OCORA

Open CCS On-board Reference Architecture

CCS Communication Network

Evaluation

Document ID: OCORA-TWS02-010

Version: 3.00

Date: 08.06.2022

Management Summary

Today the interfaces between CCS components on the vehicle are proprietary. The proprietary interfaces do not allow to exchange CCS components from different suppliers. The OCORA architecture [7] aims for plug and play interchangeability within the CCS domain through the specification of a generic and open communication backbone, the CCS Communication Network (CCN) former called Universal Vital Control and Command Bus (UVCCB). The CCN itself will be modifiable in accordance with future technological evolutions by means of strict separation of the different communication layers (OSI Layers).

This document is based on the CCN evaluation reports of former releases [10], [11]. It provides all investigations and results of all phases, containing evaluations of different communication layers, data serialization formats, cyber security and network topology.

Due to the TSI-CCS 2022 with its new established Subset 147 covering the communication layers for the CCS network, the evaluations noted in this document in the chapters 3 and 4 have to be reworked in the subsequent release.

The CCN evaluations done in former release phases proposes the CCN to be a TSN Ethernet based network with the use of SDTv2 / SDTv4 as safety layer. In order to be able to integrate the CCN on the next generation of train communication network (NG-TCN) or establish an own ECN-like network for the CCN, every hard-real-time CCS device (e.g. Safe Computing Platform etc.) should have at least one TSN-capable Ethernet port whereas for soft- or non-real-time CCS devices a single standard non-TSN-capable Ethernet port is sufficient. Hard-real-time devices can use both planes of NG-TCN with two TSN-capable Ethernet ports in order to improve reliability and availability.

On session layer TRDP 2.0, OPC-UA Pub/Sub (over TSN) or DDS/RTPS (over TSN) are suitable solutions. These three options will be further investigated considering the system architecture with platform/CCU and the subcomponents.

The currently proposed protocol stack of CCN is listed in the following table. Highly recommended standards to be used as reference for procurement in OCORA are listed in **bold** font.

Layer	Protocol for hard-real-time data		Protocol for soft- or non-real-time data
(Safety Layer[1])	**(SDTv2 / SDTv4)**		
Session Layer	TRDP 2.0, OPC-UA Pub/Sub or DDS/RTPS		
Transport Layer		UDP TCP	UDP TCP
Network Layer		IPv4	Ipv4
Data Link Layer	**Time-Sensitive Networking (TSN) IEEE 802.1**		**Standard Ethernet IEEE 802.3**
Physical Layer	**100BASE-TX or 1000BASE-T**		

Table 1: Protocol Stack CCN

The protocol stack allows safety-related and hard-real-time data traffic. For non-safety-related and soft- or non-real-time applications, standard TCP/IP or UDP/IP data traffic over standard Ethernet (IEEE 802.3) can be used on the same physical layer of the CCN.

As a result of the evaluation of data serialization formats it is proposed to use a mix of Bitstream and JSON / XML over the CCN. For time-critical CCS-applications the interfaces will anyway be specified (e.g. in ERA SUBSETs) which allows Bitstreams as a non-self-describing but rather fast format. For non-time-critical applications, such as maintenance, a human readable data format like JSON or XML would be well suited.

The investigation of network architectures respecting cyber security shows that the zoning concept of the onboard networks is not clearly defined yet. The zoning concept must be discussed and investigated in the subsequent phases of the OCORA initiative. Furthermore, the CCS and TCMS domains must jointly elaborate the same understanding of the network architecture. At the end, the results shall be incorporated into the next version of IEC 61375 standard.

[1] Safety Layer is only applicable for safety-related data traffic.

Revision history

Version	Change Description	Initial	Date of change
1.00	Official version for OCORA Delta Release	SSt	30.06.2021
2.00	Official version for OCORA Release R1	SSt	26.11.2021
2.01	Consequences of Subset 147 Draft included	SSt	14.04.2022
3.00	Official version for OCORA Release R2	SSt	08.06.2022

Table of contents

Table of figures

Table of tables

References

Reader's note: please be aware that the numbers in square brackets, e.g. [1], as per the list of referenced documents below, is used throughout this document to indicate the references to external documents. Wherever a reference to a TSI-CCS SUBSET is used, the SUBSET is referenced directly (e.g. SUBSET-026). OCORA always reference to the latest available official version of the SUBSET, unless indicated differently.

[1] OCORA-BWS01-010 – Release Notes

[2] OCORA-BWS01-020 – Glossary

[3] OCORA-BWS01-030 – Question and Answers

[4] OCORA-BWS01-040 – Feedback Form

[5] OCORA-BWS03-010 – Introduction to OCORA

[6] OCORA-BWS04-011 – Problem Statements

[7] OCORA-TWS01-030 – System Architecture

[8] OCORA-TWS02-020 – CCS Communication Network – Proof of Concept (PoC)

[9] OCORA-TWS06-030 – Preliminary Cyber Security Requirements

[10] OCORA-40-003-Beta – UVCC-Bus Evaluation, Version 1.01

[11] OCORA-40-003-Gamma – UVCC-Bus Evaluation, Version 2.00

[12] COAT-STI-BUS004 Evaluation UVCCB SBB V1.00

[13] NewTec GmbH: UVCCB Study BUS Technologies, Version A6-final

[14] Selectron Systems AG: UVCCB Technology Evaluation Report, Version 1.0.1

[15] CTA-T3.5-D-BTD-002-12_-_Drive-by-Data_Architecture_Specification

[16] CTA2-T3.4-T-SIE-019-03 – Safety Analysis SDTv4

[17] CTA2 Technical Seminar Brussels, Drive-by-Data Architecture Presentation, 24.01.2020

[18] EN 50129:2018 – Railway applications - Communication, signaling and processing systems - Safety related electronic systems for signaling

[19] EN 50159:2010 – Railway applications - Communication, signaling and processing systems - Safety-related communication in transmission systems

[20] IEC 61375-2-3: Railway Applications – Electronic railway equipment – Train communication network (TCN) – Part 2-3: TCN communication profile, 2015

[21] IEC 61375-3-4:2013 – Electronic railway equipment – Train Communication Network (TCN) – Part 3-4: Ethernet Consist Network (ECN)

[22] IEC 62443-3-3: Industrial communication networks – Network and system security – Part 3-3: System security requirements and security levels, 2013

[23] CENELEC TS 50701: Railway applications - Cybersecurity, Version D8E5

[24] ERTMS/ETCS Subset-056: STM FFFIS Safe Time Layer, Version 3.0.0

[25] ERTMS/ETCS Subset-057: STM FFFIS Safe Link Layer, Version 3.1.0

[26] OPC 10000-6 Part 6: Mappings, 2017

[27] RTI blog about DDS and TSN: The Future for Real-Time Data Exchange?

[28] RFC 8949, Concise Binary Object Representation (CBOR), 2020

[29] UIC 559, Specification "Diagnostic Data Transmission" from railway vehicles, 2010

[30] Unife TWG OB ARCHI, 25th of June – Presentation of results on actions 1.1 & 2.1

[31] UNISIG-DSG-D-ALS-006, Presentation FRMCSready ERA TWG Modular Architecture, 03/11/2020

1 Introduction

1.1 Purpose of the document

This document is based on the CCN evaluation reports of former releases [10], [11]. It provides all investigations and results of all phases containing evaluations of different communication layers, data serialization formats, cyber security and network topology.

This document is addressed to experts in the CCS domain and to any other person, interested in the OCORA concepts for on-board CCS. The reader is invited to provide feedback to the OCORA collaboration and can, therefore, engage in shaping OCORA. Feedback to this document and to any other OCORA documentation can be given by using the feedback form [4].

If you are a railway undertaking, you may find useful information to compile tenders for OCORA compliant CCS building blocks, for tendering complete on-board CCS system, or also for on-board CCS replacements for functional upgrades or for life-cycle reasons.

If you are an organization interested in developing on-board CCS building blocks according to the OCORA standard, information provided in this document can be used as input for your development.

1.2 Applicability of the document

The document is currently considered informative but may become a standard at a later stage for OCORA compliant on-board CCS solutions. Subsequent releases of this document will be developed based on a modular and iterative approach, evolving within the progress of the OCORA collaboration.

1.3 Context of the document

This document is published as part of an OCORA release, together with the documents listed in the release notes [1]. Before reading this document, it is recommended to read the Release Notes [1]. If you are interested in the context and the motivation that drives OCORA we recommend to read the Introduction to OCORA [5], and the Problem Statements [6]. The reader should also be aware of the Glossary [2] and the Question and Answers [3].

1.4 Renaming

The CCS Communication Network was formerly called Universal Vital Control and Command Bus (UVCCB). The evaluations on different communication layers concluded to use a time-sensitive Ethernet network as communication backbone. Therefore, the UVCC-Bus was renamed to CCS Communication Network.

1.5 Problem Description

Today the interfaces between CCS components on the vehicle are proprietary. The proprietary interfaces do not allow to exchange CCS components from different suppliers. The vendor lock-in created by proprietary interfaces leads to high costs. The existing proprietary interfaces do not allow to add easily new functions.

Moreover, these interfaces are implemented using heterogeneous bus technologies. This leads to increased complexity and extensive effort for the operator/maintainer to handle these heterogeneous systems.

1.6 Concept

The OCORA architecture [7] aims for plug and play interchangeability within the CCS domain through isolation of specific functions in combination with the specification of a generic and open communication backbone, the CCS Communication Network (CCN). In the following figure the final physical view of the OCORA architecture [7] is shown. The CCN connects different components of the future CCS on-board systems as for example:

- Safe Computing Platform (SCP)
- Train Display System (TDS)
- Cab Voice Device (CVD)
- National Train Control System (NTC) or Specific Transmission Module (STM)
- Gateway to Train Control Management System Network, Operator Network, Communication Network or Security Network (ECN/ECN Security Gateway)

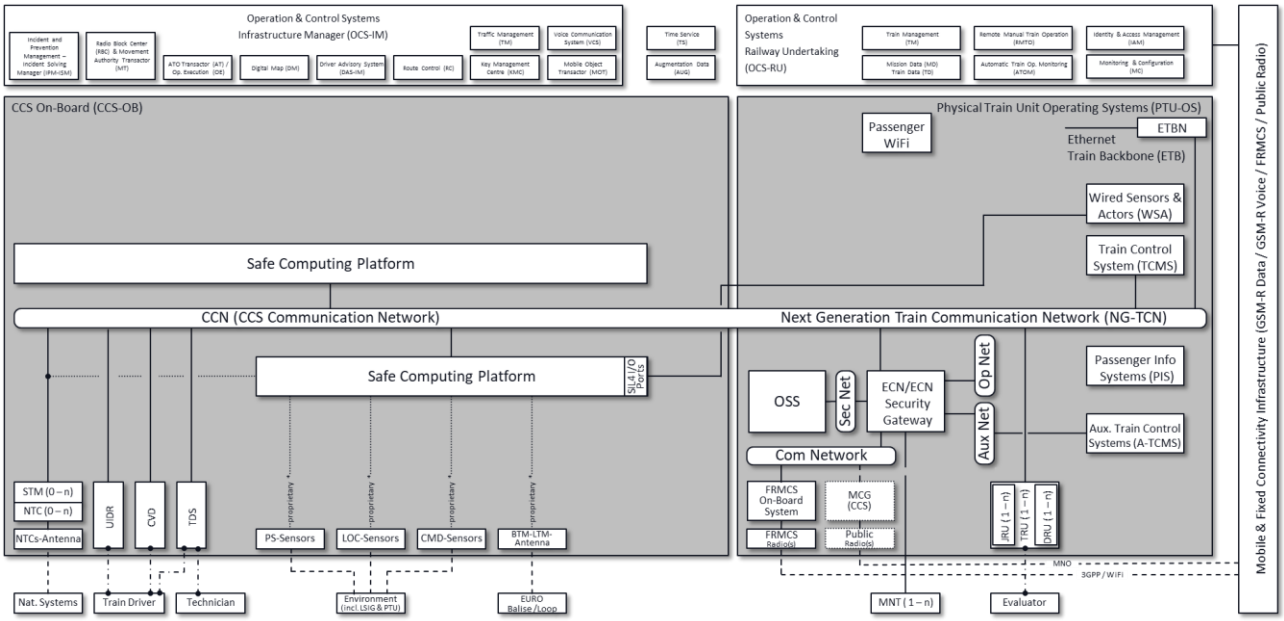

Figure 1: Technical architecture (final view) from [7]

In the final vision of the system an open standardized CCN ensures the safe data connection between all CCS components. The network allows simple upgrades of the CCS System by new functions or components. It also enables procurement on a component-based way which leads to more flexibility in the life cycle management and optimal components due to larger market size. The CCN itself will be modifiable in accordance with future technological evolutions by means of strict separation of the different communication layers (OSI Layers).

Due to the TSI-CCS 2022 with its new established Subset 147 covering the communication layers for the CCS network, the evaluations noted in this document in the chapters 3 and 4 have to be reworked in the subsequent release.

The CCN evaluations done in the former release phases proposes the CCN to be a TSN Ethernet based network with the use of SDTv2 / SDTv4 as safety layer. In order to be able to integrate the CCN on the next generation of train communication network (NG-TCN) or establish an own ECN-like network for the CCN, every hard-real-time CCS device (e.g. Safe Computing Platform etc.) should have at least one TSN-capable Ethernet port whereas for soft- or non-real-time CCS devices a single standard non-TSN-capable Ethernet port is sufficient. Hard-real-time devices can use both planes of NG-TCN with two TSN-capable Ethernet ports to improve reliability and availability.

On session layer TRDP 2.0, OPC-UA Pub/Sub (over TSN) or DDS/RTPS (over TSN) are suitable solutions. These three options will be further investigated in subsequent versions considering the system architecture with platform/CCU and the subcomponents.

The proposed protocol stack of CCN is listed in the following table. Highly recommended standards to be used as reference for procurement in OCORA are listed in **bold** font.

Layer	Protocol for hard-real-time data		Protocol for soft- or non-real-time data
(Safety Layer[1])	**(SDTv2 / SDTv4)**		
Session Layer	TRDP 2.0, OPC-UA Pub/Sub or DDS/RTPS		
Transport Layer		UDP TCP	UDP TCP
Network Layer		IPv4	IPv4
Data Link Layer	**Time-Sensitive Networking (TSN) IEEE 802.1**		**Standard Ethernet IEEE 802.3**
Physical Layer	**100BASE-TX or 1000BASE-T**		

Table 2: Protocol Stack CCN

The defined protocol stack allows safety-related and hard-real-time data traffic. For non-safety-related and soft- or non-real-time applications, standard TCP/IP or UDP/IP data traffic over standard Ethernet (IEEE 802.3) can be used on the same physical layer of the CCN.

1.7 Goals

This document is based on the CCN (UVCCB) evaluation reports of former releases [10], [11]. It provides all investigations and results of all phases. In the following chapters the goals of the respective phases are described.

1.7.1 Beta phase

A list of requirements is established for the on-board CCS communication network (CCN) and provides a summary of independent assessments of existing, open standard bus / network protocols for safe communication among the CCS components in the vehicle. The first bus / network evaluation considered OSI-Layers 1 to 6 & Safety Layer. Possibilities of existing bus or network protocols were collected and an evaluation regarding the requirements of the bus technology was performed. The goal was to decide for one specific stack of existing bus or network protocols fulfilling the requirements. This protocol stack was then defined as the chosen CCN technology within the OCORA initiative.

1.7.2 Gamma phase

The goal was to update the requirements for the network protocols on session (and presentation) layer as well as to evaluate existing protocols. Further, the technical integration of the CCN within the NG-TCN and the separation of responsibilities between the two domains was elaborated. Also, the solution for retrofit vehicles, where there will be a CCN without a NG-TCN, was developed.

1.7.3 Delta phase

The goal which was completed in delta phase was the evaluation of data serialization formats.

1.7.4 Release R1 phase

In Release R1 phase, the network architecture with detailed technical implementation of CCN in NG-TCN (network configuration) and cybersecurity was investigated. Also the work done in different working groups (e.g. IEC TC9 WG43 or ERA TWG Archi) was aligned in order to get consistent new standards and regulations (e.g. IEC 61375, TSI-CCS 2022, ERA Subsets, OCORA specifications).

Further, a Proof of Concept (PoC) was established to show the feasibility of the CCN as a logically separated network on a common physical train communication network (NG-TCN). The setup of the demonstrator helps to investigate the technical implementation details of the CCN in NG-TCN.

[1] Safety Layer is only applicable for safety-related data traffic.

1.7.5 Release R2 phase

The main work done in Release R2 phase was related to TSI-CCS 2022 with its new established Subset 147 covering the communication layers for the CCS Network.

Further, the PoC was extended with additional TSN tests as well as with the realized end-to-end TRDP tests.

This document contains the current results of the evaluation tasks. The PoC CCN part is documented in [8].

2 Requirements

The list of requirements collects all requirements to be used for the technology evaluations. They shall not be copied directly for a call for tender. After the evaluations, a proper specification will define the requirements for hardware procurements.

2.1 Functional requirements for CCN

This chapter elaborates the functional requirements for the CCN

CCN-01 Data transfer

- Requirement: The bus/network supports data exchange between different nodes (component of the CCS system).
- Type: must
- Remarks:

CCN-02 Safety

- Requirement: The bus/network supports safe data exchange for safety applications. It is possible to transmit data for safety applications with different safety integrity levels: from no safe data exchange to data exchange for safety applications from SIL1 to SIL4.
- Type: must
- Remarks: used for ETCS functions with SIL4 requirements.

CCN-03 Safety

- Requirement: The bus/network fulfils the requirements of EN 50159:2010 (Railway applications - Communication, signaling and processing systems - Safety-related communication in transmission systems) for safety applications.
- Type: must
- Remarks:

CCN-04 Safety

- Requirement: The safety targets for the different SIL of the safe data transmission are as follows (based on EN 50129 [18]):

Safety Integrity Level SIL**	Tolerable Functional Failure Rate TFFR per hour and per function	Tolerable Function Failure Rate TFFR per hour and per part of function *
4	$10^{-9} \leq$ TFFR $< 10^{-8}$	$10^{-11} \leq$ TFFR $< 10^{-10}$
2	$10^{-7} \leq$ TFFR $< 10^{-6}$	$10^{-9} \leq$ TFFR $< 10^{-8}$

* The transmission is only a part of a function. The safety target for the transmission part is estimated to be 1% of the safety target for the function.
** For functions with at least SIL1 the table row for SIL2 is applicable, for functions with at least SIL3 the table row for SIL4 is applicable.

- Type: must
- Remarks: Same failure rate allocation as in ERA Subset 057 v310 [25].

CCN-05 Determinism, Predictability

- Requirement: The bus/network shall be capable for real time applications. This means either cyclic slots for process data or a working prioritization methodology to guarantee a throughput of process data and avoidance of network capacity reduction in case of overload. It is important, that the methodology is transparent and accepted in applications with real-time needs.
- Type: must
- Remarks:

CCN-06 Data type

- Requirement: The bus/network supports process data objects (cyclic) and message data objects (event based).
- Type: can
- Remarks:

CCN-07 Latency

- Requirement: For process data, a maximum latency of 10 ms shall be provided. For message data, a maximum latency of 100 ms shall be provided.
- Type: must
- Remarks: This requirement is in line with IEC 61375-3-4:2013 [21]

CCN-08 Jitter

- Requirement: For process data, a maximum jitter of 10 ms shall be provided.
- Type: must
- Remarks: This requirement is line with IEC 61375-3-4:2013 [21]

CCN-09 Bandwidth

- Requirement: The physical layer of the bus/network supports a minimum gross data rate of 100 Mbit/s.
- Type: must
- Remarks:

CCN-10 Number of participating nodes

- Requirement: The bus/network supports a minimum of 62 nodes.
- Type: must
- Remarks:

CCN-11 Maximum physical distance

- Requirement: The bus/network system (with its topology) allows at least the physical distance between two nodes of 100 m.
- Type: must
- Remarks:

CCN-12 Topology flexibility

- Requirement: Bus/network shall be able to support different topologies i.e., Line, Star, Ring etc.
- Type: can
- Remarks:

CCN-13 Time synchronization

- Requirement: The bus/network provides a time synchronization of ± 1ms between several nodes.
- Type: must
- Remarks:

CCN-14 Independency of data streams for modularity, upgradability

- Requirement: In order to simplify the approval for updates, the bus/network provides a clear separation (independency) on data link or physical layer of data streams between nodes/applications. If a data field of one stream changes, all other streams should not be affected.
- Type: must
- Remarks:

CCN-15 Communication pattern

- Requirement: Direct communication from data producer to data consumer must be possible. This implies a communication pattern in a “Publish & Subscribe” mode. Pure “Client/Server” or pure “Master/Slave” approaches are therefore not sufficient.
- Type: can
- Remarks:

2.2 Non-functional requirements for CCN

This chapter elaborates the non-functional requirements for the CCN

CCN-16 Openness
- Requirement: The bus/network technology is open and standardized. There is no restriction regarding intellectual property. This means that technical specifications are readily available for homologation purposes, obsolescence support, upgradability and 2nd source.
- Type: must
- Remarks:

CCN-17 Independence
- Requirement: The bus/network is based on a technology with components that either produced by different suppliers (independence) or if there is mainly a single supplier that there are many customers using these components.
- Type: must
- Remarks: This prevents a supplier lock-in where components price is mainly dictated by the supplier.

CCN-18 Availability
- Requirement: The bus/network is based on a technology with components that are commonly used on the market and produced on large quantities. In addition, it uses components that continue to be available on the market and that are not end of life within the next 5 years.
- Type: can
- Remarks: Large quantities ensure that prices are convenient. There is a commercially interesting market for the suppliers and obsolescence life cycle is handled directly by the suppliers.

CCN-19 Simplicity
- Requirement: The bus/network technology allows a simple design (architecture) from a network topology perspective (HW) as well as from the software integration perspective (simplicity).
- Type: must
- Remarks:

CCN-20 Portability
- Requirement: The same bus/network technology can be used for different components, environments (subset of components) and vehicles without or with just a minimum of configuration work (portability).
- Type: must
- Remarks:

3 First evaluation communication layers and safety layer

3.1 Process

To have an assessment of the bus technologies that is as independent and neutral as possible, three separate assessments, one internal and two externals, were made based on the list of requirements from chapter 2. The results of the different assessments were considered in order to decide on the most suitable solution for the OCORA platform.

3.1.1 Internal assessment

In this chapter, the process for the internal assessment is described. The results are given in chapter 3.2.1. As it is impossible to compare all the existing standards and protocols that exist, a relevant subset of all standards and protocols was compared and evaluated for suitability with the OCORA system. The following set was chosen in order to give a comprehensive overview of available technologies.

- MVB according to IEC/EN 61375-3-1
- CANopen according to IEC/EN 61375-3-3
- Profibus FDL according to IEC 61158/IEC 61784 and Safe Time Layer and Safe Link Layer according [24] and [25].
- TRDP with SDTv2 according to IEC/EN 61375-2-3
- TRDP 2.0 over TSN according to [15] with SDTv2 / SDTv4 according to [16]
- TTEthernet (time-triggered Ethernet) according to SAE AS6802
- EtherCAT with FSoE according to IEC 61158 and IEC 61784
- PROFINET with PROFIsafe according to IEC 61158 and IEC 61784

To compare the different technologies, each one was rated for a subset of the requirements deemed the most relevant. The technologies could then be compared. After comparison, the selected technology was checked for compliance with the complete set of requirements. The requirements of Table 3 were deemed most relevant and used for the comparison of technologies.

Number	Title	Requirement
CCN-02	Safety	The bus/network supports safe data exchange for safety applications. It is possible to transmit data for safety applications with different safety integrity levels: from no safe data exchange to data exchange for safety applications with SIL2 and SIL4.
CCN-05	Determinism	The bus/network shall be capable for real time applications. This means either cyclic slots for process data or a working prioritization methodology to guarantee a throughput of process data and avoidance of network capacity reduction in case of overload.
CCN-09	Bandwidth	The physical layer of the bus/network supports a minimum gross data rate of 100 Mbit/s.
CCN-10	Number of participating nodes	The bus/network supports a minimum of 62 nodes.
CCN-14	Independency of data streams	In order to simplify the approval for updates, the bus/network provides a clear separation (independency) on physical layer of data streams between nodes/applications. If a data field of one stream changes, all other streams should not be affected.

Number	Title	Requirement
CCN-16	Openness	The bus/network technology is open and standardized. There is no restriction regarding intellectual property. This means that technical specifications are readily available for homologation purposes, obsolescence support, upgradability and 2nd source.
CCN-18	Availability	The bus/network is based on a technology with components that are commonly used on the market and produced on large quantities.

Table 3: Requirements used to compare bus technologies.

3.1.2 External assessments

For the external assessments, the list of requirements from chapter 2 was given to two companies alongside a minimum list of bus technologies to be evaluated. The external assessments were made by *Selectron Systems AG* and *NewTec GmbH*. The companies had access to the OCORA architecture and used the descriptions of the OCORA initiative contained in that release to estimate the suitability of the technologies to the OCORA initiative. The results of the external assessments are given in chapters 3.2.2.1 and 3.2.2.2.

3.2 Evaluation Results

3.2.1 Internal assessment

The table below from [12] provides an overview of the assessment of the standard protocols regarding the most relevant requirements. The evaluation is based on the information for the protocols given in chapter 3 and its subchapters in [12].

Protocol Stack	Safety	Determinism	Bandwidth	Number of participating nodes	Independency of data streams	Openness	Availability
MVB	Safety Layer tbd	++	1.5 Mbit/s	≤ 4095	-	standardized	Hardware and software available but always rail specific since MVB is a rail specific standard.
CANopen	Safety Layer tbd	++	125 kbit/s - 1 Mbit/s	127	0	standardized	COTS hardware and software available. Widely used in automation industry and rail sector.
Safe Time Layer Safe Link Layer Profibus FDL	SIL 4	+	9.6 kbit/s – 12 Mbit/s	≤ 125	+	standardized	COTS hardware and software available. Widely used in automation industry and rail sector.
SDTv2 TRDP TCP/UDP IP Ethernet	SIL 2	--	100 Mbit/s	≤ 16382	0	Stack fully standardized Open-source software for TRDP available (TCNopen)	COTS Ethernet hardware available Open-source software for TRDP available (TCNopen)

Protocol Stack	Safety	Determinism	Bandwidth	Number of participating nodes	Independency of data streams	Openness	Availability
SDTv2 / SDTv4 TRDP 2.0 TCP/UDP IP TSN	SIL 4	+	≥ 100 Mbit/s	≤ 16382 with addressing of TRDP	+	TRDP and TSN standardized. TRDP over TSN with new SDTv4 standardized in IEC 61375 in 2022. Open-source software for TRDP available (TCN)	Prototypes of network devices available. Fully certified serial products expected in 2025. Open-source software for TRDP available (TCNopen)
(SDTv2) (TRDP) TCP/UDP IP TTEthernet	Safety Layer tbd SIL 2, if SDTv2 is used	++	≥ 100 Mbit/s	≤ 16382 if addressing of TRDP is used	++	standardized	COTS hardware and software available. Used in automotive and avionics industry.
EtherCAT, FSoE	SIL 3	++	≥ 100 Mbit/s	≤ 65535	-	standardized	COTS hardware and software available. Widely used in automation industry.
PROFIsafe PROFINET	SIL 3	+	100 Mbit/s	≈4.2 Mrd. (IPv4 address space)	++	standardized	COTS hardware and software available. Widely used in automation industry.

Table 4: Evaluation of Protocols regarding most relevant Requirements

The internal evaluation in [12] shows that only one of the evaluated protocol stacks fulfils every most relevant requirement in sufficient quality. Therefore, the proposal of the protocol stack to be used as CCN is TRDP 2.0 over TSN with SDTv2 / SDTv4:

Layer	Protocol
Safety Layer	SDTv2 / SDTv4
Session Layer	TRDP 2.0
Transport Layer	UDP (for process and message data) TCP (for message data)
Network Layer	IPv4
Data Link Layer	Time-Sensitive Networking (TSN) IEEE 802.1
Physical Layer	100BASE-TX or 1000BASE-T

Table 5: Protocol Stack TRDP 2.0 over TSN with SDTv2 / SDTv4

The Shift2Rail (S2R) projects CONNECTA and SAFE4Rail elaborated the Next-Generation Train Communication Network (NG-TCN) which is one of the main building blocks of S2R's next generation of TCMS architectures. The NG-TCN is based on today's TRDP protocol stack according to IEC/EN 61375-2-3. It introduces a new TRDP traffic class (TSN-PD) for scheduled data traffic based on Time-Sensitive Networking (TSN). This traffic class is intended to be used for safety critical and latency critical data. TSN is defined in IEEE 802.1 standards.

The proposed protocol stack fulfills the most and the less relevant requirements as shown in [12] and the following table:

Property	Characteristics
CCN-01 Data Transfer	TRDP 2.0 over TSN with SDTv2 / SDTv4 is developed for data exchange between different onboard components of railway systems.
CCN-02 Safety	SDTv2 enables safe communication for functions of SIL 2. SDTv4 enables safe communication for functions of SIL 4.
CCN-03 Safety	The bus/network can fulfill the requirements of EN 50159:2010 [19]. The safety approval for the protocol stack will be done by CONNECTA.
CCN-04 Safety	The safety analysis for SDTv4 will be done by CONNECTA.
CCN-05 Determinism	TSN adds services on standard ethernet layer (layer 2) for deterministic networking with bounded latency and low jitter. TSN ensures a quite strong determinism for real-time applications. The following sub-standards of TSN describes mechanisms for stream specific bandwidth allocation and latency minimizing: • Stream Reservation Protocol (SRP) of TSN in IEEE 802.1Qat and 802.1Qcc • Per-Stream Filtering and Policing (PSFP) of TSN in IEEE 802.1Qci • Path Control and Reservation (PCR) in IEEE 802.1Qca In CONNECTA Drive-by-Data architecture specification a maximum end-to-end latency within consist network over 2 consist switches is estimated with 435 µs.
CCN-06 Data Type	TSN supports different traffic classes which can be understood as process data and message data. For time critical process data, a scheduled traffic can be used whereas for message data with low time criticality best effort ethernet traffic can be used.
CCN-07 Latency	According to [15] a maximum end-to-end latency is estimated to be below 5 ms for the maximum network topology with 64 consist switches and 63 ETBNs.
CCN-08 Jitter	According to [15] a jitter in low µs range can be expected.
CCN-09 Bandwidth (Gross Data Rate)	≥ 100 Mbit/s
CCN-10 Number of participating nodes (Address Space)	≤ 16382 with addressing of TRDP
CCN-11 Maximum physical distance	The maximum segment length of 100BASE-TX or 1000BASE-T physical layer is 100 m which fulfills the requirement of 54 m maximum physical

	distance.
CCN-12 Topology Flexibility	The UDP/IP over TSN stack is open to different network topologies. Nevertheless, if the CCN is integrated in the NG TCN of TCMS system, a common network topology of both domains must be elaborated.
CCN-13 Time Synchronization	TSN allows precision time synchronization in the range of nanoseconds to microseconds which is lower than the required value of ±1 ms.
CCN-14 Independency of data streams	TSN ensures quite strong independency of data streams. The following sub-standards of TSN describes mechanisms for stream specific bandwidth allocation: • Stream Reservation Protocol (SRP) of TSN in IEEE 802.1Qat and 802.1Qcc • Per-Stream Filtering and Policing (PSFP) of TSN in IEEE 802.1Qci • Path Control and Reservation (PCR) in IEEE 802.1Qca
CCN-15 Communication pattern	TRDP allows a direct communication from data producer to data consumer over a “Publish & Subscribe” communication pattern.
CCN-16 Openness	Standards for TRDP and SDTv2 (IEC/EN 61375-2-3 and others e.g. IEC/EN 61375-2-5) should be enhanced by TRDP 2.0 over TSN and SDTv4 until 2022. TSN itself is specified as an open standard by IEEE 802.1 TSN group. Today’s TRDP and safety layer SDTv2 is available as open-source software (TCNopen)
CCN-17 Independence	The network hardware must be compatible with TSN standard. The rest of the protocol stack will be done in software. Since the TSN standard is open and not railway specific and will be used among different domains, it is expected to have different hardware suppliers. Already the first prototypes of network devices for railway use in CONNECTA/SAFE4Rail project are elaborated by two independent hardware suppliers (Westermo and Moxa).
CCN-18 Availability	TSN IP core for railway application elaborated by TTTech. First prototypes of network devices elaborated by Westermo and Moxa in SAFE4Rail project. Network devices are used by Bombardier, CAF and SIEMENS in demonstrators of CONNECTA project. Fully certified serial products expected in 2025.
CCN-19 Simplicity	The design of a TSN network in terms of hard and software integration of the end devices will be quite simple. However, the configuration of the network devices (switches/routers) will be challenging. But with establishment of the standard, it is assumed that different configuration tools will simplify the configuration process.
CCN-20 Portability	The portability can be ensured with an adequate configuration of the CCN within the whole TSN network.

Table 6: Properties TRDP 2.0 over TSN with SDTv2 / SDTv4

3.2.2 External assessment

3.2.2.1 NewTec GmbH

This chapter contains the main parts of the executive summary of the evaluation report from NewTec GmbH [13].

The legacy bus protocols MVB, CANopen and Profibus cannot support major OCORA requirements in chapter 2. All state-of-the-art Ethernet based protocols meet most of OCORA requirements – and each has its pros and cons:

Protocol	TCMS compatible (CCN connecting to TCMS)	TCN integration (Extending CCN thru ECN)	Safety	Development Support

TRDP 2.0	✔	✔	With SDT (SIL4)	Medium / standard in progress
OPC UA	Gateway needed	With TSN	– (in preparation)	Good
Profinet	Gateway needed	With TSN	SIL 3	Medium
EtherCAT	Gateway needed	No, gateway needed	SIL 3	Good
CIP	Gateway needed	No, TSN in preparation	SIL 3	Good
FDF	✔	✔	SDT (SIL4)	Standard in progress

Table 7: High Level Comparison Chart

EtherCAT does not, and CIP does not yet support TSN – thus a CCN using those protocols and supporting real time traffic will be isolated. A hardware gateway to access the TCMS is needed.

Using OPC UA (TSN) or Profinet, the CCN can use a software gateway to access TCMS; via TSN remote CCS devices could be attached to the ECN and save additional wiring.

TRDP 2.0 with SDTv2 / SDTv4 can be used without any additional gateway but has no object modelling support – configuration is aimed for use with the TCMS.

Using the Functional Distribution Framework (FDF) could be a valuable solution, especially as it is not bound to TCMS use and is independently developed using two diverse Operating Systems / middleware. The FDF, which uses TRDP as communication protocol (the AUTOSAR AP version additionally provides OPC UA), seems currently the best option, although its development is still ongoing.

3.2.2.2 Selectron Systems AG

This chapter contains the main parts of the conclusion of the evaluation report from Selectron Systems AG [14].

The traditional fieldbuses are increasingly being replaced by Ethernet solutions. This trend can be observed in all relevant markets: Industrial automation, automotive, aerospace and railway. In addition, the technology assessment shows that especially the bandwidth requirement can only be fulfilled by an Ethernet solution. Such a solution can be divided into three layers: An Ethernet Layer, a Protocol Layer and a Safety Layer. For each layer, a different solution can be chosen according to detailed requirement of the OCORA system.

Ethernet Layer

An Ethernet network can consist of different types of architecture. In order to improve the failure tolerance against hardware faults, at least ring architecture should be chosen for the CCN. Such an architecture could easily be extended to ladder architecture which would provide even more redundancy and it would also support dual homing. Figure 2 shows an example architecture.

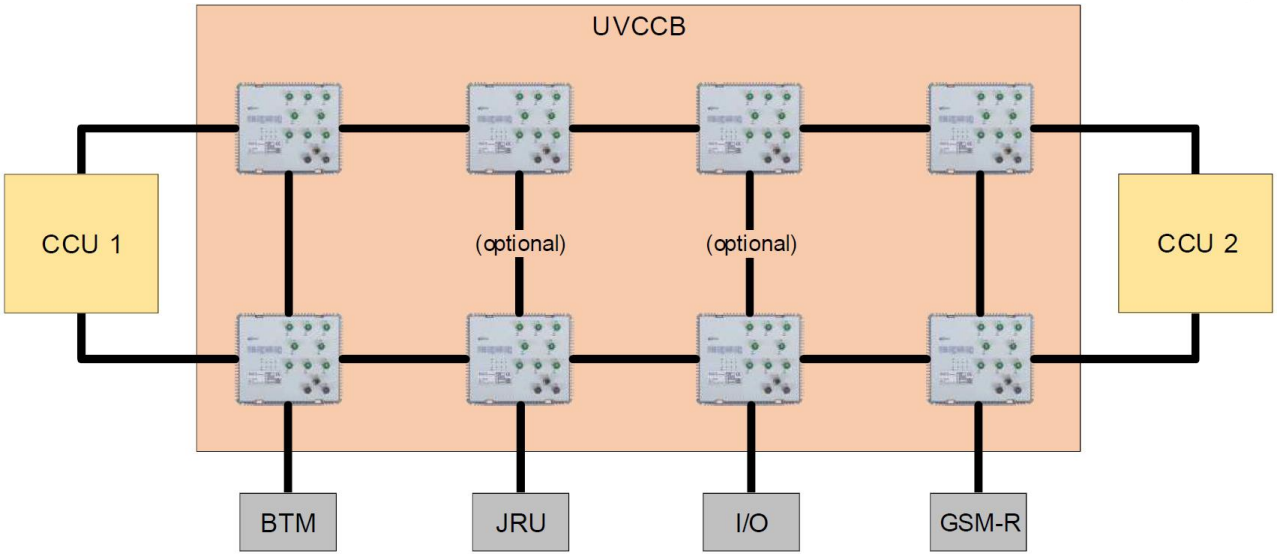

Figure 2: Possible CCN (former UVCCB) Architecture

In the future, it is recommended to use TSN Ethernet to manage the network. TSN Ethernet is seen as the future standard for most automation networks. However, not all TSN Ethernet standards are already released or only rarely implemented in serial products. Therefore, if the CCN should be implemented before these changes, a migration strategy could be defined. Such a migration strategy could look as follows:

- Build up the system with available Ethernet components. For network management, a solution which fulfils the performance and redundancy requirements should be chosen:
  - RSTP
  - MRP
  - LLAP
  - PSP
- Update the system with TSN Ethernet networking devices (switches). This step can be seen as a preparation for the next step.
- Update the system with TSN Ethernet capable End Devices. This would be the final stage for the CCN, where all defined features (low jitter, high bandwidth, traffic with mixed criticalities) could be used.

In order to have the possibility to perform the migration strategy step by step, it is essential that the Ethernet layer contains no safety critical functionality.

Protocol Layer

Most of the protocols evaluated use a master / slave approach, where one (master) device configures and manages the slave devices.

From a configuration point of view, this provides the possibility to configure all devices from one single point. On the other hand, the master / slave approach significantly complicates a slave-to-slave resp. master-to-master communication.

Of all protocols examined, TRDP causes the least restrictions for such flexible communication structures. Therefore, it is recommended to use TRDP for the CCN, but it must also be mentioned that the use of TRDP will require a lot of standardization effort by the OCORA organization. This is because for TRDP resp. in the IEC 61375 standard series, nearly no application profiles for End Devices are defined which would be required in order to provide the possibility to exchange devices from different suppliers.

Safety Layer

The CCN requirements in chapter 2 require Safety layers for SIL 2 and SIL 4. For both SIL, at least a different safety code shall be used. The safety layers SDTv2 / SDTv4 defined in IEC 61375-2-3 and Subset 056 / 057 defined by UNIFE are the only safety layers investigated which fulfil these requirements. Unfortunately, both safety layers have considerable disadvantages:

- SDTv2 / SDTv4
  Limited latency monitoring which requires measures on application level or allow only a classification as Class 1 safety protocol according to EN 50159 [19].
- Subset 056 / 057
  The safety code (CRC) of subset 057 is 48 Bit and considers PROFIBUS as non-safe communication system. The Ethernet resp. protocol layer of the CCN would have to provide the same Bit error probability as PROFIBUS, which violates the black channel approach.

Because SDTv2 / SDTv4 provides more flexibility (quantity of payload data, number of communication partners), it is recommended to use SDTv2 /SDTv4.

Additional Remarks

The suggested solution with an Ethernet based CCN which uses TRDP and SDTv2 / SDTV4 is similar to the approach chosen by CONNECTA. The goal of this project is to define the next generation TCMS. According to the OCORA Architecture, TCMS shall be connected to the CCN directly or via a gateway. This can be a chance to standardize the technology used in railway industries even more. On the other hand, this can be a risk if the OCORA and CONNECTA efforts are not coordinated. Problems which can occur if the efforts are not coordinated could be for example:

- Ambiguous definition TRDP ComID
  The same TRDP ComID could be assigned to devices / functions of CCN and TCMS.
- Multiple use of SDT SMI
  The same SDT SMI could be assigned to VDPs in CCN and TCMS. Such an error could be difficult to find and prevent the homologation of a vehicle.

Therefore, it is recommended to coordinate the activities of OCORA and CONNECTA.

3.2.3 Summary and preliminary specification

All the internal and external assessments of CCN bus technologies come to the same result. The recommended CCN is a TSN Ethernet based network with the use of SDTv2 / SDTv4 as safety layer. In order to be able to integrate the CCN on the next generation of train communication network (NG-TCN) or establish an own ECN-like network for the CCN, every hard-real-time CCS device (e.g. Safe Computing Platform

etc.) should have at least one TSN-capable Ethernet port whereas for soft- or non-real-time CCS devices a single standard non-TSN-capable Ethernet port is sufficient. Hard-real-time devices can use both planes of NG-TCN with two TSN-capable Ethernet ports in order to improve reliability and availability.

On the OSI layers 3 & 4, at least IPv4, UDP, TCP must be supported by the platform/CCU. On session layer there are several possible solutions like TRDP, OPC-UA Pub/Sub or DDS-RTPS which can coexist. The detailed evaluation of session layer protocols can be found in chapter 4.

The protocol stack of CCN is shown in the following table. Highly recommended standards to be used as reference for procurement in OCORA are listed in **bold** font.

Layer	Protocol for hard-real-time data		Protocol for soft- or non-real-time data
(Safety Layer[3])	**(SDTv2 / SDTv4)**		
Session Layer	TRDP 2.0, OPC-UA Pub/Sub or DDS/RTPS		
Transport Layer		UDP TCP	UDP TCP
Network Layer		IPv4	IPv4
Data Link Layer	**Time-Sensitive Networking (TSN) IEEE 802.1**		**Standard Ethernet IEEE 802.3**
Physical Layer	**100BASE-TX or 1000BASE-T**		

Table 8: Protocol Stack CCN

[3] Safety Layer is only applicable for safety-related data traffic.

4 Detailed evaluation of session layer protocols

As proposed in chapter 3.2.3 to use the same technology for CCN as already defined for the next generation train communication network (NG-TCN) in the TCMS domain. On the OSI layers 3 to 6 at least IPv4, UDP, TCP and TRDP (i.e., TRDP 2.0) must be supported in order to be able to communicate directly to the TCMS. For communication between CCS devices other protocols on session layer are possible which are be evaluated in this chapter.

4.1 TRDP 2.0

4.1.1 Description

TRDP 1.x is based on standard Ethernet UDP communication (for Message Data, TCP/IP is an option). In full duplex-switched Ethernet (IEEE 802.1) TRDP can be used in parallel with other Ethernet based protocols. A predecessor to TRDP 1.x is Bombardier's IPTWire protocol (realized as IPTCom), from which it inherited many features.

TRDP 1.x is standardized in IEC 61375-2-3 Annex A. Additional requirements from the development of the next generation train communication network (NG-TCN) led to some additions to the protocol and therefore also to the current open-source implementation TCNopen. The extended open-source implementation TCNOpen is known under the term TRDP 2.0. TRDP 2.0 is, except for TSN, fully compatible to the standard TRDP 1.x stack.

In the following subchapters the communication of TRDP/TRDP 2.0 is described. The content is derived from the external evaluation from NewTec during OCORA Beta phase [13].

4.1.2 Communication pattern

TRDP offers basically two classes of communication schemes:

- Process Data (PD) – Cyclic Push Pattern, aka Publish & Subscribe
- Message Data (MD) – Event Pattern, aka Client/Server or Methods

4.1.2.1 PD Push – Unicast

PD Push is the standard communication pattern where one application (the publisher) provides relatively small amounts of data on a regular basis to a remote application (the subscriber). The data sent must fit into one network frame and the data size must not change for this publish. The TRDP protocol stack sends these telegrams in regular intervals even if the payload does not change. The publisher will not know, if the telegram was received. There is no acknowledging.

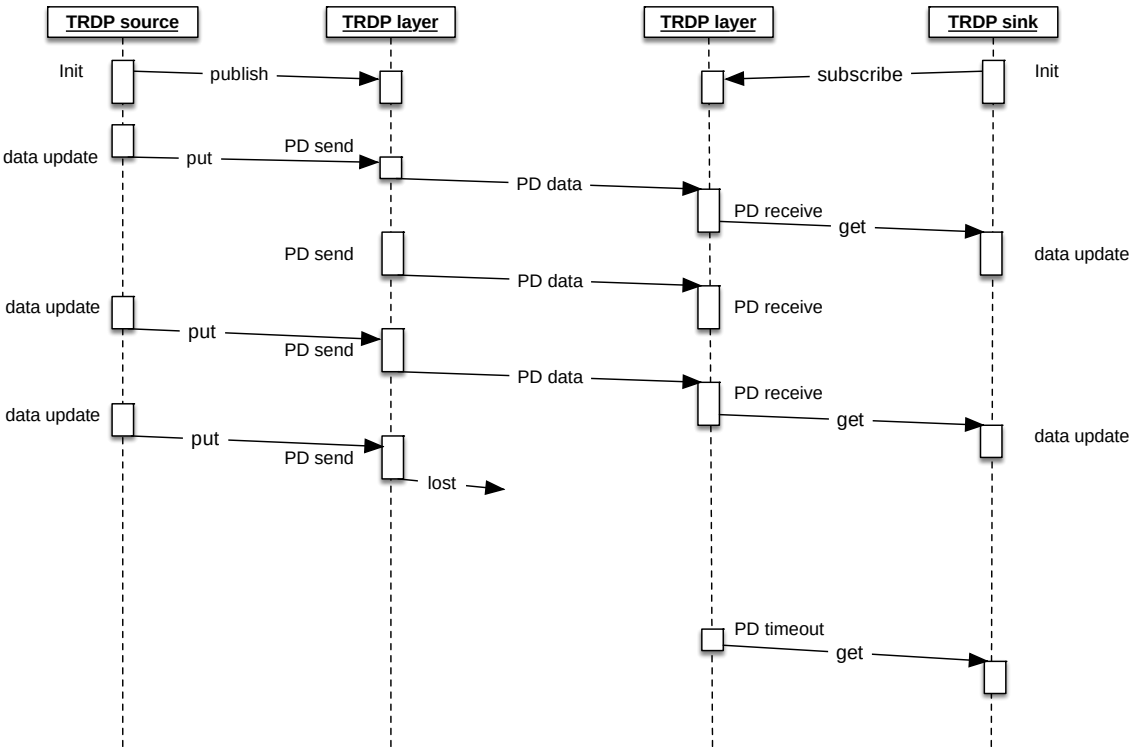

Figure 3: Publish & Subscribe

In Figure 3, 'TRDP source' is the publisher and 'TRDP sink' is the subscriber to the process data. The publisher may update the data asynchronously to the defined PD cycle time – each sent frame is marked with a sequence number and missing or duplicate frames will be detected by the subscriber stack. The subscriber will receive a timeout error, if a defined number of frames were lost – usually 2 or 3.

4.1.2.2 PD Push – Multicast

Using an IP multicast group as destination, a publisher can send one telegram to many subscribers. As with unicast addressing, the publisher will not know whether a subscriber is listening or not.

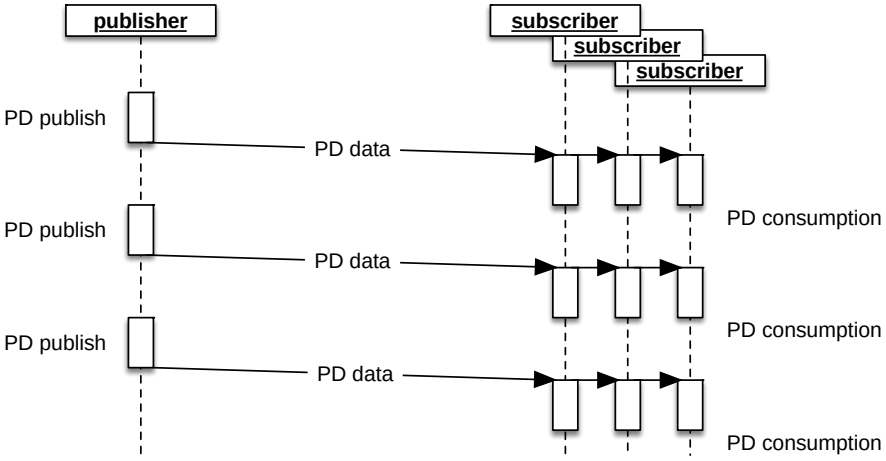

Figure 4: PD Multicast

When using TRDP 2.0 with TSN, the application is responsible to provide cyclic data in time. The TCNOpen TRDP stack supports an additional 'put' call to the application to push data directly to the network stack,

providing an additional absolute time parameter for the NIC. Synchronous TSN operation (IEEE802.1AS) can be supported.

4.1.2.3 PD Pull

The PD Pull pattern allows a subscriber to trigger a publisher to push data immediately (and not waiting for an interval). The addressing can be unicast and multicast, also for the pulled request:

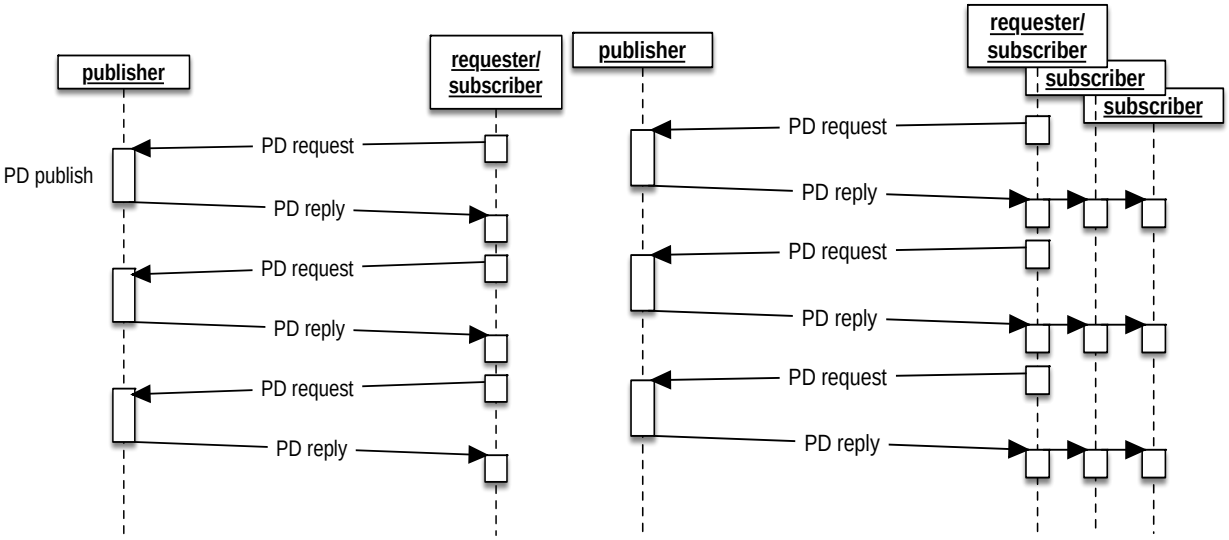

Figure 5: PD Pull: one requester

One subscriber sends a pull request for a certain telegram (ComID) with a reply IP address to a publisher. The publisher sends the requested PD immediately to the IP address (or multicast group).

Introduction of a new TRDP traffic class (TSN-PD) for scheduled data traffic based on standard IEEE 802.1Qbv (Time Sensitive Networking TSN) has been provided with TCNOpen TRDP 2.0. This traffic class is intended to be used for safety critical and latency critical data.

4.1.2.4 MD Pattern

For ‘Methods’ or RPC (remote procedure calls), TRDP offers three Message Data communication schemes:

- Notifications
- Request/Reply
- Request/Reply/Confirm

Notifications correspond to a function without return values – no acknowledge. Request/Reply correspond to a normal function call, where the reply returns requested values or an error code.

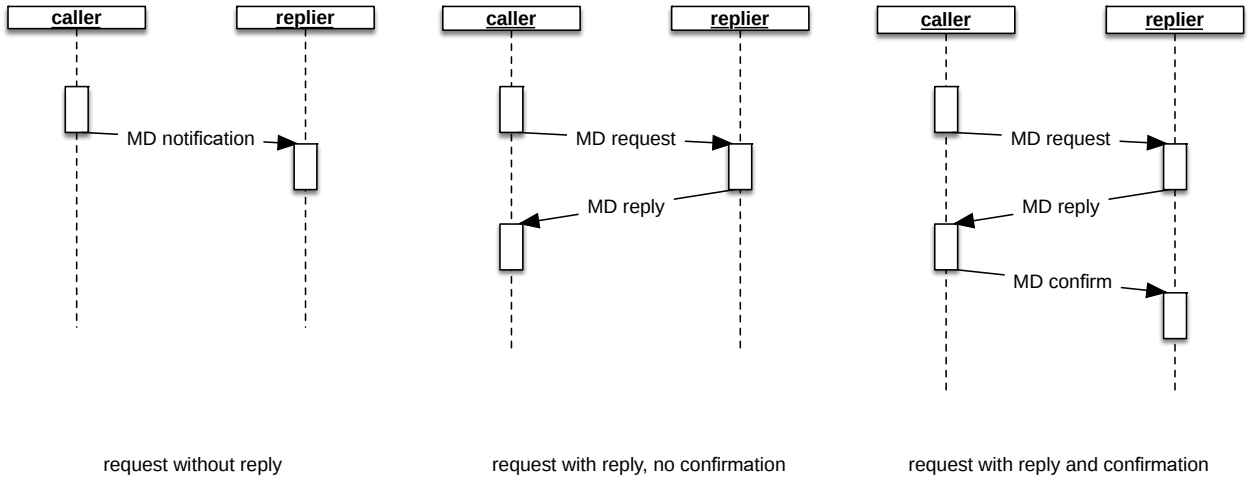

Figure 6: Message Data pattern, unicast

If the replier (server) needs to know if the reply was accepted, it can request a confirmation. This pattern can use UDP/IP and TCP/IP for transport. Because UDP will fragment frames larger than MTU size, TCP/IP should be used as the underlying protocol, because it takes care of resending of lost fragments. This can be configured for each defined telegram.

When using multicast addressing, only UDP is supported:

4.1.3 Addressing

In the TCN and TCMS, device addressing is highly dynamic, because a train composition can change by coupling and uncoupling vehicles, groups of vehicles (consists), or change of leading (preferred driving direction by changing the position of the leading cab). A change of train composition or direction needs re-assigning functional addresses of many networked devices.

Thus, dynamic handling of device and function addresses is a central feature of the TCMS and of the used protocols in the railway domain. The TRDP offers special features regarding the change of train topology. Each data packet contains in its header fields values, which allow a receiver to verify the correct addressee (topology counters). These topology counters are computed after each change of train composition by a process called ‘train inauguration’. The train inauguration ensures, that every node taking part in certain communication has the same view of the train and uses the correct device addresses.

IEC61375-2-3 defines a train-wide central function and device repository, called TTDB (Train Topology Data Base) and an addressing scheme using Unified Resource Locator (URL) and Unique Resource Identifier (URI). A central instance of a TCN-DNS (name server) translates URIs to IPv4 addresses, which are used in the TCNOpen TRDP implementation, for instance.

Figure 7: TCN Domain URL

The user part is currently not defined, but subject to the upcoming service-oriented approach. The host part contains

- a device or group: ldev, grpDoorCtl, devHMI, devECSP, grpAll…
- a vehicle: anyVeh, leadVeh, cstVeh02…
- a consist: lCst, leadCst, anyCst…

- optional closed train
- optional train: standard ltrn

Some parts of such an URI are train topology dependent, means: The train-wide IP address of the leading vehicle or consist will change depending on the position of the leading cab, for instance. Each TRDP telegram provides topography checks, which eventually invalidates the data in case the train topology (and thus the IP address of a device) changes.

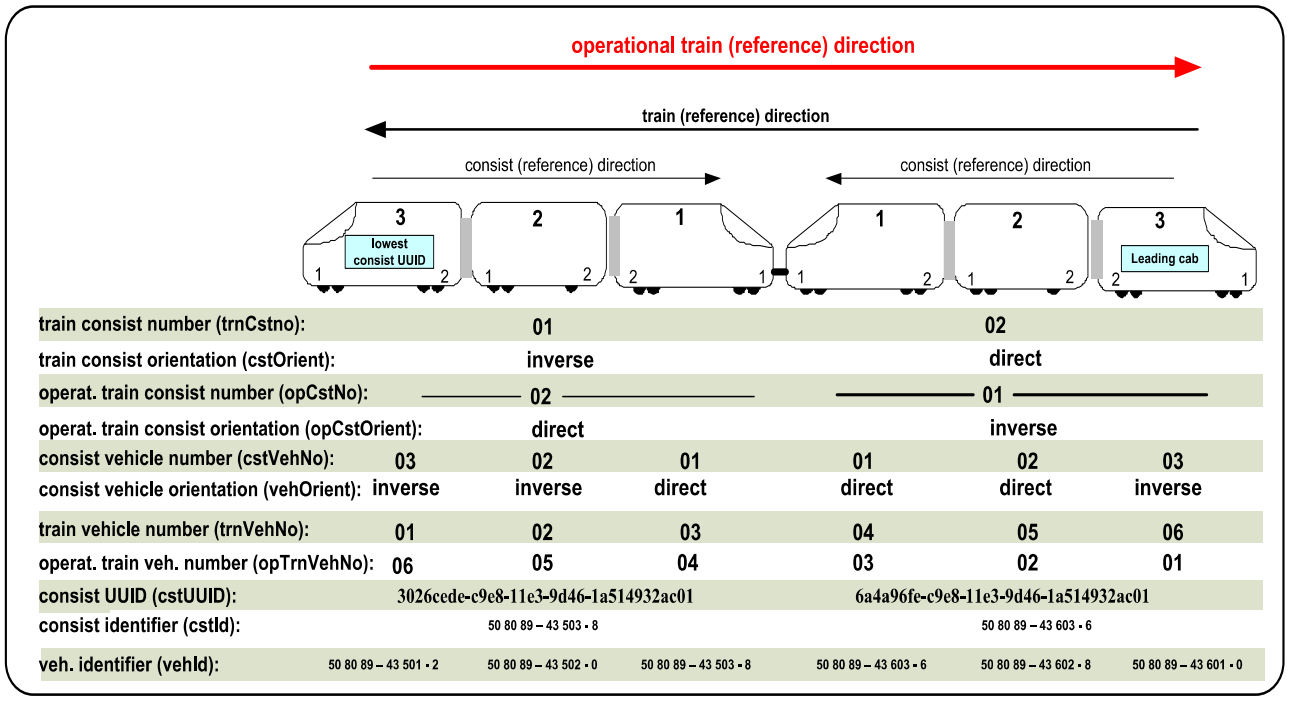

Figure 8: Numbers and IDs within the TCN

For consist-internal communication, static IPv4 addressing can be used. The range defined for the TCN consist network is 10.0.0.0/9. Communicating between consists (over the ETB) is provided by NAT and R-NAT and is managed by ETBNs. The range defined for the ETB network is 10.128.0.0/9.

4.1.4 Security

Until now, security is not covered by the TRDP protocol itself, as it is supposed to be used within a closed network. Access to the ECN is restricted by static configuration of the consist switches (during commissioning). Additional security is provided by IP-IP-gateways and firewalls, which connect (or separate) non-TCMS functions like multimedia or PIS.

Because TRDP uses the standard IPv4 protocol over Ethernet, IPsec or MACsec can be used.

The safety measure ‘Source Identifier’ SID is not transmitted over the network and must be implicitly known by both communication partners. It is actually the seed for the CRC computation (validation and verification). This can be seen as a security feature of the SDT layer (both SDTv2 and SDTv4).

4.1.5 Support

The TRDP 1.x was defined by the working group 43 of the IEC (TC 9/WG 43) and it is currently standardized in IEC 61375-2-3 Annex A. Currently TRDP is evolving to NG-TCN with support of TSN and service orientation.

TRDPs use is mandatory for communication on the ETB (connecting consist networks) and optional within the ECN.

On sourceForge.net, the TCNOpen interest group (Bombardier, CAF, Siemens, NewTec, Toshiba) are actively developing and sharing an Open-Source reference implementation. The resulting library and example applications can run on several platforms, preferably POSIX, but Desktop OSes (Windows 64, MacOS X, iOS) and RTOSes (freeRTOS, ESP32) are also supported and as demo targets included.

For configuration of datasets and communication parameters, an XML or text editor can be used. Several vendors integrated this XML generation in their already existing tools (e.g. Bombardier).

4.1.6 Application area

As a standardized protocol, TRDP was primarily defined and designed with the use on rolling stock in mind. It features special provisions to react to leading direction and train topology changes, which no other protocol provides. Each packet exchanged carries such verification values and a receiver will automatically discard misdirected information.

The Open-Source reference implementation provides standard 'C'-Bindings; compile time options allow configuring the protocol stack for

- High performance (uses more memory for index tables, for high speed/high traffic demands)
- Process Data only (simple devices without the need for Message Data)
- Service Orientation (additional for NG-TCMS)
- Hard Real-Time provisions (TSN extension)
- HW/OS Target

A C++ implementation is in use at the CONNECTA-2 WP3 urban & lab demonstrators.

4.1.7 Summary

Despite the name (TRDP = Train Real-time Data Protocol) suggests, TRDP in its original design and implementation was defined as 'soft real time', as the underlying standard Ethernet protocol for the ETB and the ECN has no hard-real-time features. Only with the advent of TSN (802.1Qbv…) and the extensions introduced in V 2.0 of the TCNOpen TRDP implementation, hard real-time behavior comparable to other field busses can be achieved.

With its Process Data Push communication pattern also direct communication from every data source to the corresponding data sink is possible.

As TRDP is an evolving network protocol, designed and standardized especially for use on rolling stock and for the railway industry. Therefore, it is suited for the use on CCN as already noted in chapter 1.6.

4.2 OPC UA Pub/Sub over TSN

4.2.1 Description

OPC UA is a platform-independent and service-oriented communication standard typically used for machine-to-machine communication. It provides client/server and publish/subscribe communication patterns. TCP (client-server) and UDP (publisher-subscriber) are used as transport protocols. OPC UA is typically used for controller-to-controller (C2C) communication. To support real-time communication, OPC UA over TSN was defined between 2016 and 2018 to further promote OPC UA as the standard for Industry 4.0 applications and the 'Internet of Things'.

In the following subchapters the communication of OPC UA Pub/Sub over TSN is described. The content is derived from the external evaluation from NewTec during OCORA Beta phase [13].

4.2.2 Communication patterns

Originally, OPC UA supported the client/server communication pattern via the HTTP and TCP, only. Pub/Sub, cyclic transmission of process data, was added later to support real-time applications. It supports point-to-multipoint and multipoint-to-multipoint communication. This is also the higher layer support needed for effective TSN usage.

One major distinguishing feature is the service-oriented architecture (SOA), which allows the addressing of functions / services independent from their physical network location.

- Service discovery: find the availability of OPC Servers on the local device and/or remote network
- Address space: all data is represented hierarchically (e.g. as single variables and datasets/containers) allowing for simple and complex structures to be discovered and utilized by OPC Clients
- On-demand: read and write data/information based on access-permissions
- Subscriptions: monitor data/information and report-by-exception when values change based on a client's criteria
- Events: notify important information based on client's criteria
- Methods: clients can execute programs, etc. based on methods defined on the server

Events and Methods map to some extend to AUTOSAR Adaptive's communication model but are not exactly the same.

4.2.3 Security

OPC UA provides some controls covering security:

- Transport: numerous protocols are defined providing options such as the ultra-fast OPC-binary transport or the more universally compatible JSON over web sockets, for example
- Session Encryption: messages are transmitted securely at various encryption levels
- Message Signing: with message signing the recipient can verify the origin and integrity of received messages
- Sequenced Packets: exposure to message replay attacks is eliminated with sequencing
- Authentication: each OPC UA client and server is identified through X509 certificates providing control over which applications and systems are permitted to connect with each other
- User Control: applications can require users to authenticate (login credentials, certificate, web token etc.) and can further restrict and enhance their capabilities with access rights and address-space "views"
- Auditing: activities by user and/or system are logged providing an access audit trail

4.2.4 Support

OPC UA is standardized in several parts of IEC 62541:

IEC/TR 62541-1	OPC Unified Architecture - Part 1: Overview and Concepts
IEC/TR 62541-2	OPC Unified Architecture - Part 2: Security Model
IEC 62541-3	OPC Unified Architecture - Part 3: Address Space Model
IEC 62541-4	OPC Unified Architecture - Part 4: Services
IEC 62541-5	OPC Unified Architecture - Part 5: Information Model
IEC 62541-6	OPC Unified Architecture - Part 6: Mappings
IEC 62541-7	OPC Unified Architecture - Part 7: Profiles
IEC 62541-8	OPC Unified Architecture - Part 8: Data Access
IEC 62541-9	OPC Unified Architecture - Part 9: Alarms and Conditions
IEC 62541-10	OPC Unified Architecture - Part 10: Programs
IEC 62541-11	OPC Unified Architecture - Part 11: Historical Access
IEC 62541-12	OPC Unified Architecture - Part 12: Discovery
IEC 62541-13	OPC Unified Architecture - Part 13: Aggregates
IEC 62541-14	OPC Unified Architecture – Part 14: PubSub
IEC 62541-100	OPC Unified Architecture - Part 100: Device Interface

The leading organization is the OPC Foundation (More than 400 supporting companies/members alone in Europe).

4.2.5 Application area

The OPC UA information-modelling framework is another major feature. Together with the service-oriented approach (see 4.2.2 Communication patterns), it turns data into information. With complete object-oriented capabilities, complex multi-level structures can be modelled and extended. It defines the rules and base building blocks necessary to expose an information model with OPC UA, providing semantics instead of pure communication.

While OPC UA already defines several core models (or profiles), which are targeted for industrial or factory use, more specific information models can be defined for e.g. railway and automotive use.

4.2.6 Summary

The OPC UA Pub/Sub with its hard-real-time capabilities over TSN and its service orientation is suited for the use on CCN. But due to its complex protocol specifications, implementations of the protocol are often incomplete and not fully compatible. Nevertheless, with the same implementation, it could be beneficial to use OPC UA Pub/Sub on CCN since it is widely adopted by the automation industry.

4.3 DDS / RTPS

4.3.1 Description

Data Distribution Service on Real-time Publish-Subscribe (DDS/RTPS) is a data centric middleware that works with a global data space. It supports real-time and QoS (Quality of Service) differentiation. With its data centric approach there is no need for centralized IT infrastructure. In order to meet hard real-time requirements, DDS/RTPS standards by the Object Management Group (OMG) will be enhanced by the usage of TSN on data link layer by the end of 2020 [27].

4.3.2 Communication pattern

DDS is a networking middleware that simplifies complex network programming. It implements a data-centric publish/subscribe pattern for sending and receiving data, events, and commands among the network nodes. Nodes that produce data (publishers) create "topics" (e.g. temperature, location, pressure) and publish "samples". DDS delivers the samples to subscribers that declare an interest in that topic.

DDS handles transfer: message addressing, data serialization, delivery, flow control, retries etc. Any node can be a publisher, subscriber, or both simultaneously. The DDS publish/subscribe mechanism is done with peer-to-peer connections which eliminates the need of a broker component needed. This decentralized publish/subscribe communication pattern makes the system more reliable since there is no broker component needed.

The idea of the new integration of TSN on data link layer is to get the configuration of the network nodes for TSN out of the DDS [27].

4.3.3 Security

DDS Security Specification defines mechanisms (authentication, access control, encryption, message authentication, digital signing, logging and data tagging) for out-of-the box security and interoperability between compliant DDS applications.

4.3.4 Support

DDS/RTPS are standards of the Object Management Group (OMG) for machine-to-machine communication using a publish/subscribe pattern. Originally it was developed by Real-Time Innovations (RTI) and Thales Group. RTI today deliver a commercial implementation of DDS RTPS. But there are also open implementations like openDDS.

4.3.5 Application area

According to the OMG's website, DDS/RTPS is one of many protocols used in industry sectors such as air traffic control, smart energy, medical services, military and aerospace, and industrial automation. DDS/RTPS is also used as communication protocol within the AUTOSAR Adaptive platform and also in the ROS2 middleware.

4.3.6 Summary

Due to its wide application area and its enhancement with TSN capability, DDS/RTPS will be suited for the use on CCN. Due to its use on different platforms and use in different industry sectors, it could be beneficial to use it on CCN.

4.4 MQTT

4.4.1 Description

Message Queuing Telemetry Transport (MQTT) is an open machine to machine publish/subscribe network protocol. Usually, the protocol runs on TCP but can be implemented on top of every lossless bidirectional connection.

4.4.2 Communication pattern

MQTT (Message Queuing Telemetry Transport) is a lightweight publish/subscribe protocol based on TCP. It is especially designed to connect remote devices with low bandwidth.

The message architecture of the publish/subscribe mechanism of MQTT needs a broker which handles the publications and subscriptions as well as the data. The approach is therefore still centralized even with the publish/subscribe mechanism. In the following picture an example of data exchange between three devices over a MQTT broker is shown.

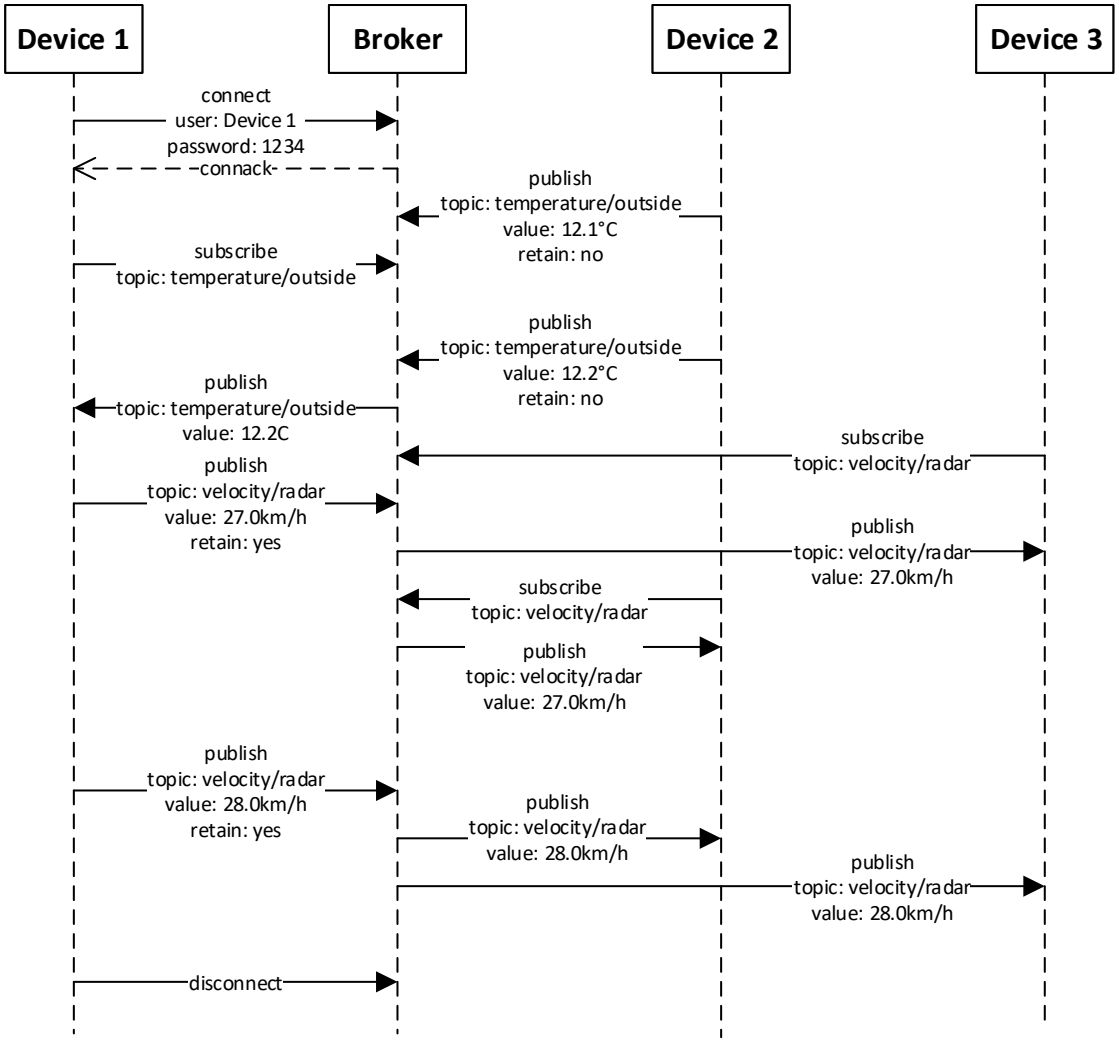

Figure 9: Sequence diagram with data exchange over MQTT

There is information that devices must know for a data exchange. The client must know about the broker that it connects to and the subscriber must know the subject it is subscribing to. A client subscribes to a specific topic, in order to receive corresponding messages. However, other clients can also subscribe to the same topic and get the updates from the broker with the arrival of new messages. Broker serves as a central component that accepts messages published by clients and delivers them with the help of the topic to the subscribed clients. With a set retain flag, the last message of a specific topic will be stored in the broker in order to immediately forward the actual message after subscription.

4.4.3 Security

MQTT does not provide encryption since it was designed as a lightweight protocol. Data is exchanged as plain text, which is clearly an issue from the security point of view. Encryption needs to be implemented as a separate feature. For instance, TLS can be used for an encryption protocol below MQTT (like HTTP or FTP

protocol). Authentication can be implemented by MQTT brokers. In this case, clients need to connect to the broker with the right credentials. Access Control Lists (ACLs) are also managed by MQTT brokers restricting some topic access to non-authorized clients.

4.4.4 Support

MQTT was released by IBM (v3.1) and later adopted by OASIS (v5.0) for internet of things (IoT).

4.4.5 Application area

MQTTs lightweight messaging protocol makes it suitable for resource constrained devices and for non-ideal network connectivity conditions, such as with low bandwidth and high latency. Because of its simplicity and small overhead, it is often recommended as the communication solution of choice in IoT where components have low power requirements.

4.4.6 Summary

MQTT is designed for low power devices with bad network connections. This is not the case for the application as CCN. Moreover, MQTT cannot benefit from the TSN features which leads to non-deterministic communication behavior. Therefore, MQTT is not suitable for the use on CCN.

4.5 AMQP

The Advanced Message Queuing Protocol (AMQP) is a very versatile, standardized binary network protocol for message-oriented middleware. It can be used for a broad variety of different kinds of messaging capabilities. But AMQP cannot benefit from the TSN features which leads to non-deterministic communication behavior. Therefore, AMQP is not suitable for the use on CCN.

4.6 ROS / ROS2

ROS (Robot Operating System) is an open-source software framework (middleware) originally developed by Willow Garage for his robot PR2. Today it is supported by the Open-Source Robotics Foundation (OSRF). Its main targets are research institutes in various areas with a focus on robotics software.

The successor of ROS, ROS2 is built on top of DDS/RTPS. Due to this fact, only the original protocol DDS/RTPS is investigated in this evaluation.

4.7 SOME/IP

Scalable Service-Oriented Middleware over IP (SOME/IP) is a service-oriented communication protocol. It is designed as part of the AUTOSAR Adaptive software platform. Since 2018 the AUTOSAR Adaptive platform supports DDS/RTPS as main communication standard. This is the reason why SOME/IP is not considered for the use on CCN.

4.8 Conclusion

The possible solutions on session layer, remaining as described in the subchapters before, are:

1. TRDP 2.0
2. OPC UA Pub/Sub over TSN
3. DDS/RTPS over TSN

MQTT, AMQP, and SOME/IP are not suitable solutions. ROS/ROS2 is covered by DDS/RTPS since ROS2 uses DDS/RTPS as communication protocol

In principle all tree suitable communication protocols TRDP 2.0, OPC UA Pub/Sub over TSN and DDS/RTPS over TSN can be used on the network – even in parallel. TRDP 2.0 is mandatory for the communication on the ETB in TCMS domain and therefore TRDP 2.0 will also be the standard option for communication between CCS and TCMS domain. All three options will be further investigated considering the system architecture with platform/CCU and the subcomponents (e.g. STM, VLS etc.).

5 Serialization formats

5.1 Introduction

At the interface between the applications and the communication network lies the problem of serializing the data. The way data is handled and structured by applications is application specific and depends on the choice of programming language and implementation. For different applications to be able to communicate data in a generalized manner, as well as transforming the data objects into a format that can be sent over a serial network, the data is transformed into a cross platform format. This process is called serialization as it also permits the data to be sent over a serial communication interface without losing information or creating ambiguities. The words serialization and encoding will be used interchangeably in the following chapters.

The question of serialization needs to be addressed and specified at the application level. However, in this document a recommendation for a data serialization format from the point of view of the lower networking layers (1-6 in the OSI-Model) is provided alongside an overview of different serialization formats. The aim is to assure the compatibility between the applications using the CCN and the CCN.

The serialization formats differ from each other according to following criteria which will be used to evaluate them:

- Readability by humans
- Data typing
- Performance of encoder/decoder
- Space needs of serialized data (directly linked to networking speed)
- Platform / language independence
- Availability of implementations
- Open / proprietary
- Flexibility for upgrades
- Complexity of supported data structures

Some of these criteria should be given more importance than others, dictated by the needs of the CCN. These are briefly discussed below.

It is important to note that the data sent over the CCN will not have a complex structure. The data structures to be sent over the CCN are mostly a small number of variables that do not include more complex structures such as arrays of varying length. E.g. SUBSET-119 uses only following variable types:

- BOOLEAN1
- UNSIGNED8
- INTEGER16, 2s complement
- BITSET8
- UNSIGNED16

For clarity or more information, one can imagine grouping together certain of these variables to give the data some structure. For instance, all variables pertaining to track conditions could be grouped in a structure and separated from other variables. Moreover, the telegram can thus be structured by separating the fields data from a telegram header and safety trail.

The CCN will be a TSN Ethernet based network ([11]) and uses thus Ethernet frames. The payload data per frame is thus limited to the 1500 octets of a standard Ethernet frame. Looking at IEC 61375-2-3 [20] and SUBSET-119, along with the data one would send a header with information about the function the data is sent to, the function that created the data, a version information and message type (process data, message data) and a safety trail. The additional information needs to be included in the data frame and some of the 1500 available octets could be taken up by other protocols.

The system will not need remote procedure call functionality as the nodes will only communicate via a specified network interface.

The CCN is designed to be the communication network of choice for CCS Systems for the foreseeable future. The choice of serialization format should thus be robust over the next 30-40 years and be flexible enough to allow for adaptations to the data and protocols that could arise during this time.

5.2 Data formats

5.2.1 Bitstream

This method of serializing data is currently used for data transmission over the CCS network. The data is serialized and transmitted using a bitstream. The format of the data is chosen when specifying the network. This specification does not need to follow a set of rules, as long as it is not ambiguous. It is however essential that all components adhere to the specification for the system to work smoothly. Using custom encodings and the knowledge that every node adheres to the specification (e.g. Subset 119), the data can be represented in a very compact way.

However, as the encoding is not standardized, a custom encoder and decoder needs to be developed for each system that communicates on the network as well as for each development environment used by the applications. This is only feasible for small systems involving a small number of nodes and applications that communicate a limited amount of data. Once set up, the format is also quite static, as changes to the specification can affect all the components. This can be slightly improved by including reserve bits set aside for future use. However, the data structure will remain very rigid. The advantage of this rigidity however is that the communication over the network is well-defined and the encoding and decoding can be tailored to the needs of the application and thus be more performant. It is also easy to recognize and discard a message that does not follow any specified data structure.

5.2.2 XML

Extensible markup language (XML) is widely used for the representation of arbitrary data structures. It is simple, human readable and very general. Specified by the World Wide Web Consortium (W3C) in free open W3C recommendation, the language is very accessible and used in a wide range of applications. (http://www.w3.org/TR/xml/, version 1.0, last issue at time of writing: 2008)

XML is a textual data format i.e., the data structure as well as the data are represented as text. Encoding of

the text using Unicode standards is supported. Typically, this is UTF-8 or UTF-16. Other encodings (ASCII, …) can be used but are not necessarily supported by every XML parser.

Thanks to its wide use, encoders and decoders are available as APIs for most of the programming languages.

Document type declaration (DTD) (with element type declarations) and schemas can be used to restrict the data types and format.

Due to the structure of XML (use of tags to delimit data) and the use of text (1 byte per character in UTF-8 for most common (i.e., ASCII) characters) this format is easily readable by humans, can however be very verbose and use a lot of space. Though it was designed to be used over the internet, it is not the most efficient format to send data over a network.

The UIC 559 Specification "Diagnostic Data Transmission" from railway vehicles [29], specifies the use of XML with an XML schema definition for diagnostic data transmission from railway vehicles to ground IT systems.

Typical syntax with DTD:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE people_list [
  <!ELEMENT people_list (person*)>
  <!ELEMENT person (name, birthdate?, gender?, socialsecuritynumber?)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT birthdate (#PCDATA)>
  <!ELEMENT gender (#PCDATA)>
  <!ELEMENT socialsecuritynumber (#PCDATA)>
]>
<people_list>
  <person>
    <name>Fred Bloggs</name>
    <birthdate>2008-11-27</birthdate>
    <gender>Male</gender>
  </person>
</people_list>
```

From <https://en.wikipedia.org/wiki/Document_type_definition>

5.2.3 JSON

JavaScript object notation (JSON) is a textual based programming language independent data format, designed to exchange data between applications. It is specified as an ISO/IEC standard (ISO/IEC 21778:2017 Information technology — The JSON data interchange syntax).

JSON format is more lightweight than XML but just as easy for humans to read and write and for machines to parse and generate. The more compact notation uses fewer characters to encode the same data, it is thus more efficient at transmitting data over networks.

The format is based on unordered sets of name/value pairs. Names must be of type string, values can be string, number, "True", "False" or "Null", objects or arrays. The format however does not include information what the type of each element should be and there is a certain ambiguity if how a number should be interpreted (type int, float or other).

A way to validate and restrict the types of data in a JSON format is to use a schema that specifies the types as well as additional restrictions for the values of the objects and variables. The schema needs to be present at validation. Including a validation step in the process however uses more computation time for decoding data.

JSON does not support comments in the data files.

As this format is also text based, it is not as compact as a binary format can be. The text needs to be encoded using Unicode UTF-8.

Typical syntax:

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

From <https://en.wikipedia.org/wiki/JSON>

5.2.4 YAML

YAML is another text-based data interchange format. It is more data oriented than XML and the latest version accepts JSON files as valid. It uses Python style syntax, defining blocks by indentation but can also use flow style with '[' and ']' or '{' and'}' to describe the data structure. Unlike JSON it supports comments. The text needs to be encoded using Unicode character sets.

YAML is an open format that is specified openly (https://yaml.org/spec/1.2/spec.html). It is however not fixed in a standard by an international body or association.

YAML includes JSON as a subset offers however further features mainly pertaining to more complex data structures (relational anchors, extensible data types or mappings preserving key order to name a few). However, for the purpose investigated here these features do not add significantly to the usefulness of YAML.

Another drawback with YAML is that it has a rather open syntax that lets the same data be represented in several ways creating different sizes of serialized data and needs a more complex encoder or decoder.

Typical syntax example:

```
---
receipt:     Oz-Ware Purchase Invoice
date:        2012-08-06
```

```
customer:
    first_name:   Dorothy
    family_name:  Gale

items:
    - part_no:   A4786
      descrip:   Water Bucket (Filled)
      price:     1.47
      quantity:  4

    - part_no:   E1628
      descrip:   High Heeled "Ruby" Slippers
      size:      8
      price:     133.7
      quantity:  1

bill-to:  &id001
    street: |
            123 Tornado Alley
            Suite 16
    city:   East Centerville
    state:  KS

ship-to:  *id001

specialDelivery:  >
    Follow the Yellow Brick
    Road to the Emerald City.
    Pay no attention to the
    man behind the curtain.
...
```

Source: https://en.wikipedia.org/wiki/YAML

5.2.5 EXI

Efficient XML interchange language is a binary format that tries to make XML more efficient. It is the binary XML encoding recommended and supported by the World Wide Web Consortium (W3C) and is specified as a W3C recommendation at https://www.w3.org/TR/exi/. It is equivalent to XML at the information set layer (will generate the same information with same structure than XML). Being binary it can reduce the verbosity of XML and with it the size of the serialized data. It also reduces parsing costs. To improve the performance further schemas can be included giving the algorithm more information on the data, however the schema must be present during serialization as well as during deserialization.

5.2.6 CBOR

Concise binary object representation (CBOR) is a binary data serialization format. It is loosely based on JSON and works using name/value pairs. The format being binary is not easily readable by humans, can however be much more efficient than text-based formats.

Data type information for major types: integers, byte string, text string, array, map, tag of number N, simple/float (length is specified, indefinite length possible) date/time strings are supported.

The format was designed to support encoding and decoding on constrained nodes (according to RFC 7228), as well as for high volume transfers. Thus, the formatted data is compact, and the encoder/decoders do not need a lot of computing or memory resources. It was developed to be used with internet of things devices which usually send a lot of data but have only constrained computing possibilities.

The format is specified by the Internet Engineering Task Force (IETF) in RFC 8949 [28].

A human readable diagnostic notation is specified which can be used during debugging.

5.2.7 CDR

Common data representation is developed by OMG (Object management group (also specify DDS protocol)) and part of OMG IDL. It is specified in this context by CORBA v3.0 (https://www.omg.org/cgi-bin/doc?formal/02-06-51). It is almost exclusively used within the CORBA environment.

This format is a binary format and thus not human readable. It assumes prior agreement on type and does not include information about types in the data representation. The OMG interface description language is used to define the data. This makes it lightweight as it includes just the data in a binary format.

An extended version of CDR, that supports evolvable types, is used within the DDS (distributed data service) middleware.

5.2.8 OPC UA Binary

The OPC UA Binary is a data format developed by the OPC foundation to meet the needs of OPC UA applications. According to OPC UA specification [26], the format is designed primarily for fast encoding and decoding while also considering the size of the encoded data on the wire. It is well integrated into the OPC UA environment and uses a broad range of primitive types including Booleans, Integers, Floating Point, String, DateTime, ByteString and several more specialized types. The format is almost exclusively used within the OPC UA framework.

Next to the binary format, OPC UA also defines OPC UA XML and OPC UA JSON formats for data serialization to be compatible with XML or JSON based applications over the web. All the data formats are specified in OPC 10000-6 Part 6: Mappings.

5.2.9 Apache Thrift

Apache Thrift does not define the serialization per se but rather an interface at program (application layer) level and forms a remote procedure call (RPC) framework more lightweight than CORBA or SOAP (XML based) as it is kept simple and uses binary.

Structure of data is defined using interface description language. This description is used for serialization and deserialization.

Several predefined serialization protocols are included in the framework: binary, compact binary, http-friendly (over JSON).

The implementation of Apache Thrift is based on the white paper: https://thrift.apache.org/static/files/thrift-20070401.pdf. Apache Thrift is not standardized but open source (Apache license 2.0) and maintained by Apache Foundation. A wide range of standard APIs are available.

Depending on the serialization used the data can be human readable (however it will then take up more space). The mandatory use of interface descriptions however makes for human friendly handling of the serialization. However, as the data serialization is thus not self-describing, a copy of the interface description needs to be present on all nodes that need to deserialize the data.

5.2.10 Protocol buffers

Protocol buffers is a framework similar to Apache Thrift for distributed applications to communicate and exchange data. It is developed and maintained by google under an open-source license. Serialization uses binary types (not human readable, an ASCII version is implemented for debugging purposes but is not backwards nor forwards compatible). It is designed to be smaller and faster than XML.

Comes natively with Code generators for C++, Python, Java, C#, Go, Ruby, JavaScript… (Protobuf 3.0), other languages have third party implementations (C, Perl, …).

As Apache thrift an interface description language is used to define a schema for serialization. The serialization

is not self-describing and needs a copy of the interface description for the deserialization of the data.

Reference: https://developers.google.com/protocol-buffers

5.2.11 Apache Avro

Apache Avro is a data serialization system that provides similar functionality to Apache Thrift or Protocol Buffers. It serializes data in a binary format based on a schema. It can also use JSON to encode the data although this is mainly intended for debugging purposes.

The schema is stored with the data in a file. The data and schema are fully self-describing facilitating thus the data can be processed in a more dynamic manner than in Apache Thrift or Protocol Buffers, which need to generate code from the interface description prior to execution. Another advantage with this system is that if the program expects a different schema, as both schemas are present this can be easily resolved (missing fields, extra fields, etc.). Avro schemas are defined with JSON. Avro comes with optimization possibilities where schemas can be exchanged and retained at the beginning of a connection between two nodes. Thus, data can then be sent without repeating the schema at every transmission.

Avro has official releases for C, C++, C#, Java, PHP, Python, and Ruby

Apache Avro is available under an Apache License 2.0 (permissive free software license). (https://avro.apache.org/docs/current/)

5.2.12 ASN.1

The abstract syntax notation one (ASN.1), is a standard interface description language for defining data structures. Together with a set of encoding rules ASN.1 can be used to serialize and deserialize the data. Implementations of the standard in the form of compilers exist that create libraries of code from the ASN.1 data description, that can encode or decode data. These tools are well established for Java, C and C++.

The standard was created by the international telecommunications union (ITU) and is specified at https://www.itu.int/rec/T-REC-X.680/en.

Different encoding rules generate different outputs so the performance of this serialization varies depending on the encoding rules (binary: BER, DER, PER,… Human readable: XER, JER,…). Custom encoding rules can also be defined using the standardized encoding control notation (ECN) which is part of the ASN.1 family of standards.

The IEC 61375-2-3 [19] standard defines data structures using a system based on ASN.1.

5.3 Evaluation of data formats:

The evaluation of the relevant criteria is shown in Table 9 where the different serialization formats are evaluated for relevant criteria using a qualitative scale (+: good, 0: neutral, -: bad). The following criteria were considered, although only the relevant ones are shown in Table 9.

- Readability by humans: Text based formats, although also encoded to bytes as UTF-8 can be directly read by most computers and programs and are thus considered human readable. It is very easy to read the data from these. All binary formats are not human readable and thus scored with a bad score for this criterium. Environments like Apache Thrift or Protocol Buffers support several encodings, including text-based ones. However, we always consider the binary format for these, as they will minimize the size of the serialized data and as the text-based formats are mainly supported for development and debugging purposes and are not always forwards and backwards compatible.
- Data typing: This criterium is used to differentiate between formats that include data types in the data serialization (+), those who use a schema or interface description language to define the data types separate from the data (0) and those who do not support data type information (-).

- Performance of encoder/decoder: The performance of encoders and decoders is evaluated regarding computation/memory needs as well as speed. The exact performances are difficult to estimate as one would need to run benchmarks for each format with typical data, as some formats could be faster for certain types but not for others for example.
- Space needs of serialized data (directly linked to networking speed): The memory size of the serialized data is evaluated here. This is also the space the data takes up on the network when sent. To ensure fast communication as well as high network throughput, the size of the serialized data should be kept small. Here binary formats are almost always better than text-based formats. However, the size of the serialized data can also depend on the actual data and would need benchmarks with typical data to be evaluated definitively.
- Platform / language independence: Serialization data formats are developed with platform independence in mind. All data formats are platform and language independent. Thus, this criterium is not figured in Table 9.
- Availability of implementations: Some data formats are more widely used than others. This generally translates to more implementations of decoders and encoders in a more diverse set of languages. The implementations can be openly available or commercial products, usually with higher performance or more development tools. Here we also evaluate if the data format is widely used in relevant industry areas (automation, networking)
- Open / proprietary: None of the data formats is proprietary. However, we distinguish here whether the data serialization format is specified or standardized by an international institution and widely used (+), the format is specified or standardized by a non-international foundation but still widely used in industrial automation (0) or whether the format is standardized by an international institution or foundation but used by few key players on the market (-). The evaluation reflects the level of trust placed in the data format for its future relevance and continued maintenance of the standard. The Bitstream format was evaluated with a good mark (++) for this criterion, even though it could be considered not very open or even proprietary from the outside. However, as the format is completely controlled and specified within the system the concerns this evaluation criteria addresses are not relevant.
- Flexibility for upgrades: Here the rigidity of the data format to updates in the data, like adding new variables, is evaluated. A flexible format needs only small changes to the applications to handle a change of the data.
- Complexity of supported data structures: Some data formats can handle more complex data structures than others, as for instance dictionaries, references to other objects or arrays of mixed typed elements. However, as the needed data complexity is rather low for this application, all data formats can support sufficiently complex data. Therefore, this criterion will not be evaluated further.

	Readability by humans	Data typing	Performance of encoder/ decoder	Size of serialized data	Availability of APIs	Open/proprietary	Flexibility for upgrades
Bitstream	-	-	++	++	-	++	-
XML	+	+	-	--	++	+	+
JSON	+	0	+	-	++	+	+
YAML	+	0	+	-	0	-	+
EXI	-	+	+	+	+	+	+
CBOR	-	+	+	+	+	+	+
CDR	-	+	+	+	+	0	0
OPC UA Binary	-	+	+	++	+	0	0
Apache Thrift	-	0	+	++	++	-	0
Protocol buffers	-	0	+	++	++	-	0
Apache Avro	-	0	+	+	+	-	+
ASN.1	-	+	+	++	+	+	0

Table 9: Comparison of different data serialization formats.

There is no data format that excels in all the criteria and is an obvious pick. Further, some criteria are more important in some use cases than others. Thus, the need to differentiate between different use cases arises. On one hand we will consider process data for time-sensitive applications going over TSN-Ethernet such as the vehicle locator data for instance. On the other hand, we will consider message data for non-time-sensitive applications going over standard Ethernet, like diagnostic messages for instance. Several data formats can coexist on the CCN that are tailored to the needs of the applications. It is however preferable to keep the data formats somewhat uniform throughout the applications to make for a more modular and coherent system. This will facilitate application development.

In general, however, due to the long lifetime of the CCN, it is preferable to have a solution that can either be completely controlled by the specifications of the CCN or that is specified in a standard by an international organization or widely used in the industry of interest such that unforeseen changes can be prevented. Thus, we will not consider formats with a bad rating in the open/proprietary criterium.

5.3.1 Time critical applications

For time critical applications a fast encoding and decoding is needed to meet the strong timing requirements. Also, the size of serialized data is important due to the limitation of the maximum payload of an (TSN-)Ethernet frame of 1500 bytes. Other criteria like readability by humans, data typing, and availability of APIs are less important. This implicit weighting of the criteria is considered in the following listed possible formats for time critical applications.

5.3.1.1 Bitstream

For process data for applications relying on fast data transmission or even real-time, it is essential to have small, serialized data sizes as well as fast encoding and decoding of the data. The perfect example for this is bitstream. They sacrifice readability and flexibility for smaller data sizes and fast encoding as the encoder can be specifically written for the data it works on.

5.3.1.2 OPC UA Binary

Another format that is also used in industry for real time applications is OPC UA Binary. OPC defines not only a serialization format but also a data exchange protocol that can work with TSN in order to deliver real time process data. It is well established in industry and supported by several key players in automation such as ABB Automation, Siemens, B&R industrial automation, Bosch Rexroth, etc. It is a modern solution that is however still evolving and due to the breadth of services provided, not all implementations are compatible. It would integrate well with the OPC UA communication protocols, is however seldom used outside of this framework.

5.3.1.3 CBOR

CBOR being a binary, self-describing language that can be encoded and decoded using limited resources could be an interesting alternative in case of constrained nodes such as embedded systems for instance. It will still be relatively short as it uses a binary format but will not manage to compete with a bit-stream or a schema informed language.

5.3.1.4 ASN.1

ASN.1 using BER or PER is another suitable standard that is quite widespread due to its early standardization and use in telecom industries. However, most of the implementations are in-house developments or commercial products. Only a few open implementations exist. The syntax of the description language is well known and used in standards relevant to the railway industry. (IEC 61375-2-3 [19] for instance).

5.3.1.5 CDR

When using the DDS middleware at the session layer, extended CDR together with OMG-IDL is used by DDS. To get to the full potential of DDS and its data-centered approach, it would be counterproductive to use another format.

5.3.1.6 Other

Other formats for that are fully schema informed could be used like Apache Thrift, Protocol Buffers or Apache Avro. As they are fully schema informed, they also have small sizes of serialized data. From these Protocol Buffers would probably be the best choice as it is well established (compared to Avro which relatively new) and well documented (compared to Thrift which has less documentation). However, Protocol Buffers were not made to be used in embedded environments (although stripped down implementations exist (nanopb)) and are a less universal standard as they are not standardized by an international body of standardization. They are maintained and specified by google which is a company not necessarily aligned with the railway industry.

5.3.2 Non-time-critical applications

For non-time-critical data such as diagnostic data, a human readable data format would be well suited. Especially for maintenance purposes the debugging and reading of messages could be made a lot easier as the data can be accessed directly and in a comprehensible manner. Moreover, the widespread use of certain text-based data formats also in other industry fields as well as their good standardization, means that they will continue to be relevant in the future. Other criteria like fast encoding and decoding and size of serialized data are less important. This implicit weighting of the criteria is considered in the following listed possible formats for time critical applications.

5.3.2.1 JSON

Due to its small size and good performance, JSON would be the most suitable solution. For explicitly introducing the data types of the data fields, the use of JSON schemas would be preferable.

5.3.2.2 XML

XML is verbose, and the size of the serialized data is higher than JSON. Even though, as XML is already used for different standardized communications, it is also solution for non-time critical applications.

5.3.2.3 others

YAML as the last of the text-based formats is not standardized by a reputable standardization body and has a very open syntax that would allow for the same data to be represented in too many ways, making the size of the format less predictable and not improving the readability of the data. Therefore, it is not proposed as a preferable solution.

5.3.3 Conclusion

Considering the differentiation between time critical and non-time critical data for the evaluation, the following possible formats remain for the corresponding applications.

Possible formats for time-critical application data over TSN-Ethernet	**Possible formats for non-time-critical application data over standard Ethernet**
Bitstream	JSON
OPC UA Binary	XML
CBOR	
ASN.1	

Table 10: Possible Data Serialization Formats considering the respective application

For safety- and time-critical CCS-applications todays interfaces specifications are defining bitstream packets. Thus, it is not necessary to have a self-describing format and one can expect all the applications to follow the specified interfaces. And with reserved parts in the data packets there is still flexibility for further updates. Therefore, Bitstream is recommended for the use of time-critical applications.

For non-time-critical applications JSON with the use of schemas is recommended considering the evaluation of the different criteria.

This is however only a recommendation. Several data formats can coexist on the same network and others could be used. E.g. CBOR can be used to have a short binary format that uses few computational resources for time-critical data. Or OPC UA Binary could be integrated into applications making use of the OPC UA communication protocol on session layer. XML can still be used too, where it is already used (e.g. UIC 559). The aim of this recommendation is to create an ecosystem on the CCN that is as uniform as possible.

As already mentioned, when discussing the OPC UA binary format, the recommendation might be influenced by the choice of session layer protocol. At the moment, TRDP 2.0, OPC-UA PubSub and DDS/RTPS are still considered. The session layer protocols OPC-UA and DDS also provide or integrate, in the case of DDS, the data encoding using OPC UA binary or extended CDR.

6 Network architecture and cyber security

6.1 Network Architecture of Next-Generation Train Communication Network (NG-TCN)

The Shift2Rail (S2R) projects CONNECTA and SAFE4Rail elaborated the Next-Generation Train Communication Network (NG-TCN) which is one of the main building blocks of S2R's next generation of TCMS architectures. The network architecture of the NG-TCN is shown in Figure 10.

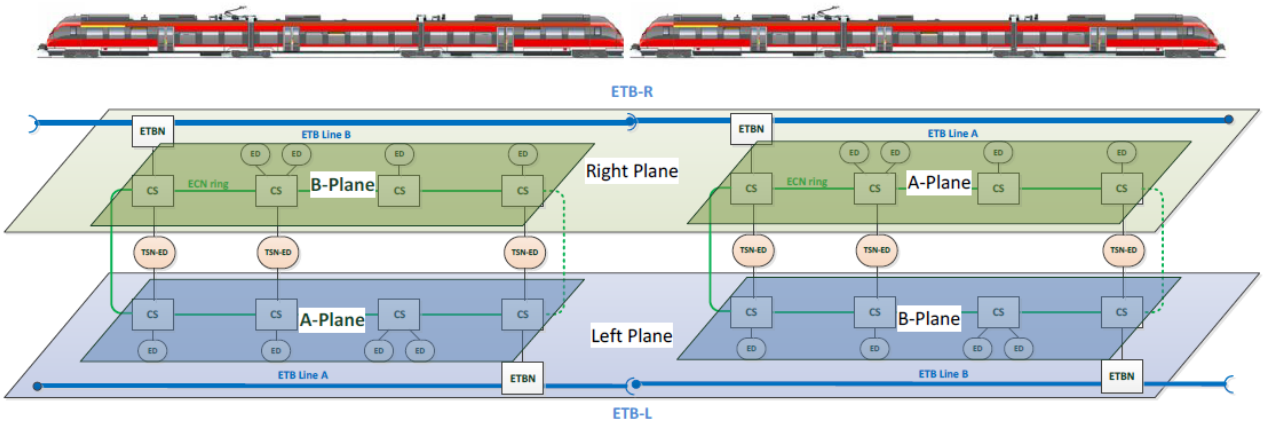

Figure 10: Network architecture of NG-TCN from [15]

The layout of the NG-TCN (ECN and ETB) will be a mixture of ring structure and the 'ladder' configuration. Non-safety-related and soft- or non-real-time devices are attached to the ECN via a single link, safety-related or hard-real-time devices will connect to the two planes of the ladder. Traffic on each 'wing' will be separated by the consist switches and will use the right respectively the left line of the ETB. This adds reliability and also eases realizing SIL4 functions on the ECN.

Safety-related or hard-real-time devices connected to the consist network thus will need two Ethernet ports (switch ports), which emit and receive duplicated frames. This procedure of frame replication and elimination is standardized in the TSN substandard IEEE802.1CB. It is shown in Figure 11.

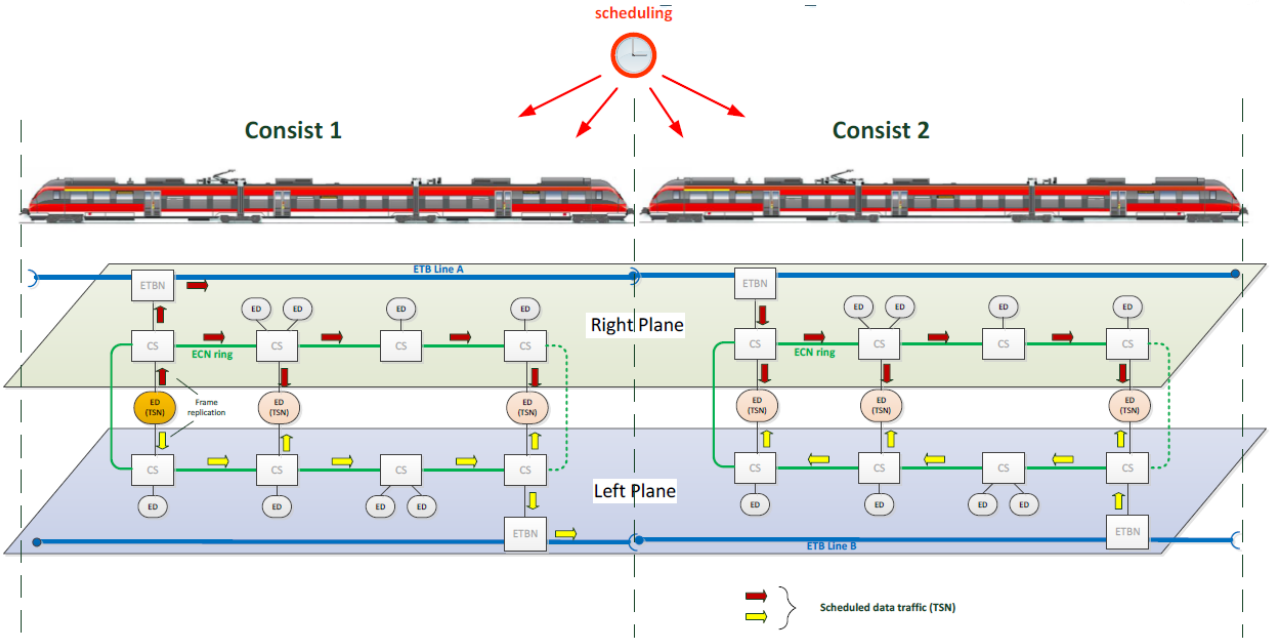

Figure 11: Data flow for TSN traffic on NG-TCN [17]

On the physical networks ECN and ETB, CONNECTA defines mainly three different logical networks TCMS, operator-oriented services (OOS) and customer-oriented services (COS). The draft of the VLAN definition from CONNECTA is shown in Table 11.

No	Name	VLAN ID	Description
1	ECN-TCMS	2	Consist level VLAN used by connected eligible devices for non-TSN TCMS data traffic.
2	ECN-OOS	16	Consist level VLAN used by connected eligible devices for non-TSN OOS (operator-oriented services) data traffic.
3	ECN-COS	20	Consist level VLAN used by connected eligible devices for non-TSN COS (customer-oriented services) data traffic.
4	ECN-TSN-A-X	X = 32 … 287 (256 IDs)	Consist level VLANs used by TSN devices for TSN data streams.
5	ECN-TSN-B-X		
6	ETB-TCMS	5	Train level VLAN used by all ETBN for non-TSN TCMS data traffic. This VLAN is configured on both ETB Line A and ETB Line B.
7	ETB-OOS	24	Train level VLAN used by all ETBN for OOS data traffic. This VLAN is configured on both ETB Line A and ETB Line B.
8	ETB-COS	28	Train level VLAN used by all ETBN for COS data traffic. This VLAN is configured on both ETB Line A and ETB Line B.
9	ETB-BEACON	6	Train level VLAN used by all VCU (Train Integrity Validator) for side selective BEACON telegrams. This VLAN is configured on both ETB Line A and ETB Line B.
10	ETB-TSN-A-X ETB-TSN-B-X	X = 288 … 543 (256 IDs)	Train level VLANs used by all ETBN for ETB TSN data streams. TSN data streams use identical VLAN-IDs on both ETB planes.

11		3 … 4, 7 … 15, 17 … 19, 21 … 23, 25 … 27, 29 … 31, 544 … 4094	Reserved for future use
12		0, 1, 4095	Reserved (not for application use)

Table 11: Predefined VLAN for NG-TCN operation (preliminary) from [15]

6.2 Cybersecurity

6.2.1 IEC 62443-3-3 [22] and TS 50701 [23]

In the industry sector the standard series IEC 62443 is established for cybersecurity. The railway sector adopted this standard series and shows in the preliminary technical specification TS 50701 [23] how to apply the industry standard IEC 62443. Based on a threat analysis, followed by a risk analysis the relevant Security Level will be derived. For CCS applications the security level is defined as SL3, as written in [9]. In the following table the protection corresponding to a specific security level is defined.

Security Level	Protection against attacker type
SL1	Protection against casual or coincidental violation
SL2	Protection against intentional violation using simple means with low resources, generic skills and low motivation
SL3	Protection against intentional violation using sophisticated means with moderate resources, IACS specific skills and moderate motivation
SL4	Protection against intentional violation using sophisticated means with extended resources, IACS specific skills and high motivation

Table 12: Security Levels from IEC 62443-3-3 [22] and TS 50701 [23]

In the IEC 62443-3-3 [22] standard requirements for different security levels are defined. Regarding network topology for CCN the following requirements on the restricted data flow are important:

Security Level	System Security Requirement 5.1 and enhancements
SL1	The control system shall provide the capability to logically segment control system networks from non-control system networks and to logically segment critical control system networks from other control system networks.
SL2	The control system shall provide the capability to physically segment control system networks from non-control system networks and to physically segment critical control system networks from non-critical control system networks.
SL3	The control system shall have the capability to provide network services to control system networks, critical or otherwise, without a connection to non-control system networks.
SL4	The control system shall provide the capability to logically and physically isolate critical control system networks from non-critical control system networks.

Table 13: System Security Requirements 5.1 – Network segmentation from IEC 62443-3-3 [22]

All system security requirements from IEC 62443-3-3 [22] are generally applicable to railway applications according to the security levels (SL-T) of the zones and conduits in the system under consideration (SuC). Nevertheless, due to the peculiarity of the railway application, the TS 50701 [23] informs about the existence of railway specific considerations as guidance. To the system security requirement SR 5.1 the following railway notes are listed.

Security Level	Title	Railway notes
SL1	Network segmentation	In response to an incident, it may be necessary to break the connections between different network segments. In that event, the services necessary to support essential operations should be maintained in such a way that the devices can continue to operate properly and/or shutdown in an orderly manner. This may require that some servers may need to be duplicated on the control system network to support normal network features, for example dynamic host configuration protocol (DHCP), domain name service (DNS) or local CAs. It may also mean that some critical control systems and safety-related systems be designed from the beginning to be completely isolated from other networks.
SL2	Physical network segmentation	Independence from non-control networks is required at SL2. In case physical segregation is technically not feasible or even an increase in cybersecurity risks, a logical segregation concept is acceptable explicitly if the following associated system security requirements [SR 1.2, SR 1.8, SR 1.9, SR 3.1/SR 3.1 RE 1, SR 3.7, SR 4.1/SR 4.1 RE 1, SR 6.2, SR 1.5 RE 1] are fulfilled.
SL3	Independence from non-railway application networks	
SL4	Logical and physical isolation of critical networks	The criticality of a railway application is determined by the risk assessment and that should influence the logical and physical isolation. The usage of segmentation methods like different fibers or colors for fiber-optic cables or the usage of cryptographic measures like those mentioned in EN 50159 are ways to implement this requirement in railway applications.

Table 14: System Security Requirement notes on SR 5.1

The system security requirement SR 5.1 for SL2 and higher from IEC 62443-3-3 [22] require physical segmentation between control system networks and non-control system networks as well as between critical control system networks and non-critical control system networks. Nevertheless, the TS 50701 [23] supports the clear physical segmentation between control system networks and non-control system networks above SL1. But it allows logical segmentation between critical and non-critical control system networks even on SL2 or SL3 if certain system security requirements (see Table 14) are fulfilled.

6.2.2 Impact of cyber security standards on network architecture

Considering the requirements from IEC 62443-3-3 [22] and the railway notes in TS 50701 [23] one physical network for control system networks like CCN or ECN and non-control system networks like operator network will not be acceptable from a cybersecurity point of view. The control system networks shall be physically segmented from the non-control system networks. Non-control system networks can be operator networks for e.g. CCTV, passenger information. Passenger networks for public internet access or entertainment on passenger devices must be physically segmented as well from control networks like CCN or ECN. Also, the

communication devices for the access to the trackside systems shall be physically separated from control networks.

Also, critical control networks shall be separated from non-critical control networks. As noted in TS 50701 [23] a logical segmentation between critical and non-critical control system networks on SL3 is necessary with associated system security requirements [SR 1.2, SR 1.8, SR 1.9, SR 3.1/SR 3.1 RE 1, SR 3.7, SR 4.1/SR 4.1 RE 1, SR 6.2, SR 1.5 RE 1].

6.3 Network architecture for new trains with NG-TCN

Considering the currently defined network architecture of NG-TCN and cyber security aspects, the following four different network architecture scenarios are derived. The scenario A shows the vehicle architecture like today's vehicles with two physically separated networks for CCS and TCMS systems. In scenario B the networks of the CCS and TCMS systems are two logically separated networks, what the NG-TCN generally supports. If the cyber security standards IEC 62443-3-3 [22] and TS 50701 [23] are applied adequately, the control and non-control systems as well as critical control and non-critical control systems should be clearly separated, which leads to scenario C. Scenario C represents a proposal for separating the systems due to their criticality: critical control systems are logically segregated from non-critical control systems. Finally, if for scenario C a physical separation between critical and non-critical control networks is needed, scenario D can be applied.

Due to cyber security aspects, the NG-TCN architecture was considered in a different manner. The networks of NG-TCN architecture (TCMS, OOS, COS) are physically segmented or even isolated, instead of only having a logical segmentation.

In order to be able to integrate the CCN in ECN of the NG-TCN or establish an own ECN-like network for the CCN, every hard-real-time CCS device (e.g. Safe Computing Platform etc.) should have at least one TSN-capable Ethernet port whereas for soft- or non-real-time CCS devices a single standard non-TSN-capable Ethernet port is sufficient. Hard-real-time CCS devices could use both planes of NG-TCN with two TSN-capable Ethernet ports in order to improve reliability and availability.

6.3.1 Scenario A: CCN as physically separated network

In this scenario, the CCN and the NG-TCN are physically separated. All communication components, all operator components and all security devices are located also on their own physically separated network. The network with connected public devices (e.g. public WLAN access point with connected passenger devices) is physically isolated. Every network represents a security zone.

The central element between all networks (except physically isolated network with public devices) is the ECN/ECN Security Gateway (GW). The ECN/ECN Security Gateway routes the traffic between the different networks and acts as firewall between them. Cyber security aspects between trackside and onboard networks are covered in the FRMCS onboard system and MCG.

With the CCN as a physically separated network, the CCS and the TCMS domain are strictly divided. However, the complexity of the network configuration (TSN schedule) will increase due to the two separated network configurations linked together. And as the two network configurations of the two domains will be done independently the hard-real-time behavior for data between the two domains will suffer (latency and jitter will increase).

In the following two figures the physical and logical network architecture of scenario A is shown.

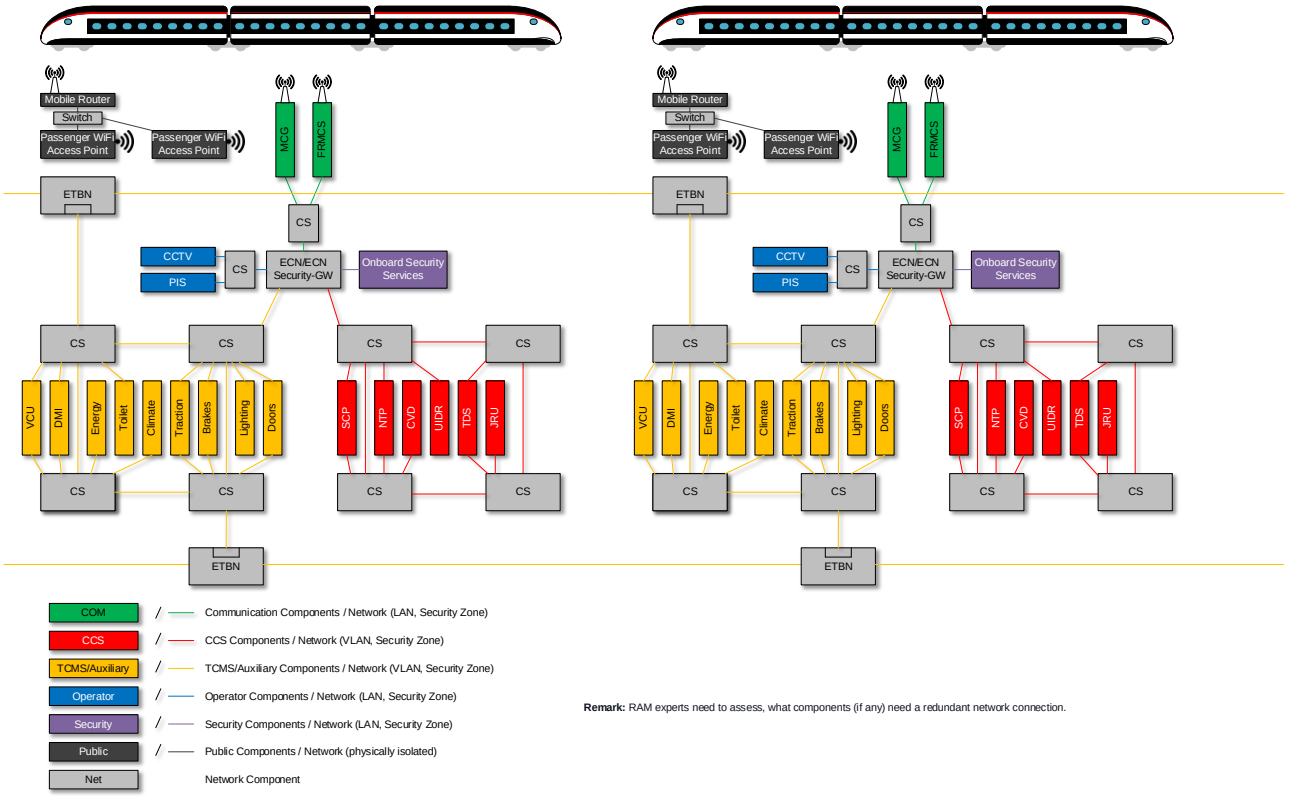

Figure 12: Physical network architecture scenario A: CCN as physically separated network

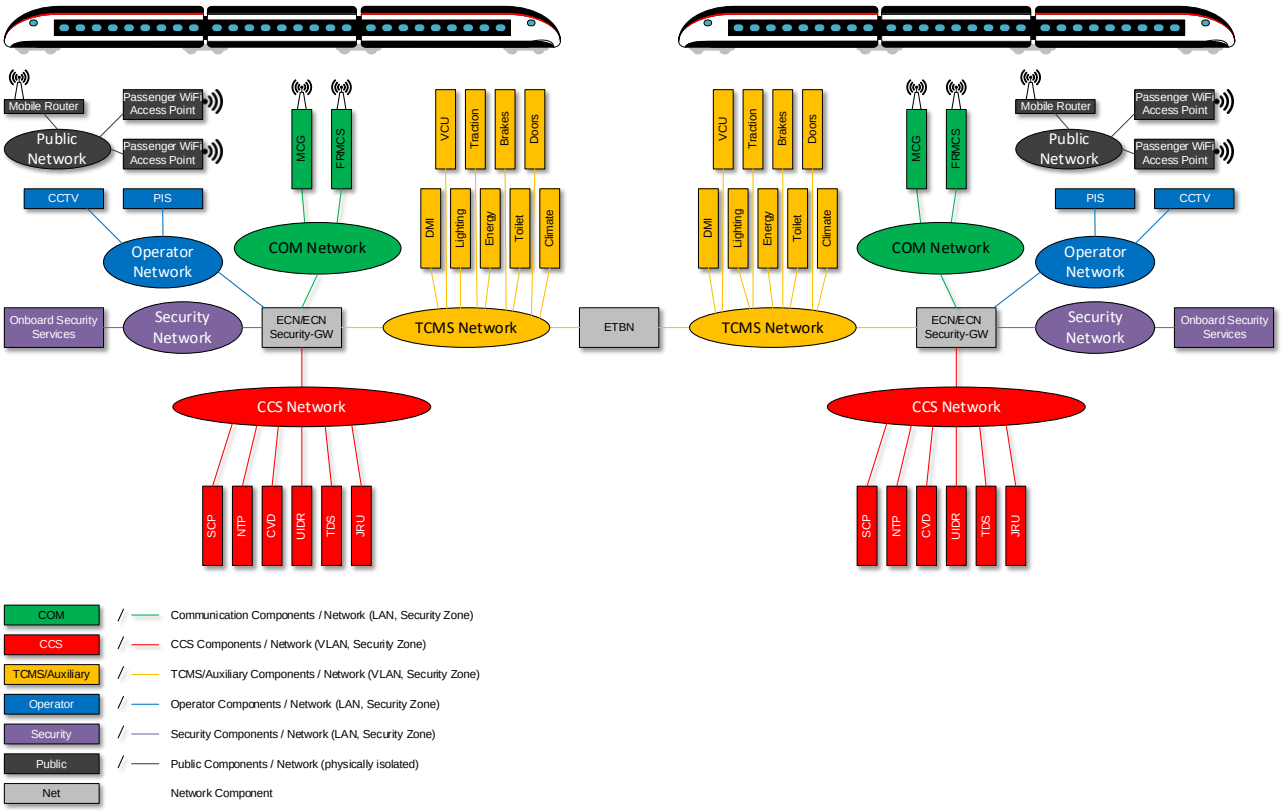

Figure 13: Logical network architecture scenario A: CCN as physically separated network

6.3.2 Scenario B: CCN as logically separated network

In this scenario, the CCN and the NG-TCN are located on the same physical network. But the CCN and NG-TCN are logically separated. So, the CCN and the NG-TCN represent an own logical network. All communication components, all operator components and all security devices are located on their own physically separated network. The network with connected public devices (e.g. public WLAN access point with connected passenger devices) is physically isolated. Every logical network represents a security zone.

The central element between all networks (except physically isolated network with public devices) is the ECN/ECN Security Gateway (GW). The ECN/ECN Security Gateway routes the traffic between the different networks and acts as a firewall between them. Cyber security aspects between trackside and onboard networks are covered in the FRMCS onboard system and MCG.

With the CCN as a logically separated network, the CCS and the TCMS domain are strictly divided. The complexity of the network configuration (TSN schedule) is low due to the fact of having only one network configuration for the common physical network (CCN and NG-TCN). The hard-real-time behavior for data between the two domains over the ECN/ECN Gateway is sufficient.

In the following two figures the physical and logical network architecture of scenario B is shown.

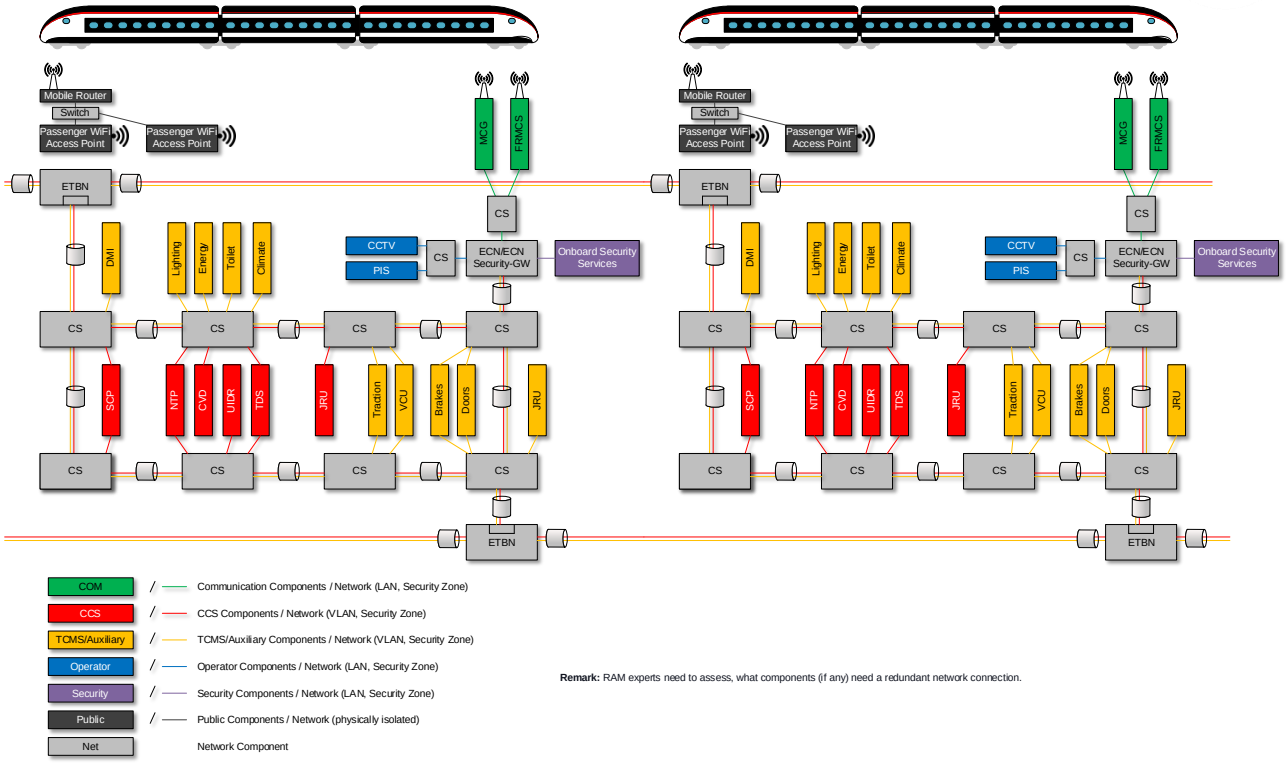

Figure 14: Physical network architecture scenario B: CCN as logically separated network

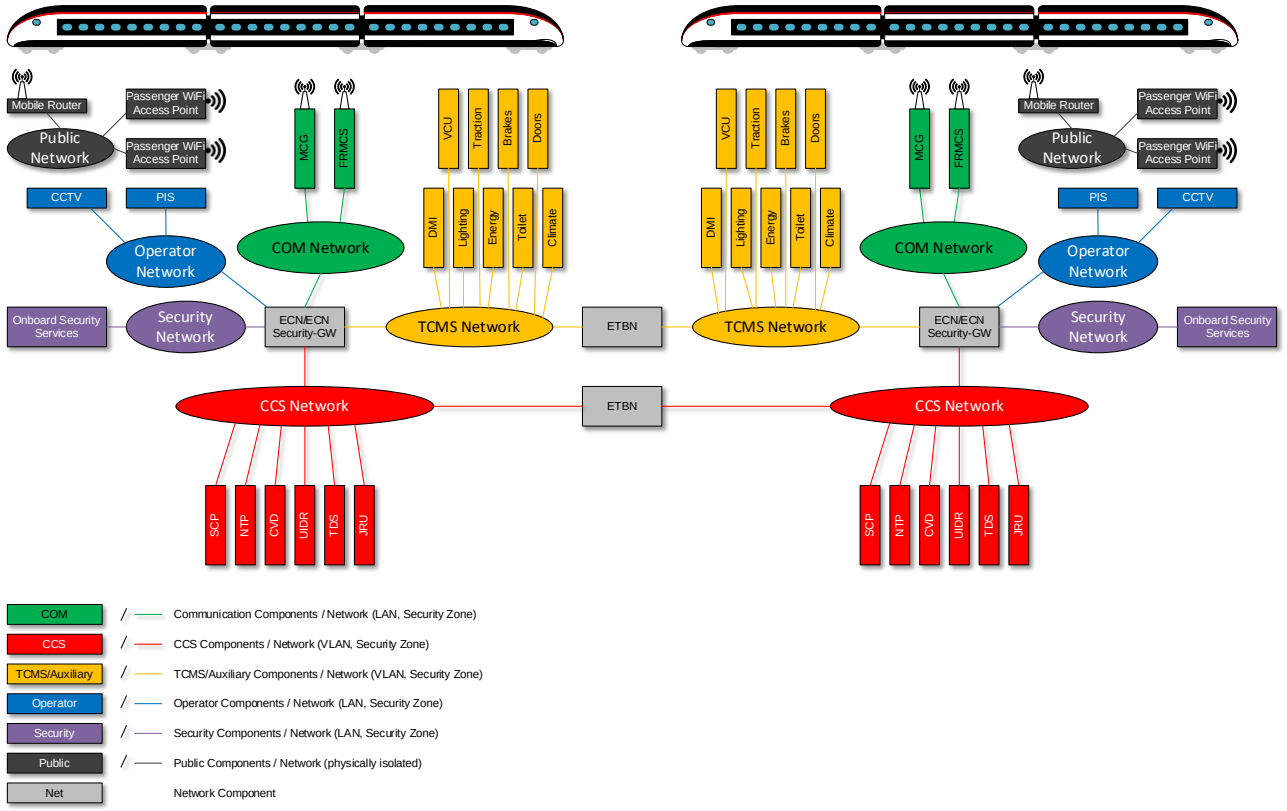

Figure 15: Logical network arch. scenario B: CCN as logically separated network

6.3.3 Scenario C: Common critical control network logically separated

In this scenario, the networks or security zones are derived from the criticality of the functions of the train. The CCS components and the critical control components of the TCMS domain (e.g. VCU, Traction, Brakes, Doors) are located on the same logical network. The non-critical control components of the TCMS domain, the auxiliary components, like e.g. toilets, climate and lighting are located in a logically separated network on the same physical network together with the critical control components. So, the CCS/TCMS and the auxiliary network represent an own logical network. The detailed split of the TCMS domain into a critical and a non-critical part should be defined by the TCMS sector (e.g. CONNECTA, X2Rail). All communication components, all operator components and all security devices are located on their own physically separated network. The network with connected public devices (e.g. public WLAN access point with connected passenger devices) is physically isolated. Every logical network represents a security zone.

The central element between all networks (except physically isolated network with public devices) is the ECN/ECN Security Gateway (GW). The ECN/ECN Security Gateway routes the traffic between the different networks and acts as firewall between them. Cyber security aspects between trackside and onboard networks are covered in the FRMCS onboard system and MCG.

In this scenario critical control systems are logically separated from non-critical control systems. The complexity of the network configuration (TSN schedule) is low due to the fact of having only one network configuration for the common physical network (CCN and NG-TCN). The hard-real-time behavior for data between the CCS domain and critical part of TCMS domain without a gateway in between is excellent (very low latency and jitter in µs and ns range).

In the following two figures the physical and logical network architecture of scenario C is shown.

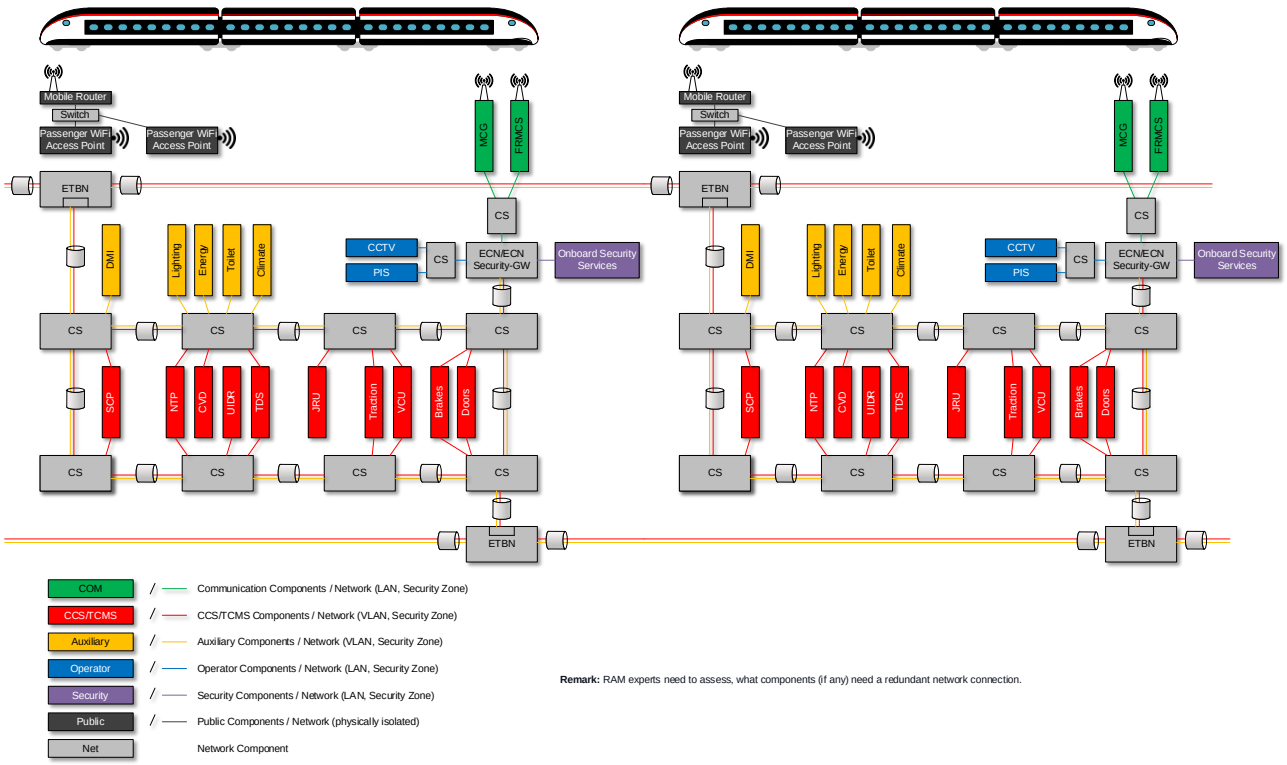

Figure 16: Physical network arch. scenario C: Common critical control network logically separated

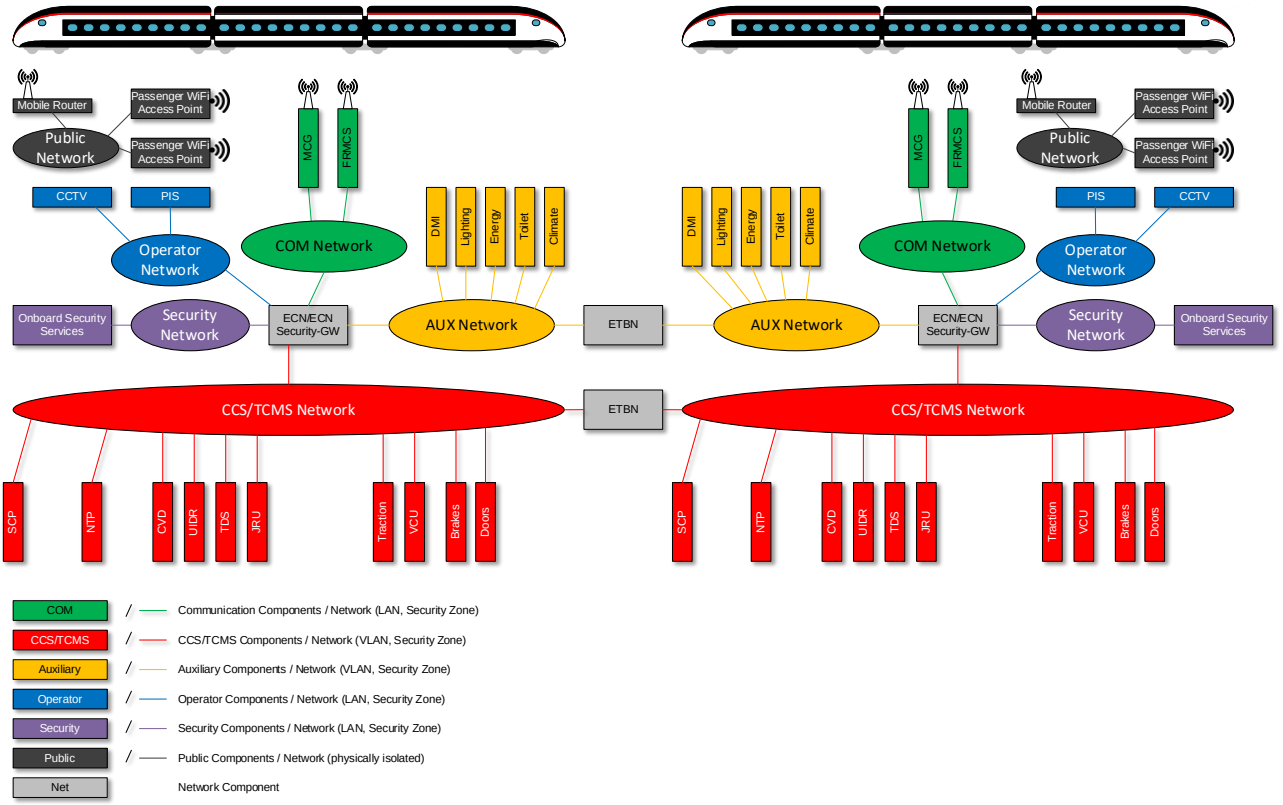

Figure 17: Logical network arch. scenario C: Common critical control network logically separated

6.3.4 Scenario D: Common critical control network physically separated

In this scenario, the networks or security zones are derived from the criticality of the functions of the train. The CCS components and the critical control components of the TCMS domain (e.g. VCU, Traction, Brakes, Doors) are located on the same logical network. The non-critical control components of the TCMS domain, the auxiliary components, like e.g. toilets, climate and lighting are located in an own physically separated network. The detailed split of the TCMS domain into a critical and a non-critical part should be defined by the TCMS sector (e.g. CONNECTA, X2Rail). All communication components, all operator components and all security devices are located on their own physically separated network. The network with connected public devices (e.g. public WLAN access point with connected passenger devices) is physically isolated. Every logical network represents a security zone.

The central element between all networks (except physically isolated network with public devices) is the ECN/ECN Security Gateway (GW). The ECN/ECN Security Gateway routes the traffic between the different networks and acts as a firewall between them. Cyber security aspects between trackside and onboard networks are covered in the FRMCS onboard system and MCG.

In this scenario critical control systems are physically separated from non-critical control systems. The complexity of the network configuration (TSN schedule) is high due to the fact of having two network configurations for the two physical networks CCS/TCMS and Auxiliary. The hard-real-time behavior for data between the CCS domain and critical part of TCMS domain without a gateway in between is excellent (very low latency and jitter in µs and ns range).

In the following two figures the physical and logical network architecture of scenario D is shown.

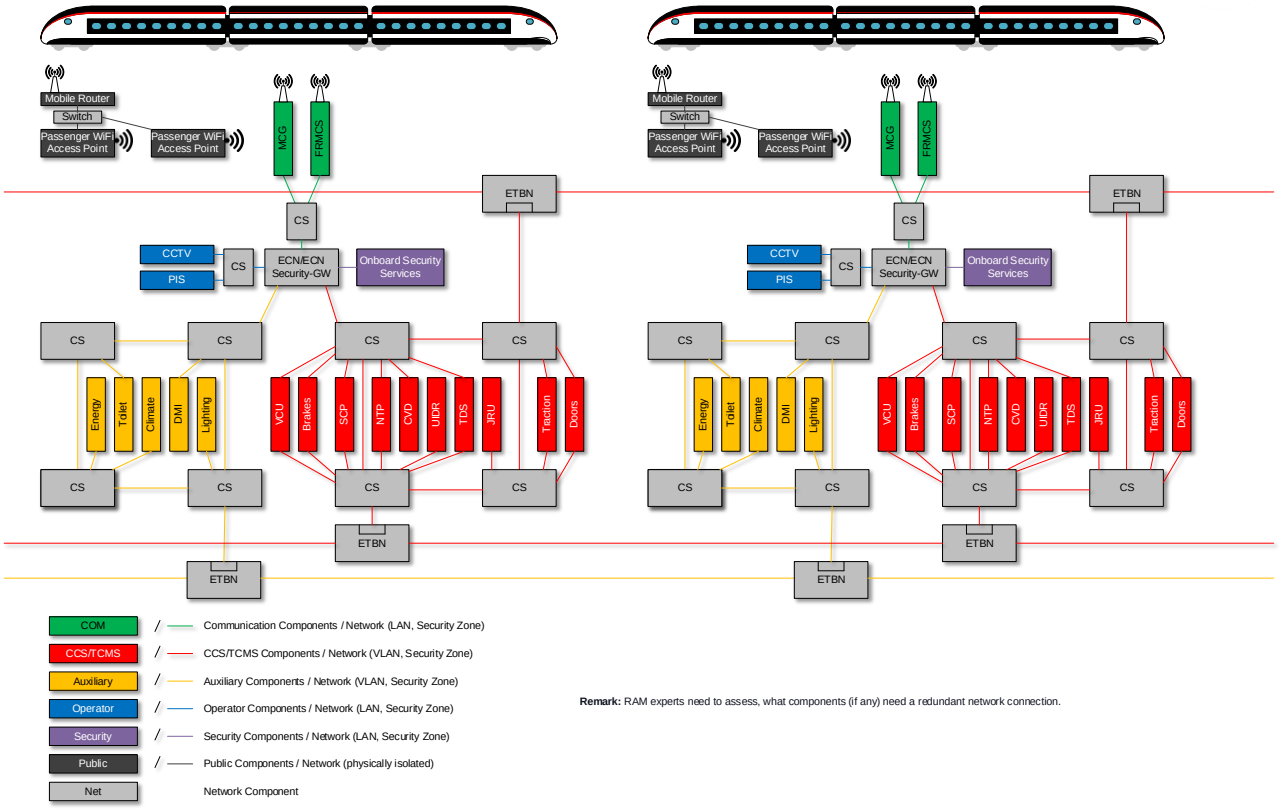

Figure 18: Physical network arch. scenario D: Common critical control network physically separated

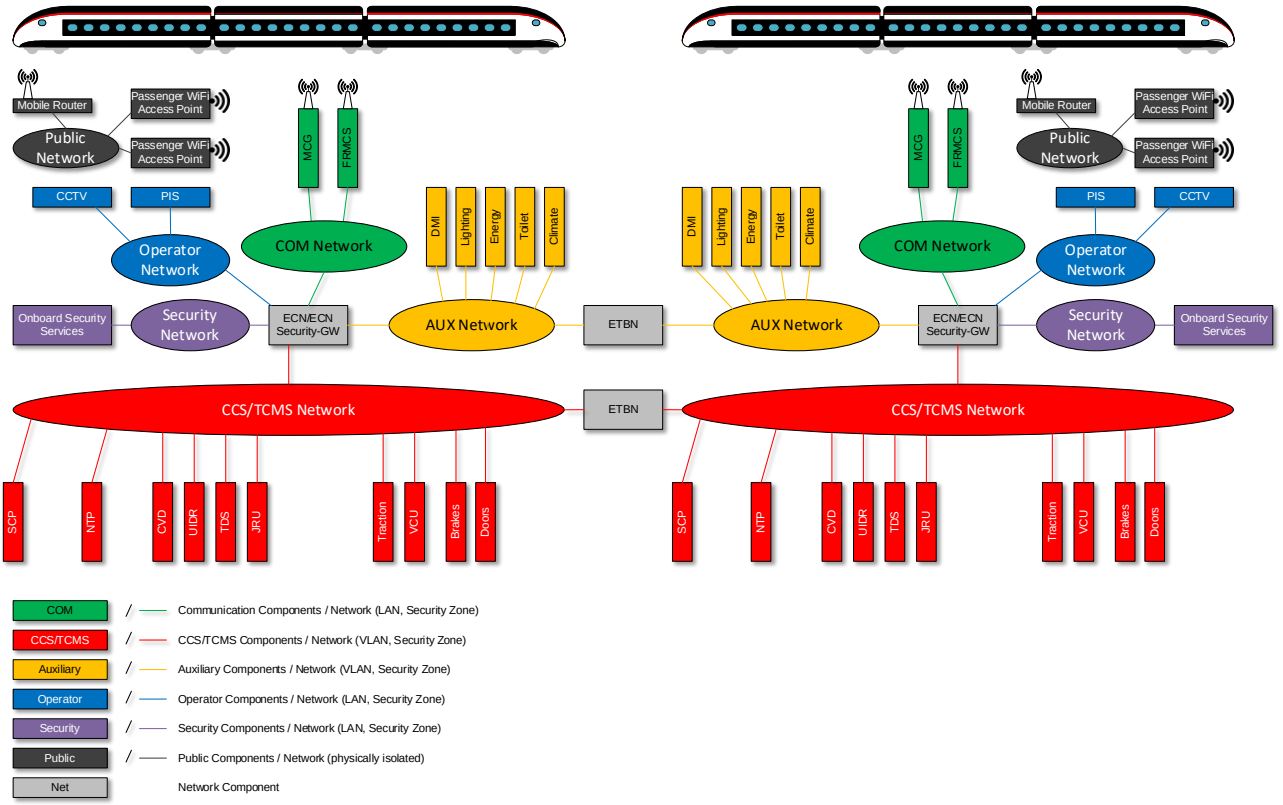

Figure 19: Logical network arch. scenario D: Common critical control network physically separated

6.3.5 Conclusion

Table 15 gives an overview of the advantages and disadvantages of the different evaluated network architectures for new trains.

Network Architecture for new Trains	Advantage	Disadvantage
Scenario A: CCN as physically separated network (Figure 12, Figure 13)	- Clear physical separation between CCS and TCMS domain. - Consist switches are not security relevant.	- No clear separation between critical control systems and non-critical control systems - Complex network configuration (e.g. for hard-real-time traffic between CCS and TCMS domain) - No direct communication between CCS and TCMS possible. Therefore, bad hard-real-time behavior. - No direct inter-consist communication for CCS
Scenario B: CCN as logically separated network (Figure 14, Figure 15)	- Clear logical separation between CCS and TCMS domain - Direct inter-consist communication for CCS possible	- No clear separation between critical control systems and non-critical control systems - No direct communication between CCS and TCMS possible. - Consist switches responsible for the logical segmentation become security relevant and have to meet certain requirements.

Scenario C: Common critical control network logically separated (Figure 16, Figure 17)	- Clear logical separation of critical control systems and non-critical control systems - Direct communication between all critical control devices of CCS and TCMS domains possible. - Excellent hard-real-time behavior ensures deterministic communication. - Direct inter-consist communication for CCS possible	- No clear separation between CCS and TCMS domain. - Consist switches responsible for the logical segmentation become security relevant and have to meet certain requirements.
Scenario D: Common critical control network physically separated (Figure 18, Figure 19)	- Clear physical separation of critical control systems and non-critical control systems - Direct communication between all critical control devices of CCS and TCMS domains possible. - Excellent hard-real-time behavior ensures deterministic communication. - Direct inter-consist communication for CCS possible - Consist switches are not security relevant.	- Complex network configuration (e.g. for hard-real-time traffic between CCS/TCMS and Auxiliary systems) - No clear separation between CCS and TCMS domain. - Direct inter-consist communication for auxiliary systems only with additional train routers and train line possible.

Table 15: Overview of network architectures for new trains

In the cyber security standards IEC 62443 and TS 50701 [23] it is required that the definition of zones shall include measures for encapsulation of functionality to keep a particular service alive in case of an incident in another zone. If all the different assets in a train are divided into different security zones from a functional point of view, it is proposed that the critical control functions traction, brakes and doors are put together to the CCS domain in the same security zone or network respectively. Especially as soon as the automatic train operation (ATO) function is added to the train, the CCS domain and the traction, braking and door functions are closely related. In case of an incident in any of these critical control systems, this would lead to a train stop. So, a further splitting of the critical control systems in two security zones (CCS and critical TCMS systems) would not improve the reliability in case of an incident.

In short, having the CCS components integrated in NG-TCN as described in scenario C or D, critical control systems are strictly separated from non-critical control systems and therefore both solutions fulfil the zoning concept of the cyber security standards. Furthermore, the direct communication between the critical control systems (e.g. CCU, VCU) ensures excellent hard-real-time behavior for different applications. For these two main reasons the network architecture of scenario C or D should be favored for a long-term vision with a new train having implemented NG-TCN. If a physical segmentation of critical control and non-critical control systems is needed, must be further investigated in subsequent phases of the OCORA initiative. Nevertheless, the scenario A could be a possible interim solution for a mid-term architecture.

Generally, the European Commission as well as many industry companies, including others than those already involved in CONNECTA or Safe4RAIL projects, have the same long-term vision of a common (TSN-) Ethernet based network for TCMS and CCS functions. Industry consortia UNIFE and UNISIG published documents with the same long-term vision, see [30] and [31].

From an organizational point of view, the network architectures in scenarios C and D represent a turning point towards a more functional architecture. The two domains CCS and TCMS will align anyway with the implementation of the ATO function. Developing the subsequent version of the NG-TCN network architecture

is a chance to start this process of joining together.

From a technical point of view, the CCS and TCMS domains must elaborate the same understanding of the common network architecture. The future network architecture must be aligned with other programs like e.g. Shift2Rail with its projects CONNECTA or X2Rail or consortia like e.g. UNISIG or UNIFE. X2Rail-3 has recently investigated the cybersecurity of the “Drive-by-Data Architecture” of CONNECTA [15], which has to be inspected by OCORA. Afterwards an alignment between all involved parties is needed. At the end, the results shall be incorporated into the next version of IEC 61375 standard.

6.4 Network architecture for retrofit vehicles

Current TCN layouts differ between vehicle manufacturers. Especially the consist networks and technologies including the used network protocols are often proprietary implementations of the manufactures. The network architecture of retrofit vehicles will be vehicle dependent and therefore project specific.

The legacy and much standardized combination of WTB and MVB is still used for the TCMS. But the need for larger usable data bandwidth led to diverse network implementations where several TCMS busses or networks coexist. Today, within consists at least these network protocols are used:

- MVB & WTB (for TCMS, legacy)
- CAN (for local subsystems, e.g. Boogie Control)
- PROFIBUS (Siemens, legacy)
- Profinet (Ethernet, Siemens)
- CIP (Ethernet, Alstom)
- IPTCom (Ethernet, Bombardier)
- TRDP (Ethernet, Stadler, Bombardier, Toshiba, Siemens, CAF)

In the legacy train the TCMS normally will not change. So, the CCN must establish its own ECN network for CCS devices only. The CCN and the legacy TCN are therefore physically separated. The CCN is connected to the legacy TCN through the OCORA Gateway (GW). All communication components, all operator components and all security devices are located also on their own physically separated network. The network with connected public devices (e.g. public WLAN access point with connected passenger devices) is physically isolated. Every network represents a security zone.

In the following two figures an example of a physical and a logical network architecture for a retrofit scenario is shown. For simplicity only one TCMS bus is outlined.

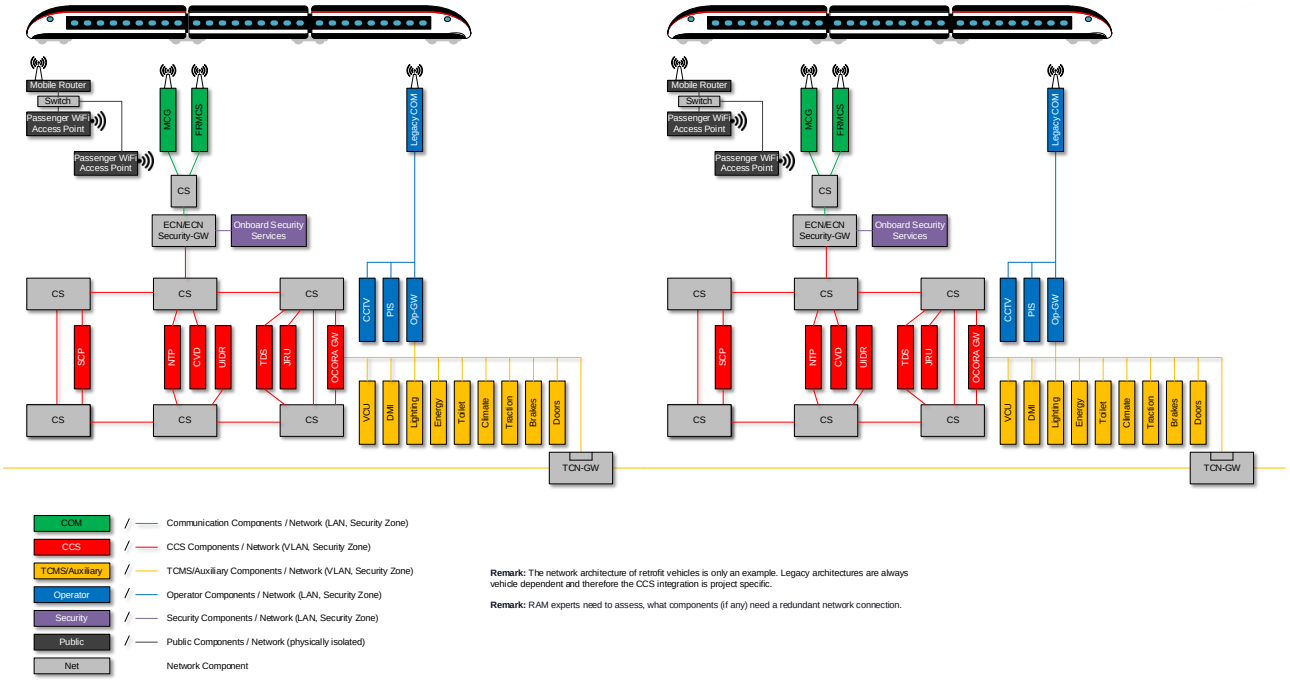

Figure 20: Physical network architecture scenario for retrofit vehicles

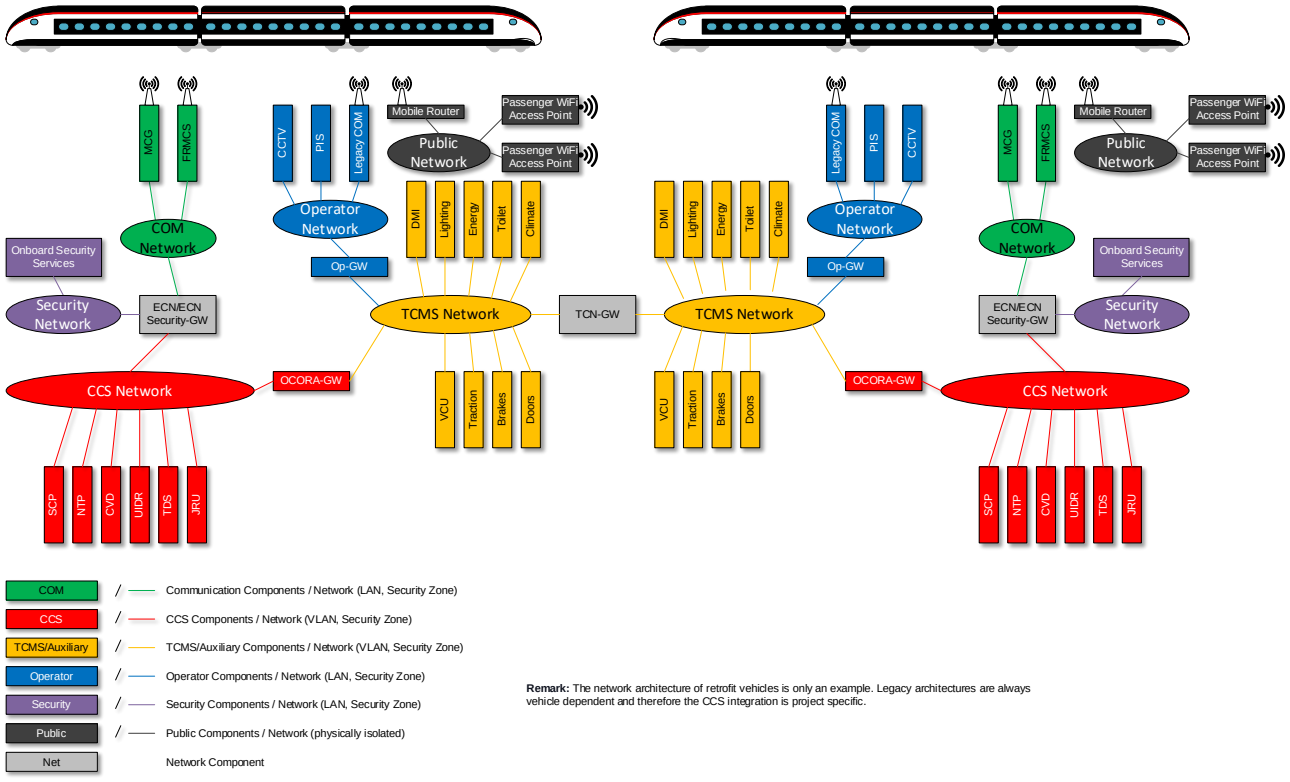

Figure 21: Logical network architecture scenario for retrofit vehicles