OCORA

Open CCS On-board Reference Architecture

UVCC Bus Evaluation
Gamma Release

Document ID: OCORA-40-003-Gamma

Version: 2.00

Date: 04.12.2020

Status: Final

Revision history

Version	Change Description	Name (Initials)	Date of change
1.10	First Draft for Gamma Release	SSt	2020-07-30
1.20	Changes due to first OCORA Gamma review	SSt	2020-11-05
1.30	Changes due to second OCORA Gamma review	SSt	2020-11-12
1.40	Changes due to third and OCORA core team review	SSt	2020-11-18
2.00	Changes based on final OCORA core team review	CG	2020-12-04

Table of contents

Table of figures

Table of tables

References

The following references are used in this document:

[1] OCORA-10-001-Gamma – Release Notes

[2] OCORA-20-003-Gamma – Technical Slide Deck

[3] OCORA-30-001-Gamma – Introduction to OCORA

[4] OCORA-30-002-Gamma – Problem Statements

[5] OCORA-30-005-Gamma – Alliances

[6] OCORA-40-001-Gamma – System Architecture

[7] OCORA-40-004-Gamma – Computing Platform – Whitepaper

[8] OCORA-40-008-Gamma – (Cyber-) Security Overview

[9] OCORA-40-012-Gamma – Modular Safety

[10] OCORA-90-002-Gamma – Glossary

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

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

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

[14] CENELEC prTS 50701 – Railway Applications - Cybersecurity

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

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

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

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

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

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

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

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

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

1 Abstract

Today the interfaces between CCS components on the vehicle are proprietary. The proprietary interfaces do not allow to exchange or update CCS components from different suppliers. The OCORA architecture [6] aims for plug and play interchangeability within the CCS domain through the specification of a generic and open communication backbone, the Universal Vital Control and Command Bus (UVCCB). The UVCCB 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 UVCCB evaluation report of beta release [23]. It updates the requirements for the UVCCB and provides further investigations on session layer protocols and integration of the UVCCB in NG-TCN or retrofit vehicles respectively.

The UVCCB evaluation report of beta release [23] proposes the UVCCB to be a TSN Ethernet based network with the use of SDTv2 / SDTv4 as safety layer. In order to be able to integrate the UVCCB on the next generation of train communication network (NG-TCN) every hard-real-time CCS device (e.g. CCU, BTM 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.

The evaluation on session layer protocols in this document come to the result that 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 proposed protocol stack of UVCCB is listed in the following table. Highly recommended standards to be used as reference for procurement in OCORA are listed in **bold** font.

<table>
<tr><th>Layer</th><th colspan="2">Protocol for hard-real-time data</th><th>Protocol for soft- or non-real-time data</th></tr>
<tr><td>(Safety Layer[1])</td><td colspan="3">(SDTv2 / SDTv4)</td></tr>
<tr><td>Session Layer</td><td colspan="3">TRDP 2.0, OPC-UA Pub/Sub or DDS/RTPS</td></tr>
<tr><td>Transport Layer</td><td rowspan="2"></td><td>UDP
TCP</td><td>UDP
TCP</td></tr>
<tr><td>Network Layer</td><td>IPv4</td><td>IPv4</td></tr>
<tr><td>Data Link Layer</td><td colspan="2">Time-Sensitive Networking (TSN)
IEEE 802.1</td><td>Standard Ethernet IEEE 802.3</td></tr>
<tr><td>Physical Layer</td><td colspan="3">100BASE-TX or 1000BASE-T</td></tr>
</table>

Table 1: Protocol Stack UVCCB

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 UVCCB.

The assessment of different network architectures shows that only by combining the UVCCB and the NG-TCN the hard-real-time behaviour is ensured for all data communication between components of the different domains. In order to still separate the two domains CCS and TCMS, the network shall be logically divided into different virtual networks (VLANs). Therefore, OCORA proposes to integrate the UVCCB in ECN as a separate virtual network.

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

2 Introduction

2.1 Document context and purpose

This document is published as part of the OCORA Gamma release, together with the documents listed in the release notes [1]. It is the second release of this document and it is still in a preliminary state.

The document is released in the OCORA Gamma release presenting the results obtained so far. This document focuses on the UVCCB, describing its requirements, evaluation steps, evaluation findings, architectural decisions and the proposed protocol stack. The content regarding the program can be found in documents [3] and [5]. Subsequent releases of this document and topic specific documentation will be developed in a modular and iterative approach, evolving within the progress of the OCORA collaboration.

2.2 Why should I read this document?

This document is addressed to experts in the CCS domain and to any other person, interested in the OCORA technical concepts for on-board CCS. The reader will gain insights regarding the topics listed in chapter 2.1, will be able to provide feedback to the authors and can, therefore, engage in shaping OCORA.

Before reading this document, it is recommended to read the Release Notes [1], the Introduction to OCORA [3], and the Problem Statements [4]. The reader should also be aware of the Glossary [10].

The System Architecture [6], the Computing Platform [7], the (Cyber-) Security [8] and the Modular Safety [9] provide further technical details to this document.

2.3 Problem Description

Today the interfaces between CCS components on the vehicle are proprietary. The proprietary interfaces do not allow to exchange or update CCS components from different suppliers. The vendor locking created by proprietary interfaces leads to high costs. The existing proprietary interfaces do not allow to add 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.

2.4 Concept

The OCORA architecture [6] 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 Universal Vital Control and Command Bus (UVCCB).

In the following figure the technical view of the OCORA architecture [6] is shown. The UVCCB connects all components of the future CCS system. The most important CCS components are:

- CCS computing units (CCUs)
- Driver Machine Interface (DMI)
- Vehicle Location System (VLS)
- Balise Transmission Module (BTM)
- Loop Transmission Module (LTM)
- National Train Control System (NTC)
- Cab Voice Device (CVD)

- Train Recording Unit (TRU)
- Input / Output Ports (I/O Ports)
- Mobile Communication Gateway (MCG)
- Gateway to Train Control Bus / Network and Passenger Information System Bus / Network

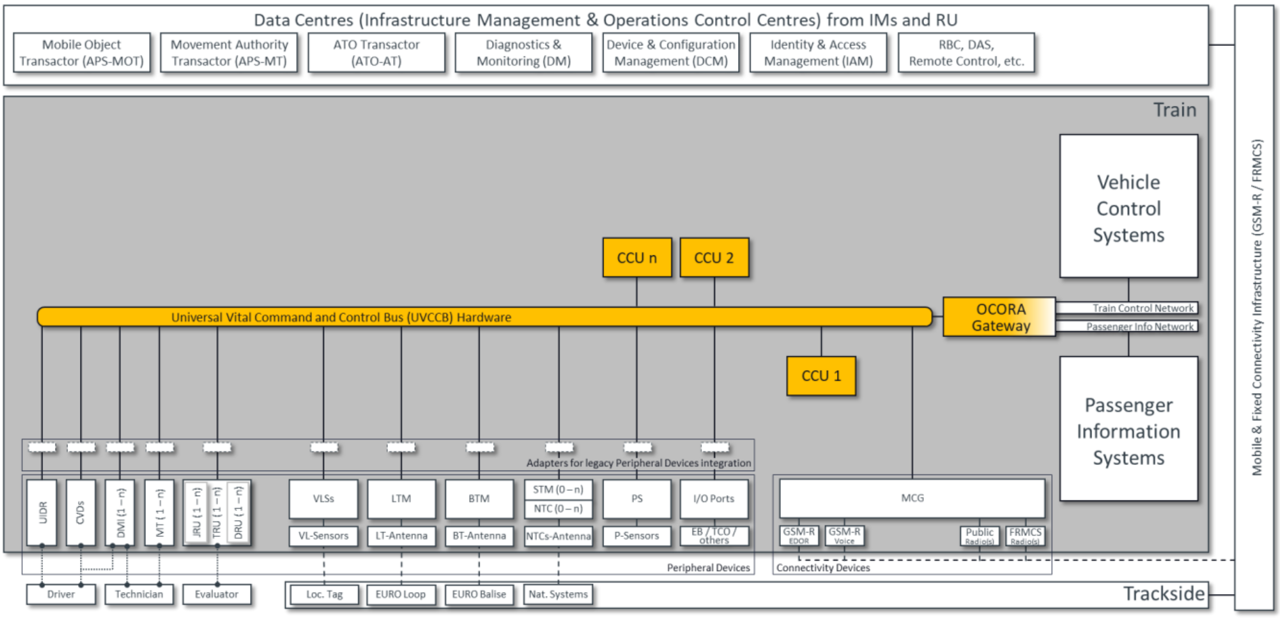


Figure 1: Technical architecture from [2]

In the final vision of the system an open standardized UVCCB (OSI-Layers 1 to 7 & Safety Layer) ensures the safe data connection between all CCS components. The bus topology 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 UVCCB itself will be modifiable in accordance with future technological evolutions by means of strict separation of the different communication layers (OSI Layers).

The UVCCB evaluation report of beta release [23] proposes the UVCCB to be a TSN Ethernet based network with the use of SDTv2 / SDTv4 as safety layer. In order to be able to integrate the UVCCB on the next generation of train communication network (NG-TCN) every hard-real-time CCS device (e.g. CCU, BTM 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 to 6 at least IPv4, UDP, TCP and TRDP 2.0 shall be supported in order to be able to communicate directly to the TCMS. For communication between CCS devices other protocols on session layer are still to be evaluated (e.g. OPC-UA Pub/Sub or DDS-RTPS).

The protocol stack of UVCCB proposed in the OCORA Beta release [23] is listed in the following table. Highly recommended standards to be used as reference for procurement in OCORA are listed in **bold** font. Protocols to be further investigated are listed in grey font.

Layer	Protocol for hard-real-time data	
Safety Layer	**SDTv2 / SDTv4**	
Session Layer	TRDP 2.0	e.g. OPC-UA Pub/Sub or DDS-RTPS
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 2: Protocol Stack UVCCB

2.5 Goal

The goal of this document is 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 UVCCB within the NG-TCN and the separation of responsibilities between the two domains shall be clearly elaborated. Also, the solution for retrofit vehicles, where there will be a UVCCB without a NG-TCN, must be developed.

The following tasks have been performed:

1. Modification of existing UVCCB requirements regarding session (and presentation) layer function.
2. Collection of the existing technologies on session (and presentation) layer.
3. Evaluation of the technologies with respect to the requirements and compatibility with lower OSI Layers.
4. Decision of the preferred technology/technologies from OCORA perspective.
5. Definition of OCORA network architecture and configuration
    a. New vehicles: Integration of UVCCB in NG-TCN
    b. Retrofit vehicles with new OCORA CCS system and existing TCMS without NG-TCN

3 Requirements

The list of requirements of OCORA Beta release [23] is updated with new requirements. The list of requirements collects all requirements to be used for the technology evaluation. The requirements shall not be copied directly for a call for tender.

3.1 Functional requirements for UVCCB

This chapter elaborates the functional requirements for the UVCCB

UVCCB-01: Data transfer

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

UVCCB-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.

UVCCB-03: Safety

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

UVCCB-04: Safety

- Requirement: The safety targets for the different SIL of the safe data transmission are as follows (based on EN 50129 [11]):

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.

UVCCB-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:

UVCCB-06: Data type

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

UVCCB-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 line with IEC 61375-3-4:2013 [13]

UVCCB-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 [13]

UVCCB-09: Bandwidth

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

UVCCB-10: Number of participating nodes

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

UVCCB-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:

UVCCB-12: Topology flexibility

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

UVCCB-13: Time synchronization

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

UVCCB-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:

UVCCB-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:

3.2 Non-functional requirements for UVCCB

This chapter elaborates the non-functional requirements for the UVCCB

UVCCB-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:

UVCCB-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.

UVCCB-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.

UVCCB-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:

UVCCB-20: Portability

- Requirement: The same UVCCB 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:

4 Evaluation of session layer protocols

For the UVCCB the OCORA Beta release [23] proposes to use the same technology for UVCCB 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 [21].

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.

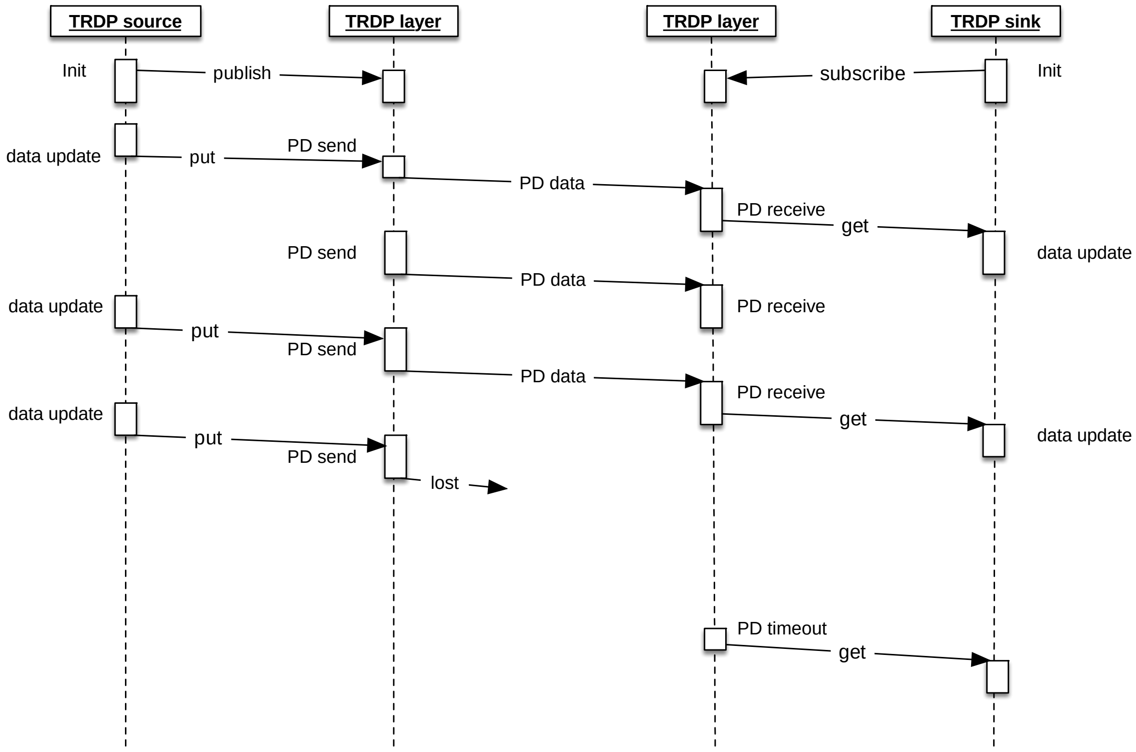


Figure 2: Publish & Subscribe

In Figure 2, '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.

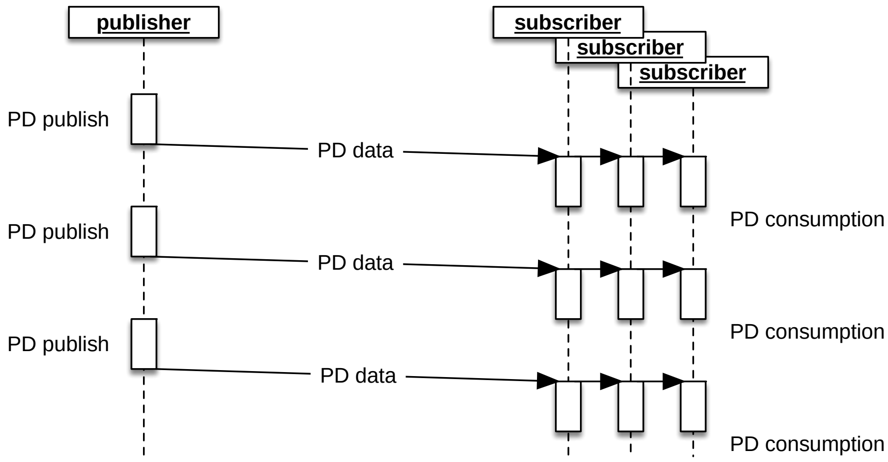


Figure 3: PD Multicast

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:

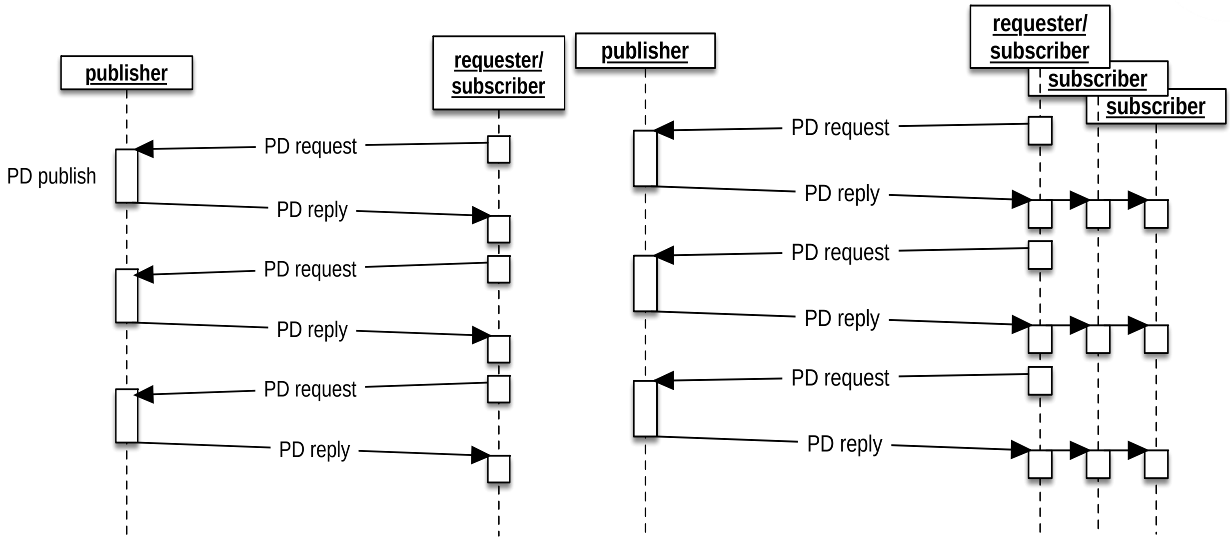


Figure 4: PD Pull: one requester

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.

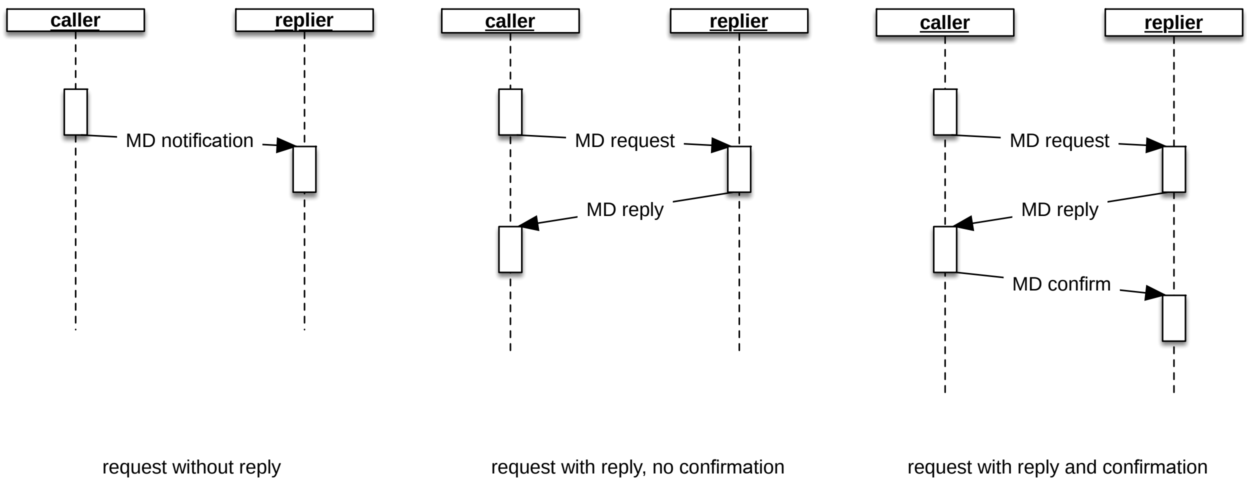


Figure 5: Message Data pattern, unicast

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.

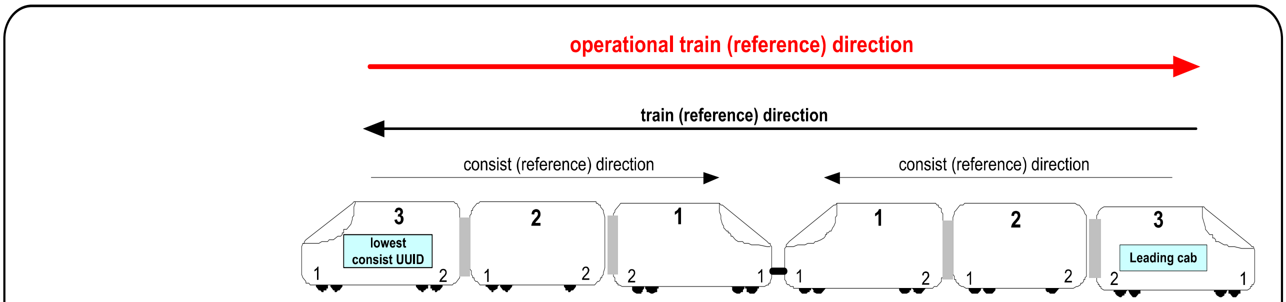
IEC61375-2-3 defines a train-wide central function and device repository, called TTDB (Train Topology Data Base) and an addressing scheme using Unified Resource Locator (URL) and Unique Resource Identifier (URI). A central instance of a TCN-DNS (name server) translates URIs to IPv4 addresses, which are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

Figure 6: 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.



train consist number (trnCstno):	01			02		
train consist orientation (cstOrient):	inverse			direct		
operat. train consist number (opCstNo):	02			01		
operat. train consist orientation (opCstOrient):	direct			inverse		
consist vehicle number (cstVehNo):	03	02	01	01	02	03
consist vehicle orientation (vehOrient):	inverse	inverse	direct	direct	direct	inverse
train vehicle number (trnVehNo):	01	02	03	04	05	06
operat. train veh. number (opTrnVehNo):	06	05	04	03	02	01
consist UUID (cstUUID):	3026cede-c9e8-11e3-9d46-1a514932ac01			6a4a96fe-c9e8-11e3-9d46-1a514932ac01		
consist identifier (cstId):	50 80 89 – 43 503 - 8			50 80 89 – 43 603 - 6		
veh. identifier (vehId):	50 80 89 – 43 501 - 2	50 80 89 – 43 502 - 0	50 80 89 – 43 503 - 8	50 80 89 – 43 603 - 6	50 80 89 – 43 602 - 8	50 80 89 – 43 601 - 0

Figure 7: 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 behaviour 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 UVCCB as already noted in chapter 2.4.

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 [21].

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 Websockets, 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 UVCCB. 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 UVCCB 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 [26].

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 [26].

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 UVCCB. Due to its use on different platforms and use in different industry sectors, it could be beneficial to use it on UVCCB.

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.

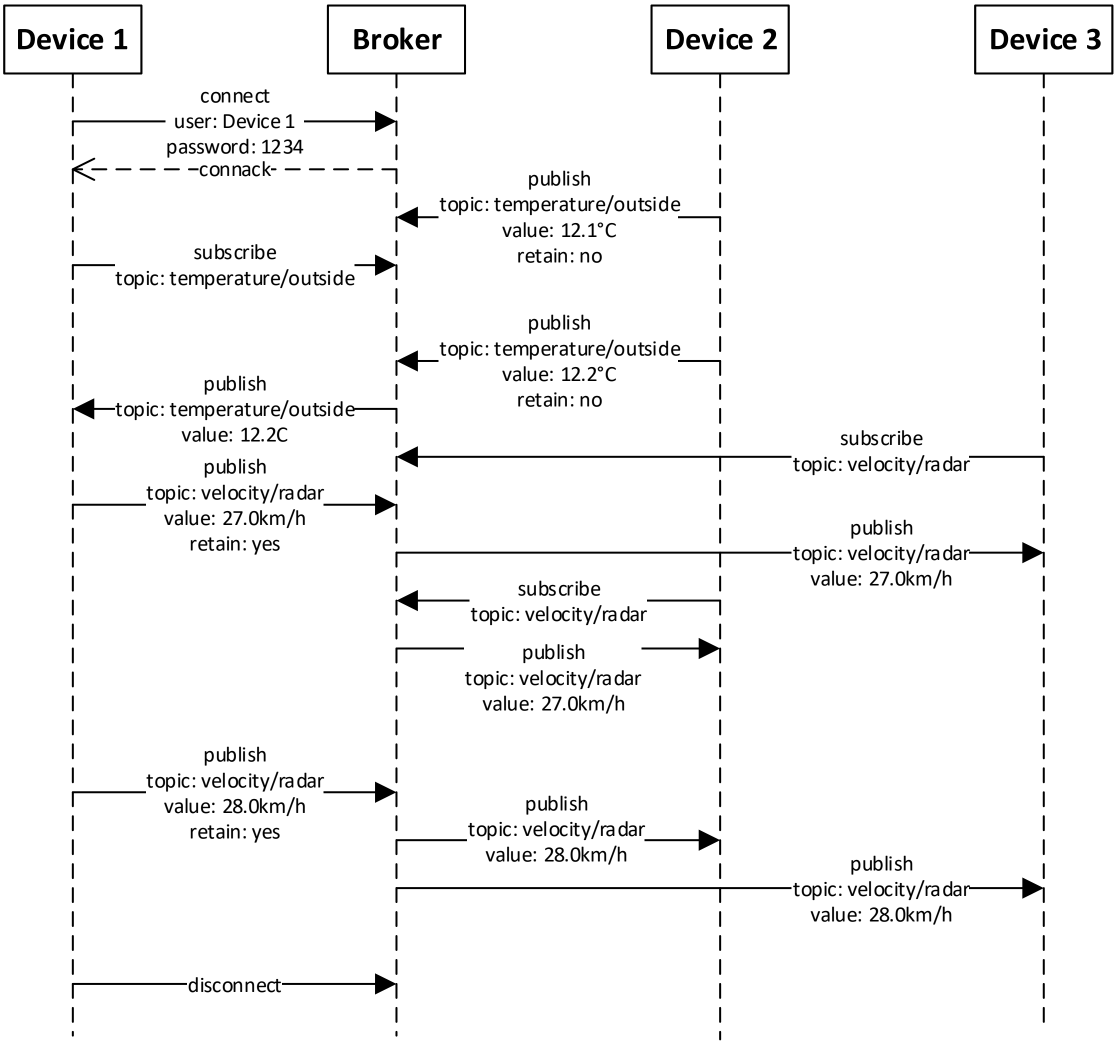


Figure 8: Sequence diagram with data exchange over MQTT

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.

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 UVCCB. Moreover, MQTT cannot benefit from the TSN features which leads to non-deterministic communication behaviour. Therefore, MQTT is not suitable for the use on UVCCB.

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 behaviour. Therefore, AMQP is not suitable for the use on UVCCB.

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 UVCCB.

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 Integration of UVCCB in train

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 showed in Figure 9.

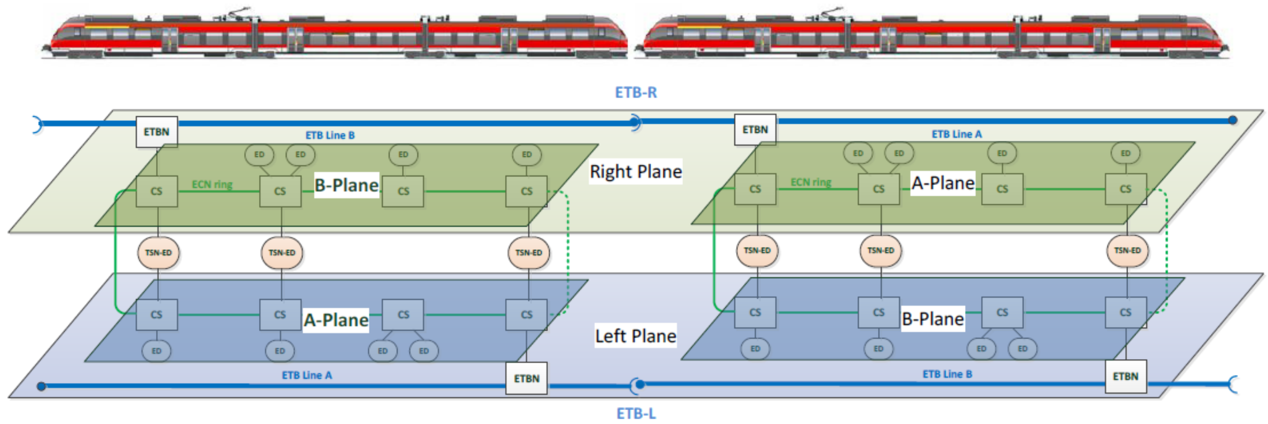


Figure 9: Network architecture of NG-TCN from [15]

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 eases also 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 a TSN-substandard IEEE802.1CB. It is shown the following Figure 10.

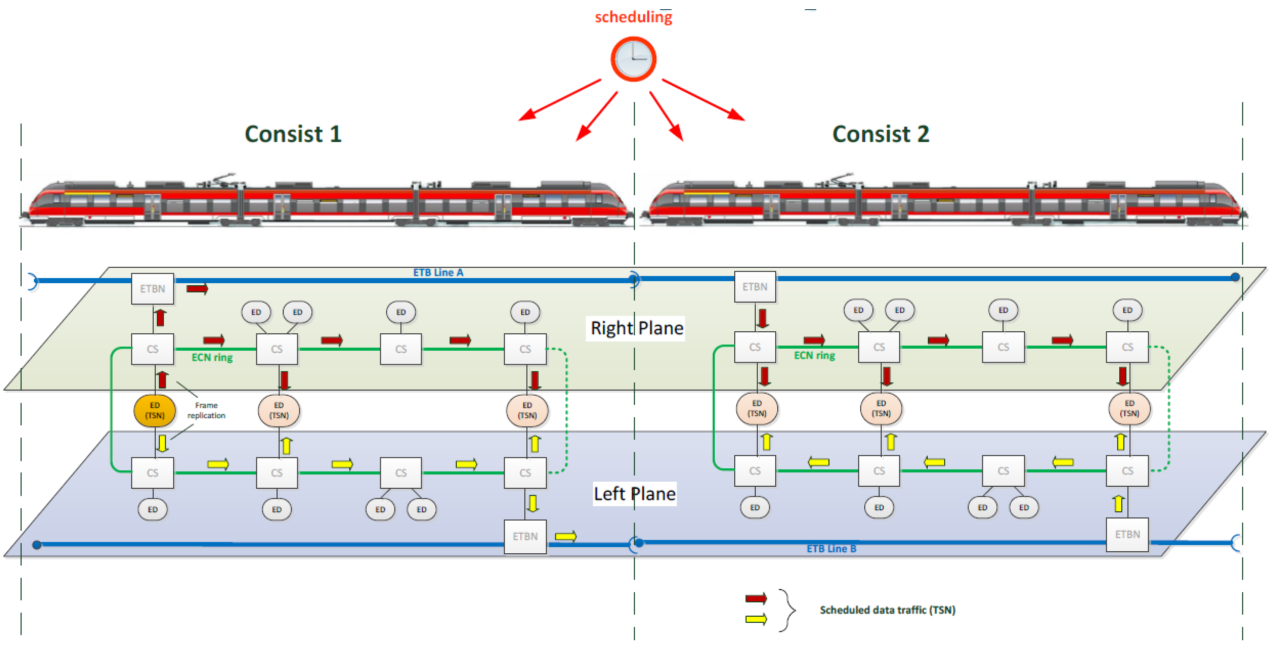


Figure 10: Data flow for TSN traffic on NG-TCN [17]

As the UVCCB is based on the same network technology as it will be used for the NG-TCN in TCMS domain, the technical integration of the UVCCB in the train must be investigated. Also, the separation of responsibilities between the two domains must be clearly elaborated. Additionally, the solution for retrofit vehicles where there will be a UVCCB without a NG-TCN of TCMS has to be defined.

5.1 Network architecture for new trains with NG-TCN

5.1.1 UVCCB as separated network

With the UVCCB as a 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. If the two network configurations of the two domains will be done independently the hard-real-time behaviour for data between the two domains will suffer.

In the following subchapters, the two options of UVVCB as a separated network are sketched.

5.1.1.1 UVCCB connected to ECN

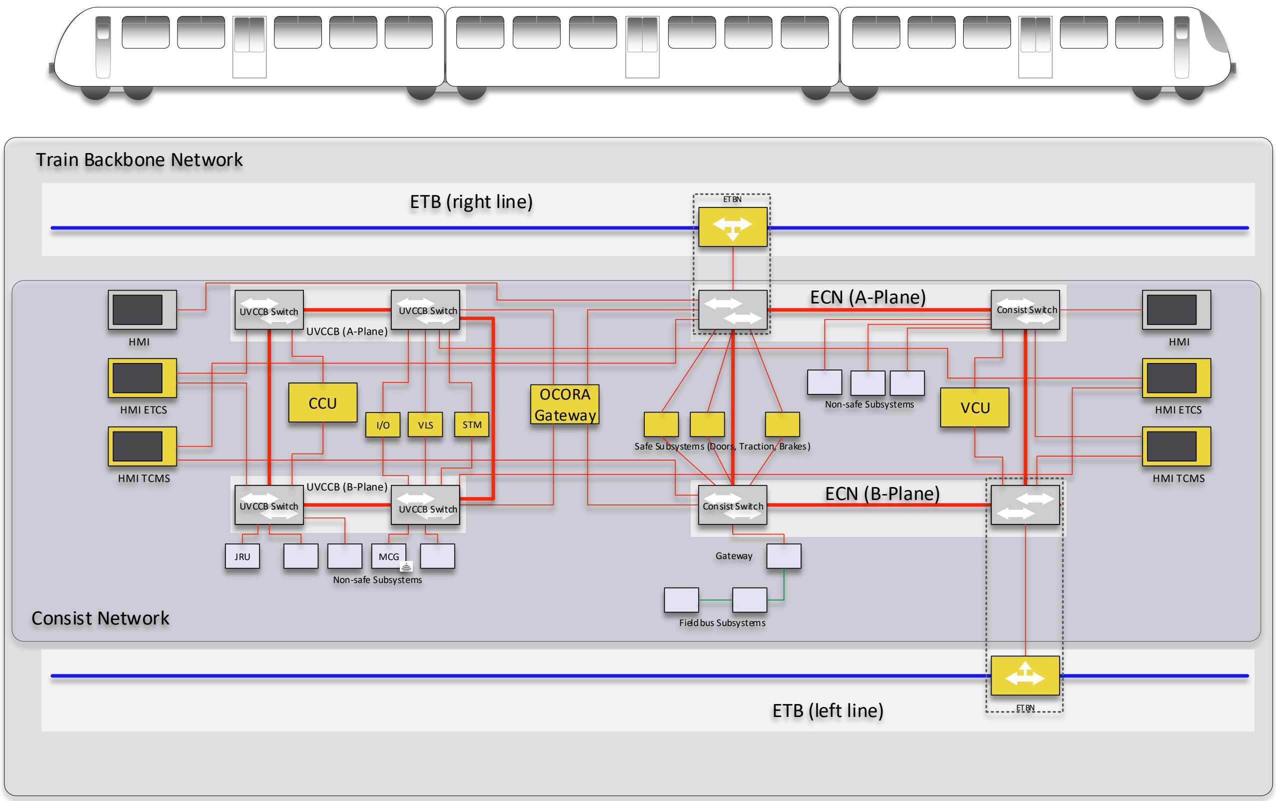


Figure 11: Physical network architecture with UVCCB connected to ECN

Figure 11 shows the situation with an UVCCB as an own separated network connected to the Ethernet consist network (ECN) part of NG-TCN. This solution needs an OCORA hardware gateway between UVCCB and ECN. The solution does not allow a direct communication from a CCU to a CCU of another consist. For such a communication the data flow must be forwarded over ECN of the home consist, Ethernet train backbone (ETB), ECN of the other consist and finally to the UVCCB of the other consist.

5.1.1.2 UVCCB connected to ETB

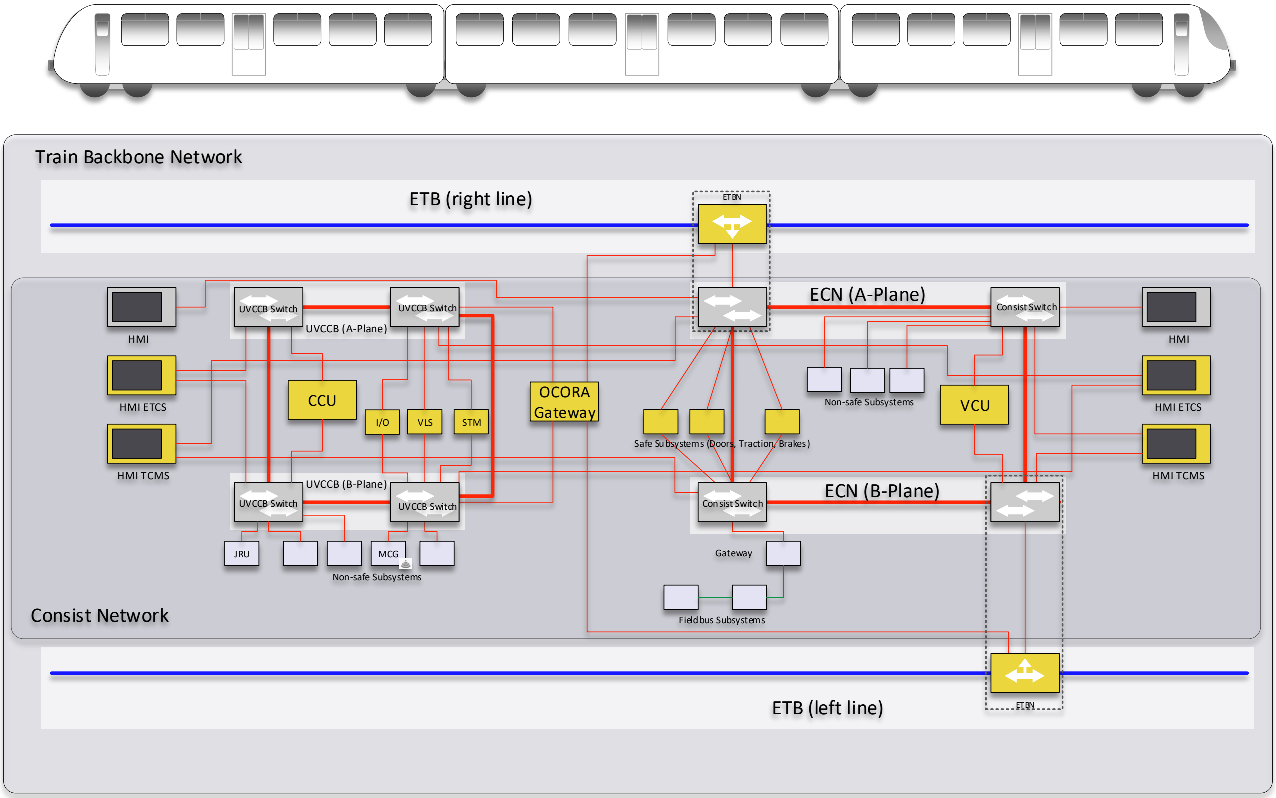


Figure 12: Physical network architecture with UVCCB connected to ETB

Figure 12 shows the situation with an UVCCB as an own separated network connected to the ETB part of NG-TCN. This solution needs an OCORA hardware gateway between UVCCB and ETB and the Ethernet train backbone node (ETBN) needs a dedicated end device (ED) port on ETB. The solution does not allow a direct communication from the CCS to the TCMS domain of the own consist. This data flow must be forwarded over ETBN. Due to IEC 61375 [13], only TRDP on session layer is allowed on ETB. This is even valid for inter UVCCB communication between different consists, if there is no dedicated CCS/UVCCB VLAN on ETB.

5.1.2 UVCCB integrated in NG-TCN

With the UVCCB integrated in NG-TCN, the network configuration (TSN schedule) of CCS and TCMS domain will be done once and the hard-real-time behaviour is ensured for all data communication.

In the following subchapters, the options of UVVCB integrated in NG-TCN are sketched.

5.1.2.1 UVCCB integrated in ECN

Figure 13 shows the physical situation with an UVCCB integrated in ECN part of NG-TCN as one common physical network. A logical separation between the domains CCS and TCMS (and others like operator services) can be done by different virtual local area networks (VLAN). The following subchapters illustrate the advantages and disadvantages of the logical separation.

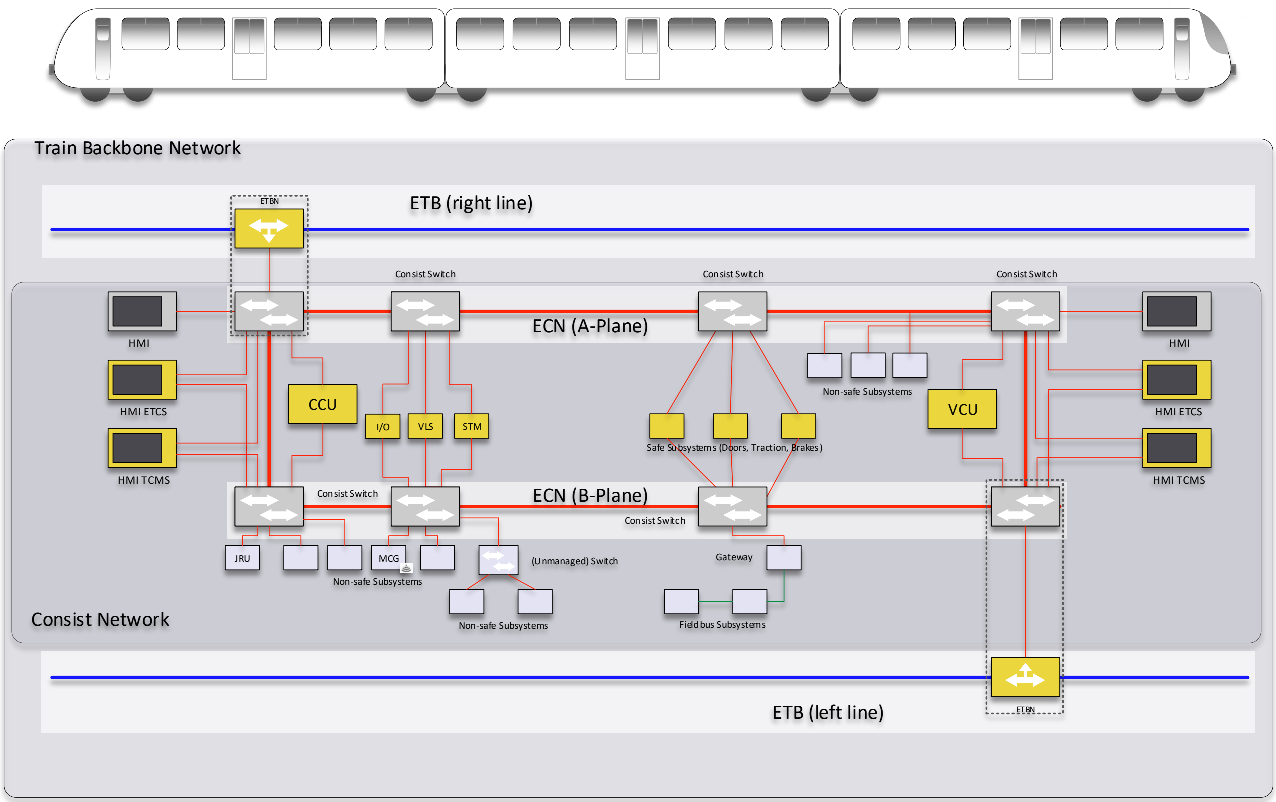


Figure 13: Physical network architecture with UVCCB integrated in ECN

5.1.2.1.1 With Separation of CCS and TCMS Domain

CONNECTA defines the VLANs for the different domains in NG-TCN. The draft of this VLAN definition is shown in Table 3.

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

For the UVCCB (CCS domain) it is possible to add new VLAN-IDs. At least for best effort Ethernet traffic two new VLANs called ECN-CCS and ETB-CCS must be reserved. For TSN-traffic a few of the already reserved VLAN-IDs 32…287 (ECN-TSN-A-X and ECN-TSN-B-X) or 288…543 (ETB-TSN-A-X and ETB-TSN-B-X) or even new additional IDs can be used.

This logical separation implies the need of a gateway between the VLANs of the CCS and TCMS domain. This can be realized as a hardware gateway or a software function on one of the computing units (e.g. CCU or VCU) for instance. The solution does not allow direct communication among all end devices of any domain (e.g. VCU to ETCS DMI).

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 logically separated by VLANs. Industry consortia UNIFE and UNISIG published documents with the same long-term vision, see [24] and [25].

5.1.2.1.2 Without Separation of CCS and TCMS Domain

Without any physical or logical separation, a direct communication among all end devices in the train becomes possible. With this solution no gateway at all is needed since CCS data traffic is located on the same VLANs as TCMS data traffic. However, several communication parameters (e.g. ComId of TRDP and SMI of SDT) must be coordinated between CCS and TCMS domains in order to prevent fatal communication errors. Additionally, a risk assessment (corresponding to prTS 50701 [14]) has to show that a single zone for TCMS and CCS domain is sufficient from a cybersecurity perspective.

5.1.2.2 UVCCB integrated in ETB

In principle the standard IEC 61375 allows end devices to be directly connected to the ETB. Therefore, all CCS components directly connected to ETB would be an option. According to [16] the addressing only allows 64 EDs or 32 ED-Ss respectively on ETB. This means in a situation of for example four consists, only 16 EDs or 8 ED-Ss per CCS system on one consist would be possible, which is definitely insufficient, see requirement UVCCB-10:. UVCCB integrated in ETB is therefore not an option.

5.2 Network architecture for retrofit vehicles

For retrofit vehicles there will be no NG-TCN network. The UVCCB must establish its own ECN network for CCS devices only It includes an OCORA Gateway to the TCMS domain.

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 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. 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)

Figure 14 shows an example of a retrofitted train with different TCMS busses/networks.

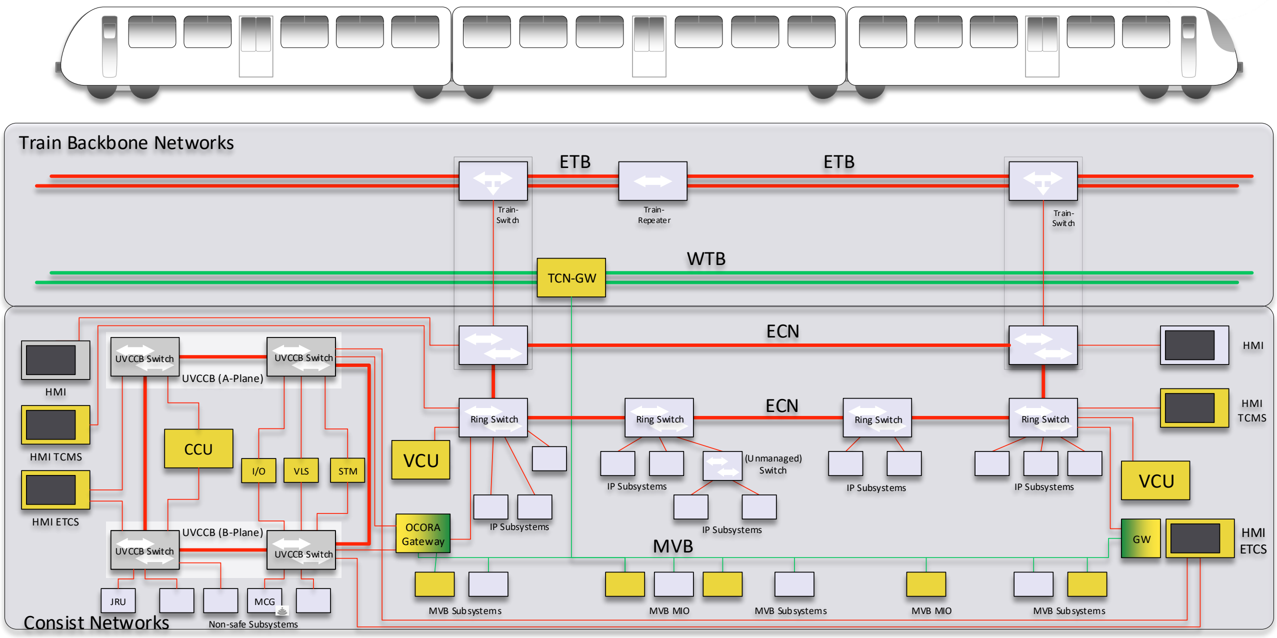


Figure 14: UVCCB for Retrofit Vehicles

5.3 Conclusion

Table 4 gives an overview of the advantages and disadvantages of the different evaluated network architectures for new trains.

Network Architecture for new Trains	Advantage	Disadvantage
UVCCB as separated network connected to ECN (5.1.1.1)	• Clear separation between CCS and TCMS domain.	• Complex network architecture with limited modifiability • Complex network configuration for hard-real-time traffic between CCS and TCMS domain.
UVCCB as separated network connected to ETB (5.1.1.2)	• Clear separation between CCS and TCMS domain	• Complex network architecture with limited modifiability • Complex network configuration for hard-real-time traffic between CCS and TCMS domain.
UVCCB integrated in ECN with VLAN separation (5.1.2.1.1)	• Simple network architecture with simple option of later modifiability • Hard-real-time behaviour is ensured for all data communication. • Clear separation between CCS and TCMS domain • Only software gateway needed	• No direct communication between all devices of any domain possible
UVCCB integrated in ECN without VLAN separation (5.1.2.1.2)	• Simple network architecture with simple option of later modifiability	• No clear separation between CCS and TCMS domain

	• Hard-real-time behaviour is ensured for all data communication. • Direct communication between all devices of any domain possible • No CCS-TCMS gateway needed	• Limitations from cyber security perspective
UVCCB integrated in ETB (5.1.2.2)		• Not an option due to limited number of devices

Table 4: Overview of network architectures for new trains

Having the UVCCB integrated in NG-TCN, the hard-real-time behaviour is ensured for all data communication between components of the different domains. In order to still separate the two domains CCS and TCMS, the network shall be logically divided into different virtual networks (VLANs). Therefore, OCORA proposes to integrate the UVCCB in ECN as a separate virtual network (see chapter 5.1.2.1.1).

In order to be able to integrate the UVCCB in ECN of the NG-TCN, every hard-real-time CCS device (e.g. CCU, BTM 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 can use both planes of NG-TCN with two TSN-capable Ethernet ports in order to improve reliability and availability.

Furthermore, the CCS and TCMS domains must elaborate the same understanding of the common network with the logical separation. The outcome shall be incorporated into the next version of IEC 61375 standard. This covers for example the topics of definition of VLAN IDs for CCS domain and bandwidth allocation.