

Rail4Future



Projekttitle:	Resilient Digital Railway Systems to enhance performance
Start Datum:	01/04/2021
Durchlaufzeit:	42 Monate
Projektnummer:	882504
Ausschreibung:	8. Ausschreibung COMET Projekte 2019

Deliverable 1.2.2 Specification of Visual Analysis Framework

Fälligkeitsdatum	31.5.2022
Einreichungsdatum	05.07.2022
Eingereicht von	Rebecca Nowak

Version	Datum	Bearbeitet von	Beschreibung
1.0	10.6.22	Rebecca Nowak	Erstellung

Deliverable released


Dr. Manfred Grafinger

by Area Scientific Manager

Deliverable released


Dr. Filip Kitanoski

by Project Scientific Manager

1 Executive Summary

This document describes the system architecture of the visualization layer of the Rail4Future platform. This is a living document. We will integrate new insights and knowledge generated in the course of this project, and adapt and extend it as required.

The visualization layer consists of modules dedicated to the different visualisation modalities. The dashboard module combines the modules for point cloud visualisation, maps, charts and simulations. It also includes a *workspace* module for asset management.

2 Inhaltsverzeichnis

1	EXECUTIVE SUMMARY	3
2	INHALTSVERZEICHNIS	5
3	SYSTEM OVERVIEW.....	6
4	SYSTEM ARCHITECTURE AND TECHNICAL DETAILS	7
4.1	INTERFACES	8
4.2	GEOSPATIAL	8
4.3	UI.....	8
4.4	SIMULATIONS	8
4.5	MAPS	8
4.6	CHARTS.....	8
4.7	POINT CLOUDS.....	9
4.8	WORKSPACE.....	9
4.9	DASHBOARD.....	9

3 System Overview

This document describes the system architecture of the visualization layer of the Rail4Future platform. This is a living document. We will integrate new insights and knowledge generated in the course of this project, and adapt and extend it as required.

The visualization layer of the Rail4Future platform consists of modules dedicated to the different visualisation modalities. The dashboard module combines the modules for point cloud visualisation, maps, charts and simulations. It also includes a *workspace* module for asset management. Interfaces connect the visualization layer to the rest of the Rail4Future platform.

Figure 1 shows an overview of the architecture of the visualization layer.

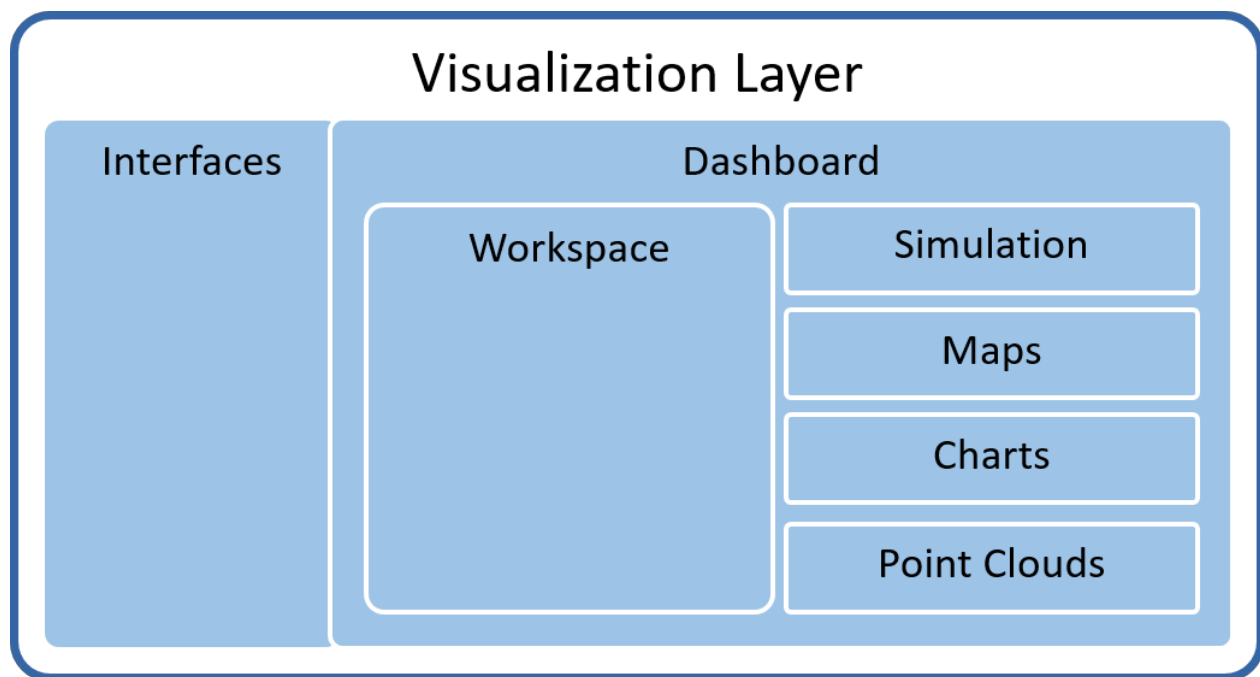


Figure 1: An overview of the architecture of the visualization layer

This architecture was chosen to allow a flexible extension of the visualization layer. Should another type of visualization be required in future, another module could be added to the current visualization modules (Simulation, Maps, Charts and Point Clouds).

4 System Architecture and Technical Details

To avoid inefficiencies like code duplication, common and basic functionality is separated from the higher level visualization modules. Figure 2 shows a detailed system architecture, also including technologies or libraries used in specific modules.

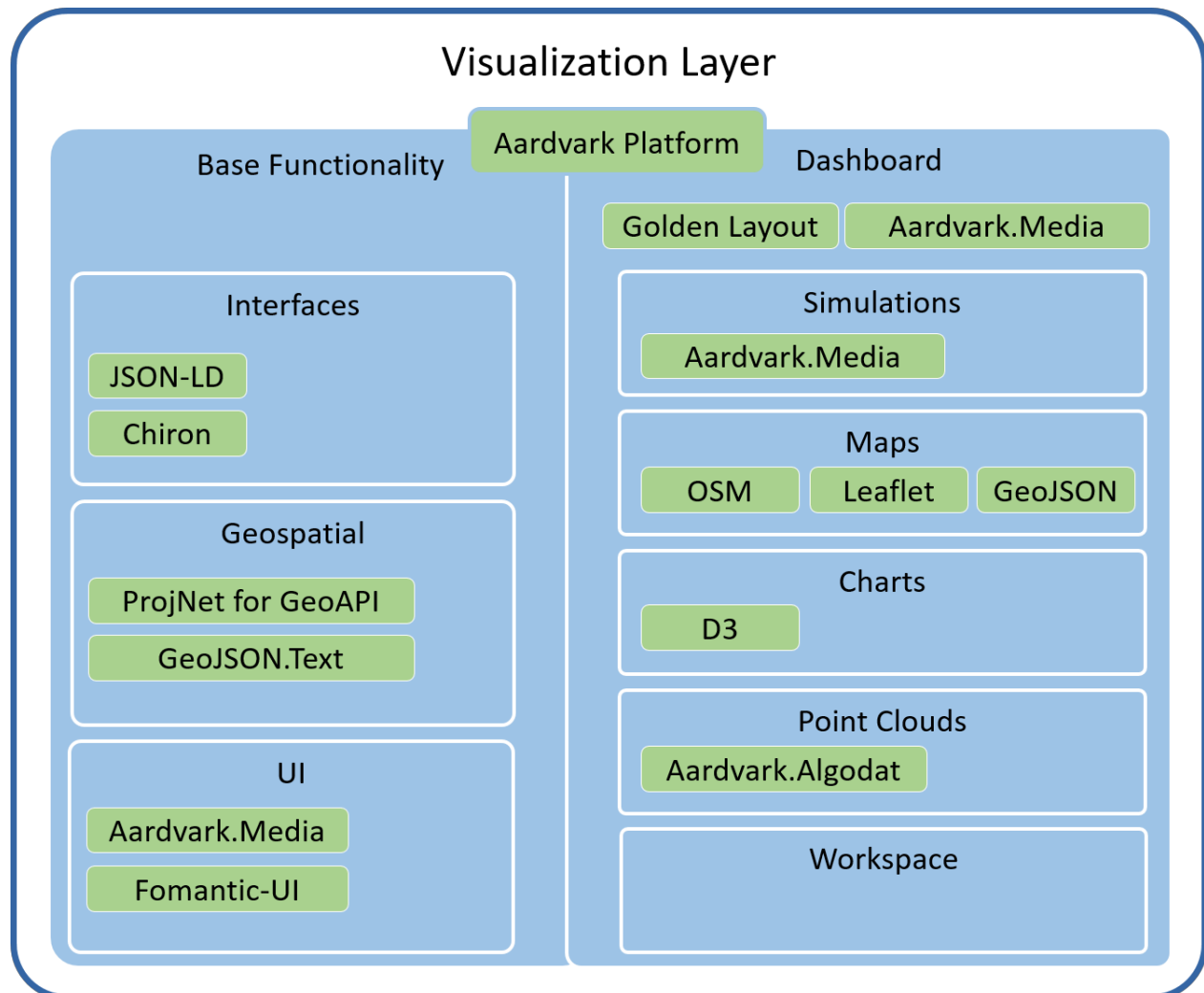


Figure 2 Detailed architecture of the visualization layer with libraries and technologies used for the modules.

We use the [Aardvark Platform](#) to develop the visualization layer. The Aardvark Platform provides us with libraries for quickly creating interactive graphics applications.

All modules listed as part of the Dashboard rather than as Base Functionality are designed so that they can be run independently of each other. This modularisation assures that single modules can be entirely left out of the dashboard, or combined as required for specific use cases.

Below, we briefly describe the individual components of the system. We start with the basic and commonly used functionality and conclude with the individual visualization modules.

4.1 Interfaces

We will use the JSON file format to interface with the rest of the Rail4Future platform. To address the need for structured metadata we intend to use the JSON-LD (JSON for linked data) format.

Data types required for specific use cases, for example a vehicle with its various parameters, are also part of this module.

4.2 Geospatial

This module contains functionality needed for working with geospatial data, for example transforming coordinates into different spatial reference systems. This module also includes functionality that deals with GeoJSON files.

4.3 UI

This module unites user interface elements required by other modules. Next to basic elements like buttons, menus, and dialogs, it also contains more complex or composite elements like a labelled time slider or drag and drop support. The user interface elements in this module are generic, and can be applied to any use case.

4.4 Simulations

This module comprises the functionality to configure simulations, start simulations, and explore their results. We build on [Aardvark.Media](#) to create a user interface that allows the user to select which data should be used as simulation input and set simulation input parameters.

Visualisations specifically designed to explore simulation results or compare different simulation runs are also part of this module.

4.5 Maps

This module deals with displaying and manipulating maps. We use the [Leaflet](#) library, which is an open-source library for interactive maps. We use the GeoJSON format for displaying data on a map. Leaflet allows the use of any map tile provider. We use [OpenStreetMap](#) for development.

4.6 Charts

This module contains the functionality necessary to display and interact with various types of charts. Examples for charts are scatter plots, bar charts, time series, and heat maps. We use the JavaScript library [D3.js](#) for creating interactive charts.

4.7 Point Clouds

This module contains the functionality required to display, interact with, and manipulate point clouds. We use the library [Aardvark.Algodat](#) to work with the large point cloud data sets we deal with in the Rail4Future project.

4.8 Workspace

The workspace module deals with loading and managing available assets. More specifically, it gives the user access to available data, simulations, and visualisations. It provides the user with a graphical user interface to browse and select these assets for further use.

4.9 Dashboard

The dashboard combines the Workspace module and any or all visualisation modules. We use the [GoldenLayout](#) library as a basis for a flexible, interactive layout. This module also deals with configuring dashboard layouts for use cases, and saving or loading previously configured layouts.