (such as a monitoring system);
- NFRs may lead to the need for additional interface specification (such as an interface from each component to a central monitoring system).

The following list can help cover all types of NFRs: Safety, RAM, Physical robustness, Security, Capacity, Time Behaviour, Scalability, Reusability, Portability, Adaptability, Modifiability, Testability, Exchangeability, Monitoring & Diagnostics, BCM, Interoperability, Usability, Form and fit, EM radiation / robustness, Environmental protection

For RCA we will have a) to identify critical requirements, b) to have a top-level strategy for dealing with these requirements and c) ensure requirements are propagated to the Working Groups.

7.2. Data management

Although mostly “invisible”, efficient handling of data plays a crucial role for the success of RCA-based system.

“Runtime” data

An RCA-based system will collect data at a new granularity and precision:

- At any moment in time (with a resolution of 1-few seconds and a few meters) position and speed of trains (and other objects) will be reliably known in a central system. In addition to its use within the CCS system itself, this data can be very valuable for customer information, logistics, optimizing the system, usage pattern (for condition-based maintenance), etc.;
- When going for ATO GoA3 or GoA4 a lot of additional sensor data will be needed and made available. Note: GoA3 / 4 has yet to be integrated in RCA.

Data for construction and maintenance phases

With RCA the following data-driven automation potentials can be achieved:

- Automated topology recording: create a safe topology data base with a high degree of automation (using automated object acquisition and mapping technology);
- Automation of the project management and documentation steps for CCS projects, including data exchange with authorities and suppliers.

These automation / efficiency opportunities rely on mastering data management processes. To make the process manageable, the RCA must clearly define the needed data items and the required data quality, which includes attributes such as completeness, consistency, timeliness and accuracy.

7.2.1. Need for a shared data model

A shared data model facilitates communication between people, processes and systems. On the other hand, efforts on “global” data models typically fail. Priorities for the data model in RCA are:

- Identifying, describing and ultimately specifying data items appearing in several RCA interfaces;
- As a special case of the above: how to reference geographical, topological data.

In RCA this is even more important than for EULYNX, since RCA has conceptually new interfaces, a richer information model (e.g. positioning information) and more complex (cascading) interactions.

The scope of the data model in RCA will go beyond the scope of EULYNX but will re-use existing EULYNX assets. RCA will define its data model considering existing effort such as S2R, RailTopoModel, etc.

7.2.2. The role of Data preparation

The importance of data modelling and data preparation has already been noted in EULYNX:

> EULYNX also standardizes data preparation. This consists of designing the data structures that capture the information a supplier needs to build a signalling system from the ground up. This is a radical improvement from the present where the parties involved in (re-)signalling projects exchange heterogeneous and proprietary datasets, often on paper. In the future, the signalling industry ingests standardized EULYNX data into their proprietary design toolset. As before, the signalling industry processes the data and then returns the enriched data, in EULYNX format, to the IMs who absorb this as-built data into their asset management systems. IMs and signalling industry retain their proprietary formats and tooling but EULYNX harmonizes data exchange. […] The case for automating the data transfer process through well-defined standard data structures is obvious.

7.3. Concept of Capability-Based Application Protocols

The first step for exchangeable and upgradable components are well-defined interfaces (meaning all interfaces of a component). This allows changing one implementation of a component for another. This, however, is not sufficient when an upgrade requires different interfaces (e.g. additional attributes, different sequences, new functionality …). In this case, we need, as a second step, the architectural capability to introduce local upgrades without requiring upgrades on all “adjacent” components. (A third, orthogonal capability of the architecture is to facilitate the upgrade of components to allow upgrading of several components at once).

Capability-based application protocols (CBAP) provide a mechanism to facilitate local upgrades without requiring upgrades on all “adjacent” components (step 2 above).

Well-known examples of the use of capability-based protocols is the Bluetooth- or the USB-standard.

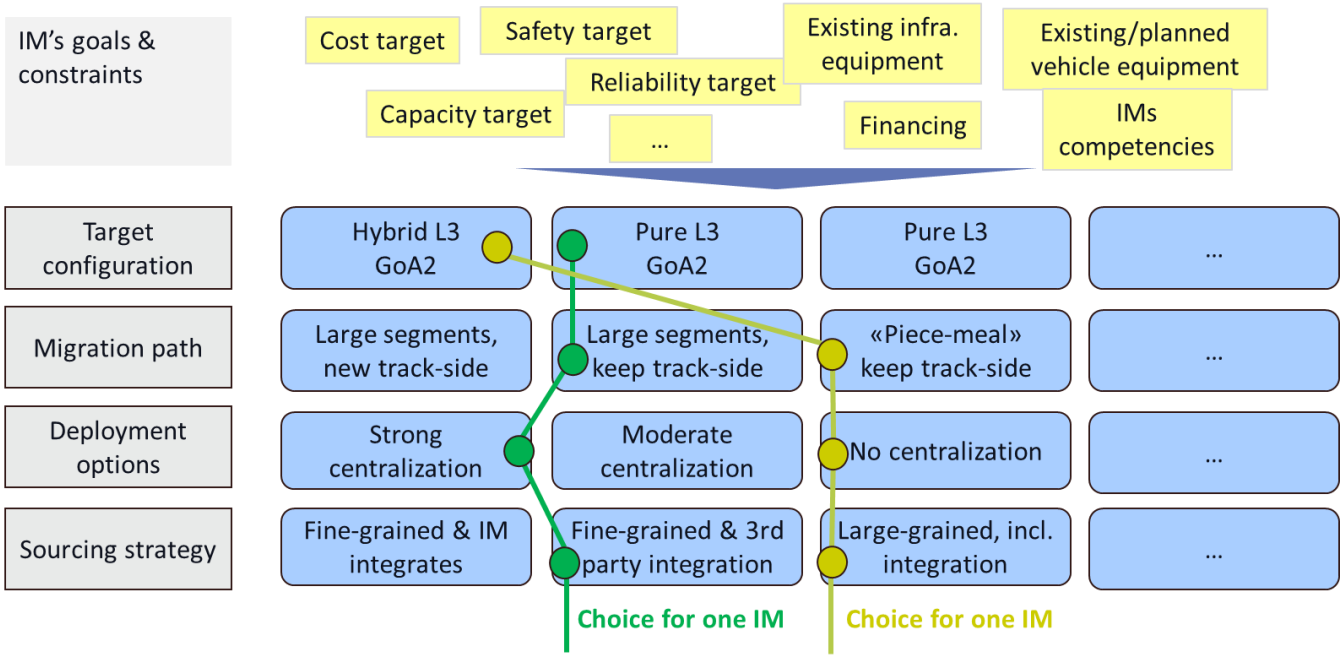
The concrete architecture needs to be developed.

8. Overview of possible RCA usage scenarios

This chapter gives an overview, how RCA can be applied by an IM. IMs may have different usage scenarios. The goal of RCA is to support different scenarios without including too much variability in RCA interfaces and components. Ideally different usage scenarios can be covered by different combinations of RCA-compatible components. In the terminology of product line engineering these are variation points and possible variants. Here we will deal with the following variation points (there are more):

1. We describe a set of **target configurations** (targets in the sense of important feature / technological choices), which the IM might want to achieve with a modernization project. This also depends on what types of lines RCA is to be applied.
2. We also describe a set of **migrations scenarios** describing steps to achieve a target configuration. The migration scenario therefore depends on the current configuration and on the chosen target configuration.
3. We describe a set of **deployment options** (deployment in the sense of how functions / components are physically distributed).
4. Different sourcing strategies (e.g. in what bundles components / systems are procured) (not included in RCA Beta).
5. Degraded modes / Business continuity (not included in RCA Beta).

The following diagram shows the idea of different usage scenarios (incomplete and simplified).



RCA is not suitable for all target configurations (i.e. a target configuration based on traditional signals) and cannot efficiently support all migration scenarios. RCA is, however, suitable for a range of target configurations and a range of migration scenarios.

8.1. Overview of possible target configurations

The following table shows (as an example) some important parameters for a target configuration of an RCA-based system. For each parameter possible choices are indicated. Each parameter is also a variation point (i.e. an accepted variability in RCA). In green as example the choices for smartrail 4.0 (SR40 release 3).

Parameter / Variation point	Choices	Comments
ATO	a) (GoA1) b) GoA2 c) GoA3 / 4	An RCA-based system offers higher localization resolution and better capacity usage (geometric interlocking). To fully

Parameter / Variation point	Choices	Comments
		use these benefits, trains must follow an optimal trajectory. This strongly suggests using GoA2 or higher.
Safe high-resolution localization by trains.	a) Only Trackside Balises b) Trackside Balises + high-end virtual balises (e.g. inertial measurement + odometry + GNSS + other technologies) c) Only high-end virtual balises d) Low-end virtual balises only based on GNSS, used for low density traffic e) other	RCA allows several combinations of localization technology, because we expect different needs and on-going technological evolution. (Remark: "Virtual Balise" is used here as name for a safe onboard function, that provides continuous localization and track information to different onboard applications like EVC).
Trackside train-detection devices	a) None b) Very few (selective) c) As-is d) Typical number e.g. for L2 e) Hybrid Level 3 setup (reduced train detection) f) New technologies (like FOS or micro balises)	Reducing train detection devices is an important contributor in the business case to reliability and cost savings. The reduction in classic train-detection devices parallels the availability of safe high-resolution localization by trains (see above).
TMS	a) Static / mostly manual b) Some automated optimizations c) Fully automated re-planning every n^{th} minute d) With / without automatic operational conflict resolution e) Geometric / precise / dynamic / adaptive real-time train control and movement / speed optimization versus block based and static train control	Although the implementation of TMS PAS (schedule planning, capacity management) is out-of-scope in RCA, it's mode of operation has an important influence on the business case (cost savings, capacity).
Interlocking / APS	a) "Geometric" safety logic, risk assessment on runtime based on a generic function b) Traditional block-based safety logic	Since the interlocking determines the "resolution" of train control, the capacity effects of an RCA-based system depend on having an APS with a geometric logic. This also reduces the effort for engineering, safety case and data prep when changing the IXL. So, the interface design will implement in the full RCA architecture only a geometric description of all transferred information objects (like movement

Parameter / Variation point	Choices	Comments
		authority). Nevertheless, a traditional “block-based” component can be used as long as a conversion “geometric description <> block description” is done at the component interface.
Communication to trains	a) FRMCS b) GSM-R (with GPRS) c) (other)	IN RCA, the mobile (and fixed) communication is largely isolated in the communication stack. Therefore, several solutions are, in principle, possible. For dense traffic with interoperability criteria, we think that FRMCS is the way to go. Since GSM-R will be end of life in the next decade it can only play a role in the short-term migrations.
Communication between object controller and trackside devices (last mile)	a) Classic cabling (“copper”) b) Energy + communication bus (fibre optic, radio)	This interface is out of scope of the RCA architecture, but it influences cost and duration of the migration quite much.

The combination of these options lead to different setups, which can be for example a “classical ETCS Level 2” configuration, or a very cost-efficient implementation of full moving block with strongly reduced trackside assets and a real-time TMS controlling the traffic fully automatically.

8.2. Overview of possible migration scenarios

In most cases RCA will be applied in a “brown-field” situation i.e. an RCA-based system is rolled-out in the immediate presence of existing (legacy) CCS systems. Migration is the process of moving from the existing situation to the target state while optimizing cost, time, risks, and impact on ongoing operation.

Considering the scope of RCA, the following interactions with the environment are crucial for migration planning. RCA-compatible system to…

- … track-side equipment (connecting and configuring re-used or new equipment);
- … vehicle equipment (network access conditions);
- … communication infrastructure;
- … planning functions (TMS);
- … associated processes & people;
- … neighbouring segments not (not yet, not ever) RCA-based.

Migration must address the following questions:

- Coordination of the RCA-compatible system with systems in its environment (see above);
- Overall migration strategy (fast / slow, lifecycle-driven / effect-driven, large-grained / fine-grained, accepted disruption to service, …);
- Migration “tactics” for migration of a given segment, including dealing with segment borders.

Typical overall migration strategies will be analysed and supported. This can be for example a complete CCS replacement or a complete replacement except trackside assets. It can be a fast industrial migration in large steps, or a slower migration in smaller steps.

8.2.1. Migration mechanisms in RCA

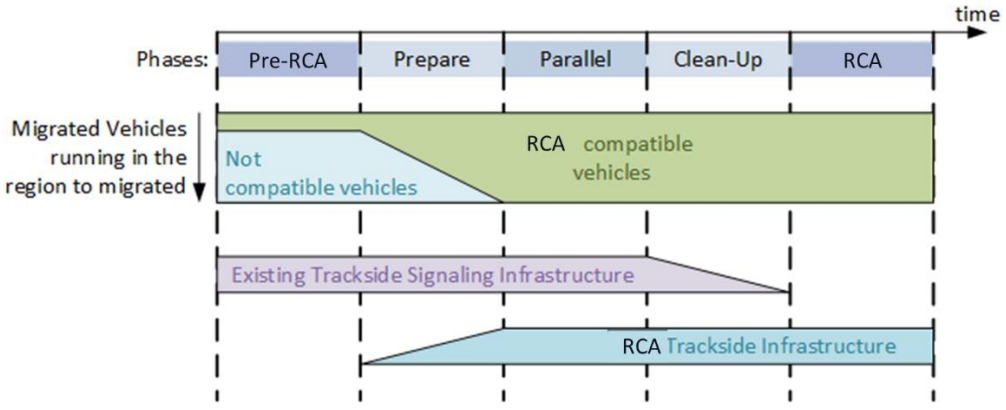
The RCA will be designed to support functionalities, that are especially designed to reduce the workload for the migration process (and also the lifecycle workload):

Migration mechanism	How it can be used
Object-Controllers with "Y-Switch" [I1]	The "Y-switch" allows to connect an old and a new interlocking to an existing trackside asset at the same time (switchable). This is useful when migrating to an RCA-based system while keeping (many) track-side devices. The (physical) network can be prepared "non-intrusively" by rolling out OCs with a Y-switch. Allows using (short) test windows. Provides fallback. Simplifies to prepare large segments of the network for the commissioning and avoids temporary interfaces, which are normally a larger cost block in the migration.
Legacy-Interface OC to neighbouring block [I2]	Connecting an RCA-based segment to a non-RCA-based segment.
Object Aggregation for Localisation information	Allows e.g. slowly reducing existing train-detection devices while ramping up auto-localisation by trains.
Legacy-Interface "Movement Authority Transactor" to RBC [R1]	Connecting an RCA-based segment to a non-RCA-based segment.
Legacy-Interface in TMS "Plan Execution" to existing interlockings [A1]	Ensures that the plan execution also works on adjacent non-RCA-based segments.
Legacy-Interface (out-of-scope RCA) of TMS "Planning" to other (neighbouring) planning systems.	Ensures that the plan synchronization also works on adjacent non-RCA-based segments.

The brackets [x] reference interfaces defined in chapter 6.

8.2.2. Principle of migration scenario

The following diagram shows a possible migration scenario with several migration phases:



Depending on the usage of the fleets on the network the duration of the "parallel" phase can vary very much.

The exact steps depend on the target configuration and how the vehicle migration can or cannot be steered. The table focuses on the generic steps needed to switch over to an RCA-based configuration, it does not completely explain migration.

Phase	Description
Pre-RCA	Before the RCA migration for that region.
Prepare	• All vehicles that will run in the region after the migration must be migrated to the RCA requirements before RCA is set into service for that region. This needs to be published early enough in the network statement. • The trackside RCA infrastructure is installed.
Parallel	• It is possible to switch between the existing trackside equipment and the RCA trackside infrastructure back and forth – only for testing reasons. There is always only one of the systems actively controlling the vehicles and trackside assets. • This is used such that the new installation can be tested and such that after the migration the existing infrastructure can be used as a fall-back. • Parallel means, that a line can be switched between a non-RCA-compatible (existing, Class-B) and an RCA-compatible state only for testing reasons. “Parallel” does not mean, that one line has a Class-B and an RCA-compatible function set at the same time (“overlay” installations like ETCS Level 2 + trackside signalling on the same line, mixed usage by different trains). These overlay architectures will not be supported by RCA because they lead to a very complex requirement and interface set, complex operations, lower availability, very high lifecycle cost and many national variants.
Clean-Up	The existing trackside equipment is built back.
RCA equipped	After the RCA migration for that region.

8.2.3. Vehicle migration / OCORA

For the commissioning of an RCA-compatible line, all vehicles must be equipped according to the chosen setup for the line, but in minimum with the ERTMS onboard equipment. Additionally, for example mobile localization / virtual balises, ATO, TIMS or safe length measurement can be made mandatory. To “solve CCS”, the vehicles (and drivers) have to be brought into the equation. While RCA focuses on the IM perspective, we bring up some discussion points regarding vehicle migration:

- Vehicle migration is already a challenge for ERTMS (without RCA). Equipment of vehicles today is slow, very expensive and not future-proof (i.e. even upgrades are costly). Too often compatibility issues hinder the migration;
- While arranging financial help for vehicle migration is important, we must also address the question, how to get to more efficiency for the CCS-related equipment of vehicles;
- Under the title “OCORA” (Open CCS Onboard Reference Architecture) several railway undertakings have started an initiative similar to RCA with the goal to set standards for the CCS-related on-board architecture. The target of OCORA is a modular architecture for the CCS functionality on the vehicle that strongly reduces the costs and workload for upgrades and extensions including the simplification of the homologation.

8.3. Migration perspective for recently completed / on-going existing ERTMS installations

ERTMS implementations based on the industrial products, that are available today, have the disadvantage of a) having an un-attractive performance- / price-ration and of b) being difficult and complex to evolve / upgrade / maintain. This is not a problem of ERTMS itself, which is a specification and rule-set. It comes from too small project volumes and from the high market segmentation in Europe, which does not allow to amortize the development of high-quality products with a high grade of automation of installation, operation, and maintenance processes. Also, the specification has not been sufficiently precise from the start to avoid integration / interoperability problems. Finally, very often complex “hybrid” architectures (old technologies connected to new technologies) are used because there was no economic path visible in the past to amortize a complete reinvention of all CCS systems.

This is the main reasons for starting RCA. At the same time, major “existing” rollouts are on-going or in advanced planning stages, for example because the age of assets requires a replacement of them today. What can RCA contribute to these projects, which do not want to wait until RCA has materialized?

In the RCA context we use the very important requirement of “Upgradeability” in three senses:

- A migration, that today starts with an existing architecture shall be able to change its migration method when RCA is available. This will be an important change, but the target of RCA is to reduce the costs as much as possible. This point has aspects concerning architecture, technologies, operational rules, engineering rules, procurement, and contracting;
- RCA as a building block architecture shall allow, that an IM or RU changes the mixture of components easily and also after initial deployments (connectivity, “plug & rail”). Example: When introducing onboard localization, the existing trackside train detection perhaps will be used in parallel for a while (simpler safety case, high performance from the start). But afterwards the trackside train detection may be reduced or not reinvested any more. These “smooth” migration steps shall be supported by the interface mechanisms and qualities of RCA;
- RCA itself expects innovations and changes because of the ongoing digitization. So, all systems and even the interface design shall include mechanisms for extensions and simple changes.

RCA shall develop and recommend requirements and methods for projects applying existing ERTMS/ETCS architecture.

8.4. Overview of possible deployment architectures

Deployment architecture describes how components (as described in the “RCA interface architecture”) are “mapped” to the real world. When mapping logical components to the real world, 2 decisions must be made:

- The “cardinality” of the component i.e. how many instances of the component are operated. Here we do not consider instances multiplied for availability or safety reasons (“standby” or M-out-of-N-configurations);
- Where the component(s) are physically deployed.

The following table shows some of the possible choices for an RCA-based system. The components refer to the “RCA interface architecture”.

Component	Cardinality / Extent (options)	Where deployed (options)
TMS-PAS	• 1 per IM • 1 per TMS-region (if needed for organizational or performance reasons) Higher multiplicity complicates overall optimization but makes attaining local optima easier. TMS-PAS (plan construction) is out-of-scope for RCA. The deployment options for TMS are, however, an important driver for RCA deployment options.	• Centralized data centre • Regionally centralized data centre • “in the cloud”
TMS-PE TMS-AE APS-SL APS-SM APS-OA APS-MT APS-MOT APS-FOT	A bundle of these components is instantiated once per APS-region. This bundling allows simple communication relationship between them. For each APS-region separated hardware nodes are used. The size of the APS-region is mainly limited by the following factors: • Performance (Scalability) For increasing the capacity of the rail network, frequent position reports (causing many messages) and fast control-loops are needed. Smaller APS-region will reduce the load per APS-region;	• Centralized data centre • Regionally centralized data centre • “in the cloud”

Component	Cardinality / Extent (options)	Where deployed (options)
	• Availability: By using multiple redundant hardware, the unavailability caused by hardware failures could theoretically be reduced to such a low number that huge APS-region could be built (whole country), but practically other factor like the following will lead to unavailability and will limit the size of the APS-region: ○ In the case of a planned maintenance work, it can happen that an APS-region must be taken (for a short time) out of service or that during maintenance the risk of an interruption increases. If an APS-region is smaller, it is easier to find a time for the maintenance work when no or only a few trains are running in the affected APS-region; ○ A redundant running software could become unavailable in the case of an SW-Bug, a security attack or an interruption caused by an operating error of the service personnel. By having smaller APS-region, the impact of such a problem could be limited and the problem can hopefully be eliminated fast enough, that it only occurs once. In addition, a new version of the software can first be deployed for a single APS-region (for a few month) such that possible SW-bugs will only impact that APS-region.	
VS, VL ATO-AV	• 1 per vehicle, possibly several per train	• On-board vehicle
OC	• 1 per trackside equipment • In some cases, grouped together (“Soft-OC”)	• trackside: included with trackside equipment • trackside: cabinets for a small set of trackside equipment • “legacy” interlocking building

8.5. Overview possible sourcing scenarios

RCA supports the following sourcing principles:

- RCA components will mostly be bought from suppliers. Some (software-dominated) components may be developed by IMs;
- RCA components can be procured “fine-grained” / “per component” and integrated by IMs OR
  RCA components can be procured “per component” and integration tasked out to a third party OR
  RCA can be procured in “large-grained” / “pre-integrated” systems.

How the system is sourced (for initial rollout and for later phases) is up to the national programs of each IM.

The planned vertical integration has, however, some influence on how RCA is applied. See chapter 4.6 “How the interface architecture is linked to procurement options”.

A more thorough treatment is necessary in the future.

