Horizon 2020
European Union Funding
for Research & Innovation

In2Rail

In2Rail

Project Title:	INNOVATIVE INTELLIGENT RAIL
Starting date:	01/05/2015
Duration in months:	36
Call (part) identifier:	H2020-MG-2014
Grant agreement no:	635900

Deliverable D7.5

Evaluation of the Proof-of-concept

Due date of deliverable	30-04-2018
Actual submission date	10-04-2018
Organization name of lead contractor for this deliverable	DLR
Dissemination level	Public
Revision	Final

Authors

		Details of contribution
Authors	**Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR)** Stefanie SCHÖNE	Overall author and editor General approach
	SIEMENS (SIE) Stefan WEGELE	Prototype Architecture, Mapping of D 8.1/D8.5, prototyping activities with IL-Explorer, IL-API, Topo and Timetable conversion.
Contributors	**HaCon (HC)** Rolf GOOSSMANN Sandra KEMPF	Architecture specification, prototyping activities with TPS, evaluation of the proof of concept.
	Ansaldo STS (ASTS) Marco GIAROLI Carlo DAMBRA	Architecture specification, prototyping activities with Asset Management, evaluation of the proof of concept.
	Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) Ulf NOYER Christian LINDER	Architecture specification, prototyping activities with IL-API, evaluation of the proof of concept.

Executive Summary

The way to cost-efficient and reliable Traffic Management Systems (TMS) goes through opening markets for innovative solutions, cost reduction for solution integration and providing innovative functions increasing the degree of automation of railway operations.

The first two aspects are covered by a common communication platform specified in In2rail-WP8 and the development of the innovative functions is subject of the future Shift2rail projects. This deliverable provides a bridge between the specification of the communication platform and future Shift2rail projects – its objective is to evaluate, if the specification is good enough for future TMS and for the future Shift2rail projects.

The evaluation is mostly a formal process: there is a set of requirements defined in former works in In2Rail, which are analysed one by one, and an assessment about its fulfilment is carried out. The results of this process are documented in this document and its annexes.

Another objective of the task is to convince the future Shift2rail partner of the maturity and quality of the specification. To cover both objectives several software prototypes were developed during the specification work of WP8. The detected issues of the specification were immediately reported to the specification team and used for improvements. On the other hand the prototypes provided enough practical experience to enable assessment of the requirements.

TABLE OF CONTENTS

Abbreviations and Acronyms

Abbreviation / Acronyms	Description
AF	Application Framework
API	Application programming interface
CDM	Canonical Data Model: Hierarchically structured object data model; developed and specified in WP8
COTS	Commercial off-the-shelf
DBMS	DataBase Management Systems
Dx.y	In2Rail Deliverable x.y from Work Package x
IL	Integration Layer
I²M *	**I**ntelligent **M**obility **M**anagement: information developed as a strategically critical asset: • A standardised approach to information management and dispatching systems enabling an integrated Traffic Management System (TMS). • An Information and Communication Technology (ICT) environment supporting all transport operational systems with standardised interfaces and with a plug and play framework for TMS applications. • An advanced asset information system with the ability to 'nowcast' and forecast network asset statuses with the associated uncertainties from heterogeneous data sources.
IMDG	In-Memory Data Grid
RailML	The original railML models (railML, railML 2) are a series of XML schemas produced by a loose consortium of railway companies, academic institutions, and consultancy firms (see www.railml.org for further information)
TMS	Traffic Management System: a traffic control-command and supervision/management system, such as ERTMS in the railway sector.
TPS	Train Planning System (system developed by HaCon)
WP7	Work Package 7: System Engineering of Intelligent Mobility Management (I²M) of In2Rail.
WP8	Work Package 8: Integration Layer of Intelligent Mobility Management (I²M) of In2Rail.
WP9	Work Package 9: Intelligent Mobility Management (I²M) Nowcasting and Forecasting of In2Rail.
XML	Extensible Markup Language

** Definition extract from §Common Glossary of the* [In2Rail D7.1] *deliverable of In2Rail.*

1 Background and Objective

This document constitutes the first issue of Deliverable D7.5 "Evaluation of the Proof-of-concept" in the framework of the project entitled "Innovative Intelligent Rail" (Project Acronym: In2Rail; Grant Agreement No 635900).

The overall objective of Work Package 7 – WP7 – is to provide the specification to validate the Intelligent Mobility Management (I²M) open integrated platform for Traffic Management Systems (TMS) and dispatching systems of the future and validate the outcome from WP8 and WP9. WP7 covers three topics, which come at different development stages of the future Traffic Management System:

- Task 7.1: to carry out the requirement analysis;
- Task 7.2: to specify a Standard Operators' Workstation allowing the display and control of all services and functions applied in an integrated traffic control centre;
- Task 7.3: to validate an integrated I²M Technical Readiness Level 3 proof-of-concept built around the Integration and Application Layers, the Demand Management functionalities and the nowcasting and forecasting of the network assets status.

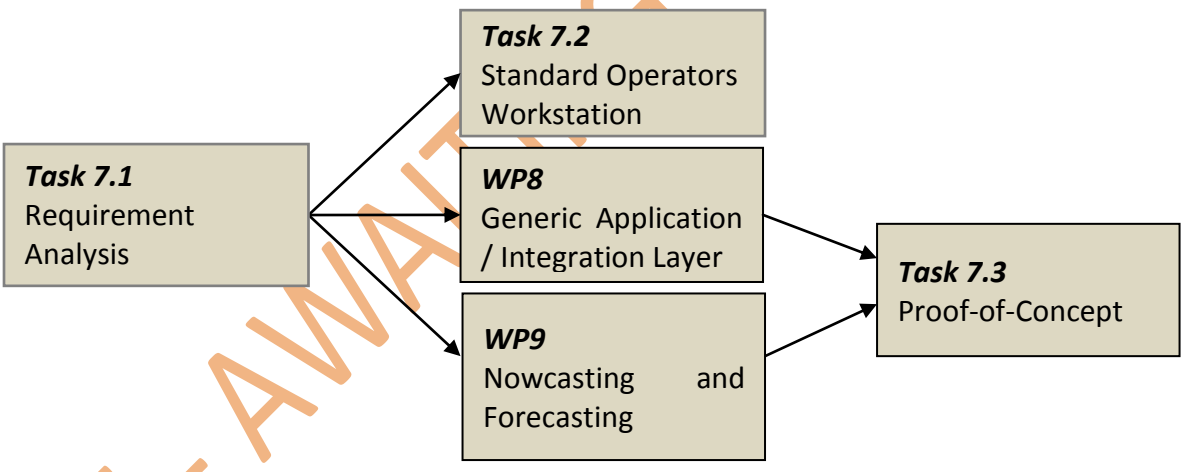

Figure 1.1: Simplified view on integration of WP7 with WP8 and WP9

The objective of WP7.3 is, as Figure 1.1 Figure 1.1:shows, to provide a reliable proof, that the selected architecture for Integration Layer (IL) and Application Framework (AF) in WP8 and the asset status functionalities developed in WP9 are sufficient to fulfil requirements of a future Traffic management system stated in Task 7.1.

Use Cases have been defined that will be demonstrated in the proof-of-concept at the end of the project In2Rail, when in the last phase of Task 7.3 the prototype shall be validated against the system requirements from D7.2, as shown in Figure 1.2.

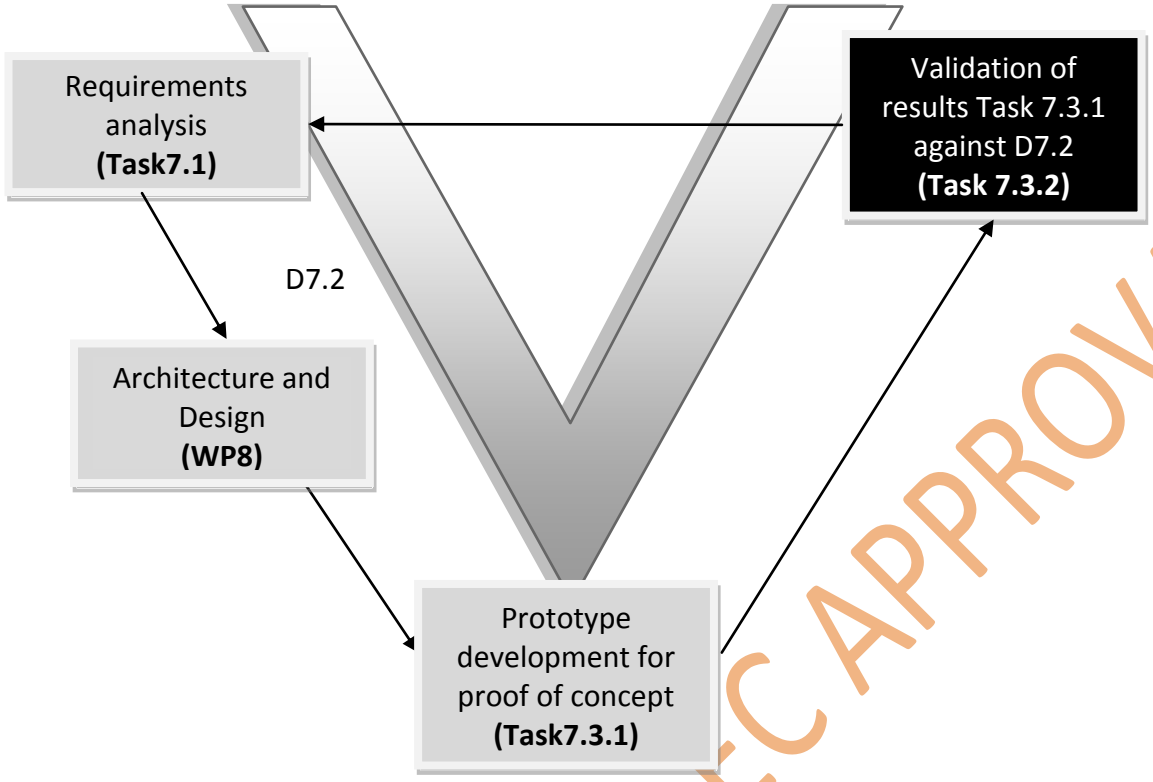

Figure 1.2: Development process as suggested in D 7.2 [In2Rail D7.2]

Deliverable 7.5 is the result of the second and final step of Task 7.3, constructing the TMS prototype following the planned architecture described in Deliverable 7.4 [In2Rail D7.4] as well as the Canonical Data Model outlined by the ongoing works in WP 8. As the work, both in the proof of concept itself, and in the presupposed WP 8 and 9 are ongoing until the end of the project, this deliverable represents a state of work of all of these three Work Packages. Thus alterations of the prototype, the interfaces and the used data until the very end of the project, exceeding the status described in this document, cannot be excluded, especially as the final versions of the WP 8 deliverables are also to be expected at the end of the project (two months after finalisation of this document).

This document is the result of the works building and evaluating the prototype, by the main Task 7.3 members (SIE, DLR, HC, ASTS) as well as ongoing regular workshops of this team, starting in June 2016.

2 Introduction

2.1 General approach

The objective of the Task 7.3, presented in this document, was to analyse the quality of the specifications provided by Work packages 8 and 9 of the In2rail project. The WP9 provided assessment of the results in a specific document D9.5. Therefore the objective of this task was to evaluate the integration of WP9 results with WP8 specifications.

As reference for the quality assessments the requirements from the following sources were used:

- D7.1 and D7.2, providing end user requirements from recent tenders and research projects;
- D8.1 and D8.5, providing end user requirements to extract specific requirements for the specifications to be developed in WP8.

To estimate to which extent the requirements are fulfilled by the specifications from WP8 a software prototype was implemented. The originally planned structure of the prototype is described in D7.4 (see [In2Rail D7.4] chapter 3). The experience collected during the prototype development was used for the requirements assessment documented in the Annexes 1 and 2.

The specifications provided by WP8 should be the basis for a common (standardised) communication platform used for technical demonstrators in Shift2rail projects and as a consequence in future Traffic management systems. The specification comprised a set of documents covering different aspects of the communication platform and having different impact and limitations on the future implementations of TMS:

- Data modelling language: is specified in D8.7 and represents a reasonable XML-notation of messages, its attributes and their relations. As the XML-schema is easily extensible, this document represents a non-critical part of the specification;
- Canonical Data Model (CDM): is the specification of data structures covering different aspects of TMS. The entire model will be quite big. In the frame of the In2rail project the intension was to define rules and patterns for CDM and to start modelling the most required parts of the TMS – infrastructure, timetable, and signalling. The achieved state is not yet able to cover a commercial TMS implementation and is the subject of modifications during the following Shift2rail projects. For the requirements assessments the current state of CDM was used to estimate, if the requirement is already covered or will be easily covered in the future;

- Data access patterns: the Application Programming Interface (API) is the most critical part of the specification, as it separates the responsibilities between client application and the communication platform (Integration Layer). Although minor modifications of the API are possible, modifications of the data access pattern would require huge effort for the client applications: if the application shifted the responsibility for reliable data management to IL, it would not be easy to integrate this responsibility back afterwards;
- Data version management: defined as a Sandbox Management Service comprises the algorithm and the data structures. It is used for keeping a history of modifications, synchronising modification requests and providing transactional behaviour to the clients. This is a critical part of the specification as well – it specifies the workflow for data modifications by editors and algorithms. Minor modifications of the workflow are possible, but changing the main approach would require modifications of all functions involved in data modification.

The idea behind the specification of the common communication platform was to provide a small API for standardised access to the data, and to use an existing solution from the market (Commercial off-the-shelf products – COTS). The main question in this context is, if the resulting combination of API+COTS-Product is able to fulfil all performance, security and extensibility requirements.

2.2 Prototype Architecture

During the planning phase of the proof of concept, a preliminary prototype architecture has been developed (s. [In2Rail D7.4] chapter 3). The main focus of this prototype architecture was to achieve a consistent, functioning overall system building a mini-TMS, so the critical parts of the specification could be identified and evaluated.

During the development the focus was partly shifted from the idea of mini-TMS to the idea of validation of the most critical parts of the specification. Some of the aspects were considered as “achievable with high probability” due to the selected software architecture:

- backward-compatibility of the clients is achieved by using a data modelling language with attribute annotation – in case of new or outdated attributes the new or outdated client would “understand” only the known part of the message;
- the completeness of the data specification is not yet achievable, as in context of In2Rail only a limited part of the Canonical Data Model is preliminarily specified. The specified part was tested during importing/exporting of existing topology and the timetable data – for these two aspects a full coverage was achieved;
- simplicity of the integration of an existing application into Integration Layer was evaluated with the development of several services.

The critical points of the Integration Layer are resulting from unusual patterns for data management and data access:

- conventional systems use two/three layer architecture, with request-reply data access patterns and heavy use of transactions. Therefore such systems strongly depend on relational database management systems, building a backbone of the data management;
- in the Integration Layer the data is managed in Topics as NoSQL-key-value-pairs with the data access by publish-subscribe pattern only. This approach scales very well providing high performance for data management and data distribution, but it does not support transactions as widely applied in conventional systems.

So the first and major critical question for the Integration Layer from the application developers was: “How should I create consistent changes to my data set?”. To evaluate this question a considerable part of the development efforts was dedicated to the transaction management approach of the Integration Layer (Sandbox management functionality) and the integration of the forecast calculating algorithm planned in D7.4 was postponed to the future Shift2rail projects (this function is part of several technical demonstrators anyway).

Figure 2.1 shows the implemented prototype architecture.

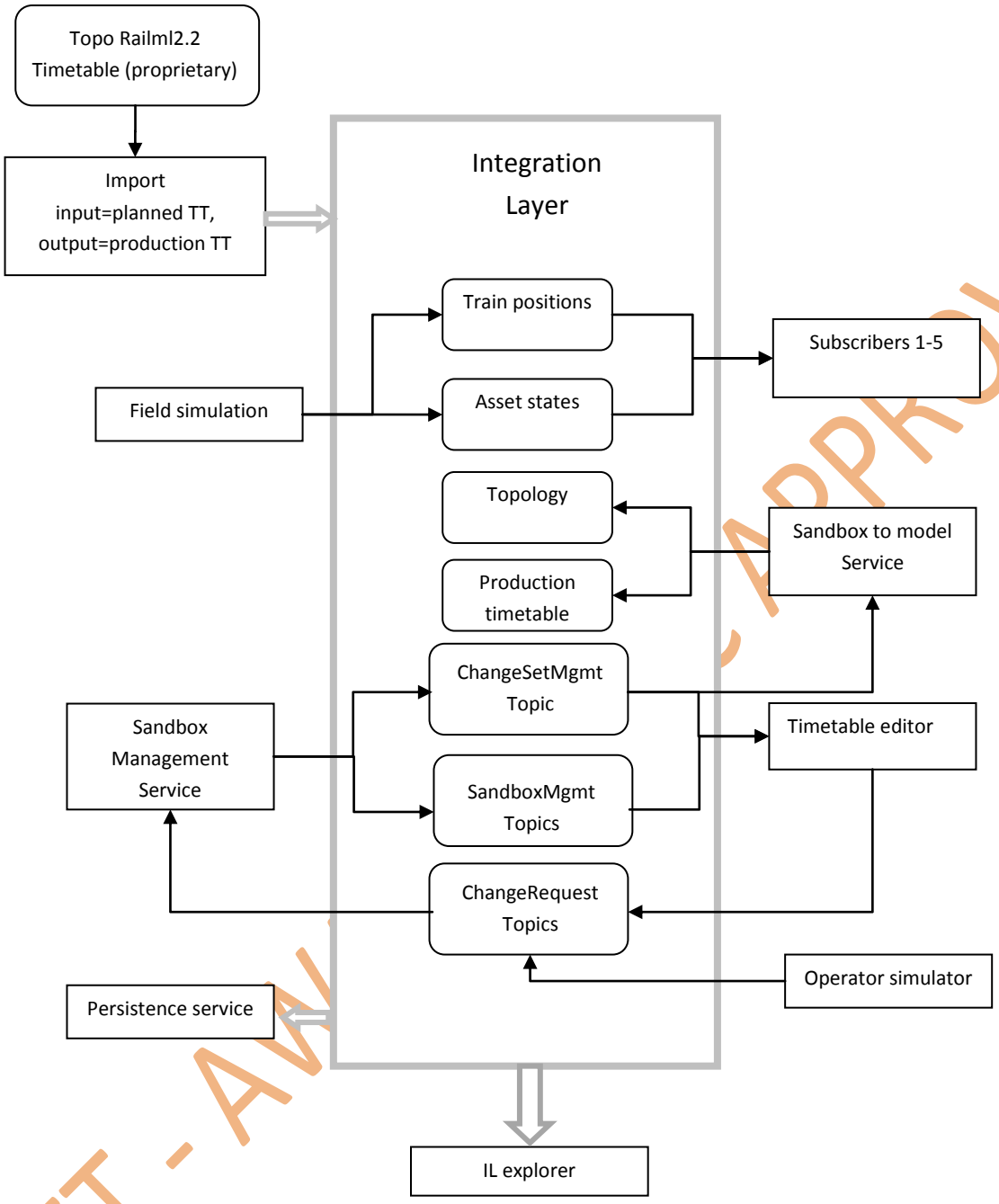

Figure 2.1: Final prototype architecture

It covers most parts of the canonical model, specified in In2Rail, enables validation of performance requirements and evaluation of flexibility due to integration of existing productive software from several companies. In the following chapter the software modules developed during the proof-of-concept are presented including findings to the IL-specification.

3 Evaluation steps for proof of concept

3.1 IL-API

The API is a specification of the functions located in a dynamic library which allow the access to the functionality of the Integration Layer. The functionalities covered by the Integration Layer are quite large therefore it was never an option to develop a new IL-implementation in In2Rail or in Shift2rail, but to select an appropriate solution from the market.

The purpose of the API is to provide a unique way to access data, hiding complexity and interfaces of the selected software product. From this point of view the API shall be small (narrow, simple) enough to be supported by as many products on the market as possible, but it must be extensive enough to cover all the functionalities assigned to the Integration Layer.

In the proof-of-concept two libraries were implemented providing the “In2rail-API” to two products from different types of the software products:

- Hazelcast as a representor of In Memory Data Grid solutions;
- Opensplice DDS as a representor of Data centric publish-subscribe solutions.

It was possible to cover all functions of IL by both products, proving the maturity level of the IL-API specification.

3.2 Data importer

To evaluate preliminary data structures and a general setup of the Canonical Data Model a data importer was developed, which converted:

- productive microscopic infrastructure data of a middle sized country with over 5000 km tracks from RailML 2 representation into CDM containing over 42.200 track elements;
- proprietary timetable representation with over 2000 trips with detailed routes into preliminary timetable part of CDM.

Both conversions were possible, proving the maturity of the CDM specification. In the meantime the CDM specification was modified, so further adjustments of the converter are required in Shift2rail. Other parts of the CDM like Restriction management or Possession management will be specified during Shift2Rail projects.

3.3 IL explorer

The IL explorer evaluated the following main aspects:

- it shall be possible to dynamically extend the CDM, so the IL explorer shall be able to represent data in some readable format dynamically (without recompiling);
- it shall provide access and modification functionality of every data-set on the IL, allowing efficient validation, testing, and debugging.

The implementation of IL-explorer covers the API-versions prior to 10/2017. It will be adjusted to the latest API specification at the end of the In2Rail project.

3.4 Sandbox management service

The Integration Layer specifies the way to handle transaction and versioning of data sets in [In2Rail D8.4]. In context of the proof-of-concept the specified algorithm was implemented and tested with the imported topology and timetable data in a Timetable Editor. Being one of the critical IL-functions it was tested by the testing approach specified in [In2Rail D8.8] with the REST-API-Testing framework Chakram. During the implementation it could be proved, that:

- the algorithm for the Sandbox Management is working properly;
- the part of CDM covering Sandbox Management is sufficient;
- the testing approach in D8.8 allows efficient automated testing.

3.5 Performance evaluation

To evaluate performance restrictions resulting from the selected API and product on the market, a test scenario was created, emulating a big disturbance during a normal traffic for the imported topology and timetable:

- one hour of the operation was selected from 8:00 to 9:00;
- for all running trains train position reports were generated every 3 seconds (assuming ETCS-2 approach);
- changes in sections' occupations influenced by moving trains' issued state-change-;
- messages for routing commands, switch state changes, releases were calculated;
- it was assumed that 20 operators concurrently work on the timetable producing one modification per second and applying their sandboxes every 2 minutes on average;
- it was assumed that a decision support system provided a new proposal to each of the operators every minute taking into account 20-50 different modifications.

All generated telegrams were saved in csv-format and created a test scenario for the performance evaluations. In total it was 265 MB in 383.000 telegrams during the hour.

Figure 3.1 shows the hardware setup. The Integration Layer is running in a one-node-setup. The services send the telegrams as specified in the scenario described above. The mirror-service doubles the IL-load by copying all TMS messages into a special topic. The Message-Monitoring-Service subscribes to this “mirror”-Topic and calculates delays from initial sending service until message arrival.

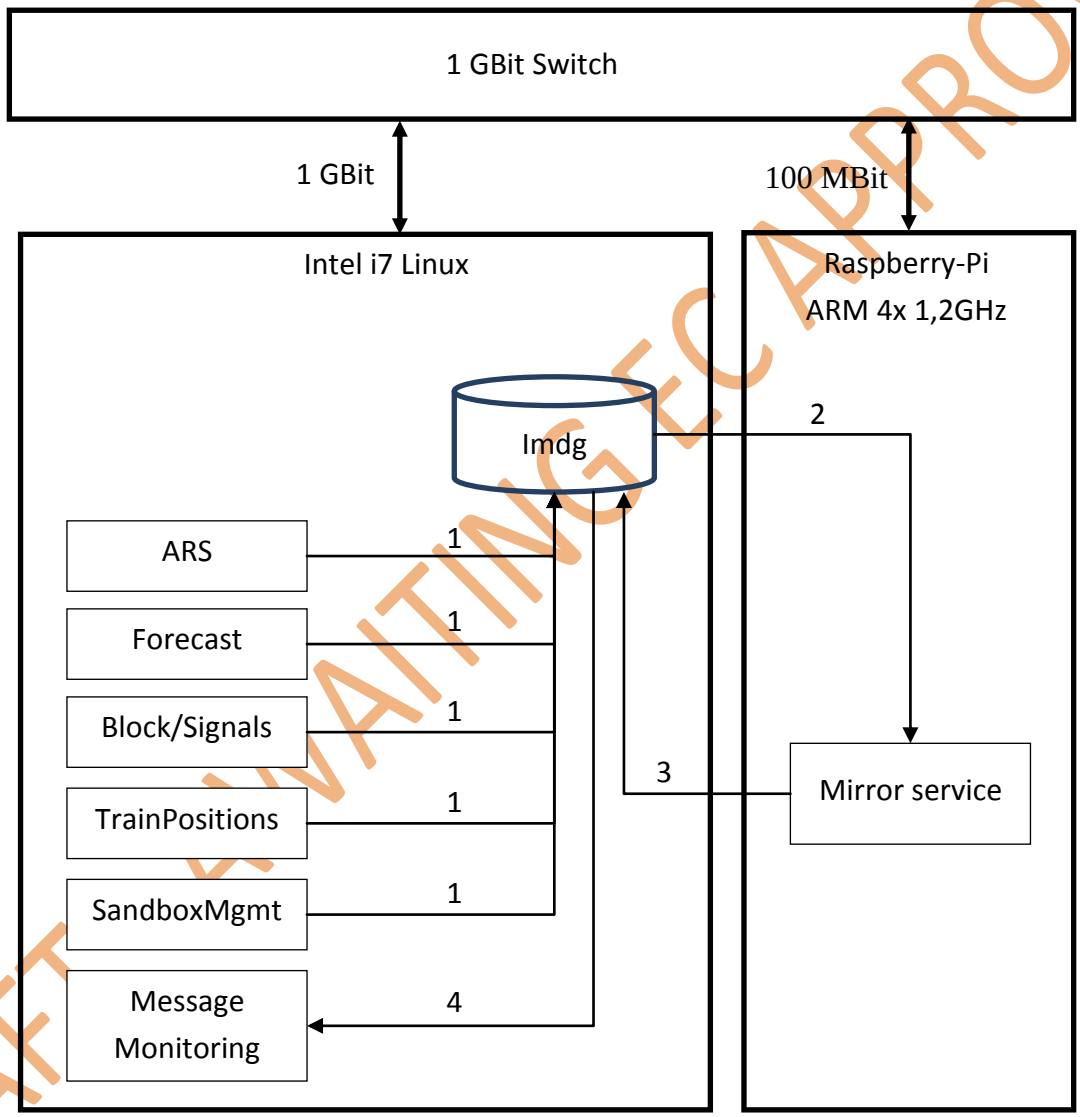

Figure 3.1: Hardware setup for the first performance tests

Therefore the measured message delay comprises four steps:

- time from source service (ARS, Forecast, etc.) to IL;
- time from IL to the mirror service;
- time from mirror service to IL;
- time from IL to the message monitoring service.

Figure 3.2 shows the summary of message delays specific to the source services with different message sizes. It represents a sample of the first three minutes of the scenario. It is obvious that the message transfer time on the 100 Mbit-interface of the mirror service dominates the delay – the bigger the message, the longer is the delay.

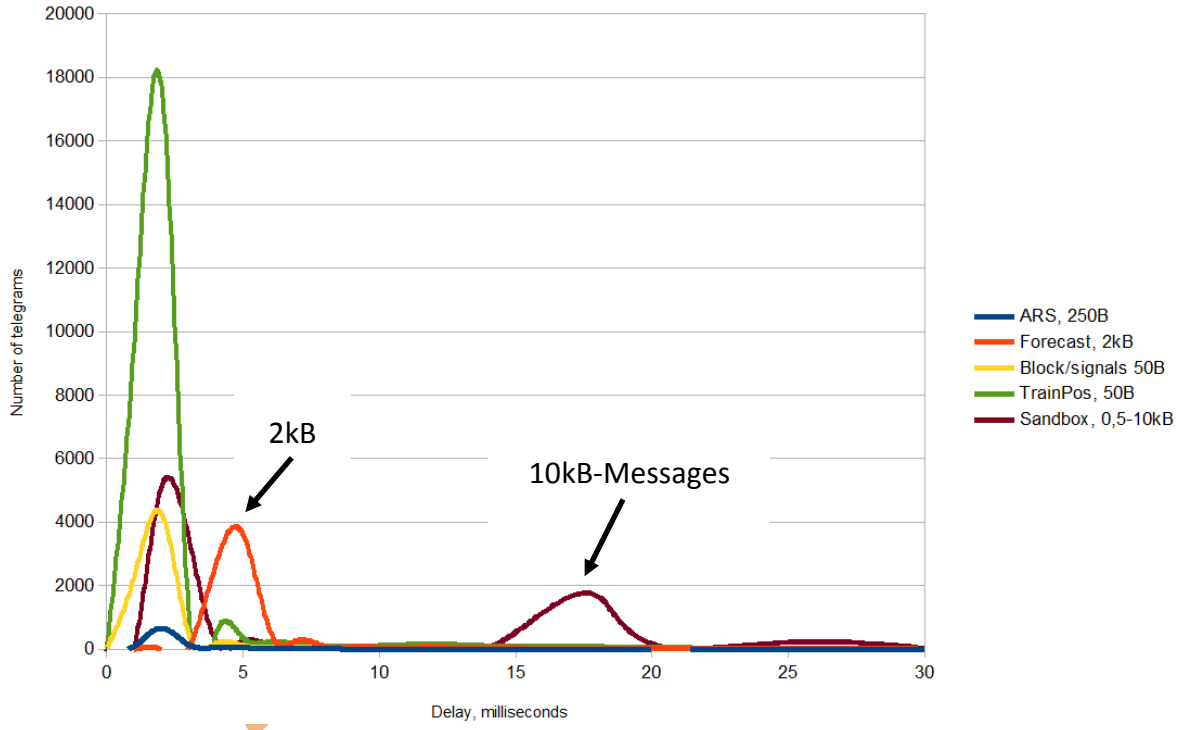

Figure 3.2: Message delays for different TMS systems with double load of the Integration Layer

In this test scenario with only two subscribing applications (mirror service and message monitoring service) only the overhead of data management inside of the IL is evaluated. As the results show, the time spent by IL for the data management and synchronisation between IL-nodes is negligible in comparison to the network delays.

During the next phase of the performance evaluation, as foreseen in Shift2rail, a setup with larger number of subscribers will be evaluated, to estimate the data distribution performance. The current assumption is that it will be limited to 99% by the available network bandwidth.

The outcome of the performance evaluation should not be overestimated, as this scenario evaluates one specific product on the market, i.e. Hazelcast. Hazelcast's main purpose is not the high performance message delivery, but a rich set of IMDG-functions.

Most probably the Opensplice–DDS implementation will over perform Hazelcast in message delivery metrics, as it uses a server-less architecture with multicast reliable message distribution protocol (biggest work of the message distribution is shifted to the network-hardware). But even in case of Hazelcast typical requirements on messaging system can be fulfilled.

3.6 Application Framework

The Application Framework is specified in [In2Rail D8.6] and [In2Rail D8.7]. It is using the Integration Layer for communication with AF-clients:

- one topic for the definition of the desired state of the services (which services shall run, how many instances, to which topics it shall be connected, etc.);
- one topic for the current state of the managed services.

The general idea for the AF implementation is similar to the IL: the functionalities shall be provided by a product from the market. The purpose of AF specification is to provide a unique way to communicate with AF-functionalities. It is done by specification of the messages and topics on the IL. In context of the proof-of-concept the product Docker-Swarm was selected as a basis for implementation. The selected architecture is shown in Figure 3.3.

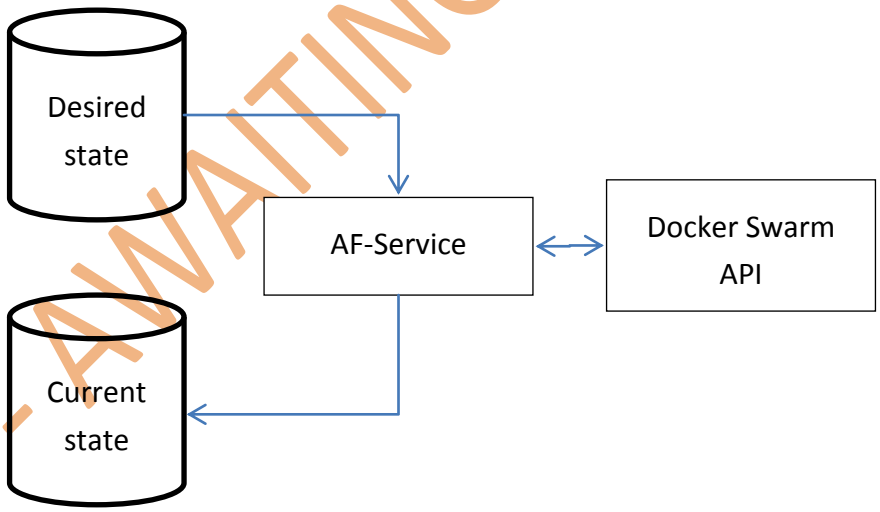

Figure 3.3: Architecture of the AF-prototype

The AF-Service is responsible for mapping the AF-protocol to Docker-Swarm-Protocol. The started implementation was not finished due to higher efforts spent on the Sandbox functionality. But the collected experience allows evaluating requirements from D8.5 for the selected AF-architecture.

3.7 Integration of Asset Forecasting Software from Ansaldo STS

The WP9 of In2Rail created software solutions for asset forecasting and nowcasting. The output of these functionalities shall be presented in the TMS for making better decisions.

The IL was used to provide all configuration data (infrastructure, graphical representation) and to receive results of the forecast algorithm.

Figure 3.4 shows a screenshots of the result.

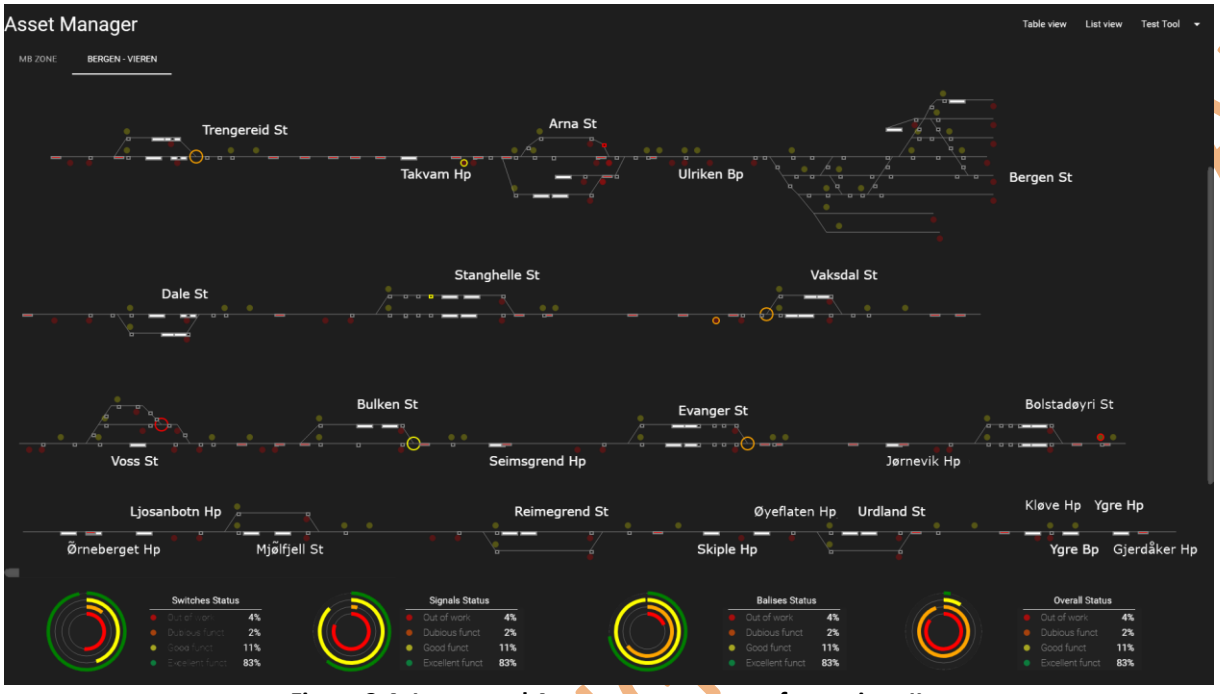

Figure 3.4: Integrated Asset management software into IL

3.8 Integration of existing Timetable Management software (Hacon)

For the development of the prototype the existing HaCon software TPS Online is connected to the Integration Layer. It fulfils the use case to connect an existing “big” TMS Application to the IL. For the In2Rail WP7 prototype the IL is represented by the IMDG (Hazelcast), so the connection is implemented only to the IMDG.

Messages containing train positions of the running trains are sent from the IMDG to TPS Online. TPS Online processes these messages by updating the timetable inside of the system. The updated running trains can be seen in the Tabular Timetable Editor and the Graphical Timetable (see Figure 3.5). The imported train positions are shown with green colour as times at specific points/stations. Comparing the actual times to the planned times provides the delay of the trains.

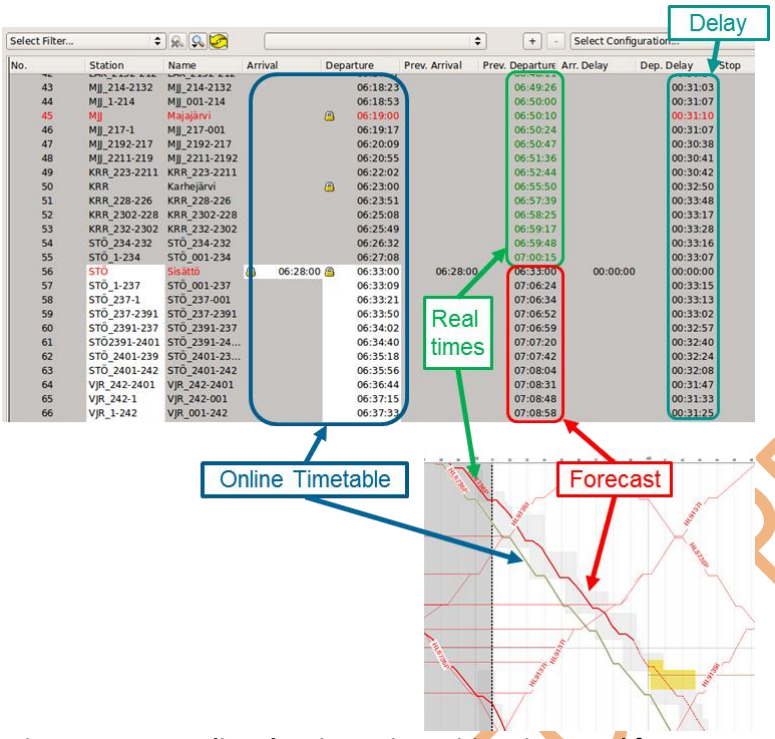

Figure 3.5: TPS Online showing train positions imported from IMDG

Additionally the infrastructure view of TPS Online shows the actual train positions and the occupied tracks (see Figure 3.6).

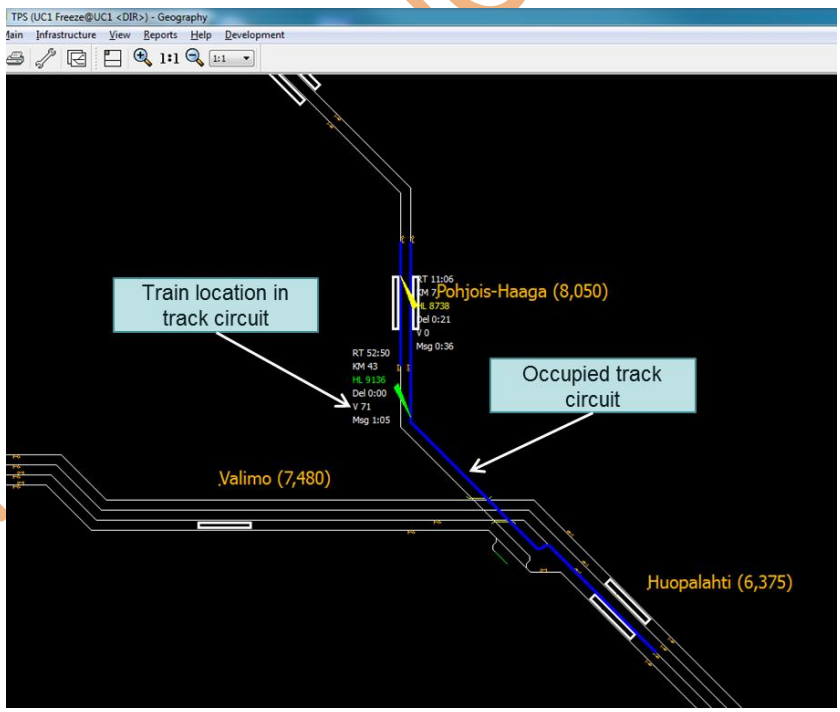

Figure 3.6: TPS Online Infrastructure view with actual running trains

3.9 Requirements fulfilment state

The experiences collected during the prototype development were used to assess to which extent the specification is able to fulfil the requirements formulated for Integration Layer and Application Framework. The assessment results are presented in the Annexes 1 and 2:

- Annex 1 containing the assessment of the Integration Layer requirements stated in [In2Rail D8.1];
- Annex 2 containing the assessment of the Application Framework requirements stated in [In2Rail D8.5].

The annexes contain the whole original files from the WP8 Deliverables, extended by additional columns in the “requirements” work sheets.

4 Conclusions and Outlook

The development of the proof-of-concept prototype was carried out based on the specification works in the other WP. This on the one hand reduced the performance of the prototype developers, as the often changing specification required some work to be redone. On the other hand the specification team received quick feed-back on modifications improving the maturity of the specification to a great extent.

The main result of the proof-of-concept is, that the IL specification is currently considered to be a good basis for the future TMS. In further Shift2rail projects bigger teams will use the specification for the development of technical demonstrators. The work done in this work package will allow them to quickly start with the main functionalities.

5 References

[In2Rail D7.1] In2Rail Project, Grant Agreement H2020-MG-2014-635900 – D7.1 State-of-the-Art and High Level Requirements – 30/09/2015, http://www.in2rail.eu/

[In2Rail D7.2] In2Rail Project, Grant Agreement H2020-MG-2014-635900 – D7.2 I2M Consolidated functional and non-functional requirements – 30/04/2016, http://www.in2rail.eu/

[IN2Rail D7.4] In2Rail Project, Grant Agreement H2020-MG-2014-635900 – D7.4 Definition of the Proof-of-concept - 30/04/2017, http://www.in2rail.eu/

[In2Rail D8.1] In2Rail Project, Grant Agreement H2020-MG-2014-635900 – D8.1 Requirements for the Integration Layer – 31/12/2016, http://www.in2rail.eu/

[In2Rail D8.4] In2Rail Project, Grant Agreement H2020-MG-2014-635900 – D8.4 Interface Control Document for Integration Layer Interfaces, external/ Web interfaces and Dynamic Demand Service – 30/04/2018, http://www.in2rail.eu/

[In2Rail D8.5] In2Rail Project, Grant Agreement H2020-MG-2014-635900 – D8.5 Requirements for the Generic Application Framework – 30/09/2016, http://www.in2rail.eu/

[In2Rail D8.6] In2Rail Project, Grant Agreement H2020-MG-2014-635900 – D8.6 Description of the Generic Application Framework and its constituents – 31/07/2017, http://www.in2rail.eu/

[In2Rail D8.7] In2Rail Project, Grant Agreement H2020-MG-2014-635900 – D8.7 Interface Control Document (ICD) for Application-specific Interfaces – 31/07/2017, http://www.in2rail.eu/

[In2Rail D8.8] In2Rail Project, Grant Agreement H2020-MG-2014-635900 – D8.8 Integration Test Plan for Application Framework and Constituents – 30/04/2018, http://www.in2rail.eu/

6 Annexes

Annex 1	Excel sheet with assessment of Integration Layer Requirements, based on [In2Rail D8.1] Annex 1.
Annex 2	Excel sheet with assessment of Generic Application Framework Requirements, based on [In2Rail D8.5].