OCORA

Open CCS On-board Reference Architecture

Proof of Concept Train Display System

Phase 1 results

Document ID: OCORA-TWS01-221

Version: 1.00

Date: 31.01.2025

Revision history

Version	Change Description	Initial	Date of change
1.00	▪ Official version for OCORA Release R6	NB	31.01.2025

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-BWS01-040 - Feedback Form

[2] OCORA-TWS01-222 - Train-Display-System_PoC_Use_Case_1

[3] OCORA-TWS01-223 - Train-Display-System_PoC_Use_Case_2

[4] OCORA-TWS01-224 - Train-Display-System_PoC_Use_Case_3

[5] OCORA-TWS01-201 - Train-Display-System-Concept

[6] OCORA-TWS01-202 - Train-Display-System-Specification

1 Introduction

1.1 Purpose and applicability of the document

The purpose of this document is to describe the TDS proof of concept (PoC) and its results, which can become inputs for the Train Display System specification activity.

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 [1].

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

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

1.2 Applicability of the document

The document is informative. 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. If you are interested in the context and the motivation that drives OCORA we recommend reading the Introduction to OCORA, the Guiding Principles, and the Problem Statements. The reader should also be aware of the Glossary and the Question and Answers.

OCORA has initiated work to standardize the TDS. It is performed as part of WP04 in the TWS01 Architecture and has started in the beginning of 2022. The first results are contained in the document OCORA TDS Concept Paper as part of the OCORA Release 4 and Release 5. This paper describes the first concepts of the TDS:

- Function allocation between TDS and systems (EVC, TCMS, CVR…).
- Preliminary architecture proposal submitted to the industry.
- Logical concepts for sharing the displays between different systems (TCMS, ETCS, Voice Communication…).
- Definition of major software and graphical components.
- Management of degraded modes.

OCORA aims at having these concepts being integrated as part of a TSI or as part of a Standard.

The industry is also interested in this standardization so they can broaden the number of suppliers for these display components. In fact, TDS should ensure a clear separation between systems and displays. This will allow the use of displays from different suppliers.

ERJU and ERA meanwhile require that new elements shall be mature enough so that they can be included in a Standard or TSI.

The supply of documents such as FFFIS, SRS and demonstrators are key elements to demonstrate the TDS concept maturity.

The demonstrators can be implemented as part of the Innovation Pillar or outside of this program.

2 Objective of PoC TDS

The purpose of the PoC will be to demonstrate the TDS concepts [5] developed for OCORA Release 5.

The first requirements to be demonstrated, as expressed in the TDS On-Board Concept Paper, are “Modularity” and “Active Configuration”. These requirements are listed below as a reminder.

OCORA-10017, D-Level – Modularity
The TDS shall enable systems to link all their input/output to any touch/display panel based on a pre-approved configuration.

Status	In Review
Classification	Requirement
Rationale	• Enhance flexibility for configuration • Rationalise the number of display panels in the cab for space and costs saving
Category	Functional
Acceptance Method	Design Review
Acceptance Criteria	-
Remark	Limitation to physical characteristics The modularity concept should be weighted depending on the pre-approved configuration ([OCORA-10015]) which takes under consideration the location of the display panel in the cab. Indeed, some systems could be only displayed in some location (e.g.: ETCS in front of the driver).
Linked Work Items	has parent: OCORA-10289 - General requirements , defines: OCORA-7587 - Train Display System (TDS) , is refined by: OCORA-10280 - Control the flow and layout elements , is refined by: OCORA-10282 - Generate View

OCORA-10015, D-Level - Active configuration
The TDS shall set the active configuration automatically (by events) or manually (by the driver) among a list of preapproved configurations.

Status	In Review
Classification	Requirement
Rationale	Only one configuration is active at a time.
Category	Functional
Acceptance Method	Certification
Acceptance Criteria	-
Remark	TDS should consider a nominal configuration by default and other configurations when required: - Nominal configuration (with all systems and HMI elements operational) - Event based configuration (e.g.: driver selection, train in motion/at standstill...) - Degraded configurations (in case of system or HMI element failure)

	The configuration shall take under consideration the mandatory systems for operation (e.g.: ETCS in a front location) and the optional systems. For each system, an evaluation of the possible location on the desk shall be performed with the associated events. The driver can select only pre-approved configuration. In case of failure, the TDS shall switch the active configuration in a fully transparent way for systems. This reconfiguration shall not require a reboot of TDS, systems, or HMI elements.
Linked Work Items	Has parent: OCORA-10289 - General requirements , defines: OCORA-7587 - Train Display System (TDS) , is refined by: OCORA-10285 - Manage configuration , is refined by: OCORA-10286 - Failover configuration

This PoC development is based on the TDS logical architecture recommended by OCORA in R5.1

3 PoC Implementation

3.1 Implementation means

3.1.1 Software

The TDS software was implemented using MATLAB, a programming and numeric computing platform developed by MathWorks, and its capabilities, especially:

- MATLAB Simulink, a tool which provides a graphical editor, customizable block libraries, and solvers for modelling and simulating dynamic systems. This tool was used to emulate:
  - The ETCS and TCMS software
  - The Display manager software
- MATLAB AppDesigner, a tool for interactively creating custom UI (User Interface) components. It was used in this PoC to emulate:
  - The Layout engine of each TDS Unit.
- MATLAB Stateflow, a control logic tool used to model reactive systems via state machines and flow charts within a Simulink model.
- Simulink Real-Time, a tool to perform rapid control prototyping and hardware-in-the-loop testing.

3.1.2 Hardware configuration

- SpeedGoat Realtime targets
  - Performance Real-Time Target Machine (1 unit)
  - Unit Real-Time Target Machine (2 units)
- Development PC: DELL Precision
  - With MATLAB R2022B installed, including the following toolboxes
    - Simulink
    - MATLAB Compiler
    - Simulink coder
    - Simulink Real-time
    - Stateflow
    - SpeedGoat I/O Blockset

- Lenovo Thinkstation P500 (2 units)
- Generic Samsung office Monitor (2 units)
- Ethernet switch TP-LINK TL-SG108 (1 unit)

3.2 PoC Architecture

Figure 1 : Bench setup

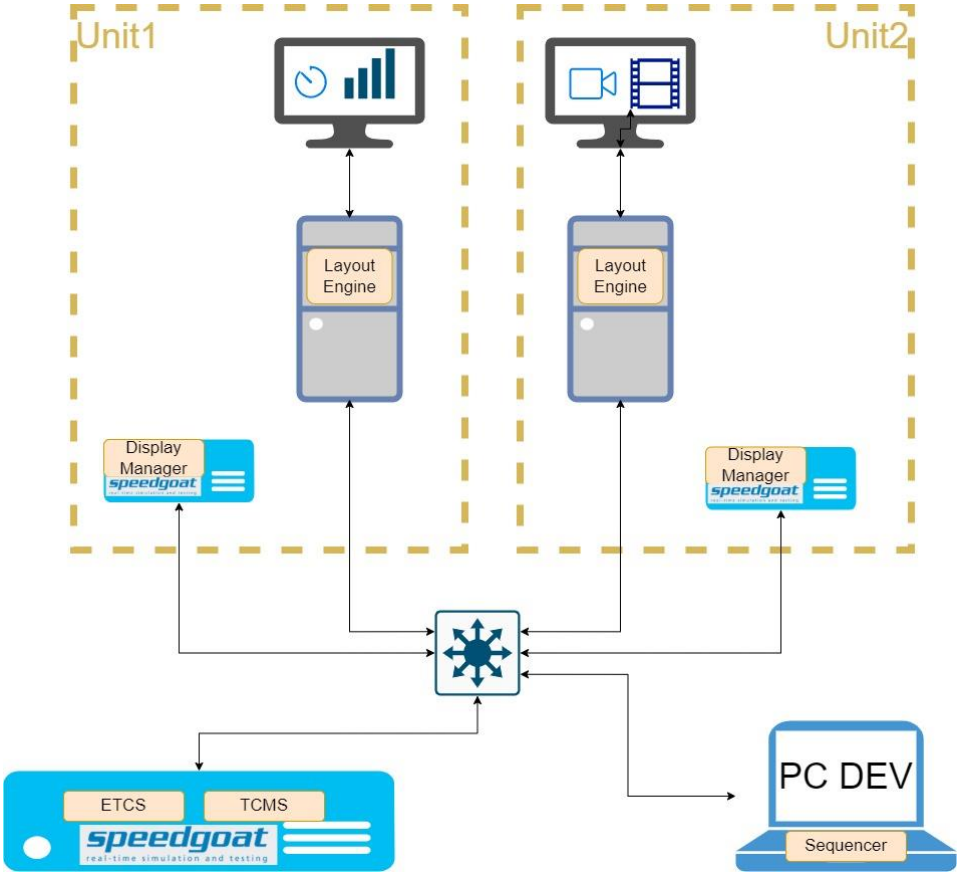


Figure 2 : PoC Hardware and functional architecture

3.3 Software

3.3.1 Display manager

Simulink model “Unit1Model.slx” generated on development PC, converted to “Unit1Model.mldatx” and loaded on SpeedGoat Unit 1 (a similar model is loaded on SpeedGoat Unit 2).

This software has two main responsibilities:

- Monitoring the other TDS Unit’s health and choosing an adequate TDS configuration among the list of preapproved configurations
    - This configuration selection function is only performed by the Display Manager of the **master** TDS unit
- Forwarding the system data sent by the performance model to its layout engine.

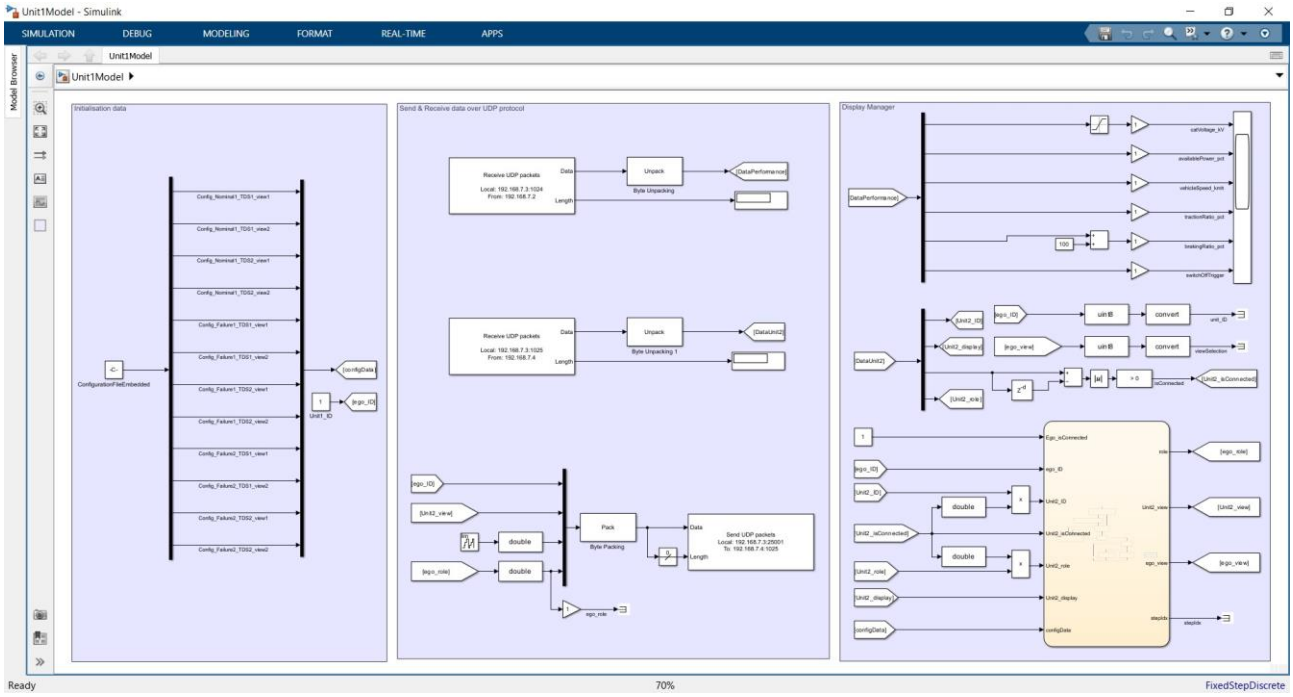


Figure 3 : Unit1Model (Display Manager)

The Display Manager's state machine is implemented as follows, and can be broken down to three phases:

1. INITIALISATION: The TDS Unit is waiting for other display units in the system to connect. After a certain time has elapsed or when all known display units are connected it moves on to the next phase: "role attribution".
2. ROLE ATTRIBUTION: The TDS Unit determines whether its role should be "slave" ("follower") or "master" ("leader") based on the following information:
    a. Is there already a display unit acting as Master? If so, the TDS Unit takes the role of slave.
    b. If not, is there a connected display unit whose ID is higher than its own? If so, the TDS unit takes the role of slave.
    c. Else the TDS Unit takes the role of master and moves on to the next phase: "configuration selection".
3. CONFIGURATION SELECTION: If the TDS Unit is attributed the role of master, it needs to select a configuration among the preapproved configurations. To each "TDS health state" should correspond an adequate TDS configuration. The Master TDS Unit is responsible for selecting this adequate configuration and ensuring it is applied by the slave TDS Units. This configuration remains valid until power off, or until the "TDS health state" is modified, e.g. upon failure of a slave TDS unit.

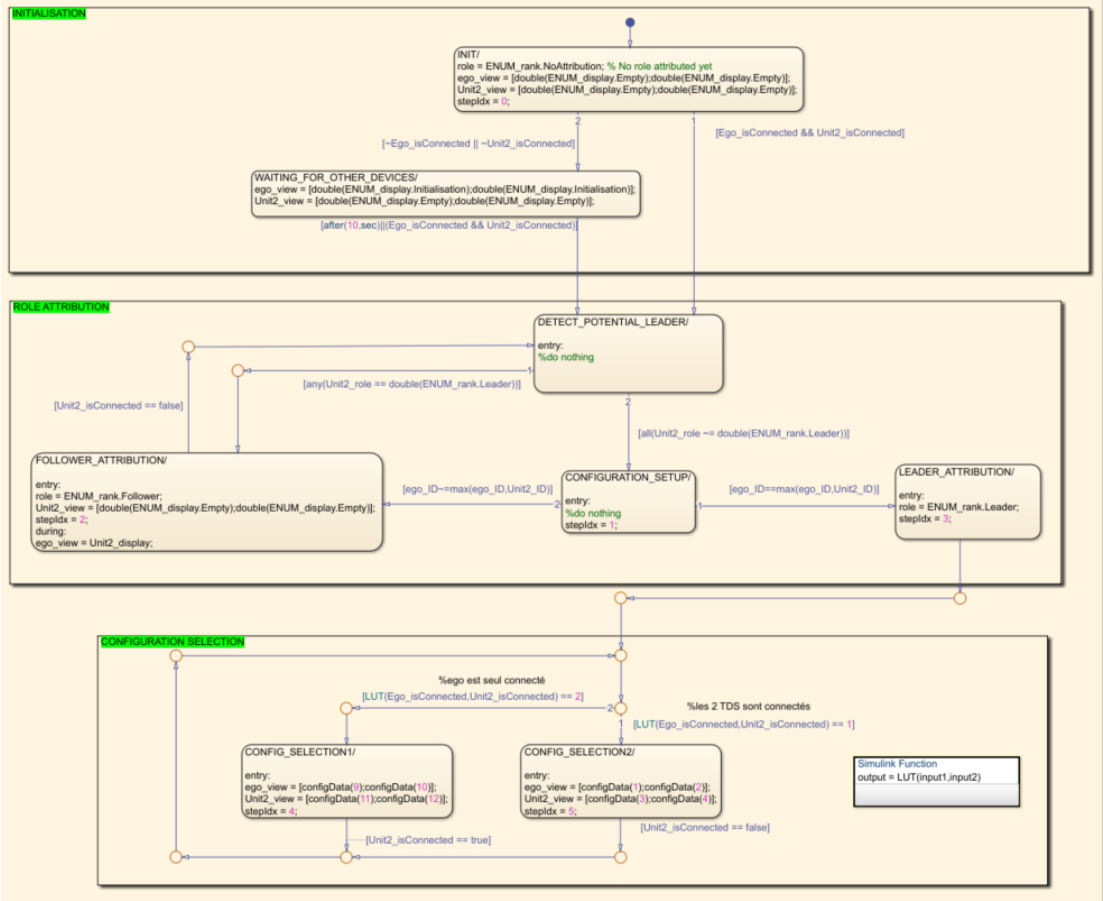


Figure 4 : Display Manager state machine developed with Simulink Stateflow

The following tables show the data exchanged by the Unit 1 Display Manager with other software in the system. Data highlighted in yellow is “proof of concept specific”, in other words this exchange of data is necessary because of our PoC architecture but is not relevant in the TDS concept.

Data	Emitter	Description	Transmission means
ETCS Data	SpeedGoat Performance	Train speed	UDP, ethernet
TCMS Data	SpeedGoat Performance	• Available power • Catenary voltage • Traction/braking ratio	UDP, ethernet
Configuration file	Layout engine PC	Description of all configurations	SLRT, ethernet host link
Other TDS Unit Data	TDS Unit 2	• ID • displayed system • role • watchdog for health monitoring purposes	UDP, ethernet

Table 1 Data received by Unit 1 Display Manager

Data	Receiver	Description	Transmission means
Systems Data	Unit 1 Layout Engine	• Available power • Catenary voltage • Traction/braking ratio • Train speed	SLRT, ethernet host link

Unit data	Unit 1 Layout Engine	For health monitoring and configuration selection purposes • ID • viewSelection • switch off trigger • ego role	SLRT, ethernet host link
Ego Data	Other TDS Units (TDS Unit 2 in our architecture)	For health monitoring and configuration selection purposes • ID • displayed system • role • watchdog for health monitoring purposes	UDP, ethernet

Table 2 Data sent by Unit 1 Display Manager

3.3.2 Layout Engine

The Layout Engine software is a MATLAB application, created with the AppDesigner add-on and exported as a standalone application on both PCs in TDS Units thanks to the MATLAB Compiler toolbox.

It exchanges information with the Display Manager Software in the corresponding SpeedGoat Unit concerning which application it should display (e.g.: ETCS), what is the value of the data to be displayed (e.g. train speed) and whether a sound shall be generated (e.g. upon reconfiguration).

The layout for both applications consists of a static background image compliant with ERA_DMI_15560 with only a few gauges controlled by the systems' input data. The instrumented gauges are:

- For the ETCS:
  - The speed circular gauge (without the speed hook)
- For the TCMS:
  - The catenary voltage gauge
  - The available power gauge
  - The traction and braking percentage gauges

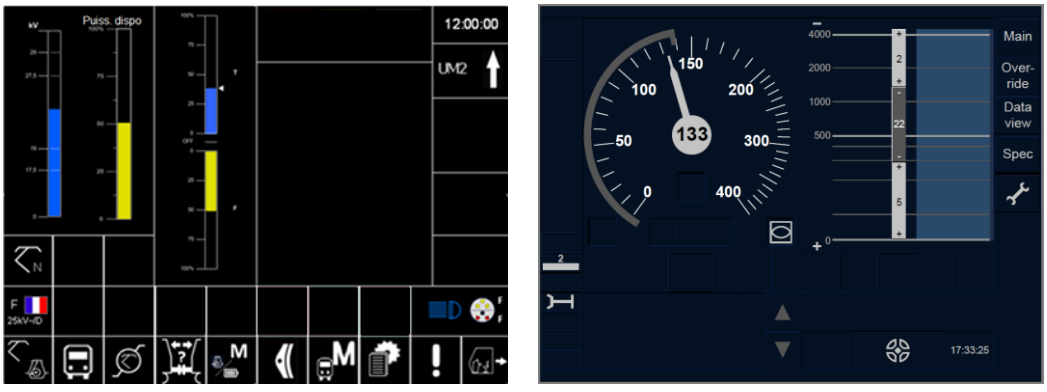


Figure 5 : TCMS and ETCS Layout

The Layout Engine Software is also responsible for reading the xml configuration file, and sending it to the display manager software hosted by the SpeedGoat Unit target.

The following tables show the data exchanged by the Unit 1 Layout Engine with other software in the system. Data highlighted in yellow is "proof of concept specific", in other words this exchange of data is necessary because of our PoC architecture but is not relevant in the TDS concept.

Data	Emitter	Description	Transmission means
Systems Data	Display Manager	• Available power • Catenary voltage • Traction/braking ratio • Train speed	SLRT, ethernet host link
Unit data	Display Manager	• ID • viewSelection • switch off trigger • ego role	SLRT, ethernet host link

Table 3 Data received by Unit 1 Layout Engine

Data	Receiver	Description	Transmission means
Configuration file	Display Manager	Description of all configurations	SLRT, ethernet host link

Table 4 Data sent by Unit 1 Layout Engine

3.3.3 System computing units (TCMS and ETCS)

The modelling of TCMS and ETCS system computing in this PoC is rudimentary. Its objective is solely to provide the TDS units with a few variables to be displayed by the adequate layout engines. At this level of implementation, no data is coming from the TDS Units to this model.

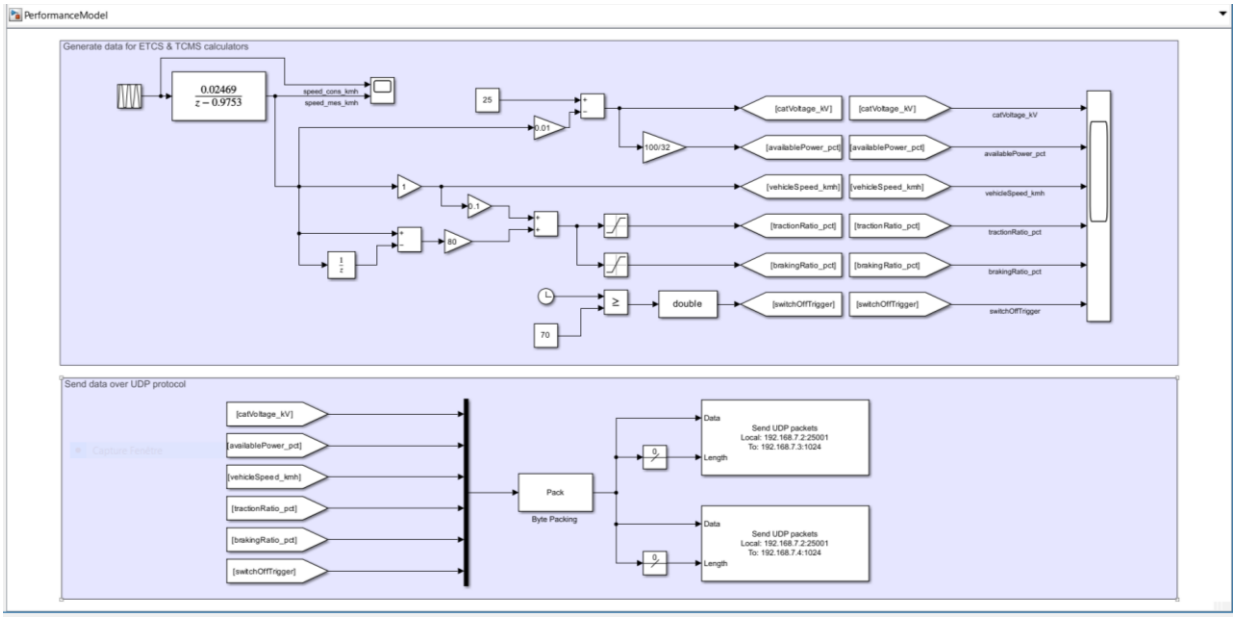


Figure 6 : System computing units model

For the PoC, a basic 80 seconds train movement scenario was set with the following characteristics:

- Acceleration from standstill to 150 km/h
- Speed maintained for 20 seconds
- Braking to standstill

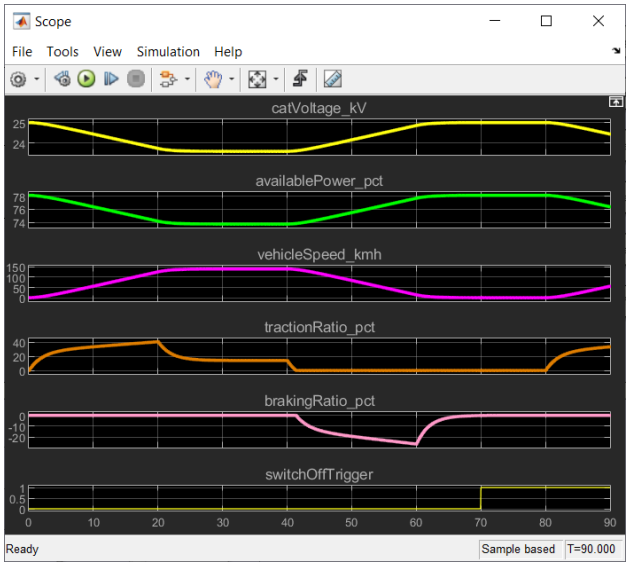


Figure 7 : Basic scenario data

The following table shows the data exchanged by the Performance model with other software in the system. Data highlighted in yellow is “proof of concept specific”, in other words this exchange of data is necessary because of our PoC architecture but is not relevant in the TDS concept.

Data	Receiver	Description	Transmission means
ETCS Data	TDS Units (SpeedGoat units)	• Train speed (km/h)	UDP, ethernet
TCMS Data	TDS Units (SpeedGoat units)	• Catenary Voltage • Available power • Traction level • Braking level	UDP, ethernet
Sequence data	TDS Units (SpeedGoat units)	Switch off trigger at the end of the scenario	UDP, ethernet

Table 5 Data sent by Performance model (system computing units)

3.4 Specification coverage

A TDS specification document has been initiated by SNCF and served as a basis for this PoC development. The following table depicts the PoC specification coverage with regards to this document.

Function	Description	Implementation	Comments
Cab Management	The aim of this function is to determine which TDS units are activated depending on the cab status	0%	Out of scope for the POC
Master Display Manager determination	The aim of this function is to determine the role of each operational Display Manager: master or slave.	100%	
Health Monitoring	The master Display Manager shall be aware about the availability of each Presentation logic and TDS units (slave Display Manager) activated in the cabin.	60%	Availability of slave DMs only, presentation logic health is not monitored as of now. No implementation of Subset 121 safe communication process.

Configuration selection and application	master Display Manager selects and shares the configuration to be set by Systems and TDS units. This action is performed at the initialisation or after a failure detection.	80%	Configuration is not sent to the systems. System data is sent to both TDS units at all times (simplification). No consistency check from the systems or the slave display units.
Communication between system and layout engine	Safety aspect of the communication	15%	No safe/unsafe communication assessment is made. The data packing has not been implemented; all system data is currently sent without discrimination of sender system from the SpeedGoat performance to the SpeedGoat units.
Building view	how the Layout Engine builds the view to display	10%	Simplified view built (see Layout engine chapter)
View swap	When many views share the same display, the driver can select the view displayed or the TDS can force the view to be display according to events of the train (train in motion, at standstill...)	70%	Function implemented for use case 3. Only swap on driver selection is implemented.
Diagnosis	Log TDS events for maintenance purposes	0%	Out of scope for the POC

Table 6 Specification coverage

4 PoC Results

4.1 Use Cases

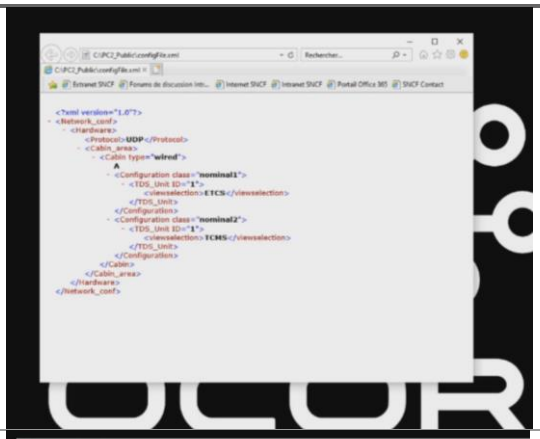
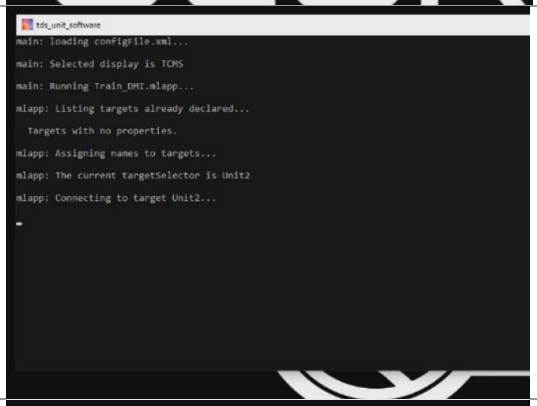
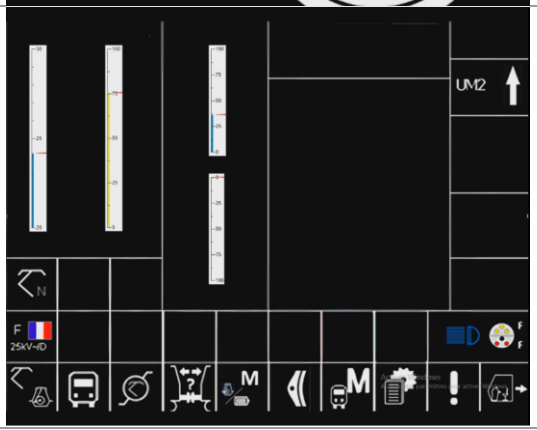
Use Case	Description	Implemented
Use Case 1	Modularity requirement	Yes
Use Case 2	Manage a nominal configuration	Yes
Use Case 3	Manage a display failure	Yes
Use Case 4	Reconfiguration on a system start or failure	No (lack of time)
Use Case 5	Reconfiguration depending on events	No (lack of time)

Table 7 Use Case coverage

The Use Case videos [2], [3] and [4] can be found on the OCORA Release R6 GitHub repository.

4.1.1 Use Case 1: Modularity requirement

The goal is to demonstrate the ability to display the specific view of any system on a single display (modularity requirement). The change of view is done by manually rebooting the TDS Unit and manually changing the configuration file.

Event	Displayed view
TDS configuration file is presented: configFile.xml. This file is loaded each time the TDS unit is powered on. Two configurations are presented. The configuration nominal1 for ETCS on display 1, and the configuration nominal2 for TCMS on display 1.	
Start-up of the TDS unit on a configuration nominal2 (TCMS view)	
Carry out a standard driving cycle: accelerate, cruise, decelerate and stop.	

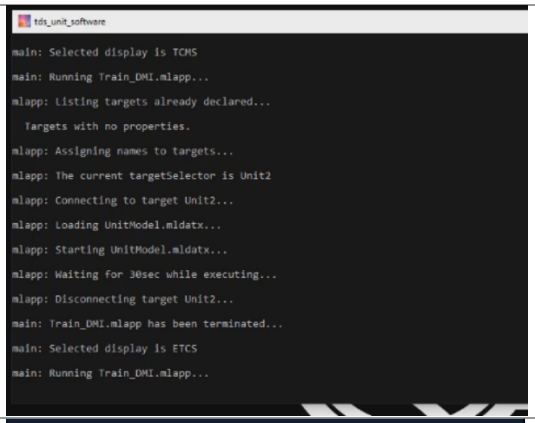
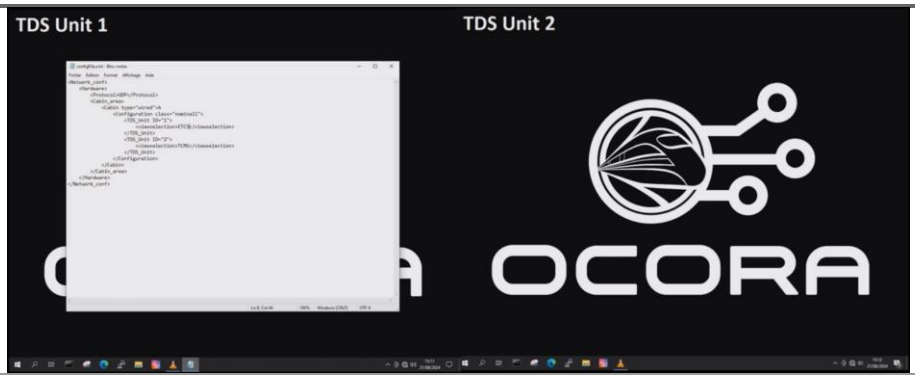
Restart the TDS unit with a new configuration nominal1 (ETCS view)	
Perform a second driving cycle	

Table 8 Use case 1 walkthrough

4.1.2 Use Case 2: Manage a nominal configuration

ETCS is shown on Display 1, and TCMS on Display 2. After a manual shutdown, the configuration file is manually modified by the tester so that ETCS is shown on Display 2 and TCMS on Display 1. The tester then boots the TDS Units to witness the altered configuration applied.

Event	Displayed views
TDS configuration file is presented: configFile.xml. The configuration defines ETCS on TDS unit 1 (left) and TCMS on TDS unit 2 (right).	

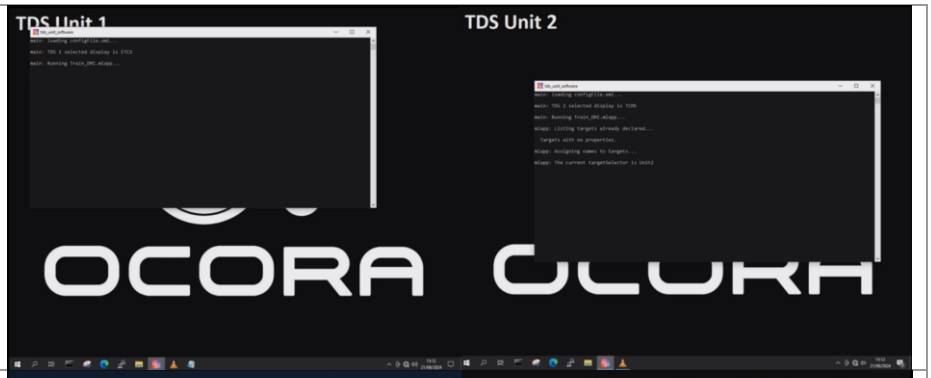
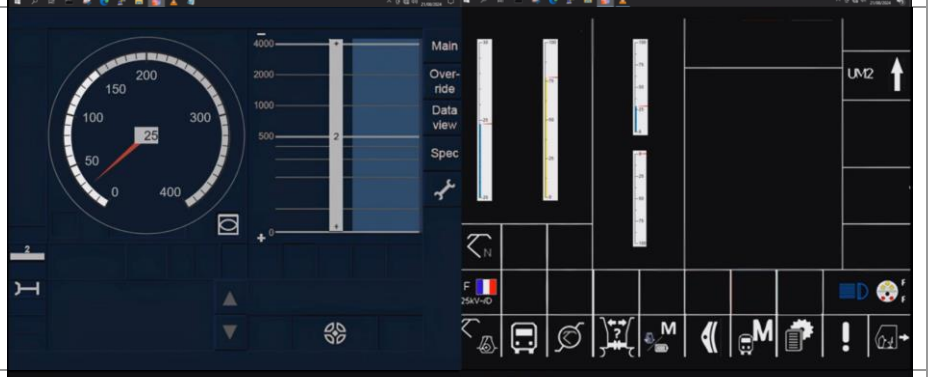
Start-up of the TDS units with a configuration selected by TDS unit 2 (master role) and shared to TDS unit 1 (slave role)	
Carry out a standard driving cycle: accelerate, cruise, decelerate and stop.	
The tester manually edits the configuration file so that the nominal configuration is now TCMS on TDS unit 1 (left) and ETCS on TDS unit 2 (right) The tester then restarts the TDS units.	
Perform a second driving cycle	

Table 9 Use case 2 walkthrough

4.1.3 Use Case 3: Manage a display failure

If a display supporting a mandatory system fails, TDS shall reallocate the system's view to another display used by an optional system.
For instance, mandatory system A uses display 1 and optional system B uses display 2. If display 1 becomes unavailable, TDS will apply a degraded configuration. System A will use display 2 and system B's view won't be displayed.

In this scenario the mandatory system A is ETCS and the optional system B is TCMS (lower priority).

The failure and reboot commands are done manually by a tester. The reconfiguration is done automatically by the TDS.

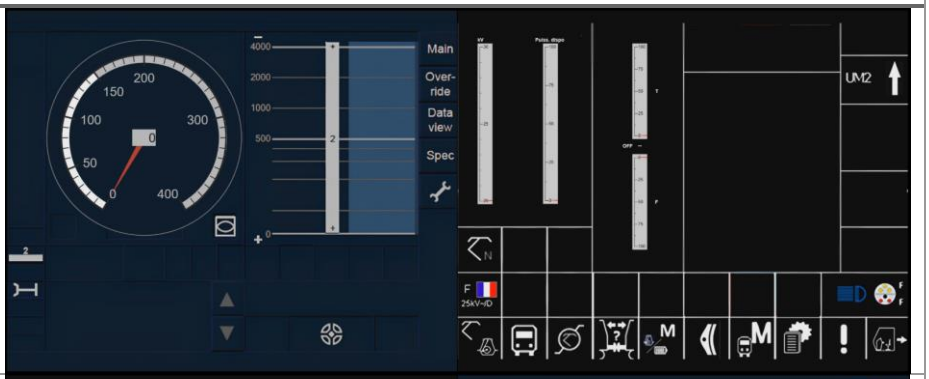
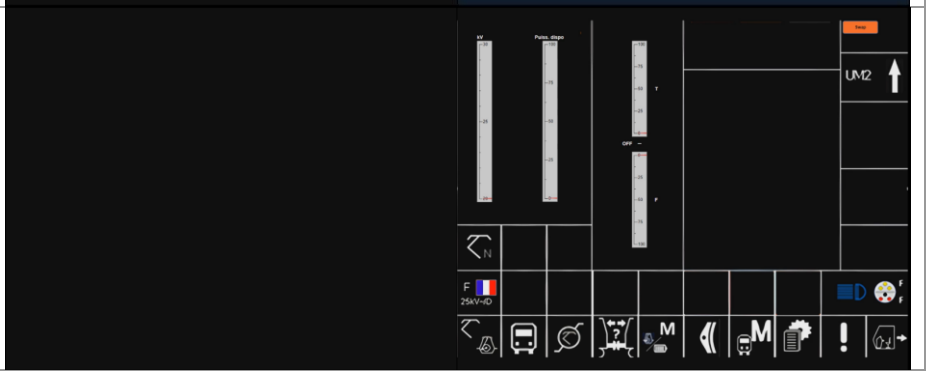
a) Use case 3.1: Cold redundancy.

Closely related to Use Case 2, in the event of a fault, reconfiguration should be performed when all systems are shut down. In a train, for example, the driver can activate a switch to change the configuration and restart all systems and the TDS. In other words, the TDS doesn't manage health monitoring of TDS units and systems. The master Display Manager only considers an external logical input that is activated by tester. The ergonomics applied to inform the driver about the presence of a stacked view will be a rectangle displayed on the top right of the view. When present, the driver will be able to swap the view by a command (touch the screen or external button activation: to be discussed).

a) Use case 3.2: Hot redundancy

As in Use Case 3.1, reconfiguration is required in the event of a failure. In this Use Case, the reconfiguration is performed automatically by the TDS without disturbing the systems. In other words, the TDS manages health monitoring of TDS units and systems. Depending on their states, TDS selects the appropriate configuration and applies it. In this use case, no driver action and systems reboot are needed.

The choice was made to only test the Use case 3.2: Hot redundancy. The mechanism in place is the same that would occur on cold redundancy, as a reconfiguration is actually the same as a configuration.

Event	Displayed views
Start-up of the TDS units with a nominal configuration shared by TDS unit 1.	
Tester switches off TDS unit 1, simulating a hardware error. TDS unit 2 automatically reconfigures the display, stacking both views and adding the possibility to switch between the views (orange swap button on the top right-hand corner)	
Tester switches from ETCS to TCMS layout on display 2 by clicking the "swap" button	

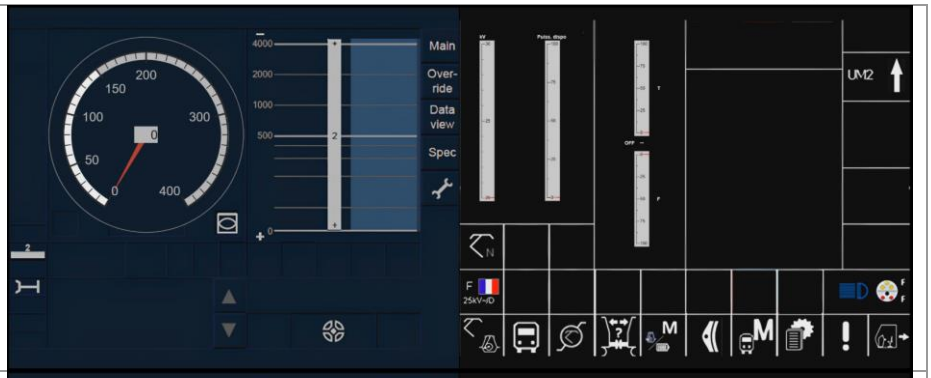
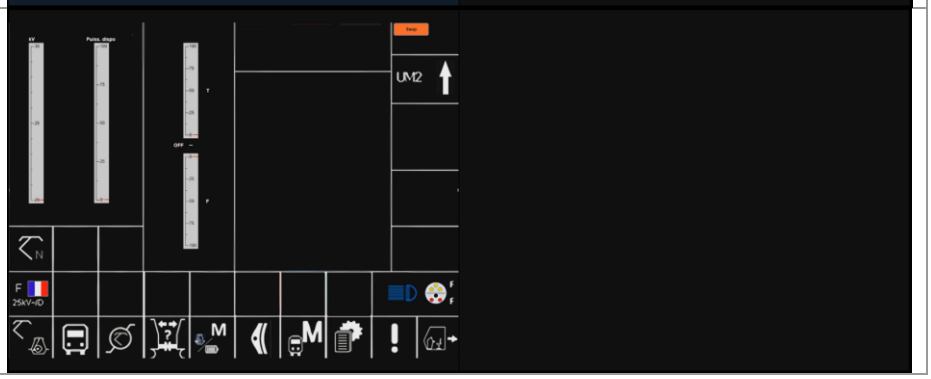
Reboot of TDS unit 1, automatic reconfiguration done by TDS unit 2 to the nominal configuration.	
Tester switches off TDS unit 2, simulating a hardware error. TDS unit 1 automatically reconfigures the display, stacking both views and adding the possibility to switch between the views. (orange swap button on the top right-hand corner)	
Tester switches from ETCS to TCMS layout on display 1 by clicking the “swap” button	

Table 10 Use case 3 walkthrough

4.1.4 Use Case 4: Reconfiguration on a system start or failure

In this use case, three systems (ETCS, TCMS and rearview system) are considered, even though only two displays are present. The aim of this use case is to show how an event can affect several displays in series (cascade reconfiguration). First, the ETCS is in No Power, TCMS and Rearview system use existing displays. Then ETCS starts by means of a command from the tester. As ETCS is a mandatory system, configuration shall automatically adapt to display it. So TCMS view is moved to the right screen and rearview system view is stacked behind it. If ETCS fails, TDS shall warn the driver. When driver acknowledges this information, a new configuration is applied.

This use case was not implemented in the POC, due to lack of time.

It would have enabled a more complete implementation of the display manager, by adding a health management of the presentation logics and complexifying its configuration choice parameters (until now only the availability of display units is considered when choosing a configuration).

4.1.5 Use Case 5: Reconfiguration depending on events (driver selection or external events)

With two systems using a single display, the configuration could be changed depending on driver selection or events related to speed (detection of standstill or motion).

For instance, if a system A shall be displayed in motion, the configuration shall change in case motion is detected and if system B is originally displayed.

This use case was not implemented in the PoC, due to lack of time.

It would have enabled a more complete implementation of the display manager, by complexifying its configuration choice parameters (until now only the availability of display units is considered when choosing a configuration).

4.2 PoC Limitations

The architecture selected for this PoC induces some necessary limitations regarding its representativity of an actual on-board system. The main identified limitations are listed in the following table.

PoC Limitation	Mitigation
The Hardware separation of what is considered a TDS display unit (SpeedGoat target + Computer + Monitor) induces unnecessary complications to the concept: • Communication between Display Manager and Layout engine is not trivial • TDS shutdown is not trivial (when you shut down the SpeedGoat target the application on PC keeps running). The option of shutting down the PC would require a long reboot time. • The configuration file is hosted on the PC, and thus read by the Layout engine software which then sends its data to its expected user: the Display Manager software hosted on the SpeedGoat unit.	Some code workarounds were found by the developer to cope with this limitation. The next PoC will feature programmable railway DMIs, which will eliminate this limitation.
This PoC only features 2 TDS Units, which leads to an oversimplification of the Master Display Manager logic. Considering 3 Units instead of 2 would already provide a more robust system, capable of handling any number of TDS units.	The developer kept in mind the need for scalability when coding, assuming whenever possible that there could be a list of other TDS units instead of just one. However, this was not always possible in the constraint of the project timeline, consequently some simplifications have been made, especially in the configuration selection phase.
The driver actions (switch between views) are done by mouse click instead of touch screen or softkey.	No major impact identified
No data is being sent from the TDS to the ETCS and TCMS systems.	In the second PoC it would be interesting to scale up the TDS Unit software and the ETCS software by adding upstream data such as driver acknowledgement of events.

Table 11 PoC Limitations

4.3 Feedback for TDS specification activity

This chapter addresses various issues that came up during the development of the PoC and that might be of interest for the TDS specification paper [6].

a) Initialization phase

The TDS initialization phase was not trivial to implement. To avoid multiple reconfiguration each time a new

TDS unit is powered on, it was decided to add a timer. The TDS units move on to their next phase (Role Attribution) as soon as:

- All known TDS units in the CAB are connected.

OR

- The initialization timer has elapsed.

After trying multiple timer values, we decided on 10 seconds in the PoC.

There is also a need to specify what should be displayed by the TDS units during this initialization phase.

b) Configuration file

c) The configuration file should be explicit enough for the Master display manager to be able to choose the adequate display configuration in relation to the active systems and TDS. One “health state” of the TDS system should not correspond to multiple possible approved configurations. This will be the maintainer’s constraint while defining the configuration file.

d) The configuration file must be identical in all TDS units, to prevent unwanted reconfiguration in case the Master role is reattributed on the fly. This could be checked by exchanging configuration file version number between Display Managers during the initialization phase.

- A configuration must specify the stacked views for each display where it is necessary. (Example in Appendix 1: Configuration File for Use Case 3, where a “layer” attribute was added to address this issue)

e) Role

f) The rule for deciding which TDS unit takes the master role needs to be shared among all TDS unit software. In our PoC the master role is attributed to the TDS Unit with the highest ID.

- Upon reboot of a display unit and while the cab remains active, no role reattribution is made even if the rebooted display would have been elected master otherwise. This is to avoid unnecessary activity which could lead to system instabilities.

g) It was decided not to add a feature indicating to the driver which display has the master role, since this information was deemed irrelevant. This issue is debatable and could evolve after a deeper human factor analysis.

- The PoC software refers to “Leader” and “Follower” roles, while the TDS specification activity uses the “Master/Slave” terminology. This terminology tends to disappear in recently concepted systems, which should be considered for the TDS specification activity.

h) Health monitoring and Reconfiguration

- Upon detection of a change in the TDS’s health state (display unit shuts down or is rebooted for instance), the previous configuration should remain applied until every unit is ready to switch to the new configuration selected by the master TDS unit.
- Make sure a display considered “down” by the TDS does not display any system data.

i) The master TDS unit should regularly send its identity and role to the whole system, to ensure that a rebooting display unit will attribute itself the role of a slave.

j) Miscellaneous

- The system’s cycle time was set to 50ms, obtaining a good balance between display fluidity and computing time.
- There was a discussion about how the Master TDS unit should communicate its configuration choice to the Slave TDS units, the two considered options being:

- Send to each TDS unit its requested system to display: (e.g.: TDS1 specifically asks TDS2 to display ETCS data).

k) Only send a configuration ID. It is then up to the slave TDS units to read their copy of the configuration file and determine which system they should display according to the configuration ID indicated by the master unit (e.g.: TDS1 asks TDS2 to apply configuration number 3). This option has the advantage of only requiring the master unit to broadcast a configuration ID, versus sending personalized assignments to each slave unit as proposed in option 1.

In this PoC, for development simplicity purposes option 1 was chosen since there is at most one slave unit. However, it is not clear which option is best suited for a more complex TDS architecture.

- The position of the “swap” button should not be altered when swapping between views, which will prove more and more complex as we will add systems to be displayed (the button needs to not overlay system data). A simpler solution might be to externalize the swap button, or alternatively reserve a screen region for TDS user interface.

Figure 8 : position of the swap button for ETCS and TCMS layouts

Further activities

Through this PoC activity we demonstrated the feasibility of important TDS features such as modularity, configuration application, reconfiguration on events and management on stacked views on a single display.

However, this demonstration was done using a simplified system (only two applications and two DMIs) and simplified processes concerning data exchange between elements as well as view construction.

That is why we have the aim to push this demonstration to a more realistic architecture using standard railway DMIs. This would eliminate the majority of PoC limitations identified in paragraph 4.2, taking us one step closer to the inclusion of the TDS concept in TSIs.

This new phase of the PoC, planned for 2025, will also be the opportunity to enhance our software, implementing more TDS functions and requirements that were ignored in this first step for simplification and time constraint reasons.

5 Appendices

5.1 Appendix 1: Configuration File for Use Case 3

```
<Network_conf>
   <Hardware>
      <Protocol>UDP</Protocol>
      <Cabin_area>
         <Cabin type="wired">A
            <Configuration class="nominal1">
               <TDS_Unit ID="1">
                  <viewselection layer="1">ETCS</viewselection>
                  <viewselection layer="2">Empty</viewselection>
               </TDS_Unit>
               <TDS_Unit ID="2">
                  <viewselection layer="1">TCMS</viewselection>
                  <viewselection layer="2">Empty</viewselection>
               </TDS_Unit>
            </Configuration>
            <Configuration class="failure1">
               <TDS_Unit ID="1">
                  <viewselection layer="1">Empty</viewselection>
                  <viewselection layer="2">Empty</viewselection>
               </TDS_Unit>
               <TDS_Unit ID="2">
                  <viewselection layer="1">ETCS</viewselection>
                  <viewselection layer="2">TCMS</viewselection>
               </TDS_Unit>
            </Configuration>
            <Configuration class="failure2">
               <TDS_Unit ID="1">
                  <viewselection layer="1">ETCS</viewselection>
                  <viewselection layer="2">TCMS</viewselection>
               </TDS_Unit>
               <TDS_Unit ID="2">
                  <viewselection layer="1">Empty</viewselection>
                  <viewselection layer="2">Empty</viewselection>
               </TDS_Unit>
            </Configuration>
         </Cabin>
      </Cabin_area>
   </Hardware>
</Network_conf>
```