Grant Agreement Number: 881778
Project Acronym: FR8RAIL III
Project title: Smart data-based assets and efficient rail freight operation

DELIVERABLE D2.3
Demonstration of enhanced and integrated line- and yard planning and possibilities for implementation

<table>
<thead>
<tr>
<th>Project acronym:</th>
<th>FR8RAIL III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting date:</td>
<td>01-09-2019</td>
</tr>
<tr>
<td>Duration (in months):</td>
<td>46</td>
</tr>
<tr>
<td>Call (part) identifier:</td>
<td>H2020-S2RJU-CFM-2019</td>
</tr>
<tr>
<td>Grant agreement no:</td>
<td>881778</td>
</tr>
<tr>
<td>Grant Amendments:</td>
<td>None</td>
</tr>
<tr>
<td>Due date of deliverable:</td>
<td>31-08-2022</td>
</tr>
<tr>
<td>Actual submission date:</td>
<td>22-12-2023</td>
</tr>
<tr>
<td>Coordinator:</td>
<td>Tomas Lidén, Swedish Nat’l Road and Transport Research Institute</td>
</tr>
<tr>
<td>Lead Beneficiary:</td>
<td>Magnus Wahlborg, Trafikverket</td>
</tr>
<tr>
<td>Version:</td>
<td>1.4</td>
</tr>
<tr>
<td>Type:</td>
<td>Report</td>
</tr>
<tr>
<td>Sensitivity or Dissemination level¹:</td>
<td>Public</td>
</tr>
<tr>
<td>Contribution to S2R TDs or WAs²:</td>
<td>TD 5.2 - Digital Transport Management</td>
</tr>
<tr>
<td>Taxonomy/keywords:</td>
<td>Logistics, Rail Transport, Rail Operations</td>
</tr>
</tbody>
</table>

¹ PU: Public; CO: Confidential, only for members of the consortium (including Commission Services)
² https://projects.shift2rail.org/s2r_matrixtd.aspx

This project has received funding from the Shift2Rail Joint Undertaking (JU) under grant agreement 881778. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and the Shift2Rail JU members other than the Union.
### Document history

<table>
<thead>
<tr>
<th>Date</th>
<th>Name</th>
<th>Affiliation</th>
<th>Position/Project Role</th>
<th>Action/Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>03-09-2021</td>
<td>TL</td>
<td>VTI</td>
<td>Coordinator</td>
<td>Document initiated.</td>
</tr>
<tr>
<td>28-02-2022</td>
<td>TL</td>
<td>VTI</td>
<td>Coordinator</td>
<td>Structure</td>
</tr>
<tr>
<td>27-06-2022</td>
<td>TL</td>
<td>VTI</td>
<td>Coordinator</td>
<td>All major texts written, reviewed and revised. Clean up: references, formatting, listings etc.</td>
</tr>
<tr>
<td>01-07-2022</td>
<td>TL</td>
<td>VTI</td>
<td>Coordinator</td>
<td>All summarising texts finished and reviewed. Final brush up. Version sent to TMT for external review.</td>
</tr>
<tr>
<td>08-07-2022</td>
<td>MJ</td>
<td>RISE</td>
<td>Co-editor</td>
<td>Adjust according to TMT comments</td>
</tr>
<tr>
<td>05-08-2022</td>
<td>TL</td>
<td>VTI</td>
<td>Coordinator</td>
<td>Some final adjustments. Set version to 1.0.</td>
</tr>
<tr>
<td>02-28-2023</td>
<td>JMR</td>
<td>Indra</td>
<td>Contributor</td>
<td>Added info in Ch. 3.</td>
</tr>
<tr>
<td>15-03-2023</td>
<td>MJ</td>
<td>RISE</td>
<td>Co-editor</td>
<td>Adjustments after comments from JU; Set version to 1.1</td>
</tr>
<tr>
<td>13-07-2023</td>
<td>MJ,</td>
<td>RISE,</td>
<td>Ch 2, 2.2, 3.3.2, 3.3.3, and 3.5. Updates and added text in Ch. 1, 3, 6 and 7. Set to version 1.2.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SG,</td>
<td>Indra</td>
<td>Ch 3</td>
<td>New text Ch 3.3, added text in Ch 7. Set to version 1.3</td>
</tr>
<tr>
<td></td>
<td>JMR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22-12-2023</td>
<td>MJ,</td>
<td>RISE,</td>
<td>Added text in Ch 7.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MW</td>
<td>TRV</td>
<td>Set to version 1.4.</td>
<td></td>
</tr>
</tbody>
</table>

### Report contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>Details of contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tomas Lidén</td>
<td>VTI</td>
<td>Coordinator/editor. Ch 1, 6, 7. General reviewing.</td>
</tr>
<tr>
<td>Martin Joborn</td>
<td>RISE</td>
<td>Ch 1, 2, 3, 7. General reviewing.</td>
</tr>
<tr>
<td>Sara Gestrelius</td>
<td>RISE</td>
<td>Ch 2.2, 3, 6, 7. Review of ch 5.</td>
</tr>
<tr>
<td>Javier Monje Rubio</td>
<td>Indra</td>
<td>Ch 3.</td>
</tr>
<tr>
<td>Enrique Gómez Gonzáles</td>
<td>Indra</td>
<td>Ch 3.</td>
</tr>
<tr>
<td>Carl-Henrik Häll</td>
<td>LIU</td>
<td>Ch 4, 6, 7.</td>
</tr>
<tr>
<td>Anders Peterson</td>
<td>LIU</td>
<td>Review of ch 3.</td>
</tr>
<tr>
<td>Christiane Schmidt</td>
<td>LIU</td>
<td>Ch 4, 6, 7.</td>
</tr>
<tr>
<td>Niloofar Minbashi</td>
<td>KTH</td>
<td>Ch 5, 6, 7. Review of ch 4.</td>
</tr>
<tr>
<td>Hans Sipilä</td>
<td>KTH</td>
<td>Ch 5, 6, 7. Review of ch 4.</td>
</tr>
<tr>
<td>Magnus Wahlborg</td>
<td>TRV</td>
<td>WP leader, general reviewing.</td>
</tr>
</tbody>
</table>
Table of contents

Executive Summary..................................................................................................................................................5
List of abbreviations, acronyms, and definitions....................................................................................................7
List of figures..........................................................................................................................................................8
List of tables .........................................................................................................................................................10

1 Introduction.....................................................................................................................................................11
   1.1 Problem description and objectives .............................................................................................................11
   1.2 Scope and outline ..........................................................................................................................................12

2 Time perspective and yard automation .............................................................................................................15
   2.1 Time perspective of real-time traffic management .....................................................................................15
   2.2 YCS and yard automation ..........................................................................................................................16

3 Demonstration of a Yard Coordination System ...............................................................................................19
   3.1 Introduction to Malmö marshalling yard and relevant actors .....................................................................19
   3.2 Open calls related to YCS ..........................................................................................................................22
   3.3 Description of YCS ......................................................................................................................................28
   3.4 Demonstration workshop ...........................................................................................................................40
   3.5 Automatic data update experiments ...........................................................................................................52
   3.6 Conclusion of YCS development in FR8RAIL III WP2 ............................................................................66
   3.7 Next step for YCS .......................................................................................................................................67

4 Application of replanning methods to selected disturbance scenarios .............................................................68
   4.1 Timetable Modification Module ................................................................................................................68
   4.2 Effect of parameter settings in TIMO on output ...........................................................................................69
   4.3 Using TIMO for replanning on the line in case of ad-hoc maintenance at the departure marshalling yard ..........................................................................................................................78

5 Interconnection of a yard prediction model with a network macrosimulation model ......................................81
   5.1 Yard Prediction Model ..................................................................................................................................82
   5.2 Network simulation model – Proton .............................................................................................................83
   5.3 Simulation setup ..........................................................................................................................................84
   5.4 Simulation runs with examples ....................................................................................................................85
   5.5 Results of evaluating yard arrival predictions .............................................................................................88

6 Usage and impact ..............................................................................................................................................93
   6.1 Possibilities for usage in support of the real-time traffic management .........................................................93
   6.2 Evaluation of KPI impact .............................................................................................................................94
Executive Summary

This document constitutes Deliverable D2.3 of the project FR8RAIL III Work Package (WP) 2 “Real-Time Network Management” within the framework of the Technology Demonstrator - TD5.2 “Digital Transport Management” of IP5 “Technologies for Sustainable & Attractive European Rail Freight” in Shift2Rail. The report documents the conducted work and results from task T2.3 and is the final deliverable from WP2.

Three methods and tools have been developed, analysed, and demonstrated on real-world operational data. These tools address three different but equally important aspects for improving the efficiency of real-time network management of railways and contribute to closing the gap between planning and operations by enabling the traffic management to have a more proactive way of working. The focus of the work is on the coordination of freight operations at lines and marshalling yards, namely:

- Coordination of all operational activities taking place at arrival/departure yards.
- Replanning of timetables for line traffic.
- Prediction of system effects and the combined operations of yards and lines.

Firstly, an integrated demonstration platform for planning operational activities at a marshalling yard is studied. The developed Yard Coordination System (YCS) itself is described as well as how it has been applied and demonstrated in a workshop with experienced participants from the three principally involved stakeholders. The demonstration has shown that a tool like YCS can improve transparency and enable cooperative and pro-active planning. The practitioners reckoned that the tool could prevent and alleviate departure delays, and they expressed a strong wish for continued development of such support. An extensive list of experiences, development suggestions, potentials and risks are reported.

Secondly, a timetable modification module (TIMO) for short-term replanning of line traffic is evaluated. The method uses a heuristic approach that aims at achieving a high bottleneck robustness, which together with algorithm runtime and several other criteria (train path deviation, change in departure time etc) are used in the evaluation. The effect of several parameters in TIMO are studied, such as iteration settings, size of allowed time windows and share of other train paths that may be adjusted—both for peak and off-peak traffic. Furthermore, how TIMO can be used in an iterative procedure to solve the replanning problem on the line in case of ad-hoc maintenance at the departure marshalling yard is demonstrated. The results show that TIMO’s performance depends greatly on various parameter settings, which delimits the (current) use cases for TIMO.
Thirdly, a proof-of-concept model framework for increasing the predictability of yard departures and arrivals is evaluated. The model framework incorporates a machine learning-based yard departure deviation prediction model (YPM) into a macroscopic network simulation model (Proton). Both the infrastructure manager and the yard operator can benefit from this model framework; the former by getting a more realistic picture of the train that runs along the line, the latter by improved yard arrival estimations.

Finally, the possibilities for real-time usage of these methods and tools are discussed along with their respective impact on the three system level performance indicators load factor, punctuality and average (transportation) speed.
**List of abbreviations, acronyms, and definitions**

<table>
<thead>
<tr>
<th>Abbreviation / Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/D-yard</td>
<td>Combined yard for arrivals and departures</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ASO</td>
<td>Automatic Shunting Operation</td>
</tr>
<tr>
<td>ATO</td>
<td>Automatic Train Operation</td>
</tr>
<tr>
<td>AWO</td>
<td>Automatic Wagon Operation</td>
</tr>
<tr>
<td>CDM</td>
<td>Canonical Data Model</td>
</tr>
<tr>
<td>ETA</td>
<td>Expected time of arrival</td>
</tr>
<tr>
<td>(F)RU</td>
<td>(Freight) Railway Undertaking</td>
</tr>
<tr>
<td>GL</td>
<td>Game leader</td>
</tr>
<tr>
<td>HCI</td>
<td>Human computer interaction</td>
</tr>
<tr>
<td>IL</td>
<td>Integration Layer</td>
</tr>
<tr>
<td>IM</td>
<td>Infrastructure Manager</td>
</tr>
<tr>
<td>IM-TC</td>
<td>Infrastructure Manager Traffic Control Center</td>
</tr>
<tr>
<td>KPI</td>
<td>Key performance indicator</td>
</tr>
<tr>
<td>LM</td>
<td>Line manager</td>
</tr>
<tr>
<td>MAE</td>
<td>Mean absolute error</td>
</tr>
<tr>
<td>ML</td>
<td>Machine learning</td>
</tr>
<tr>
<td>NMS</td>
<td>Node Management System</td>
</tr>
<tr>
<td>PANDAS</td>
<td>Python Data Analysis Library</td>
</tr>
<tr>
<td>RailML</td>
<td>Railway Markup Language</td>
</tr>
<tr>
<td>RISE</td>
<td>RISE Research Institutes of Sweden</td>
</tr>
<tr>
<td>RTYM</td>
<td>Real Time Yard Management</td>
</tr>
<tr>
<td>TAF/TAP TSI</td>
<td>Technical Specification for Interoperability relating to Telematics Applications for Freight/Passenger Services</td>
</tr>
<tr>
<td>TIMO</td>
<td>Timetable Modification Module</td>
</tr>
<tr>
<td>TM</td>
<td>Terminal manager</td>
</tr>
<tr>
<td>TMS</td>
<td>Traffic Management System</td>
</tr>
<tr>
<td>TNMC</td>
<td>Train Node Management System</td>
</tr>
<tr>
<td>TRL</td>
<td>Technology readiness level</td>
</tr>
<tr>
<td>YCS</td>
<td>Yard Coordination System</td>
</tr>
<tr>
<td>YM</td>
<td>Yard Manager</td>
</tr>
<tr>
<td>YPM</td>
<td>Yard Departure Deviation Prediction Model</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
</tbody>
</table>
List of figures

Figure 1: Schematic overview of the Yard Coordination System (YCS) and the actors in Malmö (Background image www.openrailwaymap.org) ......................................................... 22

Figure 2: Screenshots of the list view (a) and the Gantt view (b) ................................. 29

Figure 3: Interface for detailed element data. Blue fields show current data, and white fields are for modifying the data. As can be seen, not all data can be changed by this user. .. 33

Figure 4: Repository architecture for a specific node. .................................................. 34

**Figure 5: TAF TSI as interoperable protocol in the IL** .............................................. 35

Figure 6: Detailed architecture for connecting YCS to TMS ........................................ 36

Figure 7: High-Level diagram of initialization of the connection. .................................. 39

Figure 8: The game leader's sheet in the Excel document ............................................ 42

Figure 9: The line manager's Excel sheet and YCS view ............................................. 43

Figure 10: The yard manager's Excel sheet and YCS view .......................................... 43

Figure 11: The terminal manager's Excel sheet and YCS view .................................... 44

Figure 12: The participants were put in different rooms during the distributed sessions. .. 46

Figure 13: High-level diagram for updating the YCS with a new train .......................... 54

Figure 14: High-level diagram for train cancellations in YCS ...................................... 55

Figure 15: High-level diagram for when a train is modified in TMS and this has an effect on YCS. .......................................................... 56

Figure 16: High-level diagram for when a train is modified in YCS and this has an effect on TMS .......................................................... 57

Figure 17: Initial situation of YCS. .............................................................................. 59

Figure 18: Result in YCS after scenario 1 ..................................................................... 61

Figure 19: Initial status of scenario 2, before cancelling trains. .................................... 62

Figure 20: Final status of scenario 2, after cancelling trains ........................................ 63

Figure 21: Initial status of scenario 3 ............................................................................ 64

Figure 22: Final status of scenario 3 ............................................................................ 66

Figure 23: Schematic view of the considered route from Mjölby to Hallsberg .............. 69

Figure 24: Average bottleneck robustness when varying the number of path iterations and insertion iterations. The model was run 30 times with each parameter combination .... 71

Figure 25: Relative standard deviation of the bottleneck robustness when varying the number of path iterations and insertion iterations. The model was run 30 times with each parameter combination ......................................................... 71

Figure 26: Average TIMO runtime for varying path insertions over the number of insertion iterations. The model was run 30 times with each parameter combination ................. 72
Figure 27: Average bottleneck robustness and total deviation from the original timetable when running the model with different sizes of the time window \( W \). The model was run 30 times for each time window size. .................................................................75

Figure 28: Share of adjusted train paths. The model was run 30 times for each time window size. ........................................................................................................78

Figure 29: Railway line between Malmö and Hallsberg. The total distance is around 400 km. Highlighted lines are included in the simulations. .................................................................82

Figure 30: Overview of the data sources and the process used for the network simulation in Proton ..........................................................................................................................84

Figure 31: Simulated outcome for one freight train (Malmö–Hallsberg) using a general initial distribution at Malmö marshalling yard (median and percentile intervals) .........................87

Figure 32: Simulated outcome for one freight train (Malmö–Hallsberg) where initiation time at Malmö freight yard comes from Yard Prediction Model (median and percentile intervals) .88

Figure 33: Distribution of the arrival deviations .........................................................................................................................90

Figure 34 Distribution of the prediction errors .........................................................................................................................91

Figure 35: Boxplot of the most frequent trains. .........................................................................................................................92
List of tables

Table 1: Roles and responsibilities in YCS. ................................................................. 21
Table 2: Actors/roles in YCS, SMART and OptiYard. .................................................. 26
Table 3: Explanation of the symbols used in the Gantt chart view. .............................. 30
Table 4: Workshop participants .................................................................................... 41
Table 5: Agenda for demonstration workshop ............................................................... 44
Table 6: Parameters ........................................................................................................ 70
Table 7: Parameter values, in all cases: F0=120 minutes, γ=1. ................................. 73
Table 8: Results for t=11:00 ....................................................................................... 74
Table 9: Results for t=21:00 ....................................................................................... 74
Table 10: Parameter values, in all cases: Wrd=30 minutes, F0=120 minutes .......... 76
Table 11: Results for t=11:00 ..................................................................................... 77
Table 12: Results for t=21:00 ..................................................................................... 77
Table 13: General model performance indicators ......................................................... 89
1 Introduction

This document constitutes the Deliverable D2.3 in the framework of the project FR8RAIL III Work Package (WP) 2 “Real-Time Network Management” and the Technology Demonstrator - TD5.2 “Digital Transport Management” of IP5 “Technologies for Sustainable & Attractive European Rail Freight” in Shift2Rail.

This document reports the results from task T2.3 and is the final deliverable from WP2.

1.1 Problem description and objectives

Real-time network management deals with the short-term planning, control and follow up of the railway traffic that is actually performed in the railway network, including both lines and yards and the coordination of these. The plans and operations are updated according to the live situations in the railway system, also considering disruptions and actual states of the traffic, infrastructure, vehicles, etc. One core actor in real-time network management is the traffic control centre of the Infrastructure Manager that dispatch and controls the traffic in the network and interact with, for example, terminals and yards. The time perspective of the traffic control centre is the nearest 24 hours, and in particular from “now” to a few hours into the future. In Section 2, we further elaborate the relevant time perspective of the traffic control and put it in relation to the developments in this work package.

Real-time network management will improve operational processes by improved methods, information support and human interaction. The research here aims at reducing the gaps between timetable planning and operational traffic, and between yard management and line management. WP2 aims at carrying out a TRL 6 demonstration utilizing the integrated information platform developed in the project. The demonstration will include several important actors of the processes in scope, such as line planner, yard manager and arrival/departure yard planner. The aims of the demonstrations are both to show important actors the possibilities of integrated information platforms, and to get feedback on the provided concepts, use-cases, and cooperation ideas. The overarching objectives of WP2 can be summarized as follows:

a. To automate sequential planning and efficient human interaction, overcoming the current lack of synergies when handling and sharing data among infrastructure managers, yard/terminal managers, railway undertakings and maintenance contractors.

b. To improve interaction and balancing capacity utilization between infrastructure managers, yard/terminal managers, railway undertakings and maintenance contractors.

c. To reduce the gap between timetable planning and operational traffic.

d. To develop a demonstrator and perform the demonstration in TRL 6. To assess the maturity of planning and decision support methods used in this project.

The work package has been divided into three tasks (T2.1 – T2.3), with three deliveries as listed below. This document is the last of these three deliverables within WP2:
• **D2.1** Specification of innovations and scenarios for enhanced and integrated line- and yard planning.
• **D2.2** Method development for enhanced and integrated line- and yard planning.
• **D2.3** Demonstration of enhanced and integrated line- and yard planning and possibilities for implementation.

Essentially, these deliverables form a sequence of activities where investigation and specifications were reported in D2.1 (FR8RAIL III, 2021), method development in D2.2 (FR8RAIL III, 2022) and evaluation and demonstrations here in D2.3.

The objectives of the present deliverable have been to fulfill task T2.3 described by the following sub-tasks:
- Specification, planning and performing of demonstrations
- Fine-tuning of integrated information platform used for demonstrations
- Evaluation of demonstrations and conclusion of lessons-learned
- Investigating possibilities for real time usage
- Evaluation of the KPI impact of “Real time network management” concerning the KPIs load factor, punctuality, and average speed

### 1.2 Scope and outline

The freight railway network constitutes both the lines and the yards, and to achieve a good freight network management, the planning and operations of lines and yards should be integrated and not considered as two separate units. A primary focus is the situation and needs of the traffic control centre to control track capacity utilization at Malmö arrival/departure yard, and how to develop the real-time network management that is performed by them. The time perspective we work with is the same as the traffic control centre, i.e. the nearest 24 hours.

The integration of lines and yards and the traffic control centre perspective are the overall basis for the work in this document: In three different ways, contributions to the real-time network management and to the integrated planning and operations of lines and yards are made. As mentioned, the scope of this report is to summarize the demonstrations, experiments, and computational results of the work performed in WP2, while background, motivation, problem descriptions, method development, etc., were described in the previous deliverables. However, a discussion on two important aspects for this work - time perspective and yard automation - are given in Chapter 2.

The first way the performed work contributes to the real-time network management is to develop the decision support for operation of the arrival/departure yard at a major marshalling yard, which is where the lines and yard meets. Both the line and yard operations and actors are
dependent on an efficient arrival/departure yard operation and the developed decision support system integrates the operational planning and real-time control for the most significant actors. Chapter 3 outlines the user interface of a decision support system aimed at visualisation and coordinated planning, Yard Coordination System (YCS). The results from a demonstration workshop with domain experts and potential users/actors from three different organizations. The chapter also contains information on how the coordination tool could be connected to the Traffic Management System (TMS).

The second contribution is a new method for short-term modifications of the line’s timetables, Timetable Modification module (TIMO). In a freight transport setting, short-term requests for changes in the timetable are quite common, and these requests very often are connected to the yard operations, thus creating a link between yard status and line planning. In Chapter 4, the new methods are evaluated and computational experiments and results are reported.

The third contribution is a proof-of-concept model framework for increasing the predictability of yard departures and train arrivals, Yard Departure Deviation Prediction Model (YPM). Both the infrastructure manager and the yard operator can benefit from this model framework. The usability for the yard operator(s) is both applying the yard departure model and receiving the yard’s arrival time estimation from the simulation. The infrastructure manager mainly benefits from the simulation along the line which provides a more realistic picture of train runs along the line. Computational experiments and results are reported in Chapter 5.

In Chapter 6, the possibilities for real time usage of the developed methods and tools are discussed, and the impact that the results can have on important KPIs are commented. Finally, remarks and conclusions are given in Chapter 7.

Referring to the overarching objectives of WP2 in Section 1.1, objective a) is primarily treated by the yard coordination system described in Chapter 3, as improving interaction, data sharing, and visualisation between mentioned actors are all important drivers in the development and are also confirmed achievements. In particular, the possibility for the actors to share an integrated platform overcomes the lack of synergies and lack of data sharing that characterizes the current situation (without any shared platform). This objective is also treated by methods described in Chapter 4 and 5, as they automate sequential planning and create synergies. The short-term line replanning methods is of special interest for coordination with maintenance contractors since maintenance work is an importance cause for short-term replanning needs and for trains deviating from their timetables and also increasing the importance of having reliable predictions of arrival times.

Objective b) is handled in multiple ways in chapters 3, 4 and 5, as they all are means to improve
interaction and balance capacity utilization. The balancing of capacity utilization between traffic and maintenance was further treated in Deliverable D2.2 (Chapter 3).

Objective c) is also treated in various ways: In Chapter 4, the gap between timetable planning and operational traffic is reduced by enhanced methods for making adaptations to the timetable and thereby achieving better conformance to the actual operational needs, and in Chapter 5, by improved prediction and analysis methods that can increase the understanding of how the railway system operates. Also, the operational arrival and departure planning in the demonstrator reported in Chapter 3 is a significant contribution to the reduction of this gap. Finally, objective d) is handled by the demonstration workshop presented in Chapter 3, and also from the conclusions of all chapters and the related comments on the usage and impact in Chapter 6.
2 Time perspective and yard automation

In this chapter we describe and clarify the interpretation of the time perspective of “real-time traffic management” used in Fr8Rail III WP2 and also how the demonstrator YCS relates to research and development initiatives regarding yard automation.

2.1 Time perspective of real-time traffic management

Real-time network management deals with the short-term planning, control and follow up of the railway traffic that is performed in the railway network, including both lines and yards and the coordination of these. The plans and operations are updated according to the live situations in the railway system, including disruptions and actual states of the traffic, infrastructure, vehicles, etc. The core actor in real-time network management is the traffic control centre of the Infrastructure Manager. Other actors are staff working with operative control and planning, i.e. management, at RUs or service provider companies, e.g., at railway adjacent multimodal terminals. Important stakeholders are, e.g., traffic locomotive drivers, shunting locomotive drivers and staff working with various tasks (inspection, (de)coupling, (un)loading, switching etc.) at the yards or terminals.

The time perspective of the traffic control centre is the nearest 24 hours, and in particular from “now” to a few hours into the future. Real-time network management is thus both the control of what is happening right now, and the adaptation of the short-term plans that are directly affected by the current state on the railway network, and also the internal and external communication processes with relevant stakeholders, such as IM, FRU, yard- and terminal operators, and maintenance contractors. Thus, the time perspective includes the “true real-time” (now), but also the nearest future. We could call this “operational-time”. A focus within this work package has been to create different tools and methods for supporting the traffic controllers at the traffic control centre. All tools and methods work within the operational timeframe. However, for historic reasons we still refer to this as “real-time traffic management”. It is worth noticing that even the “true real-time” (now) in yard management is not as time sensitive as the tasks generally considered when talking about “real-time computing”. In real-time computing, ticks are often in terms of milliseconds, or even microseconds, a timeframe which is not useful for traffic controllers at yards as the state of the yard does not change that fast. For yard operations, a time tick of a minute, or even a few minutes, is more appropriate.

In the perspective of this work package, the traffic management works with three different questions:

- Q1: What is the situation now?
- Q2: What is the preferred situation in the short-term future?
- Q3: How to transition from the “now” situation to the preferred future situation?

All three questions are important. Historically, traffic control has had a focus on the first question.
and has worked reactively to what is happening in the railway system, i.e., track access has been granted on a first-come first-served basis rather than based on analysis of Q2 and Q3. A strategy from IMs is to move towards a more proactive traffic control, i.e., increase focus on Q2 and Q3. At Trafikverket, the traffic control should work with a “control by planning”-strategy, in contrast to a “control by execution”-strategy, which was the previous, more reactive, way of working. Relating this to the time perspective, Trafikverket want to shift from a “real-time/now”-focus to operational-time focus. To support the change towards an operational-time perspective and “control by planning”, the research and development activities in WP2 is focused on Q2 and Q3, rather than Q1.

The scope of the YCS-demonstrator, described in this report, is the arrival/departure yard at a major marshalling yard. At this yard, there are several actors from different organizations and today there is very limited information sharing possibilities regarding needs and planned actions except from phone and mail (FR8RAIL III, 2021, 2022). However, the current situation is relatively easy for the traffic controller to assess. This fact makes it a high priority to create support for Q2 and Q3, rather than Q1; the operational planning perspective is higher prioritized than real-time monitoring of current status. Therefore, the YCS-demonstrator aim at supporting traffic control at handling Q2 and Q3. Future development of YCS will also include Q1 by, e.g., connecting to live data sources like the TMS. Nevertheless, as described in this report, YCS is already prepared for being connected to live data sources.

2.2 YCS and yard automation

In this section, we comment on how the demonstrator YCS connects to the ongoing work related to automation and in particular yard automation.

Automation of train traffic has been an important topic in Shift2Rail and will continue to be so in Shift 2 Europe’s Rail (EU-RAIL). In EU-RAIL-project FP5 TRANS4M-R, automatic yards is one of the studied topics (TRANS4M-R, 2023). While ATO refers to Automatic Train Operation, i.e., taking a train on the line from A to B, AWO refers “Automatic Wagon Operation”3, i.e., the activities on yards related to the formation of trains. For freight trains, AWO includes all operations on wagons that are performed on the yards, e.g.:

• Coupling and decoupling of locomotives
• Coupling and decoupling of wagons
• Movements of wagons within the yard and over the hump
• Sorting
• Brake tests

3 Also denoted Automatic Shunting Operations, ASO.
It can be noted that the arrival and departure yards are the meeting and handover points of the two automatic systems, ATO and AWO. This means that also in a future with ATO and AWO, the arrival and departure yards will be a special type of meeting point between different systems. The arrival/departure yard is not only meeting point between two automatic systems, but also handover between two parts of the network management control: the train management and the wagon management. This means that if only one of the systems ATO and AWO are automated, the arrival/departure yard will be handover point between manual and automatic operations. Further, as different service providers may operate on the arrival/departure yards, e.g., terminal operators, their level of automation must also be considered. As the arrival/departure yards will remain a resource that is shared between different actors, and potentially also different levels of automation, the track allocation will be an important decision. Automation may reduce the time it takes to complete certain tasks and may also enable more tasks being performed in parallel (as yard staff availability is no longer a limiting factor), but it will not eliminate the need for trains and wagons to be allocated to tracks. Nor will it solve the coordination problems that arise when multiple actors want to use the same resource. In fact, partial automation may even make the coordination problem harder to solve as it is not only the coordination or actors but also of different levels of automation that needs to be dealt with. As the scope of YCS is track allocation and coordination among actors, it will be very relevant also in a future setting with automated operations in the railway system.

YCS will not only remain important but may also be part of the automation advancement. In the current version of YCS, there is no automation but from automatic planning rule fulfilment, rule conflict detection and the previously mentioned preparations connecting YCS to live data sources. However, in EU-RAIL FP1 WP 4 the possibilities for including more automatic planning support will be investigated. As already reported by the OptiYard project though, it is very hard to capture all relevant planning aspects in a mathematical model and planners sometimes make decisions based on knowledge that can’t easily, or maybe at all, be digitalized (OptiYard, 2019). Therefore, the best planning tool may be one where planners remain in control of the planning but can use automatic planning functions to make the planning faster and better.

For an automated system, the planning and control of it is dependent on:

- Q1: What is the situation now?
- Q2: What is the preferred situation in the short-term future?
- Q3: How to transition from situation “now” to the preferred future?

These are the same “questions” as stated for traffic control in Section 2. The scope of the current version of the YCS demonstrator is mainly to gather and digitalize information regarding Q2 and support the decision making regarding Q3. The real-time monitoring that is related to Q1 is very important for a fully automated system, but with the current status at the yards, a tool that supports answering Q2 and Q3 are more prioritized since it can be of practical use faster, also
without full automation of the operations at a yard. YCS decision support will help the yard to stay in a stable and manageable mode by enabling them to discover and solve problems in a proactive way before they occur.
3 Demonstration of a Yard Coordination System

One of the major tasks in WP2 is to develop a demonstrator for coordinated planning of activities at a marshalling yard. The scope is the combined arrival and departure yard (A/D-yard) in Malmö, Sweden. The characteristics and challenges of this yard are described in Deliverable D2.1 (FR8RAIL III, 2021) and a specification of the YCS is provided in Deliverable D2.2 (FR8RAIL III, 2022). RISE has been responsible for specifying and evaluating the demonstrator, and Indra for implementing it. Researchers from Swedish National Road and Transport Research Institute also contributed during the evaluation work. For the sake of completeness, this chapter starts off with a brief introduction of Malmö marshalling Yard and the concerned actors, as well as relevant lessons learned from Open Call projects. Then follows an overview of the YCS including its architecture and plans for connecting to a TMS. The TRL6 demonstration with actors is presented in chapter 3.4, including results and conclusions from the demonstration are also included. Experiments with automatic data updates are presented in chapter 3.5. Finally, conclusions and next steps for YCS are outlined.

The YCS demonstrator is also presented in a short video, available on youtube.  

3.1 Introduction to Malmö marshalling yard and relevant actors

The combined A/D-yard in Malmö (marked in yellow in Figure 1) is interesting as more than one actor perform shunting activities and require track capacity there. It’s also a significant bottleneck for the whole marshalling yard. In fact, one company performs marshalling operations and is responsible for the activities in the classification bowl (marked in green) while another company operates the multi-modal terminal (marked in blue). There are also other actors, e.g., companies that perform shunting to and from the harbour, but they are not considered in the demonstrator as they require access to the A/D-yard more rarely.

In this report, the actor responsible for the marshalling operations in the classification bowl is called the yard manager (YM), while the actor responsible for shunting to and from the multi-modal terminal is called the Terminal Manager (TM). The YM and TM are companies that are completely separate from the infrastructure manager (IM) 5. The actor responsible for allocating the tracks in the A/D-yard is called the line manager (LM), and they work for the IM. Note that

---

4 Demonstration of enhanced + integrated line- and yard planning and possibilities for implementation: https://youtu.be/LR_QJG3OvXU?feature=shared

5 This was the case at Malmö Marshalling Yard during the duration of the project. However, this is not always the case and the YM may, e.g., be part of the IM. The YCS will be useful also if the YM is not a separate organization. However, if the YM belongs to the IM some of the LM rights may be transferred to the YM.
while we call the actor a line manager, this person is actually only responsible for controlling and dispatching movements on the A/D-yard and its adjacent tracks. Calling this person a “line manager” may seem unnecessarily confusing, but the A/D-yard is simply one of many control areas that make up the entire network to be controlled. The control areas rotate among the line managers, and the person controlling the A/D-yard today may have controlled a line yesterday, and his/her title is not changed just because he/she happens to be allocated to a control area consisting of only a yard. Further, while the YCS LM is only responsible for controlling the yard, he/she cooperates with the line managers controlling the lines adjacent to Malmö marshalling yard and will consider the line-situation whenever making decisions on, e.g., suggesting changed departure times. That is, the YCS LM does have a responsibility to consider the line, even though he/she only controls the arrival/departure yard.

There is no standard way of operating a marshalling yard in Europe, and in OptiYard D5.1 four different basic models are described: Marshalling Yard operated by RU, Marshalling yard operated by IM, Marshalling Yard operated by RU and IM, and Marshalling Yard operated by a special operator (called MY). The three first basic models are defined using a table showing which actor that is responsible for which operations. None of the basic modes correspond to the one modelled in YCS, although the closest one is the one where the marshalling yard is operated by a special operator. To help explain the relationship between the actors considered in FR8RailIII we include the same table as used in OptiYard but filled in to reflect the YCS situation. Note that while YCS may be most needed when there are different actors, it would still be useful if, e.g., the IM is the YM. In this case, the people working as IM- YM would most likely still only be responsible for the classification bowl and would have to cooperate with the IM-LM controlling the arrival/departure yard. Also, the TM would still be external.
Demonstration of enhanced and integrated line- and yard planning and possibilities for implementation

Table 1: Roles and responsibilities in YCS.

<table>
<thead>
<tr>
<th>Item</th>
<th>MY (YM/TM)</th>
<th>RU</th>
<th>IM (LM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permission for train arrival</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Technical (and commerce*) inspection</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decoupling</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shunting + shunting engine</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Hump service**</td>
<td>X***</td>
<td>X***</td>
<td></td>
</tr>
<tr>
<td>Coupling</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technical (and commerce*) inspection</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Train documentation and permission to depart****</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Permission for train departure</td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

*In the FR8Rail3 project the actors talked about an “arrival inspection” rather than separate technical and commercial inspections.
**Terminal service for terminals.
***While the MY is responsible to performing the shunting, and planning how cars should be sorted/containers transferred, the shunting routes often go on both the IM and MY infrastructure, which means that they both have to do movement control.
****When the train departs it’s the RU that has to submit all the required documentation. But the MY can help generate it.

The maintenance contractor does not have a separate actor/role in YCS for two reasons; firstly, the LM, YM and TM are much more central with respect to coordination of the daily operations at the yard, and secondly, the maintenance work on the yard – with respect to track access – is coordinated and planned by the LM. However, YCS includes a forth role called “guest”. Users of the guest role can see all data and all planning without being able to change any data. In fact, this role can be used by both maintenance contractors and yard personnel to provide situation awareness about the yard operations.

The most central actor is the LM, who belong to the traffic control centre of the Infrastructure Manager. The LM is a crucial link between yard and line planning. The timeframe considered by the LM is the current day of operation, with a special focus from “now” to a few hours into the future. In fact, it is an outspoken strategy of Trafikverket network management to move from “controlling by execution” to “controlling by planning”, i.e., to become more proactive and less reactive in the whole approach to the network management. That is to say, Trafikverket want to put emphasis on short-term planning (sometimes also very short-term) instead of reacting to the real-time events. YCS contributes to the concept of “controlling by planning” by making it possible for the actors to communicate intentions and plans before actually going to action. Nevertheless, it is of course important that a tool like YCS reflect the “now” status of the yard.
and traffic so that it supports good situation awareness. The situation awareness can both come from automatic updated data streams and from users updating the data in the system. The YCS demonstrator is prepared for using updated real-time data streams for timetable data. However, because of Trafikverket IT solution requirements for sharing operational data it was not possible to connect YCS to the real-time data streams in this project. Later in this chapter, we illustrate how YCS can handle updated timetable data (which could originate from such a live data stream). In the demonstration, though, we use a static data import which is updated by the users and game leader (cf. chapter 3.4) to reflect traffic and operation developments.

Figure 1 shows Malmö marshalling yard and which tracks the different actors are responsible for. A schematic view of the YCS is also included.

![Figure 1: Schematic overview of the Yard Coordination System (YCS) and the actors in Malmö](Background image www.openrailwaymap.org)

### 3.2 Open calls related to YCS

The development of YCS in FR8Rail3 builds upon the results from ARCC and Fr8Rail2. However, the open call projects SMART and OptiYard also provide knowledge relevant to YCS. Both projects will be briefly introduced in this sub-chapter, as will the lessons learned that are relevant for YCS. Finally, some important differences between the projects are pointed out.

The SMART project’s working stream for development of a real-time marshalling yard management system focused on providing “optimization of available resources and planning of marshalling operations in order to decrease overall transport time and costs associated with cargo handling.” The workstream developed a web-based tool. The characteristics of a marshalling yard can be entered, and its layout and current placement of wagons can be
visualised. The tool can determine if a deviation effects any yard operations, and if so, provide different alternatives for re-planning. The SMART deliverables can be found here: https://projects.shift2rail.org/s2r_ip5_n.aspx?p=SMART.

Lessons learned from SMART are:

- **D6.1 (Smart, 2017)** includes an overview of commercial tools for supervision and management of marshalling yards. None of the tools focus on the type of collaborative planning that YCS is aimed at supporting. Rather, most of the tools are designed for helping yard operators keep track of wagons and other yard resources.

- **D4.1 (Smart, 2019)** presents the ways-of-working in the marshalling yard Niš-Ranžirna in Serbia and Karnobat in Bulgaria. The tasks carried out at the yards differ, but they still follow the same type of structure (train arrive -> arrival inspection and preparation for shunting -> shunting -> collecting and compounding cars to new trains -> departure preparation and inspection -> departure). The high-level modelling in YCS can accommodate both ways-of-working. Also, difficulty of synchronization following the vertical separation between the FRU and IM is mentioned as a problem for the Niš yard.

- The different ways-of-working at different yards is visible also in the data collection in **D5.3 (Smart, 2019)** as not all yards can fill in all data fields. Additionally, the challenges faced by yards differ. For example, the yard in Niš is oversized and has no track capacity problems, while track capacity is one of the main problems in Swedish marshalling yards. Finally, none of the sorting strategies implemented in the SMART RTYM system correspond to the one used in Swedish yards.

- **D5.3 (Smart, 2019)** presents data from different marshalling yards, including “level of automation of switches and communication”. While most yards have stated that the level of automation is “high”, there is one where the level of automation is “middle”. This indicates that the level of automation is not the same at all European marshalling yards, and that some yards have a long way to go to become fully automated yards.

- The SMART RTYM system assumes “operational time” rather than “real time”. All use-cases are tested based on manual data entry, and the system is only updated when a deviation trigger occurs. In all use-cases, this deviation trigger is entered manually although automatic updating should also be possible according to the report.

The **OptiYard** project developed an optimization module for planning yard resources (scheduling of tracks, shunting locomotive, and staff). They also implement a microscopic simulation of the yard and surrounding network, and connect the optimization module to this simulation. To the best of the authors knowledge, this is the first time a closed loop between a yard optimization module and a microscopic simulation has been developed. A connection between the optimization module and the simulation is implemented, and in **D2.1 (OptiYard, 2018)** a thorough overview of different data standards and existing data
Lessons learned from OptiYard are:

- In D2.1 (OptiYard, 2018) processes at a generic marshalling yard are described. The processes described are very similar to the “control by execution”-strategy currently operated in Sweden. This indicates that the lack of planning is a generic problem.

- As the OptiYard optimization module is connected to a yard simulation tool, messages have been defined for relaying all information needed by the simulation tool. An overview and analysis of different standards for data exchange within the railway industry is also provided in D2.1 (OptiYard, 2018). They also conclude that TAF TSI is an important tool for freight trains and should play a central role in the project.

- In D5.1 (OptiYard, 2018) different ways of organising the operation of marshalling yards are presented. The division of labour differs depending on whether it’s the IM, the RU or a special service provider that operate the marshalling yard. None of the cases outlined in the report corresponds to the Swedish situation.

- Some of the KPIs used in D6.3 (OptiYard, 2019) could be used to measure the effect of YCS. In particular, the KPIs related to operational reliability, resilience and quality are relevant for YCS, as is the sum of distance travelled by shunting locomotives on the A/D yard.

- In D6.2 (OptiYard, 2019) the optimized plans are compared with field data. The optimization does not seem to improve the operations, which at least partly can be explained by the fact that the planners in the field considered more future events than what was included in the optimization model’s horizon. Also, the field resources were used more flexibly than what had been formalized in the mathematical model. In their “Lessons Learned” section, the authors call these types of challenges “the difficulty of the problem definition”.

It is important to note that the projects SMART, OptiYard and FR8RailIII have made different assumptions when it comes to the actors’ responsibilities. Notably, the starting point of YCS is the fact that the actors come from different organisations and have different responsibilities and rights. For example, the service provider on the terminal should not have to make its internal planning public, but should still have to communicate information that is required to ensure efficient use of the A/D-yard tracks. Further, SMART and OptiYard take a more comprehensive
view of the planning, and plan, e.g., yard staff (which is considered the YM/TMs responsibility in YCS) as well as track allocations (which is considered the IMs responsibility in YCS). The table below show how the YCS roles correspond to the roles defined in SMART and OptiYard. For a more thorough discussion on the roles in YCS, see ARCC D2.2 (ARCC, 2018).
Table 2: Actors/roles in YCS, SMART and OptiYard.

<table>
<thead>
<tr>
<th>Role, Organisation</th>
<th>Responsibility</th>
<th>YCS</th>
<th>SMART*</th>
<th>OptiYard**</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Manager (LM), IM</td>
<td>The line manager is the IM traffic controller responsible for the A/D-yard. They sit in the IM control centre and can easily talk with the traffic controllers of adjacent lines. The LM decides, e.g., when trains are allowed to arrive and depart, and allocates A/D-yard tracks. Wagons can not be shunted to the A/D-yard without consent from the LM.</td>
<td>1. Make track reservations on the A/D-yard. 2. Update arrival/departure times. Also, 3. Update the time when the driver will arrive to a departing train. This should really be the responsibility of the freight rail undertaking, but there is no FRU role in this initial version of YCS.</td>
<td>Dispatcher, Unclear</td>
<td>Infrastructure Manager, IM</td>
</tr>
<tr>
<td>Yard Manager (YM), Service provider</td>
<td>The Yard Manager is the marshalling service provider in the marshalling yard. They are responsible for planning the yard staff and can request access to A/D-yard tracks from the LM. They are also responsible for ensuring that “their” wagons are moved to/from the A/D-yard in a timely manner.</td>
<td>1. Enter track reservation needs for activities relating to arriving trains, departing trains and other shunting activities. 2. Enter how much time the driver will need for departure preparations (as this depends on whether, e.g., a brake test has already been performed or not).</td>
<td>Dispatcher, Service provider</td>
<td>Infrastructure Manager, IM</td>
</tr>
</tbody>
</table>
| Terminal Manager (TM), Service provider | The Terminal Manager is the multimodal terminal service provider. They are responsible for planning the yard staff working for the multi-modal terminal and can request access to A/D-yard tracks from the LM. They are also responsible for ensuring that “their” wagons are moved to/from the A/D-yard in a timely manner. | 1. Enter track reservation needs for activities relating to arriving trains, departing trains and other shunting activities.  
2. Enter how much time the driver will need for departure preparations (as this depends on whether, e.g., a brake test has already been performed or not). | Dispatcher, Service provider | Prioritise yard operations.  
Coordinate yard staff. | N/A |
3.3 Description of YCS

The purpose of the YCS demonstration is to assess if a tool like the YCS could support pro-active and collaborative planning and improve the coordination of activities in a multi-actor marshalling yard, and thereby also reduce the number of resource conflicts and subsequent delays. The planning in YCS and its visualization should also facilitate communication between actors.

At large, the YCS consists of:

- a database for storing planned activities,
- a user-interface,
- a set of rules for identifying resource conflicts, default values, actor access and control rights, and so on.

3.3.1 User interface and user rights

YCS is a web-based demonstration tool, and the first view when reaching the YCS webpage is a log-in page. Each user enters the login and password. There can be different users with the same role logged in at the same time. Once a user has logged in, they are taken to a list-view where data relevant for the role of the user is listed (see Figure 2a). The user can also choose to look at a Gantt-chart view (Figure 2b) where the A/D-yard plan is visible. In the Gantt-view, time is on the x-axis and tracks on the y-axis. The Gantt-chart view contains several symbols. The symbols are explained in Table 3. For details regarding the data in YCS, see (FR8RAIL III, 2022).
Figure 2: Screenshots of the list view (a) and the Gantt view (b)

The user-interface is rather similar for the different actors (since they are interested in the same kind of information) but the functionality differs. Most notably, there is data integrity between the actors and an actor can only change data that he/she is responsible for. In the user interface, the trains and activities that are not connected to the actor (and that he/she cannot change) will
be semi-transparent. We call these semi-transparent trains/activities “background events” for this actor. The YM and TM communicate “track requirements”, i.e., when they would like to access a track. The LM is responsible for making “track reservations”, i.e., deciding which track that should be used for which activity and how long that activity can take. If there are track requirements that fall outside of their track reservations, the system indicates this as a conflict. There is also a conflict if a train is planned on a track that for some reason is not feasible for that train. This could, e.g., be because the train is longer than the reserved track.

Table 3: Explanation of the symbols used in the Gantt chart view.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrival (شرع) of train 36750.</td>
<td>The yellow-green frame indicates that the actor needs to add data to the train. If the train has all data required the yellow-green frame disappears. The train then has complete data. The colour of the train indicates which service the train’s waggons require, i.e., if waggons should be sent to the classification bowl, to the multi-modal terminal or not moved at all. Green = YM, blue = TM, black = both YM and TM, pink = neither YM nor TM (e.g., the train stops for a driver change).</td>
</tr>
<tr>
<td>Departure (شرك) of train 65254.</td>
<td>The colour of the train indicates which service the train’s waggons require, i.e., if waggons should be sent to the classification bowl, to the multi-modal terminal or not moved at all. Green = YM, blue = TM, black = both YM and TM, pink = neither YM nor TM (e.g., the train stops for a driver change).</td>
</tr>
<tr>
<td>A through train (i.e., an arriving train that continues further).</td>
<td></td>
</tr>
<tr>
<td>A train arrival with waggons for the YM. Track reservation interval from the LM (box with solid black lines). Track requirement from the YM (semi-transparent red box on the top line of the track reservation box).</td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td><img src="image1.png" alt="Symbol" /></td>
<td>A train arrival with wagons for the YM and the LM. Track reservation interval from the LM (box with solid black lines). Track requirement from the YM (semi-transparent red box on the top line of the track reservation box). Track requirement from the TM (semi-transparent grey box on the bottom line of the track reservation box).</td>
</tr>
<tr>
<td><img src="image2.png" alt="Symbol" /></td>
<td>A train that departs from the A/D yard. The box with black solid lines is the track reservation interval allocated by the LM. The semi-transparent red box on the top line of the track reservation box indicated the time interval during which the marshalling yard staff move the wagons to the A/D yard (i.e., the track requirement from the YM). The diamond with blue lines is the driver arrival time and the blue box is the time the driver needs to finish any final preparations. Note that both the driver and the cars must be on the A/D yard before the final preparations can begin.</td>
</tr>
<tr>
<td><img src="image3.png" alt="Symbol" /></td>
<td>A train that departs from the classification bowl. The box with black solid lines is the track reservation interval allocated by the LM. The red semi-transparent diamond is when the wagons are ready. The diamond with blue lines is the driver arrival time. The blue box is time needed for any final preparations after the wagons and driver are in place. The semi-transparent red arrow is when the train rolls out of the classification bowl.</td>
</tr>
<tr>
<td><img src="image4.png" alt="Symbol" /></td>
<td>Trains that are not relevant for an actor are dimmed in that actor’s view. This is a screen-shot from the YM view, where a TM train is dimmed out.</td>
</tr>
<tr>
<td>Symbol</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td><img src="sh1.png" alt="Shunting Request" /></td>
<td>A shunting request (▬) from the YM. The black arrow towards the track line from above indicates that the wagons are moved to the track by the YM. The black arrow from the track line to above indicates that the wagons are moved from the track by the YM. The red semi-transparent box is the YM track requirement. The box with dashed lines is the track reservation interval allocated by the LM (which in this case is much longer than the requested time).</td>
</tr>
<tr>
<td><img src="YMtoLM.png" alt="Moving Wagons" /></td>
<td>A shunting request for wagons to be moved from the YM to the LM. The black arrow towards the track line from above indicates that wagons are moved to the track by the YM. The black arrow from the track line to below indicates that the wagons are moved from the track by the TM. The red semi-transparent box above the track line is the YM track requirement. The grey semi-transparent box below the track line is the YM track requirement. The yellow frame means that the LM has not yet added any track reservation interval.</td>
</tr>
<tr>
<td><img src="Reserva1.png" alt="Track Reservation" /></td>
<td>A track reservation (▬) added by the LM. The track reservation can be for, e.g., parking or maintenance work.</td>
</tr>
<tr>
<td><img src="4903.png" alt="Conflicts" /></td>
<td>Conflicts are marked with red frames. The rules for conflict detection have been reported in Deliverable D2.2 (FR8RAIL III, 2022).</td>
</tr>
<tr>
<td><img src="44203.png" alt="Background Train" /></td>
<td>A background train, i.e., a train that is not connected to the actor logged in.</td>
</tr>
</tbody>
</table>

By selecting an element in the interface (train or maintenance activity), the user is able to see details of the information related to this element. All users can see all data related to the
element, but a user can only modify data that he/she has permission to modify (see Figure 3). More information is available in Deliverable 2.2 (FR8RAIL III, 2022).

![Figure 3: Interface for detailed element data. Blue fields show current data, and white fields are for modifying the data. As can be seen, not all data can be changed by this user.](image)

In relation to real-time data management, the data handling in the interface it is differentiated between:

- **Scheduled information**: it includes scheduled arrival/departure time and scheduled track (as input from timetable). It is provided by the RailML in the initial plan and cannot be changed by any actor.
- **Target information**: it includes target arrival/departure time and target track, updated according to latest information. It can be changed by Line Manager or by updated data imports, representing the real-time data flow.

By default, the target arrival/departure time and target track are equal to the scheduled arrival/departure time and scheduled track. In case a modification is performed, the operators can see it by comparing the values in the scheduled and target fields. In the section 3.5, we illustrate the functionality of automatically updated data.

### 3.3.2 Architecture of YCS

The YCS is a web-based system developed to allow CRUD (Create, Read, Update and Delete) operations through API-REST architecture style over HTTPs protocol. This means that the system is developed to be connected through internet and allows dynamic information exchange...
between the involved actors in the YCS.

(X2RAIL-4, Submitted but not yet published.) (X2RAIL-4, 2022).

Currently, YCS is an isolated environment from TMS that mainly focuses on the yard management. For testing the system, the planning of the different trains is manually updated through a RailML-file, which is obtained from the initial plan of the TMS. Below is the high-level architecture for updating the plan.

![Diagram](image)

**Figure 4: Repository architecture for a specific node.**

In the repository all trains at the A/D-yard train in the initial plan are stored, obtained from TMS through a RailML-export. This repository allows the users to update the operation times for performing their activities without affecting or taking into account the scheduled plan, as the project focusses on improving the operational times in the yard, allowing to identify conflicts during operation.

3.3.3 Automation and connection to TMS

(X2RAIL-4, 2022).

The current YCS uses an interoperable protocol that respects the TAF TSI regulation. TAF TSI is suitable because it describes the communication between marshalling and reservation systems that are in the core of the demonstration. It can be integrated within the Integration Layer, which means that the TAF TSI data models can be used by other systems (and data provided by
other applications that are available). By this architecture, a publish/subscription-notification-of-changes-interface could be implemented in order to automatically receive changes. The TAF TSI messages are essential to enable the demonstrator to new systems that demand information such as the Estimated Time of Arrival.

The Common Interface component in the architecture delivers early TAF TSI benefits by utilizing existing systems, IT platforms and communication processes whilst allowing scalability. The Common Interface architecture allows the translation of internal data to and from TAF TSI messages and the management and transmission of the messages according to the requirements of the TSI between any actor, independent of the internal systems or business processes in use by that actor.

![Diagram of TAF TSI as interoperable protocol in the IL](image)

**Figure 5: TAF TSI as interoperable protocol in the IL**

Focusing on YCS, the following architecture needs to be deployed.
The architecture above shows three independent modules that can perform the optimal connection to TMS and, thereby, provide live update of train data:

- **Node Management System (NMS):** it is the module deployed in AWS where the YCS is located. It is completely adapted for each node and needs to be configured depending on the characteristics of the yard (number of tracks, length, connections to classification and/or terminal yards, track connected to main line,…). It is the module explained in previous deliverables and sections, as it allows to manage the operations in each yard.

- **Train Node Management System (TNCM):** It is the gateway that connects the YCS to the TMS. It shall be developed in AWS or any other platform that allows connection to internet and shall allow CRUD operations and publish/subscribe capacity. Depending on the rules, the functionalities can be adapted. It is continuously comparing trains IDs between YCS and TMS. The standard requirements and data flow are explained below.

- **Traffic Management System (TMS):** It is the system that manages the trains by the IM. It is where real-time plan is available, and needs to be connected to the gateway. It provides the updated timetable for each train and can receive modifications (following the acceptance protocol explained in Chapter 3.3.3.1). In addition, it can send commands for creating/deleting trains to the gateway.

To have a clear idea of the use of TAF TSI messages, some use cases are detailed. These are related to the forecasted times in the train related messages. The messages are related to the arrival/departure of a train at a certain point, which may be a handover point, interchange point, the train destination or another reporting point. These are Train Estimated Times of Arrivals...
(TETA) and Train Estimated Times of Departures (TETD)

<table>
<thead>
<tr>
<th>Use case</th>
<th>Train Estimated Time of Arrival (TETA) originated in the TMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short description</td>
<td>Train Estimated Time of Arrival (TETA) to the track based on targets timetable changes. These changes are due to disruption events and are transmitted to the YCS.</td>
</tr>
<tr>
<td>Input data</td>
<td>TAF TSI messages following the schemes:</td>
</tr>
<tr>
<td></td>
<td>element ArrivalTrackAtLocation</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;ArrivalTrackAtLocation&quot; type=&quot;LocationIdent&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:annotation</a></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:documentation</a> Identifies the track of the arrival of a train at a reporting point. This is indicated in the LocationSubsidiaryCode in conjunction with the LocationPrimaryCode.</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:documentation&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:annotation&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:element&gt;</td>
</tr>
<tr>
<td></td>
<td>and EstimatedArrivalTimeAtLocation</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;EstimatedArrivalTimeAtLocation&quot; type=&quot;xs:dateTime&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:annotation</a></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:documentation</a> The estimated arrival date and time at the defined location&lt;/xs:documentation&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:annotation&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:element&gt;</td>
</tr>
<tr>
<td>Output data</td>
<td>Message acknowledgement</td>
</tr>
<tr>
<td>Test procedure</td>
<td>TMS publishes in the integration layer (IL) changes in the train targets timetable due to disruption events. YMS is subscribed to this topic and update the Train Estimated Time of Arrival information. This information in YMS is shared among the YMS users/actors through the interface.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case</th>
<th>Train Estimated Time of Departure (TETD) originated in the TMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short description</td>
<td>Train Estimated Time of Arrival (TETD) to the track based on targets timetable changes. These changes are due to disruption events and are transmitted to the YCS.</td>
</tr>
<tr>
<td>Input data</td>
<td>TAF TSI messages following the schemes:</td>
</tr>
</tbody>
</table>
Demonstration of enhanced and integrated line- and yard planning and possibilities for implementation

<table>
<thead>
<tr>
<th>element DepartureTrackAtLocation</th>
<th>&lt;xs:element name=&quot;DepartureTrackAtLocation&quot; type=&quot;LocationIdent&quot;&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="">xs:annotation</a></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:documentation</a>Identifies the track of the departure of a train at a reporting point. This is indicated in the LocationSubsidiaryCode in conjunction with the LocationPrimaryCode.</td>
</tr>
<tr>
<td>and EstimatedDepartureTimeAtLocation</td>
<td>&lt;xs:element name=&quot;EstimatedDepartureTimeAtLocation&quot; type=&quot;xs:dateTime&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:annotation</a></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:documentation</a>The estimated departure date and time at the defined location&lt;/xs:documentation&gt;</td>
</tr>
</tbody>
</table>

Output data | Message acknowledgement
---|---
Test procedure | TMS publishes in the integration layer (IL) changes in the train targets timetable due to disruption events. YMS is subscribed to this topic and update the Train Estimated Time of Departure information. This information in YMS is shared among the YMS users/actors through the interface.

3.3.3.1 Train Node Management System

In this section the gateway between the TMS and YCS, the TNCM, is explained in more detail. TNCM is the functional interface between the IL (which is in the TMS) and the YCS, being the system that establishes a connection between both systems. The TNCM shall not store any information.

Each time the YCS is initiated, e.g., due to a failure, the TNCM shall automatically perform the following steps:

- Connect to YCS and obtain the internal configuration parameters:
  - YCS node to be connected (in this case, Malmö)
Operation days to work: in the current version there are 6 days, which is the amount of data provided in the RailML-file. By connecting to the TMS, the whole year (or even more) may be available.

- Connect to IL (TMS) and obtain the current plan which involves the affected trains in YCS
- Send to YCS the obtained trains’ plans
- Subscribe to changes in the TMS RTTP (Real Time Timetable Plan)
- Maintain the connection with the YCS for requests

The high-level diagram for initialize the YCS is shown below:

![Diagram](image)

**Figure 7: High-Level diagram of initialization of the connection.**

Once these parameters are configured, the TNCM will dynamically obtain/provide the following information through the YCS API:

- The TNCM shall provide to YCS each update in the target A/D times of the trains which are involved in the corresponding node. This creation of trains shall come from TMS.
- The TNCM shall provide to the YCS each cancelled train for trains that have connection to the corresponding node. This cancellation shall come from TMS.
The TNCM shall provide to the YCS each new train for trains that have connection to the corresponding node and runs in the operation days established.

The TNCM shall receive from YCS changes in the A/D trains’ target times. These changes in timetable shall be send to the TMS as forecasted times. These changes shall not have any conflict in the YCS.

The TNCM shall, for each modification of target time from the YCS, ask the TMS to perform this change.

The TNCM shall, for each created train from TMS, order the YCS to create the train.

The TNCM shall, for each cancelled train from TMS, order the YCS to cancel the train.

Thereby, from these requirements on the connection between YCS and TMS the following can be concluded:

1. TMS has completely the rights for adding/deleting trains in RTTP. YCS is not able to perform this functionality as it is responsibility of the IM.
2. Through the TNCM, the YCS will have the updated plan. This means that if a train is added, the YCS will receive and add it in the plan.
3. Each modification of the RTTP in the TMS will be performed automatically in the YCS.
4. When the YCS perform a modification in the RTTPs, it shall be confirmed by the TMS before including this modification in the RTTP.

The data flow of these conclusions is explained in detail in the high level diagrams of section 3.5.

3.4 Demonstration workshop

The purpose of the workshop was to evaluate the YCS in a simulated but relevant environment, which a as far as possible had been made realistic by including real-world data, real operators as participants, operational scenarios reflecting standard days including realistic small and large deviations and disturbances, regular means of communication between the participants, fair situational awareness and premises that are plausible (including separation of the participants). The demonstration workshop took place in the RISE office in Lund and consisted of four sessions spread over two days. The aim of the workshop was to learn how intended users would use the system, and to get their input on the advantages and potential of using the tool, as well as on disadvantages and development needs. As the demonstration was made in a relevant environment, with real users using a mixture of real and realistic data, we judge the demonstration to be of TRL6.

The following three sub-chapters contain a detailed description of the workshop. The purpose of the thorough description is to provide the reader with background information on how to the YCS was evaluated. This will be useful when assessing the results and conclusions presented in
3.4.1 Participants

The workshop participants are listed in Table 4. The participants, who are the real operators (LM, YM1, TM) and potential users of such a system in real operations, acted as themselves during the demonstration, and used the YCS as they would have if this had been a real workday. The assistants (A1, A2, A3) were mainly there to provide technical support (if needed) and to observe how the tool was used. The researcher playing the “game leader” was responsible for advancing time, announcing train arrivals and train departures, and making calls to the LM that train drivers would make in a real-world situation.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Job</th>
<th>Role during workshop</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL</td>
<td>Researcher</td>
<td>Game leader</td>
</tr>
<tr>
<td>A1</td>
<td>Researcher</td>
<td>Line manager assistant</td>
</tr>
<tr>
<td>A2</td>
<td>Researcher</td>
<td>Yard manager assistant</td>
</tr>
<tr>
<td>A3</td>
<td>Project Leader</td>
<td>Terminal manager assistant</td>
</tr>
<tr>
<td>LM</td>
<td>Line manager</td>
<td>Line manager</td>
</tr>
<tr>
<td>YM1</td>
<td>Marshalling yard staff</td>
<td>Yard manager</td>
</tr>
<tr>
<td>YM2</td>
<td>Business developer at the company operating the marshalling yard</td>
<td>Observer</td>
</tr>
<tr>
<td>TM</td>
<td>Terminal director</td>
<td>Terminal manager</td>
</tr>
</tbody>
</table>

3.4.2 Preparation

All external participants (except YM2) had been introduced to the YCS before the workshop. YM2 was mainly there to observe and was not responsible for using the tool during the scenarios. The practitioners were asked if they needed some extra equipment or data to be able to simulate their part of the operations. All practitioners asked for pen and paper, but nothing else.

An Excel document with macros was developed to support the game leader and provide information (situational awareness) to the other actors. Each actor had their own sheet in the Excel document with information and control functions relevant for that actor. Overall, the Excel document (see Figure 8) contained:

1. A list of train arrivals and departures that the game leader should announce (marked in green).
2. A list of driver calls that the game leader should make (also marked in green).
3. An arrival yard simulation (marked in orange).
4. A clock keeping track of the simulated time (marked in blue).
The current simulated time was visible in all sheets but could only be controlled from the GL sheet. The simulated time moved forward in normal speed unless all actors had marked that they were ready for the time to be advanced to the next event. The LM, YM and TM could also specify specific time points that they wanted the time to be advanced to. Figure 8 shows the GL sheet. The GL controlled the time using the controls in the blue box and via the “ready markings” (x) next to the events in the event list in the green box. The orange box is the arrival yard simulation. The arrival yard simulation could be controlled from both the LM sheet (see Figure 9) and the GL sheet. The LM and GL could see which trains that were allocated to which tracks and could add and remove them. The YM and TM could only see if a track was occupied or not (see Figure 10 and Figure 11).

Figure 8: The game leader's sheet in the Excel document
Demonstration of enhanced and integrated line- and yard planning and possibilities for implementation

Figure 9: The line manager’s Excel sheet and YCS view

Figure 10: The yard manager’s Excel sheet and YCS view
The research team practiced running the simulations with the Excel document and the YCS during two separate occasions before the workshop. During these practice sessions it became clear that it would be useful if the current (simulated) time could be marked in the YCS. Since this could not be implemented in the YCS, post-it notes were provided during the workshop that could be used to mark the current time.

Four different workshop sessions were prepared. Two where the participants were in the same room, and two where the actors were in different rooms. To mimic the real situation, in which the actors work in different locations, they were placed in separate rooms. The sessions are described in more detail in the next section. Before the workshop, the participants were informed about the agenda, and were asked to fill in a data consent form. The agenda for the workshop days was as follows:

### Table 5: Agenda for demonstration workshop.

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:30 - 12:15 Lunch</td>
<td>08:30 - 08:45: Everybody arrives</td>
</tr>
<tr>
<td>12:15 - 12:30 Introduction</td>
<td>08:45 - 09:00 Introduction Day 2</td>
</tr>
<tr>
<td>12:30 - 13:30 Scenario 1</td>
<td>09:00 - 10:30 Scenario 3 and 4</td>
</tr>
<tr>
<td>13:30 - 14:00 Break</td>
<td>10:30 - 11:30: Evaluation Day 2</td>
</tr>
<tr>
<td>14:00 - 15:30 Scenario 2</td>
<td>11:30 - 12:15 Lunch</td>
</tr>
<tr>
<td>15:30 - 16:30 Evaluation Day 1</td>
<td></td>
</tr>
</tbody>
</table>
Two lists with evaluation questions were prepared. The first list was intended to be used after the first day and mainly focussed on whether the participants had understood how to use the YCS tool. The second list of questions was intended for the evaluation at the end of the second day and contained questions about the actors’ experiences of using the YCS.

3.4.3 The workshop days

During two half-days, practitioners took part in simulation sessions. The workshop started with an introduction of the YCS and the Excel document, followed by the first session that was intended to re-introduce the actors to the YCS functionalities and give them some time to figure out how YCS could support planning and communication. The overall structure of the scenarios (including type of deviations and disturbances) was discussed and agreed with the participants beforehand and deemed reasonable and relevant. The timetable of the trains, including arrivals and departures to Malmö, was a real timetable representing a week in October 2020, and the deviations of the arrivals and departures were based on samples from a realistic distribution of deviations.

During the first scenario one of the researchers announced different actions (e.g., “train 10003 arrives” or “You learn that train 1004 will be delayed by 40 minutes”) and the actors then had to change the plan accordingly. The participants were in the same room, so everyone had the same information, and it was easy to discuss and figure out how problems should be solved.

For the second scenario, the participants were put in different rooms and could only talk the way they normally do – by calling each other on the provided cell phones (see Figure 12). The simulation was a “normal” day. That is, the arrival time of trains varied but there were no major problems to overcome.

The evaluation session at the end of day one turned into more of a syncing session as there were some ambiguities about responsibilities and timings with respect to how the Excel document should be used (e.g., when should a train that is rolled into the classification bowl be removed from the A/D-yard track and should the LM or the YM remove it from the A/D-yard?). It was decided that the Excel document should be developed to the next day to let the YM/TM remove trains from the A/D-yard tracks. The plan was that the TM/YM should remove trains from the A/D-yard in the Excel document if they were responsible for shunting the cars away from the A/D-yard in reality.

The next day started with the third scenario. It was a distributed session where the actors were once again put in different rooms. However, the terminal manager and the yard manager were
put in the same room to make the session more interesting for the terminal manager (who generally had little to do during the simulations). The third session was a snow removal scenario, and all tracks but from two had to be closed for snow removal for an hour before they could be used. Also, there was only one snow removal team and therefore only one track at a time could be cleared of snow. The arrival time of trains followed the same distribution as the day before.

Last but not least, in the fourth scenario, the bridge to Denmark had to be closed for a few hours. For this session the participants were once again all in the same room and could discuss freely. Once this session was over, a group interview with all the participants was carried out. This final interview was also audio recorded for later analysis.

Figure 12: The participants were put in different rooms during the distributed sessions.

3.4.4 Data collection and processing

The final discussion was recorded, and the researchers took notes during the full two days. The research notes were compiled in a bullet point list right after the workshop. More insights were added to the bullet point list when listening to the recording of the final discussion a few days later. The document with the bullet point list was sent to the workshop participants and they were asked to check if they agreed with the statements in the bullet points. No participant provided any feed-back on the bullet point list.

3.4.5 Results of YCS-workshop

3.4.5.1 General experience of setting up and running the workshop

Running a workshop provides many learning opportunities and is a good way to involve practitioners and get their input. However, a fair amount of preparation is needed for the
workshop to run smoothly. For example, while the two practice runs carried out by the researchers greatly improved the workshop material, even more practice runs would have been beneficial, as would a practice run with the practitioners before the workshop. There were issues that had to be solved during the workshop, which took some time during the sessions intended for evaluation of the YCS during the first day. In fact, most of the first day was spent getting comfortable with the way the simulation was run and figuring out problems and responsibilities. In general, more preparation on which participant should be responsible for what would have been useful, especially during the sessions when everybody was in the same room. During the distributed sessions the participants naturally took on the responsibilities they have in real life, but when everyone was in the same room, and there were no clear time increments, the roles and division of responsibility became less clear. Further, a more thorough explanation of the differences between the distributed sessions and the common sessions would have been beneficial. Part of the confusion over who should be responsible for what might have been caused by the fact that a move from reactive execution to pro-active planning will require new ways of working, but part was probably caused by the fact that the simulation in itself was confusing.

When the participants had started to find their feet in the simulations, they spontaneously added extra events to the scenario. This type of practitioner driven adaptations made the scenario more realistic and allowed the participants to test events that they were interested in. This type of spontaneous additions would have been encouraged if we would have had more time.

In our case, all actors using the YCS had had lessons on how to use it before the workshop. This was good as no time had to be spent on learning the tool during the workshop. Also, the practitioners generally understood and learned how to use the tool faster than non-practitioners (e.g., researchers that were only involved during the evaluation phase of the project and therefore had to learn how to use YCS after it had been developed).

The question lists that had been prepared for the evaluation were not rigorously followed during the evaluation sessions. This was to avoid disrupting the discussions about the tool that emerged naturally, and also as some of them seemed quite outdated given the developments during the day. Rather, the lists, and in particular the list for the second day, were used as a check-list to ensure the discussions had covered all aspects.

3.4.5.2 Experience of using the YCS

The practitioners were very happy with using the YCS and wanted an operative tool to be developed as soon as possible. Some statements (translated from Swedish) about the tool are: "At a grass roots level, it is clear that this will be usable", "This is absolutely excellent as a
planning tool", "It is a good model for a very good future tool", "It increases transparency in the freight yard", "For those who work in production, you might gain a greater understanding of why you have to wait for 10 minutes... usually that information is missed out", "... and that you can proactively act instead of react... it's hard to act if you don't have information."

The YCS improves the situational awareness, both for planners and for the people working “on the tracks”. In fact, personnel actually performing the inspections and shunting (as opposed to working in the control tower) was identified as a new actor during the workshop. Currently, the yard personnel have limited information about current and planned yard activities. If the yard personnel do not know what is going on, or what is going to happen in the near future, it is very hard for them to problem-solve and make improvement suggestions during operations. Improved situational awareness would give everybody involved a better understanding of what is happening (e.g., knowing the answer to the question “why do I have to wait?”). This would not only enable the yard personnel to take part in the problem-solving but would also decrease the number of phone calls - both calls from the YM/TM to the LM, and YM internal phone calls between yard personnel and the planning and control personnel. Last but not least, it would be easier to do hand overs at the end of work shifts for everybody. There is a message function where the planners can leave notes, e.g., on why a certain train is planned on a certain track.

A better situational awareness would also enable more pro-active planning. In real life, the order of shunting movements is more or less first-come first-served. But given a better tool and earlier information sharing a more effective order can be planned. This will reduce the number of shunting movements, which once again means less phone calls, but also less wear and tear, and less disruption risk. The planning could also become more cooperative as the transparency between the organisations improve. For example, during the “bridge closure” scenario the LM was happy to let the YM put more trains on the arrival/departure yard than the YM had first thought. However, while the YCS enables more pro-active planning, this is not how the actors are used to work, and sometimes they forgot to look ahead in time and caused problems for themselves. There will be a learning curve when moving from reactive execution to pro-active planning, and there must be clear benefits of the pro-active planning for the staff to make the extra effort in terms of data entry and planning. Good training could also make the transition easier.

Another possibility identified by the practitioners was the analysis of past planning and execution data. Currently, there is no way to follow up the planning, and therefore it is hard to develop best practices. But if the YCS tool saves planning and execution data, the data could be analysed and best practices put in place. The YCS tool and simulations similar to the ones carried out in FR8Rail III WP2 could also be used to practice for large disruptions. Lastly, the practitioners thought that the improved and pro-active planning that the YCS supports
could give better arrival punctuality to customers and more reliable railway transportations – not least as more wagons can catch their intended trains. They also pointed out that better planning may open up for more trains to use the Malmö Marshalling Yard.

None of the practitioners thought that it was hard to use the YCS, and they thought that it would be useful both during normal operations and during large disruptions. They wanted to start testing the tool in small scale pilots as soon as possible. They also pointed out that a tool like YCS could be used for tactical planning.

While the practitioners were very happy with YCS and thought that with a few tweaks the tool could be used in small pilots, they also identified several development possibilities. These are presented in the next section. The most important development is semi-automatic updating of data (e.g., if the ETA changes this should be propagated to the YCS without the LM having to do anything, and the track allocation should automatically follow the ETA).

### 3.4.5.3 Development suggestions

The following developments were suggested at some point during the workshop:

1. **More developed view-only role** for staff that need to see the plans but not enter or change data, e.g., dispatch engine drivers, staff working with inspecting and preparing wagons and trains. There is a view-only log-in in the demonstrator, but the view-only roles were not analyzed when the demonstrator was specified.

2. More tasks need to be (semi-)automated, e.g., data entry, replanning in case of changed arrival time/departure times. The current default values worked well, but there is still too much manual entry required. In particular, track reservations are subordinate to arrival/departure times and should move with them. Similarly, it would be good if start and end time of reservations/requirements could be moved in one go (rather than having to first change the start time and then the end time).

3. The **time required** for arrival inspection and preparation depends on the **length of the train**. The default time requirement could therefore be calculated based on train length.

4. The system could **suggest solutions** to conflicts.

5. **Visualize also the current operational situation.** Functionality should also be developed for identifying and handling planned events that have changed or not occurred (e.g., if a train has not arrived as planned). Part of this solution may be the “To-do list” below.

6. **To-do list** – a place to put trains that the planner intend to plan later. This list can, e.g., be used to "save" trains that have not arrived on time and where there is no information on whether the train is late or cancelled, and for trains that have had to be moved from their planned track but have not yet been allocated to a new one. It should be possible to sort the To-do list based on original arrival/departure time. The YM also wants to be able to
add priorities to trains in the To-do list. Note that it is the YM that knows when trains will be able to depart in case of, for example, a major disruption.

7. The trains’ **originally scheduled times need to be recorded**. One way to accomplish this is to communicate the change in arrival/departure time in **+/- minutes relative to the original time** rather than overwriting the original arrival/departure time.

8. The drag-and-drop planning functionality is very good. However, the new times need to be visible when moving a planning object.

9. A **line indicating the present time**.

10. Put **track lengths** under the track names.

11. Include the **transport requirements** for shunting movements. In general, implement **more shunting rules**.

12. Add a **dangerous goods marker**. The tool is useful also without this marker, but the marker would provide valuable information.

13. The tool should be **touch-screen compatible**. The terminal staff already use touch-screens and the LMs also want to use them.

14. Functionality for **through trains** need to be designed and implemented. It should be possible to connect an arrival and a departure to a through train, and vice versa, decouple the arrival and departure of a through train.

15. Train **cancellation** functionalities need to be developed.

16. The trains **stop code** should be visible.

17. The **background events** need to be more visible. It’s good that they are not as visible as events relevant for the actor, but it should still be possible to read the information. In particular, track closures are of interest to the other actors and should not count as background events.

18. **Traceability**. In current practice, there is almost no possibility to trace planning decisions. This makes quality analyses hard, and hinders the development of best practices. It would be good if the planning and operations data were saved for later analysis.

19. In the (distant?) future, it would be good if **shunting clearance** could be requested through the system.

20. Extend the scope with certain **locomotive parking tracks, and locomotives** parked on these tracks.

21. When there is a large disruption, it might be good for the LM to “peek into” the classification bowl and the terminal to better coordinate the planning.

22. The list window was not used, and it is unclear if it serves any purpose.

---

6 These suggestions bring the tool closer to being an operations tool rather than a planning tool. Such functions should be appropriately analysed before they are implemented.
3.4.5.4 Risks

There are always risks when implementing new software. In this section we highlight some risks that were identified as particularly relevant for the YCS.

There is a risk that the development of a real YCS tool does not receive any funding – in particular if the motivation for implementation must be based on cost savings rather than quality improvements. There is also a risk that the resources put towards implementing the tool are not large enough, and that, e.g., extra resources are not allocated to the traffic management during the introduction of the tool. There was also a discussion on the importance of KPI follow up before and after the implementation to see if the intended quality improvements were realised. Examples of KPIs are fill rate, track utilization, disturbances and connection times. If possible, it would also be good to try to measure how the yard actors are affected by the YCS, e.g., measure time spent on planning, perceived situational awareness, perceived levels of cooperation and understanding for, and from, other actors.

During the workshop the risk of increased vulnerability was discussed. No traffic safety risks were identified, but a more thorough analysis during, e.g., pilot studies is necessary. The risk of competence loss in case planners rely too much on the tool was identified though, as was the risk that the planners started to rely on the view of the plan in the YCS as the real situation without double checking. This can become problematic if the situation in the YCS and the real situation differ. The risk of vulnerability in case of system down time was also discussed. However, if the system goes down the current planning process can be used instead, so this risk was not considered to be particularly important.

Another challenge is the move from reactive planning to proactive planning. This requires all actors to work differently – they must enter data and think about and discuss how to solve future resource conflicts. Also, someone must, at some point, initiate the work required to, e.g., change planned track allocations. This fact especially comes to play in already quite stressful situations resulting from events such as some sort of disruption. Even in such cases, the proactive planning should still be carried out, in order for the tool to reach its full potential. The extra work and responsibility will most likely only be accepted if the actors can see clear benefits from using the YCS and proactive planning, e.g., that there are fewer delays or that disruptions are easier to handle. The tool may also “gamify” the planning work, which can be rewarding in itself. The latter is particularly true for the Malmö marshalling yard, where the shunting can be performed in many different ways and good planning is considered a challenge. Last but not least, failure to gain user acceptance, and thereby enjoy updated data and plans, is a self-reinforcing vicious circle: poor tool usage -> bad data quality -> bad plans -> low incentive to use
YCS -> poor tool usage.

Yet another risk is that there are many “small” actors that have not been included in this project, actors that only occasionally perform tasks at the yard. There has to be a light-weight version of the system that these small actors can use, or some way for the actors to use the track resources without having to use the YCS system. The last risk may also be an opportunity, and that is that more people on the yard start having opinions about why certain planning decisions have been made. As everyone gets more planning information, they may also become more interested in understanding why the plan looks like it does and have more suggestions on how to do things. This may lead to an increased number of phone calls. However, it may also lead to better planning decisions, and/or a better understanding of each other’s requirements, abilities and objectives.

3.5 Automatic data update experiments

3.5.1 Use cases for TMS data exchange and High level diagrams

In this section it is explained the high level diagrams related to the data flow and permissions between the different systems and the involved actors for performing changes in the RTTP. These diagrams are based on the data flow followed in Figure 6.

During the usual workday, the YCS and the TNCM are configured to be subscribed to timetable changes. Once the systems have the initial timetable, the TNCM compares for each operation day if the main parameters (trains date and ID, Origin/destination in Malmö and timetable) are equal in TMS and YCS. In case these parameters are equal, there are not changes. On the other hand, in case these parameters are not equal, there might be the following possibilities:

- A new train is detected in TMS repository
- An existing train disappear from TMS repository
- A train timetable has a modification from TMS
- A train timetable has a modification from YCS (request and confirmation)

These four possibilities involve different operations performed by the systems and actors.

The following systems are involved in the diagrams:

1. Traffic Management System (TMS)
2. Train Node Coordination System (TNCM)
3. Node Management System (NMS)

In addition, the following actors are involved in the diagrams:

a. Infrastructure Manager Traffic Control Center (IM-TC)
b. Line Manager (LM)
The TMS is used by staff in the Infrastructure Traffic Control Center (IM-TC). There will be different people with different roles/responsibilities that work in the IM-TC\(^7\), but we will not specify which role in the IM-TC that is responsible for what but rather view the entire department as an actor. There is one exception to this rule though, and that is the TM controlling the arrival/departure yard, i.e. the YCS LM. The LM works in the IM-TC, but will be viewed as a separate actor. It’s also worth highlighted that only the LM has rights in YCS to modify target arrival/departure times.

3.5.1.1 Use Case 1: New train from TMS

Once the YCS is connected to the TMS through the TNCM, it is common that trains' timetables change according to disruption events. In addition, it is possible to include new trains not previously planned in order to increase capacity. For including a new train and represent it in YCS, this order shall come from the TMS as the responsible actor is the IM-TC.

The YCS is designed to be connected to TMS through the TNCM and being able to add this train to the target plan. It shall be highlighted that the YCS does not have permission in its configuration to deny this operation. This capability only corresponds to TMS, as it has all trains. As pre-configuration requirements, the system shall be initialised as explained in section 3.3.3.1, so that the current plan is available for YCS.

Below is the high-level diagram for updating the YCS with a new train:

---

\(^7\) For example, in Sweden there is a specific role for handling new trains. This person is however not responsible for controlling and dispatching the traffic in any control area. On the other hand, the responsibility of deciding whether a train can depart early/late falls on the traffic controllers of the control areas that the train will run through.
3.5.1.2 Use Case 2: Deleted train from TMS

Once the YCS is connected to the TMS through the TNCM, it is common that trains’ timetables change according to disruption events. In addition, it is possible to cancel trains previously planned as they have lost their place in the RTPP. For cancelling a train and represent this cancellation in YCS, this order shall come from the TMS as the responsible actor is the IM-TC. The YCS is designed to be connected to TMS through the TNCM and being able to cancel this train to the target plan. It shall be highlighted that the YCS does not have permission in its configuration to deny this operation. This capability only corresponds to TMS, as it has all trains. As pre-configuration requirements, the system shall be initialised as explained in section 3.3.3.1, so that the current plan is available for YCS.

Below is the high-level diagram for train cancellations in YCS:
3.5.1.3 Use Case 3: Update timetable from TMS

Once the YCS is connected to the TMS through the TNCM, it is usual that trains timetable change according to disruption events. In addition, due to these timetable modification it is possible that some trains have changes in their target timetable which are, in general, delays.

For modifying this train and represent this modification in YCS, this order shall come from the TMS, being the responsible person the IM. As in previously use cases, the YCS has not rights to deny this modifications, so that they will be automatically represented in the interface.

As pre-configuration requirements, the system shall be initialised as explained in section 1.2.1.2, so that the current plan is available for YCS.

Below is explained the high-level diagram for a modified train in TMS that affects YCS:
3.5.1.4 Use case 4: Update timetable from YCS

Once the YCS is connected to the TMS through the TNCM, it is common that trains’ timetables change according to disruption events in the yard operations, or as secondary delay events. In general, these modifications are related to timetable changes that represent delays. These modifications are provided by the LM through the YCS interface, but need to be accepted by the IM-TC before implementing the changes in the timetable. In addition, track changes in the A/D-yard do not need to be accepted by the IM-TC, as the person in charge of the yard is the LM.

Below is explained the high-level diagram for a modified train in YCS that affects TMS:
3.5.2 Experiments with manual data update

The YCS has been developed and deployed to manage and visualise real-time data exchange not only between the involved actors, but also (in future) with the TMS. The prototype is developed so that it is possible to make real-time updates of the information that comes from the TMS (that is, arrival/departure times for trains coming from the main line, trains ID, cancelations, new trains, etc.). This far, it has not been possible to connect YCS to a real TMS, but YCS is still prepared to handle such functionalities. The starting point is the initial import including the scheduled plan, and based on that, the IM-TC can update information in new RailML files that simulates new data realization coming from TMS system. This means that at each moment, the IM can update the plan so that it corresponds to the recent status of the current and future situation of the A/D-yard. In this section we illustrate the simulated real-time data flow and data updates in three different scenarios.

To test the functionalities for updated data, different scenarios are defined. The scenarios include modified data that represents the real-time data flow, and we show how YCS behaves and how the actors are affected. Different types of trains with different services
(arrival+multimodal, departure+classification, throughtrain+none, etc.) have been used to test data consistency.

In addition, the system is developed so that when an actor registers an operation, data associated to this operation is kept even if new data is provided. The only data that is overwritten is the arrival/departure times and the track assignments (and track assignments are only overwritten only in case the LM has not manually changed it).

The test scenarios were performed on 1\textsuperscript{st} February 2023, which thus represents “now” in the experiments. The initial imported RailML-file and the RailML updates cover the dates from 1\textsuperscript{st} February to 6\textsuperscript{th} February, so that the operations performed are in real time. The data updates are managed by a person using the guest role of YCS, who loads the updated RailML-files into YCS, and the other roles as LM, TM and YM are involved in the scenarios to complete their respective operations.

The starting point for the scenarios is the scheduled plan for the selected days. No operations on the data have yet been performed by any actor, so everything is in its original state as defined in the initial RailML file.

3.5.2.1 Scenario 1. New train added

In this scenario, a train arrival not included in the initial data is added, representing that the IM-TC has added another train that should be operated and scheduled. The test case emulates the addition of different trains by the TMS (by updating the RailML) in the scheduled plan. The updated data is received by YCS, which refresh the interface and illustrates the added (new) train.

A new train should have the following data (date of the train is included in the Scheduled Arrival/Departure time):

- Train number: train identification
- Track reference: track of Malmö (1-10 or 59)
- Service: Classification, Multimodal, Mixed or None (None by default)
- Scheduled Arrival/Departure time (type):
  - Scheduled Arrival time: only for arrival trains
  - Scheduled Departure time: only for departure trains
  - Scheduled Arrival/Departure times: for through trains
- Length: length of the train
- Dangerous good: if has or does not have dangerous goods (No by default)

Thereafter, YCS behaves as follows:
• The train is created on the track and time as determined in the received data.
• YCS keeps all data for operations of other trains. This means that when the update is received, the operations (data changes) already performed by the actors remains so that these changes are not reset or modified.

The initial interface for the test cases is shown in Figure 17.

Figure 17: Initial situation of YCS.

For the different test cases, the following trains will be added in the marked squares in the figure above:
• Train 6160: Departing train, has been handled by the YM. It will be added in the green square.
• Train 46256: Arriving train, to be handled by the TM. It will be added in the black square.
• Train 45517: Through train (with no connected service). It will be added in the blue square.

All test cases are explained before the conclusion/results of the scenario, see also Figure 18.

Test Case 1: Initial data before updating YCS with new trains
This test case illustrates the operations performed by the actors before obtaining a RailML updates. Figure 17 shows the initial status.
• Trains with complete data (not circled in yellow).
• There are conflicts (circled in red).
• There is a maintenance activity.
Test Case 2: New train on a track at a time when there is no other train
This test case shows an addition of a train 6160 on track 2 (in the green square in Figure 17) when there is no other train previously planned. In this test case, the train is correctly created, and it appears in the train pending scheduled table. Also, it appears circled in yellow, as no data has been manually added to it, i.e., it has not complete data.

Test Case 3: New train in conflict with another train that has no manually entered data
This test case shows an addition of train 45517 on track 1 (in the blue square in Figure 17) where there is a train previously planned but no data has been manually added to it. In this test case, the train is correctly created, and it appears in the train pending scheduled table. There is no conflict as the track reservation interval is not defined.

Test Case 4: New train in conflict with another train with complete data and a conflict
This test case shows the addition of train 64256 where there is another train previously planned, with an associated conflict on it (in the black square in Figure 17). In this test case, the new train is correctly created, and it appears in the train pending scheduled table. There is no conflict between trains as the track reservation interval of the new train is not defined, but the conflict of the initial train is kept.

Results of scenario 1
The final result in YCS of scenario 1 is illustrated in Figure 18.
As it can be seen in Figure 18, every requirement of the test cases is successfully achieved:

- Trains with previously added data keep their data (also trains with conflicts).
- Train 6160 (green square) is successfully created.
- Train 45517 (blue square) is successfully created over train 5124 without conflicts.
- Train 64256 (black square) is successfully created over train 41972, which has data and an already existing conflict which remains.
- New train are detected and shown in train pending schedule table (red square) (i.e., the list of trains that actors should add data for).

### 3.5.2.2 Scenario 2. Cancellation of trains

In this scenario, the cancellation of initially planned trains is illustrated, representing that the IM-TC has cancelled a train that will not arrive/depart from the yard. The test case emulates the cancellation of different trains in the TMS and the resulting updates of the RailML-files to YCS. The updates are received by YCS, which refresh the interface and shows the real-time situation of the yard. As starting point, the scenario is the showed in Figure 19 (final situation of scenario 1).
For cancelled trains, the YCS deletes all data related to the involved trains, including conflicts associated to them but keeps data associated to other trains. For the different test cases, the following trains will be cancelled in the squares painted in Figure 19:

- Train 42773: Departing multimodal train. All data has been added, and it has no conflicts (blue square).
- Train 4613: Arriving classification train. All data has been added and it has a conflict with train 44732 (black square).

All test cases are explained before the conclusion/results of the scenario.

**Test Case 1: Initial data before cancelling trains**

This test case shows the initial status before obtaining RailML-updates with train cancellations, see Figure 19.

- Trains with complete data (not circled in yellow).
- There are conflicts (circled in red).
- There is a maintenance activity

**Test Case 2: Cancellation of a train with complete data**

This test case shows the cancellation of a train 42773 with data associated to it that has no conflicts. In this test case, the train is correctly cancelled and disappear from YCS (blue square).
Test Case 3: Cancellation of a train with complete data and a conflict
This test case shows the cancellation of train 4613 with an associated conflict (black square), the conflict is between train 4613 and train 44732. In this test case, train 4613 is correctly removed, and the conflict disappear, and data for train 44732 is kept.

Results of scenario 2
The final scenario is shown in Figure 20.

Figure 20: Final status of scenario 2, after cancelling trains.

As it can be seen in Figure 20, every requirement of the test cases is successfully achieved:

- Trains with previously added data keep their data (also trains with conflicts).
- Train 42773 (blue square) is successfully cancelled.
- Train 4613 (black square) is successfully cancelled, the operations associated to it are removed and the conflict with train 44732 disappear.
- The conflict is removed from the conflicts table (green square).

3.5.2.3 Scenario 3. Modification of arrival and departure times in the timetable
In this scenario, we illustrate the modification of arrival and departure times of trains or change of track, representing a disturbance or a reschedule operation. The test case emulates the reschedule of different trains by the TMS (by updating the RailML to YCS) in the original plan by modifying the arrival/departure time or track. This update is received by YCS, which refresh the interface and shows the real-time situation of the yard. The starting point of the scenario is
illustrated in Figure 21 (also the final situation of scenario 2).

For modified trains, YCS behaves as follows:

- A train not involved in any modification keeps data associated to it.
- When a train has no manually changed data associated to it, the scheduled and the target arrival/departure time and tracks are modified according to updated data.
- When a train has manually changed data associated to it, the scheduled and the target arrival/departure time are modified, but the operations are kept. This imply:
  - The train will have the target arrival/departure time updated.
  - In case LM has moved the train to a track other than the original, the train stays on the track LM has assigned (even if data from TMS implies a change).
  - The associated operations are kept (also for changed trains), and in case a conflict takes place, it is detected.

![Figure 21: Initial status of scenario 3.](image)

For the different test cases, the following trains will be modified:

- Train 4172: it has no manually added data in YCS (blue square).
- Train 44737: it has complete data in YCS (green square).
- Train 9825: the track has been modified by the LM before update (black square).

All test cases are explained before the conclusion/results of the scenario.
Test case 1. Initial data before updating YCS due to reschedule of trains
This test case illustrates the status of the system before handling updates representing rescheduling of trains, as shown in Figure 21.

- Trains with complete data (not circled in yellow).
- There are conflicts (circled in red).
- There is a maintenance activity.

Test case 2. Reschedule of a train without manually added data
This test case shows that, when obtaining a RailML update which implies rescheduling of train 4172 with no manually added data, the train modifies its arrival/departure time. In this test case, the train schedule is correctly modified without creating any conflict (blue square).

Test case 3. Reschedule of a train with complete data
This test case shows that, when obtaining a RailML update which implies rescheduling a train (44737) which has manually added data associated to it, the train modifies its arrival/departure time. In this test case, the train schedule is correctly modified but it creates a conflict, as the new arrival/departure time is outside of the track reservation interval. As can be seen, the planned operations related to the train are kept, and the conflict is correctly created (green square).

Test case 4. Changing arrival time and track allocation
This test case shows that, when obtaining a RailML update which implies rescheduling a train (9825) arrival time and moving it to a new track, the arrival time is modified but the track is not changed, since track allocation is the responsibility of the LM using YCS (i.e., updated track input from TMS is overruled) (black square).

Results of scenario 3
The final scenario is shown in Figure 22.
Figure 22: Final status of scenario 3.

As can be seen, every requirement of the test cases is successfully achieved:

- Trains with previously manually added data keep their data (even the ones that have conflicts)
- Train 4172 (blue square) is successfully modified. Departure time is updated by “TMS (RailML)” from 20:16 to 22:18
- Train 44737 (green square) is successfully modified. Departure time is updated by “TMS (RailML)” from 19:28 to 22:00. The planned operations are kept and, as the new departure time is out of the track reservation interval, a conflict is detected and indicated in red.
- Train 9825 (black square) is successfully modified. Arrival time is updated according to “TMS ( RailML)” from 7:00 to 4:20, but the track is kept since it was previously modified by the LM.

3.6 Conclusion of YCS development in FR8RAIL III WP2

A YCS tool would be a first step towards digitalisation of the marshalling yard planning. The yard actors welcome the tool and would like pilot studies to be started as soon as possible. They also see many ways in which the YCS could be further developed to provide even more benefits. In general, there are certain development steps that can be taken. They are (in order of importance):

1. **Make information visible for all actors** (the current version of the YCS). For this to work automatic updating of ETAs must be developed (so the YCS needs to be connected to some other system providing this information), as well as some default rules for how
requirements and reservations should be moved when their start/end times change. Planning data should be saved to enable quality analysis and support development of best practices.

2. See which activities that have been performed (e.g., has the driver arrived, have all cars been rolled to the classification bowl...). This requires the YCS to be connected to other systems, e.g., via Deplide.

3. Automatic conflict resolution (suggestions). More intelligent support for conflict resolutions could be implemented.

4. Safety critical functions. Last but not least, safety critical functions such as, e.g., dangerous goods markers could be implemented.

3.7 Next step for YCS

Based on the discussion in the previous section, four areas of future research have been identified.

3.7.1 Pilot studies

The yard companies involved in the development and evaluation of the YCS are very interested in running smaller pilots with the system. Certain system improvements are needed before this can be done, but the sooner the pilots can be run the better. There may be possibilities for running pilots within the Europe’s Rail programme.

3.7.2 Data sharing input and output

An important area for improvement is data input and output. In particular, the manual data entry tasks need to be reduced. Figuring out which systems the YCS should get its data from and implementing this connection is an important next step.

3.7.3 Automation and semi-automation

Once again, the manual tasks need to be reduced through automation or semi-automation. The YCS could also suggest ways to resolve conflicts and thereby further reduce the amount of work the planner has to do. Specifying and implementing these functionalities is an area for future work.

3.7.4 HCI-questions

As always when introducing new tools, the user aspects must be considered. This has not been the main focus during this project but is indeed important.
4 Application of replanning methods to selected disturbance scenarios

Any disturbance to the originally planned timetable impacts the operation of one or several trains. One possibility of such disturbance is given by maintenance in case of urgent incidents, that is, maintenance that was not planned beforehand. Depending on where the disturbance occurs, different trains are impacted. Maintenance activities at marshalling yards influence freight trains. In the situation of maintenance activity without pre-planning, the traffic control centre must both request maintenance contractors and handle the operational traffic situation. Here, a particular emphasis lies in finding a feasible train path for each impacted train quickly (as this happens very close to operation) and only to a lesser extent in finding an optimal solution that may require a longer time to compute—which is available, for example, for the planning of pre-planned maintenance. The fast computation of feasible and “good” train paths motivated the development of our timetable modification module “TIMO”.

4.1 Timetable Modification Module

TIMO, a timetable modification module, is based on a ruin-and-recreate heuristic that uses a greedy insertion algorithm for recreation—we will refer to the complete approach as heuristic in the remainder of this section. In the original application, we are given a single train path that should be added to an existing timetable. For that insertion, we are allowed to perform small adjustments to a subset of the existing train paths, \( T' \), in the timetable (where we consider train paths that are temporarily close to the time window in which we would like to add the new train path). The subset \( T' \) can include all trains from the same operator in that time window, or larger or smaller train sets. We remove all train paths in \( T' \) from the timetable, and we aim to reinsert all train paths from \( T' \) and the new train, \( t^n \):

1) For \( j=1,\ldots,J \) (where we refer to \( J \) as the number of insertion iterations)
   a) We pick a random train \( i \) from \( T'U t^n \).
   b) We create several reinsertion options for the train path of train \( i \). Let the set of these possible train paths for \( i \) be denoted by \( T_{\text{options}} \). We denote \( |T_{\text{options}}| \) as path iterations.
   c) Each train path in \( T_{\text{options}} \) is evaluated using an objective function \( z \).
   d) Pick the best train path in \( T_{\text{options}} \) w.r.t. \( z \).
   e) Remove \( i \) from \( T'U t^n \). If \( T'U t^n = \emptyset \): STOP, else: go to a.

b) Evaluate the timetable of iteration \( j \) using objective function \( f \)
c) Let current be the best solution from insertion iteration 1,\ldots,j

2) Output current
When we create the reinsertion options in $T_{\text{options}}$, we do not consider the objective function $z$. That is, we do not create reinsertion options that optimize $z$, but create options and then use $f$ to evaluate all obtained options. Hence, the objective function is used to compare the created train paths and to pick the best out of these (w.r.t. $z$). But, as we do not take $z$ into account when creating the possible train paths, we will generally not obtain the optimal insertion for the train path of train $i$ w.r.t. $z$—though by chance it may happen.

Currently, the evaluation in TIMO is based on the bottleneck robustness, that is, our objective function is the minimal temporal distance between the new train path for $i$ and any other existing train in the timetable, and we aim to maximize this objective function. Hence, in step iv, we pick the train path in $T_{\text{options}}$ that has the largest bottleneck robustness of all train paths in $T_{\text{options}}$. Moreover, also $f$ is chosen as bottleneck robustness—here for all inserted train paths.

With TIMO, we aim for being able to find a “good” train path fast, we will not yield a provably optimal solution:
- In each iteration (for each train $i$), we do not find the train path that maximizes the bottleneck robustness, but we pick the best train path in $T_{\text{options}}$ w.r.t. bottleneck robustness (step iv). Thus, already for a single train path, we cannot guarantee an optimal insertion.
- We iterate over all trains in random order (step i).

4.2 Effect of parameter settings in TIMO on output

For the evaluation, we use scenarios on the route between Mjölby and Hallsberg marshalling yard (see Figure 23). For $T'$, we pick all trains that travel in the same direction as the train for which we aim to add a train path and which have a departure time from the first station within a symmetric time interval, $F_0$, around the preferred departure time for $t^n$. See Erlandson, Häll, Peterson, & Schmidt (2021; 2022) for the full description of this evaluation.

Figure 23: Schematic view of the considered route from Mjölby to Hallsberg

We use a base scenario and evaluate several input parameters:
- The departure time window for the temporarily removed train paths in $T'$, denoted by $W_{i,d}$; this time window is symmetric around the train’s original departure time (in the base scenario, we choose 30 minutes)
- The arrival time window for the temporarily removed train paths in \( T' \), denoted by \( W_i^p \); this time window is symmetric around the train’s original arrival time (in the base scenario, we choose 30 minutes)
- Share of removed trains within the given time frame: we randomly remove \( \gamma| T' | \) trains for \( \gamma \leq 1 \) (in base scenario 100%)
- The considered time frame is 120 minutes

See Table 6 for an overview of the used parameters.

**Table 6: Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_d^p ), ( W_a^p ), ( t )</td>
<td>Departure and arrival time windows</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>Time of the center of ( F_0 )</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>Share of trains that are removed within the time frame</td>
</tr>
</tbody>
</table>

### 4.2.1 Path insertions vs. iteration insertions

Figure 24 shows the average robustness of the final modified timetable when we run the heuristic with different combinations of path and insertion iterations. Figure 25 shows the relative standard deviation (RSD) of the timetable robustness (the objective function value, \( f \)), when running the heuristic sequentially 30 times for each combination of path and insertion iterations. All parameters are chosen as in the base-case scenario.

We observe that the RSD at first decreases rapidly with an increasing number of insertion iterations, it then fluctuates around 15–25%. From this parameter study, we deem 5 path iterations and 5 insertion iterations to be an appropriate choice to generate robust solutions without large deviations.
Figure 24: Average bottleneck robustness when varying the number of path iterations and insertion iterations. The model was run 30 times with each parameter combination.

Figure 25: Relative standard deviation of the bottleneck robustness when varying the number of path iterations and insertion iterations. The model was run 30 times with each parameter combination.

4.2.2 Runtime

We explicitly aim for finding “good solutions” fast with TIMO. Hence, in this subsection, we
analyze the runtime. Of course, we expect that the runtime increases both with an increasing number of insertion iterations (the number of times that we try to insert all trains in random order) and with an increasing number of path iterations (the number of possible paths we compute and evaluate for the insertion of each train path during an insertion iteration).

To determine the impact of the number of each iteration type on the total run-time, the heuristic is run sequentially, while varying the number of path and insertion iterations. The result is shown in Figure 26. We observe that the runtime increases linearly with the number of insertion iterations—whenever we increment the number of insertion iterations by one, a fixed number of train paths is reinserted in the timetable, which results in a linear increase in runtime. The larger the number of path iterations, the larger is the increase in runtime for an additional insertion iteration.

![Figure 26: Average TIMO runtime for varying path insertions over the number of insertion iterations. The model was run 30 times with each parameter combination.](image)

**4.2.3 Effect of the size of the time window**

We aim to study the effect of the choice of departure time window size. See Table 7 for the different scenarios. In each scenario, we evaluate the generated timetable based on bottleneck robustness, i.e., the objective function value, $f$; the total train path deviation compared to the original timetable, $δ$; the average change in run-times, $ϱ$; and the average change in departure times from the starting station, $ξ$. In each scenario, and for each test case, we run the heuristic model 30 times, then we calculate the average value and the standard deviation for each evaluation criterion. In each scenario, we perform a linear-regression analysis, which determines
the correlation between the parameter that is varied and each evaluation parameter. The correlation is given by the coefficient of determination, $R^2$. Note that it measures the correlation between the average evaluation parameter value and the parameter that is varied. There is a standard deviation in each data point that must be considered when analysing the results.

Table 7: Parameter values, in all cases: $F_0=120$ minutes, $\gamma=1$.

![Table 7](image)

We present the results for the analysis of the time-window size in Table 8 and Table 9. We compare results for peak hours, in this case 21:00, and off-peak hours, here 11:00. In the test cases A1 and A9, we cannot produce a modified timetable, because the time window is too small. We can observe how changing the time window size affects the average run-time changes compared to the original timetable $\varrho$. In peak hours, the coefficient of determination $R^2$ is -0.70, indicating that increasing the time window size has a significant impact on the average run-time change, and that the run-time decreases. In off-peak, the correlation is weak with $R^2 = 0.32$, which means that in timetables with sparse traffic, the run-time is not significantly affected by changes to the time window size. The correlation between the time window size and the average departure time change, $\xi$, is strong in both peak and off-peak hours, with $R^2 = 0.96$ and 0.98 respectively. The larger the departure time changes, the more re-planning work will be necessary at both the departure- and arrival marshalling yards. Despite that the departure time change increases with an increasing time window size, the increase is limited. With a time window, $W^d_t$, of 60 minutes (+/- 30 min), the average departure time change is 14 minutes at $t=11:00$ and 14.6 minutes at $t = 21:00$. 
Table 8: Results for $t=11:00$

<table>
<thead>
<tr>
<th>Test case</th>
<th>$W_d^t$</th>
<th>$t$</th>
<th>$f$</th>
<th>$\Delta f$</th>
<th>$\sigma_f$</th>
<th>$\delta$</th>
<th>$\sigma_\delta$</th>
<th>$\rho$</th>
<th>$\sigma_\rho$</th>
<th>$\xi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>30</td>
<td>11:00</td>
<td>235.0</td>
<td>43.5</td>
<td>92.4</td>
<td>18.5</td>
<td>-4.8</td>
<td>1.2</td>
<td>7.6</td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>20</td>
<td>11:00</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>25</td>
<td>11:00</td>
<td>213.0</td>
<td>-9.7 %</td>
<td>31.0</td>
<td>81.9</td>
<td>20.7</td>
<td>-5.2</td>
<td>1.0</td>
<td>6.1</td>
</tr>
<tr>
<td>A3</td>
<td>35</td>
<td>11:00</td>
<td>242.9</td>
<td>+3.0 %</td>
<td>52.0</td>
<td>90.0</td>
<td>28.4</td>
<td>-5.0</td>
<td>0.7</td>
<td>8.1</td>
</tr>
<tr>
<td>A4</td>
<td>40</td>
<td>11:00</td>
<td>249.3</td>
<td>+5.7 %</td>
<td>41.0</td>
<td>93.0</td>
<td>26.0</td>
<td>-5.0</td>
<td>0.3</td>
<td>9.9</td>
</tr>
<tr>
<td>A5</td>
<td>45</td>
<td>11:00</td>
<td>247.0</td>
<td>+4.7 %</td>
<td>46.6</td>
<td>90.5</td>
<td>24.2</td>
<td>-4.9</td>
<td>0.6</td>
<td>9.7</td>
</tr>
<tr>
<td>A6</td>
<td>50</td>
<td>11:00</td>
<td>254.8</td>
<td>+8.0 %</td>
<td>57.8</td>
<td>102.4</td>
<td>32.9</td>
<td>-5.2</td>
<td>0.9</td>
<td>10.3</td>
</tr>
<tr>
<td>A7</td>
<td>55</td>
<td>11:00</td>
<td>258.6</td>
<td>+9.6 %</td>
<td>54.7</td>
<td>99.6</td>
<td>33.5</td>
<td>-5.0</td>
<td>0.7</td>
<td>11.4</td>
</tr>
<tr>
<td>A8</td>
<td>60</td>
<td>11:00</td>
<td>253.8</td>
<td>+7.6 %</td>
<td>43.2</td>
<td>93.0</td>
<td>28.3</td>
<td>-4.7</td>
<td>0.8</td>
<td>14.0</td>
</tr>
<tr>
<td>$R^2$</td>
<td></td>
<td></td>
<td>0.86</td>
<td></td>
<td>0.72</td>
<td></td>
<td>0.32</td>
<td></td>
<td>0.96</td>
<td></td>
</tr>
</tbody>
</table>

Table 9: Results for $t=21:00$

<table>
<thead>
<tr>
<th>Test case</th>
<th>$W_d^t$</th>
<th>$t$</th>
<th>$f$</th>
<th>$\Delta f$</th>
<th>$\sigma_f$</th>
<th>$\delta$</th>
<th>$\sigma_\delta$</th>
<th>$\rho$</th>
<th>$\sigma_\rho$</th>
<th>$\xi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>30</td>
<td>21:00</td>
<td>172.8</td>
<td>40.0</td>
<td>104.4</td>
<td>22.3</td>
<td>4.1</td>
<td>1.5</td>
<td>6.6</td>
<td></td>
</tr>
<tr>
<td>A9</td>
<td>20</td>
<td>21:00</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>A10</td>
<td>25</td>
<td>21:00</td>
<td>176.6</td>
<td>2.2 %</td>
<td>31.1</td>
<td>95.6</td>
<td>25.5</td>
<td>4.2</td>
<td>0.9</td>
<td>5.7</td>
</tr>
<tr>
<td>A11</td>
<td>35</td>
<td>21:00</td>
<td>161.4</td>
<td>-6.6 %</td>
<td>50.6</td>
<td>98.8</td>
<td>26.3</td>
<td>3.9</td>
<td>1.2</td>
<td>7.0</td>
</tr>
<tr>
<td>A12</td>
<td>40</td>
<td>21:00</td>
<td>174.5</td>
<td>+1.0 %</td>
<td>29.3</td>
<td>106.0</td>
<td>30.9</td>
<td>3.8</td>
<td>1.5</td>
<td>8.2</td>
</tr>
<tr>
<td>A13</td>
<td>45</td>
<td>21:00</td>
<td>174.6</td>
<td>+1.0 %</td>
<td>30.5</td>
<td>115.6</td>
<td>30.5</td>
<td>2.4</td>
<td>1.8</td>
<td>9.9</td>
</tr>
<tr>
<td>A14</td>
<td>50</td>
<td>21:00</td>
<td>183.3</td>
<td>+6.1 %</td>
<td>24.3</td>
<td>131.1</td>
<td>35.7</td>
<td>1.3</td>
<td>2.4</td>
<td>12.8</td>
</tr>
<tr>
<td>A15</td>
<td>55</td>
<td>21:00</td>
<td>186.6</td>
<td>+8.0 %</td>
<td>4.4</td>
<td>135.0</td>
<td>32.1</td>
<td>1.4</td>
<td>2.6</td>
<td>13.3</td>
</tr>
<tr>
<td>A16</td>
<td>60</td>
<td>21:00</td>
<td>187.2</td>
<td>+8.3 %</td>
<td>1.4</td>
<td>141.4</td>
<td>45.4</td>
<td>3.3</td>
<td>3.8</td>
<td>14.6</td>
</tr>
<tr>
<td>$R^2$</td>
<td></td>
<td></td>
<td>0.71</td>
<td></td>
<td>0.96</td>
<td></td>
<td>-0.70</td>
<td></td>
<td>0.98</td>
<td></td>
</tr>
</tbody>
</table>

In Figure 27(a), we present the bottleneck robustness values. The bottleneck robustness is defined as the minimum temporal distance between a train path and any other train path along its route (Ljunggren, Persson, Peterson, & Schmidt, 2020). An increased size of the departure time window, $W_d^t$, results in an increase of the minimum bottleneck robustness. Larger time windows enable the model to investigate more solutions to the problem. This effect is more evident in off-peak ($t=11:00$), compared to the peak ($t=21:00$), with a coefficient of determination $R^2$ of 0.86 compared to 0.71 in peak hours. A reason could be that in off-peak, the traffic density is lower and there are fewer meeting trains to consider, which entails that there are more feasible ways to modify the timetable. In peak hours ($t=21:00$), the robustness increase is limited since the trains in the opposite direction restrict the number of ways each train path can be modified. At $W_d^t = 60$ minutes, the standard deviation of the minimum bottleneck robustness is low, indicating that the most critical point, i.e., the point in the timetable with the shortest temporal distance between a train and any other train, is similar each time the model runs. Therefore, the model does not find any new solutions that significantly improve the minimum bottleneck robustness.
In Figure 27(b), we observe that the total train path deviation increases with an increasing time window size, $W_r^d$. In peak (t = 21:00), the increase is larger than in off-peak (t = 11:00), resulting in a large coefficient of determination, $R^2 = 0.96$. The corresponding $R^2$ value for the off-peak period is 0.72, which indicates that the time window size has a larger effect on the magnitude of the total train path modifications in peak hours than in off-peak (see Table 8 and Table 9).

We can observe that increasing the departure-time-window size increases the bottleneck robustness both in peak and in off-peak hours. However, especially in peak hours, larger time windows result in larger path modifications, $\delta$. In case of passenger traffic this has the negative effect that passengers might lose connections at intermediate stations and need to re-plan their journeys. However, when developing TIMO, the focus has been on freight traffic, and enabling freight operators to add a train path by adjusting some of their own surrounding train paths, without affecting other operators. In such a case, $\delta$ is not of particular importance, if there is no need for a planned stop at some location along the route at a certain time. The average departure time deviation from the first station, $\xi$, however, affects the planning at the marshalling yard. A very strong positive correlation between the time window size, $W_r^d$, and $\xi$, means that an increase of the time window size would likely result in a proportionally large
increase of the departure time deviation from the first station.

4.2.4 Effect of the Share of Adjusted Train Paths $\gamma$

We aim to study the effect of the choice of the share $\gamma$ of trains that we allow to be modified within $F_0$. In Table 10 the different scenarios are shown. In each scenario, we evaluate the generated timetable based on bottleneck robustness, i.e., the objective function value $f$; the total train path deviation compared to the original timetable, $\delta$; average change in run-times, $\varrho$; and the average change in departure times from the starting station, $\xi$. In each scenario, and for each test case, we run the heuristic model 30 times, then we calculate the average value and the standard deviation for each evaluation criterion. In each scenario, we perform a linear-regression analysis, which determines the correlation between the parameter that is varied and each evaluation parameter. The correlation is given by the coefficient of determination, $R^2$. Note that it measures the correlation between the average evaluation parameter value and the parameter that is changed.

In each evaluation parameter, $R^2 < 0$ indicates an increase of the parameter value, as $\gamma$ decreases. Analogously, $R^2 > 0$ represents a decreasing evaluation parameter value, as $\gamma$ decreases.

Table 10: Parameter values, in all cases: $W_{rd}=30$ minutes, $F_0=120$ minutes.

<table>
<thead>
<tr>
<th>Test case</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>100 %</td>
</tr>
<tr>
<td>B1</td>
<td>83 %</td>
</tr>
<tr>
<td>B2</td>
<td>67 %</td>
</tr>
<tr>
<td>B3</td>
<td>50 %</td>
</tr>
<tr>
<td>B4</td>
<td>33 %</td>
</tr>
<tr>
<td>B5</td>
<td>17 %</td>
</tr>
</tbody>
</table>

Table 11 shows the results in off-peak, i.e., $t = 11:00$. The minimum bottleneck robustness, $f$ increases as the share of removed trains within the time frame, $\gamma$, decreases, but peaks at $\gamma = 50\%$. The total deviation from the original timetable, $\delta$, naturally decreases as $\gamma$ decreases, i.e., fewer train paths are adjusted. In all test cases B1 to B5, $\varrho$ is negative, i.e., in average, the runtimes for the adjusted trains decrease. A decreasing $\gamma$ has a very limited effect on the average departure time change from the starting station, $\xi$, averaging circa 7 seconds. In test case B5, in which only 17% of the train paths within $F_0$ are adjusted, $\xi$ was 11.0 seconds. Table 12 shows the results from test case B6 to B10, i.e., the peak period $t = 21:00$. Unlike at $t = 11:00$, $\gamma$ has no significant effect on the bottleneck robustness $f$. However, the deviation compared to the original timetable, $\delta$, decreases as fewer train paths are modified. The average change in runtime per train, $\varrho$, is not as clearly affected by the share of removed trains, $\gamma$. A coefficient of
determination, $R_2$ of 0.88 indicates a relatively strong correlation. However, the changes are very small, and the standard deviation for each parameter setting, $\sigma_\theta$, is significant, which makes it impossible to determine if a strong correlation exists or not. The average change in departure time from the starting station, $\xi$, is relatively constant around 6 minutes in test case B6 to B9, but decreases to 2.8 minutes in test case B10, in which 17% of the train paths within $F_o$ are adjusted. The results of $f$ and $\delta$ for all test cases are visualized in Figure 28.

Table 11: Results for $t=11:00$.

<table>
<thead>
<tr>
<th>Test case</th>
<th>$\gamma$</th>
<th>$t$</th>
<th>$f$</th>
<th>$\Delta f$</th>
<th>$\sigma_f$</th>
<th>$\delta$</th>
<th>$\sigma_\delta$</th>
<th>$\rho$</th>
<th>$\sigma_\rho$</th>
<th>$\xi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>100 %</td>
<td>11:00</td>
<td>235.9</td>
<td>43.5</td>
<td>92.4</td>
<td>18.5</td>
<td>-4.8</td>
<td>1.2</td>
<td>7.6</td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>83 %</td>
<td>11:00</td>
<td>234.6</td>
<td>-0.6 %</td>
<td>47.4</td>
<td>73.3</td>
<td>22.6</td>
<td>-4.9</td>
<td>0.8</td>
<td>7.2</td>
</tr>
<tr>
<td>B2</td>
<td>67 %</td>
<td>11:00</td>
<td>291.4</td>
<td>23.5 %</td>
<td>45.3</td>
<td>63.2</td>
<td>29.5</td>
<td>-4.6</td>
<td>1.0</td>
<td>7.5</td>
</tr>
<tr>
<td>B3</td>
<td>50 %</td>
<td>11:00</td>
<td>306.3</td>
<td>29.8 %</td>
<td>33.4</td>
<td>39.5</td>
<td>29.9</td>
<td>-4.6</td>
<td>0.8</td>
<td>7.6</td>
</tr>
<tr>
<td>B4</td>
<td>33 %</td>
<td>11:00</td>
<td>293.3</td>
<td>24.3 %</td>
<td>44.5</td>
<td>35.2</td>
<td>31.1</td>
<td>-4.9</td>
<td>1.1</td>
<td>7.9</td>
</tr>
<tr>
<td>B5</td>
<td>17 %</td>
<td>11:00</td>
<td>286.3</td>
<td>21.4 %</td>
<td>48.2</td>
<td>9.5</td>
<td>1.1</td>
<td>-3.9</td>
<td>0.2</td>
<td>11.0</td>
</tr>
<tr>
<td>$R^2$</td>
<td></td>
<td></td>
<td>-0.76</td>
<td></td>
<td>0.99</td>
<td>0.61</td>
<td>-0.71</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 12: Results for $t=21:00$.

<table>
<thead>
<tr>
<th>Test case</th>
<th>$\gamma$</th>
<th>$t$</th>
<th>$f$</th>
<th>$\Delta f$</th>
<th>$\sigma_f$</th>
<th>$\delta$</th>
<th>$\sigma_\delta$</th>
<th>$\rho$</th>
<th>$\sigma_\rho$</th>
<th>$\xi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>100 %</td>
<td>21:00</td>
<td>172.8</td>
<td>40.0</td>
<td>104.4</td>
<td>22.3</td>
<td>-4.1</td>
<td>1.5</td>
<td>7.6</td>
<td></td>
</tr>
<tr>
<td>B6</td>
<td>83 %</td>
<td>21:00</td>
<td>185.5</td>
<td>7.3 %</td>
<td>14.5</td>
<td>91.1</td>
<td>25.8</td>
<td>3.6</td>
<td>2.1</td>
<td>6.5</td>
</tr>
<tr>
<td>B7</td>
<td>67 %</td>
<td>21:00</td>
<td>188.0</td>
<td>8.8 %</td>
<td>12.8</td>
<td>54.0</td>
<td>37.4</td>
<td>0.7</td>
<td>4.0</td>
<td>6.6</td>
</tr>
<tr>
<td>B8</td>
<td>50 %</td>
<td>21:00</td>
<td>180.1</td>
<td>4.2 %</td>
<td>35.0</td>
<td>47.9</td>
<td>33.3</td>
<td>2.1</td>
<td>5.2</td>
<td>6.5</td>
</tr>
<tr>
<td>B9</td>
<td>33 %</td>
<td>21:00</td>
<td>177.6</td>
<td>2.8 %</td>
<td>41.3</td>
<td>42.3</td>
<td>35.6</td>
<td>0.6</td>
<td>5.3</td>
<td>6.3</td>
</tr>
<tr>
<td>B10</td>
<td>17 %</td>
<td>21:00</td>
<td>180.4</td>
<td>4.4 %</td>
<td>37.6</td>
<td>7.4</td>
<td>0.76</td>
<td>-4.0</td>
<td>0.0</td>
<td>2.8</td>
</tr>
<tr>
<td>$R^2$</td>
<td></td>
<td></td>
<td>-0.06</td>
<td></td>
<td>0.97</td>
<td>0.88</td>
<td>0.80</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
As expected, the total deviation from the original timetable, δ, decreases the fewer trains inside the time frame $F_0$ are removed. As mentioned, if this heuristic would be used by an operator to add a train path close to operation, a realistic situation would be that only the train paths that belong to that operator would be allowed to move. TIMO can handle the removal of train paths belonging to specific operators, but for the sake of generality, random train paths were removed within $F_0$. We observe that only removing a share of the trains within $F_0$ resulted in a more robust solution compared to removing all train paths.

4.3 Using TIMO for replanning on the line in case of ad-hoc maintenance at the departure marshalling yard

In this subsection, we employ TIMO to replan line management in case of ad-hoc maintenance at the departure marshalling yard. In the scenario we consider the departure yard is impacted by ad-hoc maintenance activities up to limited time horizon (a few hours). As described in Subsection 4.1, TIMO aims at adding a single new train path. Thus, here we investigate how iterative runs of TIMO can solve the problem at hand.

4.3.1 Using TIMO to solve the replanning problem

*Scenario description:* Departure yard $y$ blocked in $[t_1, t_2]$  
Set $T$: trains that would leave $y$ in $[t_1, t_2]$  
*Input:* timetable without $T$, forbidden departure interval for $y$  
*Output:* feasible timetable, that is, a timetable in which no trains depart from $y$ during $[t_1, t_2]$
**Strategy** (for applying TIMO):

Iteratively:

- Pick a train i in T (in order or randomly)
- Use TIMO to reinsert i into the timetable with departure from y infeasible in [t1, t2] (possibly allow to wiggle a subset of the remaining trains). This means we create several options, $T^\text{options}$, and pick the best of these.
- $T:=T\setminus\{i\}$

### 4.3.2 Example for TIMO usage to solve the replanning problem

In this section, we illustrate a small example showing how to use TIMO according to the iterative method described in Subsection 4.3.1. We illustrate how this approach finds feasible solutions to the replanning problem in case of a blocked departure yard because of ad-hoc maintenance activities. This example is based on the line Malmö marshalling yard – Hallsberg marshalling yard. This line is used by a mix of various trains, from multiple undertakings, using different parts of the line.

The underlying assumption of the example is based on is that all departures from Malmö marshalling yard are blocked from 2:00 to 3:00. During that time, three trains were scheduled to depart from Malmö marshalling yard. This gives us that set T contains three trains: $T=\{\text{train}_1, \text{train}_2, \text{train}_3\}$. These three trains do not go the full stretch from Malmö marshalling yard to Hallsberg marshalling yard. Both train_1 and train_2 only travel a very short stretch from Malmö before they deviate to another line, while train_3 continues to its end station Hässleholm (some 20% of the way to Hallsberg). Currently, TIMO is used for replanning on one line only, hence, for train_1 and train_2 we only consider their train path on the Malmö-Hallsberg stretch. This is a delimitation of the method, as for the both trains possible conflicts could appear in the other parts of the rail network they are using. However, this example is mainly used to illustrate the TIMO capability for the given problem which focusses on a single railway line.

The following description shows what happens when the trains in T are inserted to the timetable. Each “step” is the insertion of one train. In this experiment, we pick the trains from T in order of original departure time.

**Step 1:**
Using the original timetable without the trains in T, we insert train_1. This is done by first reading the original timetable data for train_1, and then using TIMO’s function to change the departure time of train_1. In this way, TIMO has the original RailML-data description of train_1’s train path. The output is the new train path (i.e., the new times at all stops) for train_1.
Step 2:
The timetable including the output from step 1 is used as input, and we add the original timetable data for train_2, for which we must change the departure time. When changing the departure time, the train paths of two other trains (train_4 and train_5) are changed (other trains possible to wiggle according to the strategy described in Section 4.3.1). The output is the new train paths of train_2, train_4 and train_5.

Step 3:
The timetable including the output from step 2 is used as input, and the original timetable data for train_3 is added. The departure time of train_3 is changed and when doing so, the train path of train_4 is changed once more. The output is the new train paths of train_3 and train_4.

As could be seen in Step 2 and Step 3, other trains were affected (that is, we allow changes to these trains). How other trains are allowed to be affected when changing the departure time of one train is described in FR8Rail II, Deliverable D3.4 (2021).

This example illustrates that TIMO can be used for line management in case of ad-hoc maintenance at the departure marshalling yard, simply by inserting one train at a time from the set of trains that originally were to leave the departure yard during the closed time window.
5 Interconnection of a yard prediction model with a network macrosimulation model

In Deliverable 2.1 (FR8RAIL III, 2021) a concept for connecting a Yard Prediction Model with network simulation was proposed. In Deliverable 2.2 (FR8RAIL III, 2022), a conceptual model framework was presented to incorporate machine learning-based departure deviations from one yard into a macroscopic network simulation tool (Proton) and predict the arrival deviations to the next yard. Also, an example with preliminary results was presented in Deliverable 2.2 (ibid). The Yard Prediction Model was implemented by using operational yard data for one year (2019) from Malmö marshalling yard. The framework was first tested by using prediction output from one week in fall of 2019 which was incorporated as input to Proton to simulate and predict arrivals to the next yard.

The initial results indicated that this model framework provides the infrastructure manager (IM) a yard departure deviation prediction model with a R-squared of 92%. In addition, the yard operator can receive the estimation of arrivals from the network to the next yard with a mean absolute error of 32 minutes, which gives possibility to the yard operator dealing with the planning of wagon rebooking and upcoming departures, especially in the case of larger deviations. The case study consisted of yard data from Malmö marshalling yard and the network simulation was applied on the section Malmö–Hallsberg.

This model framework is now applied on longer time periods, using around 7.5 months of data from year 2019 (February to May and September to mid-December). All days within these periods are simulated in Proton (weekdays as well as weekends). The operational yard data comes from Malmö marshalling yard and the focus is on freight trains running from Malmö marshalling yard to Hallsberg freight yard. Malmö marshalling yard is a single-ended yard with a combined arrival and departure yard in the south of Sweden, which makes handling arriving and departing trains more complex compared to other yards with the conventional yard layout. In addition, Malmö yard shows large disturbances among other yards in Sweden (FR8HUB, 2018). A map of the simulated railway line is presented in Figure 29. Freight trains between Malmö and Hallsberg (and other locations) may occasionally be scheduled partly on other lines than the main line and this is the reason for including some lines in addition to the Southern mainline in the southern part of Sweden as indicated on the map.
Figure 29: Railway line between Malmö and Hallsberg. The total distance is around 400 km. Highlighted lines are included in the simulations.

5.1 Yard Prediction Model

The yard prediction module is implemented by developing a machine learning-based approach using the data from the yard manager. The output from the yard prediction model is the departure
deviations which will be entered to the network simulation model to simulate the running of freight trains along the line.

The data for model implementation is a combination of three different datasets provided by the yard operator: wagon connections, train punctuality, and train properties. The datasets were merged and pre-processed using PANDAS (Python Data Analysis Library). Merging the data from three different datasets required handling some discrepancies. For example, the wagon connections and train features are planned prior to operations; wagon connections are from the wagon booking system called BRAVO and train features are from the annual timetable plan system called TrainPlan, whereas train punctuality data are measured when trains are operated. In total, there were 161162 wagon connections from which 10% were duplicates and removed. The final data used for the application modelling comprised of 30548 train connections.

5.2 Network simulation model – Proton

The network simulations are carried out in Proton (Punctuality and Railway Operation Simulation), which is originally a macroscopic railway traffic simulation tool developed by DB Analytics within PLASA and PLASA-2 projects included in Shift2Rail. One main goal has been for the tool to be able to simulate large networks with many trains within a short execution time. It was formerly known as Prism (PLASA Railway Interaction Simulation Model). The main inputs required by Proton are:

- Macroscopic infrastructure consisting of nodes and edges
  - Nodes typically represent operational control points or other timing points in a railway network. The nodes handle arrivals and departures but the track layout within the nodes is not modelled.
  - Edges connect nodes and have properties such as length, number of tracks and average block length.
- Timetable – specifies the sequence of nodes with associated scheduled arrival and departure times for each individual train along with train ID and a train type (timing load).
- Technical minimum running times for different drive modes for all relevant train types and edges. The difference between scheduled and technical running time gives available allowance, which can be used for reducing delays in the simulation.

Running stochastic simulations also requires input in the form of different types of delay distributions. These are mainly for initiations of trains (entry), dwell and running time extensions or disruptions. The main purpose with introducing these delay distributions is to introduce primary delays which then, depending on the magnitude of these and the timetable, can result in secondary delays.
Train conflicts in Proton are modelled based on minimum headway times. These are calculated from block occupation times by considering some of the infrastructure edge properties and the train running times. Proton and its functionality are described in more detail in (Zinser, et al., 2019; PLASA, 2018).

5.3 Simulation setup

The simulation framework uses data from different sources. The macroscopic infrastructure is mainly converted from a national microscopic railway network model (RailSys model). Timetables are converted from an operational timetable which for 2019 was planned in TrainPlan. Technical running times come from another system (Tigris) which also feeds TrainPlan with running times. Basic vehicle data (timing loads) are also converted from RailSys. An overview of the data sources and simulation framework is given in Figure 30.

![Figure 30: Overview of the data sources and the process used for the network simulation in Proton](image)

General delay distributions for modelling mainly primary delays of different types are compiled from historical data, which holds registration times for trains passing operational control points (mostly stations). However, this data does not distinguish between delay causes and therefore different scaling levels for distributions are applied for the purpose of calibrating the simulation model. Output data from the simulation is compared to historical data, in this case punctuality for different train categories at different locations is the main indicator used.

Scaling levels from 0% to 40% in 5%-intervals are used, the percentage refers to the share of total additional delays (increase of delays on stations or between stations) considered as primary
delay. The conclusion from the calibration runs was that a 25% scaling yielded a punctuality output that best fit the empirically observed punctuality, i.e., 25% of the empirical run and dwell time delays are primary and the remaining 75% are secondary. This is by no means a general conclusion since the scaling factor can naturally vary for different timetables and other parts of the network, and it is also depending on the simulation tool itself and the parameters used. The dispatching scheme is based on train priorities, the decision to choose this over other available schemes is based on previous simulation studies.

In an earlier study, comparative simulations were carried out in both RailSys and Proton for the same section (Malmö–Hallsberg) but with a timetable from year 2016. Here a scaling factor of 33%, 66% and 100% was applied and 33% gave a good fit to empirical data (Johansson, Palmqvist, Sipilä, Warg, & Bohlin, 2022). In a partially similar study but using a section of the Swedish Western mainline (Hallsberg–Gothenburg) a scaling factor of 30% yielded results closest to the empirical punctuality (Johansson, Sipilä, & Palmqvist, 2022).

The delay distributions are categorized based on the following main train categories: long distance passenger trains, regional passenger trains, local passenger trains and freight trains. Non-revenue trains (empty runs) use distributions from the long-distance passenger trains. In addition to these categories, the distributions have been divided into directions.

5.4 Simulation runs with examples

The version of Proton used here simulates 24 h periods (from 0 to 24) which means that trains running over midnight are cut. This leads to the fact that the selection of freight trains (from Malmö to Hallsberg) for evaluation is reduced since trains running over midnight will not make full runs in the simulations. Therefore, only trains scheduled within the same date (day) from Malmö to Hallsberg, and which are also included in the operational yard data set and were found in historic arrival data to Hallsberg are considered in the evaluation. From this group only train IDs which have run on a sufficient number of dates are finally used in the evaluation. The list gives some basic statistics on the number of trains between Malmö and Hallsberg.

- Trains within same date AND in YPM data AND in arrival data AND have run on a sufficient number of dates: 402
- Trains within same date AND in YPM data: 781
- Trains within same date but NOT in YPM data: 243
- Overnight trains AND in YPM data: 382
- Overnight trains but NOT in YPM data: 48

Simulations are run with 100 cycles for each day. The first period is from February 4th to June 2nd and the second period from September 2nd to December 7th (year 2019). The periods correspond to the operational yard data set from Malmö marshalling yard. Within the process of generating
input files to simulations, all freight trains departing from Malmö marshalling yard are checked against the data coming from the Yard Prediction Model. If there is a match either on the train mission ID or on the technical train ID for the date, the train will be initiated according to this data. If no match is found, the trains are assigned to the general freight train initial distribution for the relevant direction. Note that train initiation does not have to coincide with train departure, whether a train departs on the initiation time depends on the current situation and dispatching in the simulation. In the setup used, freight trains and non-revenue trains may be initiated both ahead, on and after scheduled time, passenger trains only on or after scheduled time.

With the aim of showing differences between using general initial distributions for freight trains or times coming from the Yard Prediction Model, the simulation outcome for one freight train is shown in a graphical timetable. Figure 31 shows the simulated outcome with median and four percentile intervals for one freight train using a general initial distribution. For the other trains, only the scheduled paths are shown. In this case a general initial distribution is used at Malmö marshalling yard meaning that the departures are largely controlled by the design of the distribution. In some simulation runs this train departs ahead of schedule. The train may encounter primary delays (coming from distributions) along the route and most likely also secondary delays due to having to give way to higher prioritized trains (in this case passenger trains). This means that the freight train may be overtaken at nodes where overtakings are enabled, thus suffer from secondary delays (knock-on delays). Some overtakings are scheduled, but these may for example take longer time than planned and additional overtakings may be introduced depending on the situation.
Figure 31: Simulated outcome for one freight train (Malmö–Hallsberg) using a general initial distribution at Malmö marshalling yard (median and percentile intervals)

The spread for this train increases with distance. This is a quite typical behaviour, especially for freight trains, although it is not always the case since it depends on the train’s ability to catch up time by using scheduled timetable allowances and the operational situation with other trains. The scheduled train path (also shown in Figure 31) shows that the train has a couple of scheduled stops, mainly for letting other faster trains pass. The train is occasionally departing with a large delay which most likely follows from an initiation in the same magnitude (percentile interval 0 – 100).

Figure 32 shows the same train but this time the initiation time at Malmö marshalling yard comes from the Yard Prediction Model. This means that for each cycle when simulating this specific date this train gets the same initiation time. As the figure shows, the train can depart on or close after the initiation time. Similarly, as in Figure 31 the spreading increases along the route. In this case, the departure prediction time is around 25 minutes after the scheduled departure time. By this time, if using the general distribution (Figure 31), this same train has already departed in most of the simulated cycles. Departing on or close to on time (as is the case for half of the simulation runs, 25 – 75 percentile) gives in the example also a smaller time window at arrival to Hallsberg marshalling yard for this percentile as if compared to Figure 32 where the arrival time window for the same percentile is clearly larger. Since the train in Figure 32 is departing around 25 minutes late, and missing its scheduled train slot, it will most likely
have a larger probability for secondary delays to begin with in all, although it depends on the timetable, other trains, and their whereabouts.

Figure 32: Simulated outcome for one freight train (Malmö–Hallsberg) where initiation time at Malmö freight yard comes from Yard Prediction Model (median and percentile intervals)

Figure 31 and Figure 32 illustrate the concept with connecting the YPM to a network simulation tool. Using a departure prediction value instead of applying a more general distribution for initiating trains gives a better prediction of the arrival to the next yard, since Yard Prediction Model is implemented using a data-driven approach with a R-squared of 0.92 which shows a high capture of variance in the data and a mean absolute error (MAE) of less than three minutes which is based on the previous performance for trains. The input delay distributions and their allocation used for the simulations can be further improved, for example by doing further analyses on historical data and performing more comparative simulations.

5.5 Results of evaluating yard arrival predictions

In this part, we discuss the overall performance evaluations of the combined model framework. We have also compared the performance of the combined approach in terms of the arrival prediction accuracy to a basic machine learning model (Basic ML). The basic machine learning model uses the same concept and predictors applied in yard departure prediction model to predict the arrival deviation to the next yard instead of departure prediction from the origin yard. It is regarded as a basic approach since it does not consider any parameter regarding the
running of the freight trains on the line. General performance indicators for comparison between these two models are presented in Table 13. In terms of R-squared, the combined model captures 45% of the variation in the original data which is more than double of what the basic ML (20%) captures. The MAE between the two methods is close, the combined model provides a slightly better MAE. The current performance is satisfactory; however, improvements may be added which we will explain later.

**Table 13: General model performance indicators**

<table>
<thead>
<tr>
<th>Model</th>
<th>R-squared</th>
<th>MAE (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combined model</td>
<td>0.45</td>
<td>39</td>
</tr>
<tr>
<td>Basic ML model</td>
<td>0.20</td>
<td>40</td>
</tr>
</tbody>
</table>

In terms of predicting the spectrum of deviations, we examined the model performance by depicting the distribution of the arrival deviations to Hallsberg in Figure 33. We can see that the actual arrival deviations span a longer spectrum than the simulated ones, which means that the simulation does not capture extra-large deviations. In addition, in the simulation, smaller deviations have higher probabilities than the actual small deviations. Another interesting difference between the simulation and reality is early arrivals; the simulation, in general, shows smaller probabilities for early arrivals than the actual ones.

One reason for the smaller spread in simulated arrivals compared to historical data is that the distributions used for primary run time extensions (and in some cases primary dwell extensions) are aggregated from a large set of historical data and from all edges and nodes within the model area. In addition, all primary delays (perturbations) are applied stochastically. In reality some primary delays are systematic, i.e., affecting multiple trains for some time span at some locations or affecting the same train along most of the route systematically.
Figure 33: Distribution of the arrival deviations

In Figure 34, we show the distribution of the prediction errors. The smaller the predictions the better the model performance would be; we can see that in our case smaller prediction errors have higher probability which shows an appropriate model performance. In the meantime, the highest probabilities are mostly negative prediction errors which means that our model framework is predicting less than the reality or in other words, have a tendency towards predicting less than more. The probability of the larger prediction errors is less which may also be due to the less occurrences of larger deviations in general. It is worthy to note that with the current setup of the model framework due to aggregations may not predict well very large deviations.
To evaluate the performance of the combined model per train, we depict the boxplot of the most frequent trains in the simulation in Figure 35. It shows how the spectrum of the deviations is different in simulation and reality. The simulation captures the spectrum of deviations better for some trains than other trains. In general, the simulation is less responsive the larger the deviations are. When this happens, the spectrum of the prediction error becomes bigger. These boxplots can be a good source of performance evaluation for practitioners; it is possible to evaluate trains individually to identify some common patterns or investigate larger deviations. The same figure can be depicted also for running times of freight trains. For future real-time applications the prediction error part can give an average error of the previous runs so that the practitioners can have a picture of the common performance of the train.

Figure 34 Distribution of the prediction errors
Figure 35: Boxplot of the most frequent trains.
6 Usage and impact

In this chapter we analyse the usage of the demonstrated models and how they can support the real-time traffic management. Further, we discuss the impact of the demonstrated models and tools regarding load factor, punctuality and average speed.

6.1 Possibilities for usage in support of the real-time traffic management

The methods and tools presented here should ideally be implemented and put into practical use in an operational-time setting in the future (see Chapter 2 for a discussion on real-time vs. operational-time). This far, the methods and tools have been applied in off-line settings. In this section we comment on the suitability and possibility of each respective method and tool to really be applied in an operational-time setting.

The YCS has been developed for operational-time usage, and in the evaluation workshop the simulated time progressed in normal speed. The results showed that the tool was fast and easy enough to use for re-planning in a real time setting. However, it is important to ensure that the planning is automatic whenever feasible, and that planning tasks can be carried out with as few manual actions as possible. The YCS is to some extent already prepared to be connected to a live data stream and is able to handle the updated data that could be a result from that, but within this project we did not have the possibility to connect the system to a live data source. Thus, the YCS-demonstrator is this far dependent on manual updates regarding, e.g., changed arrival times. Still, the conclusion of the demonstrations is that YCS is very suitable for supporting real-time network management and should be further developed to fit even better to the users’ needs. It is also important that a potential final tool is seamlessly integrated into the dispatchers current working tools to avoid introducing yet another system that the traffic controllers have to handle.

TIMO is also a tool that could support the real-time network management by improving the short-term line planning. To make such a system usable in a operational-time setting, the response time and runtime of the algorithms must be fast enough. The runtime for TIMO depends on several factors: the number of stations in the considered line, the number of path iterations, the number of insertion iterations, the considered time frame for rescheduling, and the number of trains that need to be rescheduled. Depending on the value of all of these, we can obtain runtimes in the range of a few seconds to hours. Hence, the potential of operational-time usage for replanning depends heavily on the actual use case. Thus, the use-cases and algorithms of such a system should be further studied before making definite conclusions how such a system can support the real-time traffic management.

Also for the integrated yard prediction and train traffic simulation (YPM/Proton), the
computational speed is critical for its applicability in an operational-time setting. The experiments have shown that the computational speed is suitable for a operational-time environment, in particular since such a tool could also be run more or less continuously by using updated real time train position information. This would of course require connection to a live data flow and auto preparation of all relevant input data. Arrival time windows could in this way be predicted with smaller spread and facilitate the operational planning at the arrival yard(s). In other words, the arrival yard might get a first predicted arrival interval for some train say 5 hours ahead, maybe new ones 4, 3, etc. hours ahead when the train is enroute, meaning that the prediction could be updated.

All in all, the possibility, as well as the usefulness, for applying these tools in a real time setting to support the real-time traffic management seems favourable.

6.2 Evaluation of KPI impact

One of the sub-tasks of T2.3, as described in Section 1.1, is to evaluate the KPI impact of “Real time network management” concerning the KPIs load factor, punctuality, and average speed (as an input to Task 8.3) upon request. Although no such request has yet been made, we include a brief discussion here about the possible KPI impact that the methods and tools presented in this delivery may have.

A similar analysis was presented in Delivery D2.1, Section 5.3 (FR8RAIL III, 2021), where estimates were given according to four problem or planning scenarios. Here the focus lies on the developed methods and tools per se rather than the scenarios. The discussion is mostly qualitative but supported by practical experiences from domain experts regarding YCS, and some quantitative results from the computational experiments with TIMO as well as yard prediction coupled with Proton.

YCS—Load factor increased; Punctuality increased; Average speed increased

Motivation: During the workshop described in Chapter 3, improved punctuality and predictability of trains and transports, as well as more efficient use of yard resources, were identified as benefits from using the YCS. Train departure punctuality and predictability are expected to improve as resource conflicts can be identified early on and resolved before they become a problem. The improved control and situational awareness should also reduce the number of wagons that don’t catch their connecting trains, which will further improve the punctuality of the transport service. This should also improve fill rates. In the long run, the improved punctuality and predictability of the yard operations may allow for a shorter planned connection times (and thereby also planned transport times). As stated above, a more efficient use of yard resources is also expected. This efficiency improvement comes from, e.g., fewer shunting movements, and...
should decrease the costs for both the service providers and the IM. Improved transport quality in terms of punctuality and transport times, in combination with less expensive yard operations, would make railways a more attractive mode of transportation, which could improve the fill rates even further.

**TIMO—Load factor not affected; Punctuality increased; Average speed marginally affected**

*Motivation:* The main envisioned improvement from TIMO usage is increased punctuality in case of disturbances because we limit the disturbance impact by quickly obtaining new feasible train paths for all trains impacted by the disturbance—in particular, we consider the disturbance of maintenance activity at the departure marshalling yard of freight trains. The load factor and the average speed are not impacted: we reschedule existing freight trains and for that we use the existing runtime templates for these trains, this can, but must not increase the average speed—it may even decrease the average speed, as we pick information from the runtime template according to the conditions of the existing timetable. Looking at the experiments from Subsection 4.2.1, we observe an increased average speed (a negative average runtime change in comparison to the original timetable, see Table 8) in the off-peak scenario and a decreased average speed (a positive runtime change in comparison to the original timetable, see Table 9) in the peak scenario.

**YPM/Proton—Load factor increased; Punctuality increased; Average speed increased**

*Motivation:* Connecting YPM and Proton improves the predictability of departures/arrival from/to the yards. This improvement, in the long run, enhances the flexibility and agility in freight train operations for both infrastructure manager and yard operator, which impacts three main KPIs: First, the load factor is expected to increase since train replanning and wagon rebooking is speeded up through better utilization of freight trains. Second, when the predictability increases the yard operator’s ability to act upon disturbances becomes faster, thus, the punctuality of freight trains, especially in terms departures, will be increased. Third, enhanced predictability in yard departures and the requirement to decrease to the arrival deviation to the next yard assists in better control of freights’ speed throughout their runs between the yards, which improves the possibility to increase the overall average speed.
7 Conclusions and future work

This report, which is the final deliverable from WP2, documents the conducted work and results from task T2.3.

Chapter 2 gives important background to the demonstrators regarding (1) which time-perspective that is considered, and (2) how YCS fits in with yard automation. The move towards a way-of-working governed by a “control by planning”-strategy rather than a “control by execution”-strategy and its effect on the relevant time perspective is discussed. The conclusion is that “operational time”, i.e., a perspective that includes “true real-time” (now) but also the nearest future and where ticks are in the order of minutes, is the most relevant. Further, the challenges tackled by YCS will remain relevant also when automatic train operation and automatic wagon operation have been implemented as the arrival/departure yard will still be a “meeting point” for actors from different organisations. In fact, the arrival/departure yard will not only be the hand-over point between different organisations, but also between the two automatic systems.

In Chapter 3, results from a demonstration and evaluation workshop for the YCS are reported. The YCS makes it possible to share data and visualises plans between the most important actors at an arrival/departure yard. The actors come from different organizations and have different responsibilities and rights. At the demonstration, domain experts and practitioners from different organizations were involved in creating realistic and relevant scenarios for the utilization of the tool. The demonstration showed that a tool like YCS can improve transparency and enable cooperative and pro-active planning. The practitioners reckoned that the tool could prevent and alleviate departure delays, and they wanted to run small pilots as soon as possible. The tool was considered easy to use, but less manual entry would be required in real operations. Further areas of development were also identified during the workshop, e.g., visualizing the current operational situation, saving the planning data to enable analysis and development of best practices, and adding more conflict rules and data. All involved practitioners agreed that the ideas and concepts within YCS can be considered as a “low-hanging-fruit” for making improvements both for the integration between line and yard planning and operations, and for the coordination between the actors at the yard. The demonstration with YCS has therefore moved the state of the art forward by showing how a support tool can enable cooperative and pro-active planning. Finally, initial experiments show that YCS can handle automatically updated data. Future research will focus on integration with surrounding systems and simplified handling by automated data flows and (semi-)automatic planning and including optimization aspects.

In Chapter 4, an evaluation of the performance of the timetable modification module TIMO for adding a single train path to an existing timetable was performed and it is exemplified how this
module can be used to handle the replanning task in case of ad-hoc maintenance on a marshalling yard. It shows that TIMO’s runtime depends linearly on the number of insertion iterations, where the constant of the linear increase depends on the number of path iterations. The observation was that the effect of the time-window size on the average runtime changes (for affected trains) is higher in peak than in off-peak hours—in peak hours we see a significant impact. However, the correlation between time-window size and average departure-time change is significant both in peak and off-peak. Moreover, it was found that an increased time-window size yields a larger minimum bottleneck robustness—an effect that is more prominent in off-peak than in peak hours. Even the total train-path deviation correlates with the time-window size—this effect is more prominent in peak hours. Additionally, even with a higher share of other train paths that may be adjusted, the minimum bottleneck robustness decreases, while the deviation from the original timetable increases. The average runtime change is not clearly affected by this share of trains. With this, the use cases for TIMO are delimited.

Finally, TIMO can be used in an iterative procedure to solve the replanning problem on the line in case of ad-hoc maintenance at the departure marshalling yard: both the general approach and a demonstration on how this can be used for an example where the departure from Malmö marshalling yard is blocked because of ad-hoc maintenance activities are given.

In Chapter 5, a proof-of-concept model framework for increasing the predictability of yard departures and arrivals is evaluated. The model framework incorporates a machine learning-based yard departure deviation prediction model (YPM) into a macroscopic network simulation model (Proton). Both the infrastructure manager and the yard operator can benefit from this model framework; the usability for the yard operator is mainly applying the yard departure model and receiving the yard arrival estimations from the simulation, and the infrastructure manager mainly benefits from the simulation along the line which provides a more realistic picture of train runs along the line. The framework at its current state can improve utilization of yard and line capacity, increase accuracy and responsiveness in ad-hoc replanning of yard operations, and improve wagon re-bookings which may decrease train delays due to waiting for certain wagons. However, future research is required to improve handling of large deviations in freight train simulations and better picturing of freight train earliness.

In summary, three methods and tools have been developed, analysed, and demonstrated on real-world operational data. These tools address three different but equally important aspects for improving the efficiency of freight operations on railways, namely:

- Coordination of all operational activities taking place at arrival/departure yards.
- Replanning of timetables for line traffic.
- Prediction of system effects and the combined operations of yards and lines.

All the methods, tools and concepts contribute to closing the gap between timetable planning
and operational traffic by increasing the possibilities for the traffic management to have a proactive working strategy and take control over the traffic, in contradiction to a reactive strategy in which the traffic management works in response to what is occurring in the traffic system.

The developed concepts presented in the report will be further enhanced in the future. In particular, Trafikverket sees a lot of potential in the promising results and user response related to YCS. Thus, the developments of YCS will continue within the project EU-RAIL FP1 Motional.

The planned enhancements of YCS include the following:

- **Connection to TMS**: YCS will be integrated with the TMS at Trafikverket (denoted “Digital graf”). Use cases for the user interaction regarding the relation TMS and YCS will be specified and implemented. Conclusions and requirements for the technical integration will be documented. Specifications and development will be made in Motional WP11 and demonstrations are made in WP12.

- **Operational planning data from TMS**: YCS will receive data regarding the real time traffic plan (RTTP) from TMS, and other relevant data from other data sources. The data need for YCS in will be analysed in relation to the TAF/TAP TSI. Development is made in Motional WP11 and conclusions are made in WP12.

- **Real time data sharing**: YCS will be integrated with a platform for data sharing. This platform, called Deplide, connects to several actors' data sources, and (in cooperation with FP5 TRANS4M-R) the possibilities for reusing available data on the data sharing platform in YCS will be investigated. Deplide is also the basis for the data sharing initiative Railway CDM, and implementations such as YardCDM. Investigations and development regarding connection YCS-Deplide-YardCDM are made in Motional WP11 in collaboration with EU-RAIL FP5 TRANS4M-R WP32, and demonstrations are made in Motional WP12 together with TRANS4M-R WP33.

- **Enhance functionality for operational planning**: YCS will be enhanced with functionality for more user-friendly operational planning so that the number of manual interactions in the systems will be reduced. This might include semi-automatic planning in which YCS calculates and proposes modifications to the track assignment which the user can accept or deny. Specifications and development are made in Motional WP4 and demonstrations in WP5.

- **Tactical planning**: The needs and possibilities for using YCS in the tactical planning will be investigated. Investigations and specifications are made in Motional WP4.

- **Requirements from dynamic dispatching**: In FP5 TRANS4M-R WP27, the dynamic dispatching aspects at the arrival/departure yard, such as intersecting train routes and movements of locos, will be investigated so that requirements related to yard track allocation in a dynamic dispatching perspective can be formulated. These requirements are delivered to Motional WP5, where the requirements are analysed, feasibility checked and prioritized from a YCS-perspective.

Ongoing work Europe Rail, System Pillar task 3 and Rail Net Europe:

- **Europe Rail Showcase**: FP1 Motional and FP5 TRANS4M-R are planning for a common showcase along the railway corridor Malmö – Oslo Alnabru. Several of the innovations performed in FP1
Motional and FP5 TRANS4M-R will be demonstrated in this showcase, including YCS. Other innovations and use cases that will be demonstrated include decision support and data sharing for the corridor Malmö – Oslo Alnabru.

- **System Pillar (SP):** To spread the results for EU-RAIL and also to certify that important results evolve into standards, there is an ongoing dialogue between FP1, SystemPillar (in particular SP Task 3 Capacity Management Systems/Traffic management Systems) and RailNet Europe. One focus area is timestamps in nodes and processes for sharing operational data. This is expected to have impact, e.g., on future TAF/TAP TSI.

- **Rail Net Europe and Railway CDM:** As mentioned above, YCS will be connected to the data sharing platforms and concepts Deplide and Railway CDM. RNE is working with a CEF Proposal for creating further Railway CDM Pilots and to make guidelines/handbook in Railway CDM area. This activity is planned for 2024 – 2025.
8 References


FR8RAIL II. (2021). Deliverable D3.4 Demonstrator on improved planning - verification of demonstrators.


FR8RAIL III. (2022). Deliverable D2.2 Method development for enhanced and integrated line- and yard planning.


från https://projects.shift2rail.org/s2r_ip5_n.aspx?p=OPTIYARD

PLASA. (2018). *Deliverable D3.2 Description of enhanced model prototype.*


X2RAIL-4. (Submitted but not yet published.). *D10.1 “Report on Demonstrators TRL6”.*