Horizon 2020
European Union Funding
for Research & Innovation

Deliverable D 6.1
Overall high-level architecture

Project acronym:	FR8RAIL
Starting date:	12.06.2019
Duration (in months):	24
Call (part) identifier:	2020-S2RJU-CFM-2015-01-1
Grant agreement no:	730617
Due date of deliverable:	July 2019
Actual submission date:	11-07-2019
Responsible/Author:	Norbert Kahl (DB Cargo)
Dissemination level:	PU
Status:	Final Report

Reviewed: Yes

Document history		
Revision	*Date*	*Description*
1	28.06.2019	First issue
2	30.04.2020	Second issue revised based on JU comment

Report contributors		
Name	Beneficiary Short Name	Details of contribution
Norbert Kahl	DB Cargo	All Chapters
Molley Morgan	DB AG	Review
Jan Bergstrand	TRV	Review

Table of Contents

1. Executive Summary

In order to achieve the goals of Shift2Rail IP5, the highl-level architecture discussed in FR8RAIL aims to make use of modern technologies such as Internet of Things, with significant business impact. Asset & Maintenance Digitization based on this high-level architecture will support various functionalities to increase the competitiveness of rail freight, as shown in figure 1. It is illustrated that both, locomotives (1) and wagons (3), should be equipped with sensors and further technologies in order to diagnose the condition of the assets. In addition, with wayside monitoring captures and recognises the assets (4). All data are stored in an Asset Intelligence Center (5), which allows further analytics (6, 7 & 8) as well as live usage (9) and the development of further use cases. Also, the high-level architecture paves the way for a maintenance platform that enables digital fleet control and maximizes the utilisation for the assets.

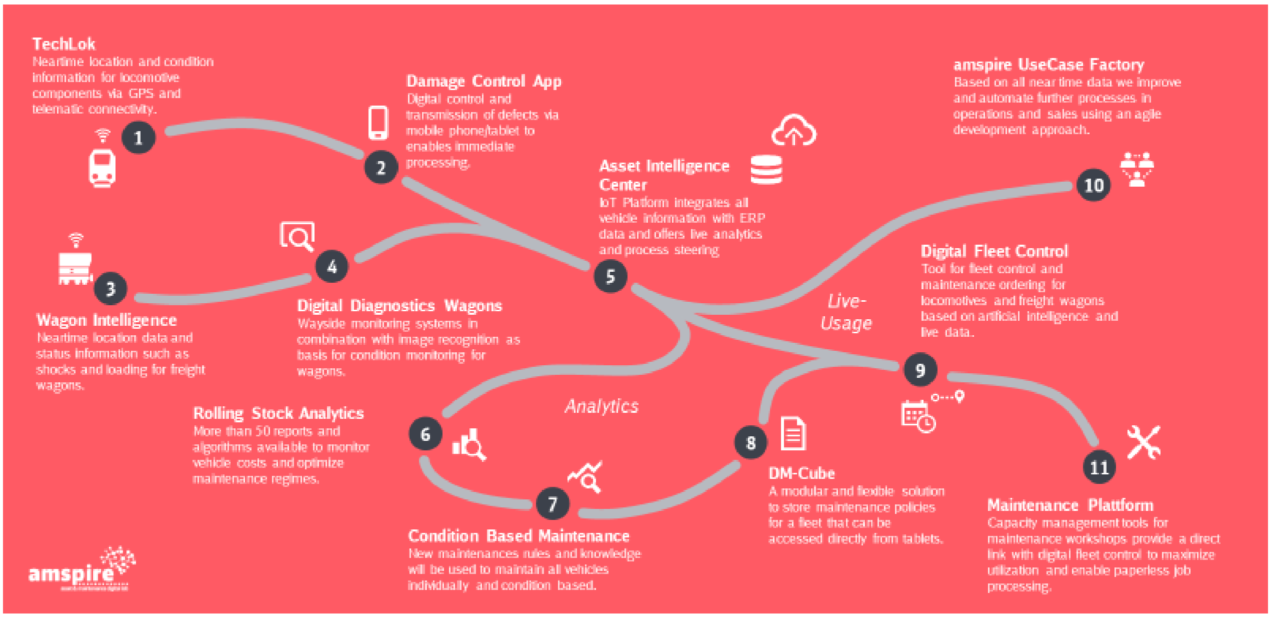

Figure 1 High Level Architecture: Asset & Maintenance Digitization

In sum, the core of the high-level architecture for Asset & Maintenance Digitization will be an Asset Intelligence Center as central analytics and processing engine. This same Asset Intelligence Center is flagged out as a central element in the Shift2Rail IP5 vision.

The Asset Intelligence Center is enabling other projects to collect, process and store their data in a flexible and scalable big data environment in the cloud. The environment is fully virtualized in processing and data storage terms and based on current opensource software.

The high-level architecture will lay a foundation for application in the Fr8Rail work packages. The relevance for the technical developments of FR8RAIL has been proven. Furthermore, it can serve to illustrate centralized analytics in rail freight, as has been done with this deliverable.

2. Abbreviations and acronyms

Abbreviation / Acronyms	Description
DB	German Railways (Deutsche Bahn)
UML	Unified Modeling Language
API	Application Interface
ESB	Enterprise Service Bus
IoT	Internet of Things
Rest	Representational State Transfer
CICD	Continuous Integration and Continuous Delivery/Deployment
HDFS	Hadoop distributed filesystem
NiFi	Niagara Files
API	Application Interface

3. Background

The present document constitutes the Deliverable D 6.1 “Overall high-level architecture” in the framework of the WP6 - High level System architecture and integration of FR8RAIL. This WP relates to TD 5.1 to TD 5.3 within IP5.

As described in chapter 6 various occasions gave opportunities for alignment processes. Also, to have an overall alignment further discussions were taken on IP5 level.

4. Objective/Aim

This work package (WP) is focused in two areas:

1. The information system, which combines telematics applications and condition based and predictive maintenance, and
2. The technical and process systems, which includes running gear, core and new market wagons, and automatic couplings.

The objective for these two areas is the alignment and synchronization of the high-level architecture design between each relevant WP following a bottom-up approach.
The input for this work package is the defined architecture of each WP, that is to say, the requirements and the architecture of the services for freight transportation planned to be developed using the information system, and the functional and performance requirements for the technical and process systems (bottom-up approach).

The objective for this WP is the alignment and synchronization of the high-level architectures between each WP. For this reason the main activity of this WP will be the coordination in terms of content on a work level basis. In a first step the WP-based architectures will be compared and in a second step verified in order to ensure a common picture and match/alignment of the business view (e.g. operating model and business processes), the information view (e.g. functional architecture) and the technology view (infrastructure, technical architecture incl. interfaces etc.). Step one and step two will be executed in an iterative process between all involved stakeholders. The whole process is supported through several synchronisation meetings and discussions, both online and offline.

The result will be a common understanding and an alignment of the overall high-level architecture which will enable and ensure future integration.

The output of this work package will be an aligned, synchronized and coordinated overall high-level architecture across the WPs to ensure an overall picture and will enable the future integration of the developed subsystems among them and with other systems in later projects within the Shift2Rail initiative. The alignment will follow the agile approach to continuously coordinate during the entire project with all WPs to ensure a state-of-the-art solution in a rapid technology progress, e.g. new ways for analytics/big data, new available sensors, new IT solutions for data transfer etc.

5. Overall high level architecture

The following architecture is designed by the c4model that is separated in four different levels like Context, Containers, Components and Code.

A System Context diagram provides a starting point, showing how the software system in scope fits into its environment. A Container diagram zooms into the software system in scope, showing the high-level technical building blocks. A Component diagram zooms into an individual container, showing the components inside it. A code (e.g. UML class) diagram can be used to zoom into an individual component, showing how that component is implemented.
This deliverable is focused on the system context and an example container to describe the fundamentals of the architecture.

5.1. Asset & Maintenance Digitization System Context

The first level of these documentation is represented by the system context of Asset & Maintenance Digitization as high level overview. It shows the involved user-groups and the cooperation between the containers of the system like the *Things* and an *API*.

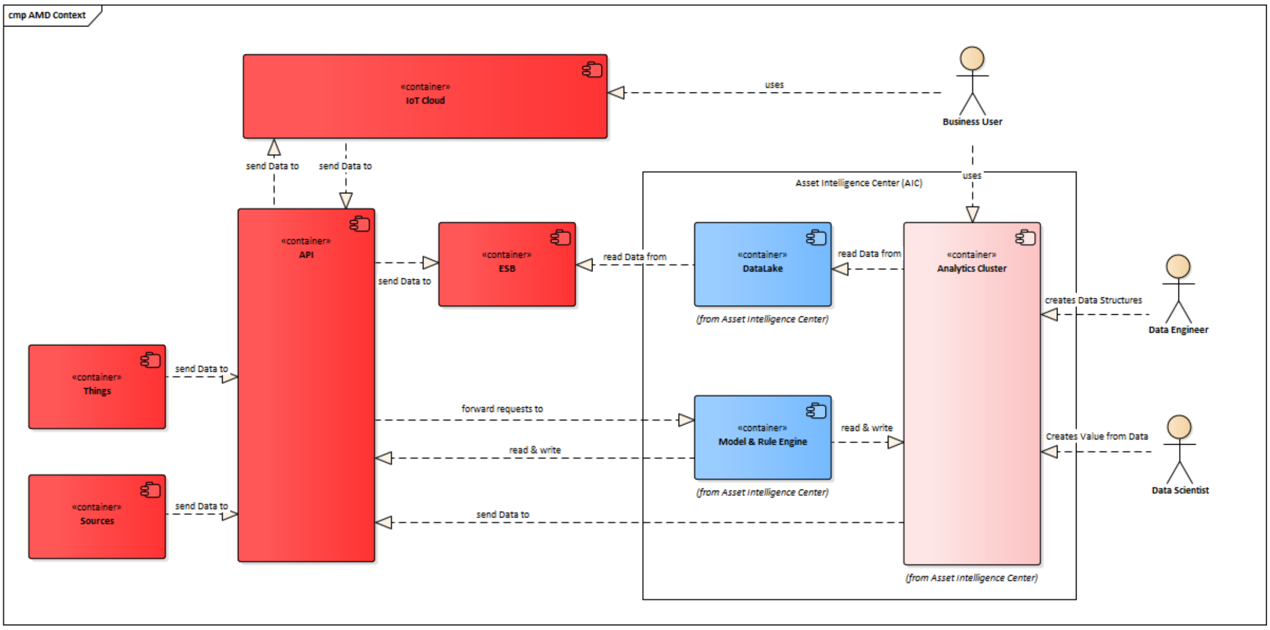

Figure 2 System Context

The *Things* contain intelligent locomotives and freight wagons, that are sending its data over an *API* to the *IoT Cloud*, from where it is processed and copied to an *enterprise service bus (ESB)*. These decoupling of Source- and Target-System allows a low level availability for non-critical container like the *Datalake*, that don't has to be "always on" and fault tolerant.
In Addition to that, other *sources* like SAP and Database Systems send its data over the *API* to *ESB*, too.

The *Datalake* as part of an *Asset Intelligence Center* will pull the data from the *ESB* and stores it for later processing in a fault tolerant storage like Amazon S3-Buckets.
The *Analytics Cluster* is pulling the data from the *Datalake* into its own storage formats, optimized for analytical queries.

When the data has to be enriched or transformed into another data model, the *Model & Rule Engine* is triggered over the *API* as a standalone REST-Service that reads and writes from and to the *Analytics Cluster* and/or reads and writes over the *API* from other systems like *IoT Cloud*.
Business Users, Data Scientists and Data Engineers can access specialized frontends, provided of the *Analytics Cluster*.

5.2. Analytics Cluster Container

The second level of these documentation is represented by the Container of The *Analytics Cluster* that is enabling the other projects to collect, process and store their data in a flexible and scalable big data environment in the cloud. The environment is fully virtualized in processing and data storage terms, and based on current opensource software.

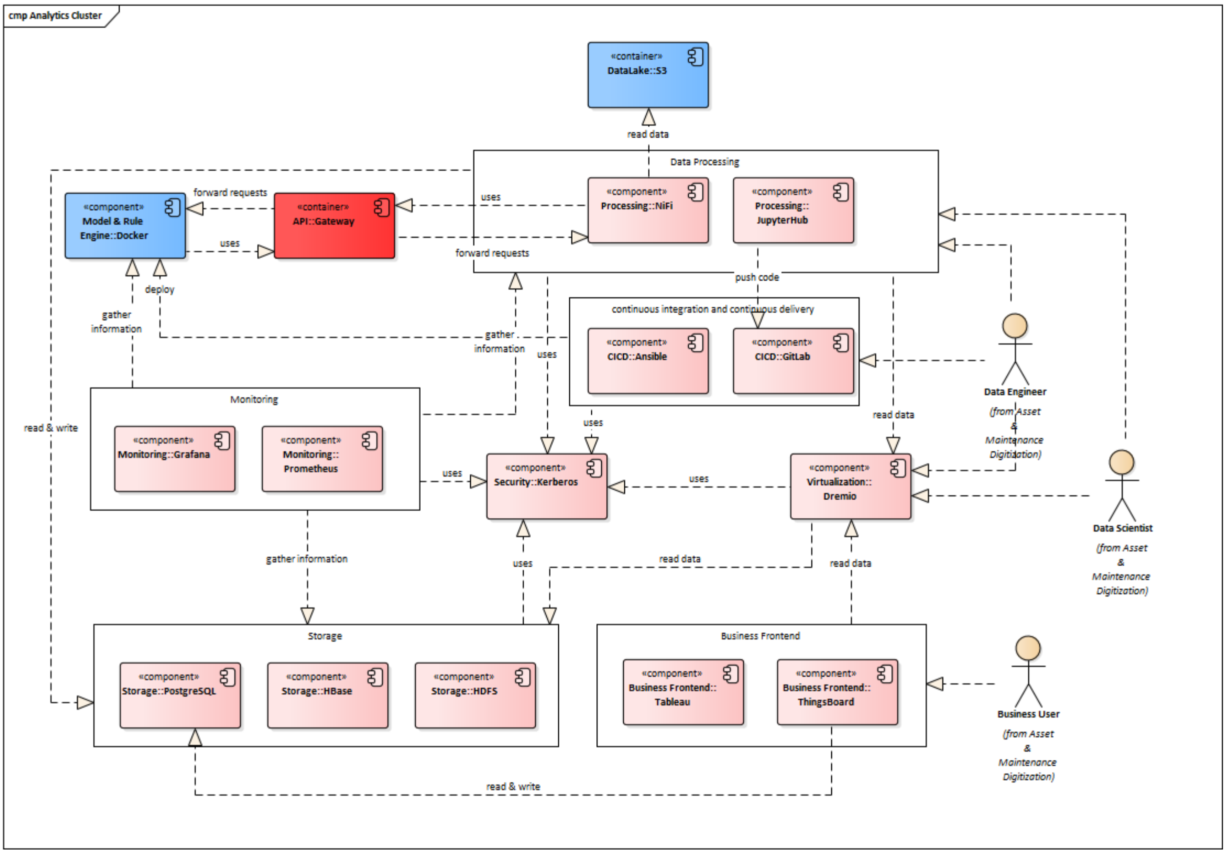

Figure 3 Analytics Cluster

The basic intention of the Analytics Cluster is to empower the *Data Engineer*, *Data Scientist* and *Business user* to work as efficient as possible with the Data.

5.2.1. Virtualization

Heart of this solution is a data virtualization platform like *Dremio* that is located between the *Business Frontends*, the *Data Processing* components like *NiFi* and *Jupyter*, and the *Storage* systems like *HDFS* and *HBase*. With this data virtualization system it is possible that a *Data Engineer* is creating the structure of a new data bucket out of different storage systems, that is analysed by a *Data Scientist* who is not interested in these details. After the *Data Scientist* has detected something interesting for the *Business User*, he´s able to present these data in an optimized format at the *Business Frontend* like *Tableau*.

5.2.2. Storage

With that virtualization concept the *Analytics Cluster* is able to combine the batch and the real-time stream of a lambda-architecture in a common table-oriented way for the *Business User* without knowledge of the different behaviour of the storage systems.

Batch processes like daily deliveries are traditionally stored within *HDFS* in Parquet files, so that analytical queries can access the data very efficiently. Realtime or near-realtime processed data is stored in *HBase* for fast writing access so that the data stream is never blocked by the latency of the storage system.

To present the combined view of these two streams, *Dremio* creates his own views for the engineers, scientists and business users, so that the data can be accessed in an optimized way.

5.2.1. Processing

The data processing is based on a flow-engine like Apache *NiFi* that uses a virtualization engine (in this architecture *Dremio*) to access data and process the data flows. In addition to that, *NiFi* is also responsible to pull the data from its source system *Datalake* initially. In the case that *NiFi* has not the right processor in its default toolset, it is using the *Model & Rule Engine* for further processing like adding rules to the flow or mapping a data model to the processing stream. In that context, the *API:Gateway* is used for a central repository of Rest-Services, placed in the *Model & Rule Engine*.

5.2.2. Monitoring

The monitoring of all processes is built around the application *Prometheus* that is gathering all the necessary process information and stores it on a central place for the frontend *Grafana*. All necessary information is created by the applications, services and images of the *Model & Rule Engine* independent of *Prometheus*, so that there is no direct coupling between the business application and the monitoring tasks.

5.2.3. Continuous Integration / Continuous Deployment

To enable a good development environment, there should be a complete *CICD* pipeline with at least something for storing code like *Gitlab* and something for deploying the code to the cluster and orchestrate the environments like *Ansible*. In combination with *Jupyter* as development frontend for writing the code, and a connection to *Gitlab*, all developers are able to write, document and deploy software to a distributed cluster in the *Model & Rule Engine*.

5.2.4. Security

All of these "Containers" are built around a common security system to ensure a clean authentication mechanism between application and client.

6. Relevance for FR8RAIL technologies

The above described High-Level Architecture for asset and maintenance digitization has the business objective to increase the cost competitiveness of rail freight by making better use of data in operational processes of rail freight. To reach this overall objective, the architecture has the technical objective to provide the basic structure and define the interaction between technological elements to ensure the required functionality and usability. Therefore, looking at figure 1, domain-specific requirements by the technical FR8RAIL WPs 2-5 have been considered as follows:

<u>WP2 Condition based and predictive maintenance:</u>
Mentioned as point No. seven in figure 1, CBM has played a key role in the primary discussion of the FR8RAIL Overall high-level architecture. Following requirements by the technical WP2 have been considered:

- Locomotive status data from element No.1: What data will be made accessible for CBM
- Data transfer from field elements / rolling stock to asset intelligence center: What data will be transferred(e.g. behavioral measurements of locomotive components)
- Scalable big date architecture for No. 5: Where data will need to be stored
- Centralized data formats for rolling stock analytics, No. 4: How data will need to be stored
- Data analysis: What input/ output relation exists for the CBM use cases
- Date utilization at process level – How data will feed information to maintenance

<u>WP3 Telematics & electrification:</u>
Linking into the point No. 3 "Wagon Intelligence" in figure 1, Telematics and Electrification has been discussed as an important enabler function with relevance for the FR8RAIL Overall high-level architecture. Following requirements by the technical WP3 have been considered, in addition to the point regarding WP 2:

- Component status data from wagons: What data will be made accessible for future usage (e.g. behavioral measurements of wagon components)
- Data transfer from wagons to the locomotive, element No.1., for a wireless train backbone: What data will be transferred
- Data transfer from the train to the asset intelligence center, No. 5: What data will be transferred

WP4 Running Gear, Core and Extended market wagon:
Linking into the point No. 3 “Wagon Intelligence“ in figure 1, concerning both the Core and Extended market wagon and their running gear, WP4 requirements have been discussed in the setup of the FR8RAIL Overall high-level architecture. Following requirements by the technical WP4 have been considered:

- Running gear status data: What data will be made accessible for future usage (e.g. behavioral measurements of wagon components)
- Date requirements from and to Core and Extended market wagon functionalities

In addition, the results from WP2 and WP3 are integrated into the wagon concepts, with view to the wagon architecture.

WP5 Automatic Coupling:
Linking into the point No. 3 “Wagon Intelligence“ in figure 1, Automatic Coupling has been discussed as an important enabler function for digitalization and automation at large, with relevance for the FR8RAIL Overall high-level architecture. Following requirements by the technical WP5 have been considered, in addition to the point regarding WP 2:

- Data transfer from wagons to the locomotive, element No.1., by wire based on the data and power lines connecting the train: What, when, how data will be transferred
- Data transfer from the train to the asset intelligence center, No. 5: What, when, how data will be transferred

In the following meetings alignment processes have been discussed:

- Technical Management team meeting (TMT) 18/02/2019
- Workshop and rehearsal before review 04/04/2019
- TMT: preparation for Final Conference 14/06/2019
- IP5 Steering Committee (SC) 28/06/2019
- SC and TMT 10/10/2019
- IP5 TMT 19/06/2019
- TMT: CBM and WP6 focus 27/09/2019

7. Conclusions & Summary

In this document the high-level architecture was described by concentrating on the system context and an example container to describe the fundamentals of the architecture. This helped to understand that a high-level architecture supports the developments in FR8RAIL (WP2, WP3, WP4 and WP5) but also generally those in IP5. figure 1 shows that with technologies such as the digital automated couplers or behavioural data of components both the wagons and the locomotive can be analysed, monitored and controlled accordingly. Hence, the high-level architecture is an example of a centralized analytical platform for different projects in FR8RAIL II (incl. CBM, telematics and electrification) and FR8RAIL III (incl. Intelligent Video Gate/Wayside Monitoring, CBM). These analytical Platform consists of state of the art, big data technologies to store, analyse, prepare an access the wagon and locomotive data for business users, data scientists and data engineers.